

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ІНСТИТУТ ПРОБЛЕМ МАТЕМАТИЧНИХ МАШИН І СИСТЕМ  
НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ**

Кваліфікаційна наукова  
праця на правах рукопису

**ГРИБКОВ СЕРГІЙ ВІТАЛІЙОВИЧ**

УДК 004.9:004.2:65.012.2:664

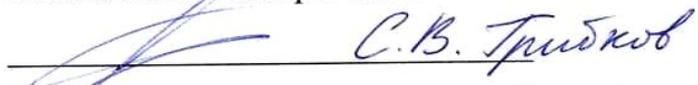
**ДИСЕРТАЦІЯ**

**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ  
МОДИФІКОВАНИХ ЕВРИСТИЧНИХ І ЕВОЛЮЦІЙНИХ МЕТОДІВ ТА  
АЛГОРИТМІВ ДЛЯ ХАРЧОВИХ ПІДПРИЄМСТВ**

Спеціальність 05.13.06 – інформаційні технології

Подається на здобуття наукового ступеня доктора технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

  
(підпис, ініціали та прізвище здобувача)

Науковий консультант Литвинов Валерій Андроникович доктор технічних наук,  
професор

Київ – 2021

## АНОТАЦІЯ

*Грибков С. В.* Інформаційні технології прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів для харчових підприємств. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Національний університет харчових технологій МОН України. – Інститут проблем математичних машин і систем НАН України, Київ, 2021.

Загальним результатом дисертаційного дослідження є вирішення науково-прикладної проблеми підвищення ефективності процесів управління харчовим підприємством шляхом розробки інформаційних технологій підтримки прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів.

У роботі розроблено математичну модель процесу планування виконання замовлень, яка, на відміну від традиційних, що передбачають розбиття за рівнями управління та підрозділами виконання, забезпечує оцінку ефективності сформованого варіанта плану з урахуванням зовнішніх та внутрішніх впливів на основі запропонованих часткових критеріїв та зазначених обмежень. Математична модель враховує усі умови діяльності харчового підприємства та процесу планування виконання замовлень: терміни зберігання сировини та готової продукції; можливість виникнення та необхідність переробки некондиційної продукції; особливості виконання кожного окремого замовлення; особливості використання технологічного обладнання тощо. Для формування узагальненої цільової функції використовується адитивна згортка критеріїв. Розроблена математична модель дозволяє сформулювати план виконання замовлень із урахуванням усіх операцій технологічного процесу при виготовленні продукції. Створена математична модель дозволяє коригувати та оцінювати ефективність виконання замовлень у залежності від об'єктивних та суб'єктивних переваг, наданих ОПР, а також забезпечує як

урахування, так і виключення певних часткових критеріїв у залежності від певної ситуації.

В роботі запропоновано модифікації алгоритмів «мурашиної колонії» та «зграї вовків», що дозволяє при їх застосуванні проводити реконфігурацію планів при зменшенні часу пошуку оптимуму. Модифікація алгоритму «мурашиної колонії» дозволяє скоротити час за рахунок використання підходів локальної оптимізації, а модифікація алгоритму «зграї вовків» – за рахунок виділення шаблонів із найкращих альтернативних рішень задачі.

Проведено дослідження застосування таких алгоритмів для розв'язання задачі планування виконання замовлень: алгоритм «бджолиної колонії», алгоритм «хаотичного кажана», алгоритм «кажана на основі стратегії пошуку польоту Леві», алгоритм «кажана на основі коефіцієнта скорочення», «генетичний алгоритм», алгоритм «косяку риб», алгоритм «зграї вовків», модифікований алгоритм «зграї вовків», алгоритм «мурашиної колонії з використанням елітних мурах», модифікований «алгоритм мурашиної колонії». При кількості замовлень менше 25 усі алгоритми знаходять однакові значення, але кращі показники за часом пошуку мають стандартні версії алгоритмів; при кількості замовлень 50 і більше кращі показники мають модифіковані алгоритми. За результатами дослідження запропоновано модифікації алгоритмів «мурашиної колонії» та «зграї вовків», які дозволяють при їх застосуванні проводити реконфігурацію планів при зменшенні часу пошуку оптимуму майже на 20% у порівнянні з іншими версіями алгоритмів.

Розроблено комбінований метод прийняття рішень на основі алгоритмів «косяку риб» та модифікованого алгоритму «зграї вовків», який стабільно працює на задачах малої та середньої розмірності й дає зменшення часу пошуку до 35% при умові врахування до 5–6 часткових критеріїв. За рахунок того, що загальна кількість ітерацій розподіляється між різними алгоритмами, відбувається загальне скорочення часу пошуку. Тобто, якщо для здійснення кожного алгоритму необхідно 100 операцій, то для усіх необхідно 200 операцій. На основі проведених експериментів було виявлено, що кількість ітерацій між алгоритмами доцільно розділити у пропорції 30/70. Для алгоритму «косяку риб» виділяється 30 ітерацій, а для

алгоритму «зграї вовків» – 70. Таке розподілення підібрано експериментально. У багатокритеріальних випадках комбінований метод забезпечує прискорення пошуку на 33,9% у порівнянні з модифікованим алгоритмом «зграї вовків», на 35,1% у порівнянні з алгоритмом «мурашиної колонії», на 33,8% у порівнянні з алгоритмом «косяку риб». Тому запропонований комбінований алгоритм доцільно застосовувати для підприємств із невеликим асортиментом та з мінімальним парком технологічного обладнання, адже він має більшу ефективність при пошуку обмежено критеріальних альтернативних оптимальних планів (до 5–6 часткових критеріїв оптимальності). Тобто, для підприємств, що використовують автоматизовані технологічні комплекси, які забезпечують серійне виготовлення продукції.

Розроблено комбінований метод на основі алгоритмів «мурашиної колонії», «зграї вовків» та «генетичного алгоритму», який стабільно працює на багатокритеріальних задачах різної розмірності й має швидкість пошуку на 10–45% більшу, ніж інші алгоритми. Для отримання первинного плану застосовуємо модифікований алгоритм «мурашиної колонії». Далі використовуємо модифікований алгоритм «зграї вовків», де обираємо як «альфа», «бета» та «дельта» вовків найкращі варіанти, отримані при застосуванні алгоритму «мурашиної колонії». Якщо за задану кількість ітерацій варіант плану не покращується, тоді застосовується «генетичний алгоритм». Якщо таке рішення знайдено, воно стає поточним, і починається нова ітерація. На кожній ітерації використання операцій мутації, кросовера та схрещування відбувається швидке отримання нових модифікованих варіантів планів виконання замовлень. Фактично застосування «генетичного алгоритму» забезпечує формування нових альтернативних планів на основі трьох найкращих. При повторному проході через усі алгоритми кількість ітерацій зменшується вдвічі для кожного алгоритму. Це продовжується до тих пір, поки не буде виконана задана кількість ітерацій. Таким чином, ми отримуємо нові варіанти, до яких застосовується алгоритм «мурашиної колонії». Поєднання трьох алгоритмів у один модифікований метод забезпечує знаходження складних варіантів виконання розкладу на багатоміноменклатурних підприємствах харчової галузі, що

підтверджено результатами випробування на ретроспективних даних за планами виконання замовлень за різні попередні періоди діяльності харчових підприємств. За інформацією, отриманою після проведення експериментів, є підстави стверджувати, що для багатокритеріальних задач комбінований метод забезпечує більш швидкий пошук, ніж усі інші алгоритми. Його доцільно застосовувати для підприємств, які виготовляють великий асортимент багатокомпонентних видів продукції і потребують складних технологічних операцій, розрізнених на певні ділянки виробництва.

Розроблена інформаційна технологія планування виконання замовлень, що забезпечує формування та оперативну реконфігурацію планів за короткий період часу за рахунок поєднання інтелектуального аналізу даних для попередньої класифікації, а також використання модифікованих алгоритмів та методів, які базуються на комбінуванні алгоритмів. Запропонована технологія дає можливість підібрати сукупність алгоритмів та методів, адекватну для певної предметної області. Використання інтелектуального аналізу даних забезпечує збільшення швидкості знаходження рішень за рахунок аналізу даних за попередні періоди, що дає змогу виявити найкращі послідовності використання конкретного технологічного обладнання для виготовлення продукції згідно з актуальними замовленнями. Застосування запропонованої технології дає змогу знайти додаткові альтернативні оптимальні плани виконання замовлень, а представлення результатів їх порівняння у вигляді наочної пелюсткової діаграми допомагає відповідальній особі швидко прийняти ефективне рішення.

Розроблено архітектуру гібридної web-орієнтованої системи підтримки прийняття рішень, яка реалізує запропоновану інформаційну технологію і базується на гнучкій інтеграції в «озері даних» інформації із статистичних та експертних джерел. До основних компонентів системи належать: «озеро даних», засоби завантаження даних, сховище даних, бази знань, серверна частина, web-інтерфейс користувача. Важливо зазначити, що інтерфейс користувача виконано у вигляді web-сторінок для відображення у браузері користувача. Джерелами даних для сховища даних слугують бази даних інформаційних систем та «озеро даних», що

забезпечують адекватність і достовірність інформаційного забезпечення для підтримки прийняття управлінських рішень. Накопичення інформації в «озері даних» та у сховищі даних відбувається в автоматичному режимі за рахунок використання засобів завантаження даних. Запропоновано обрати багаторівневу архітектуру сховища даних, яка забезпечує інтеграцію та узгодження даних із «озера даних» та інших джерел спочатку у «віртуальне сховище даних», а вже звідти буде завантажуватися у «вітрини даних».

Запропонована архітектура дає можливість поєднати декілька інформаційних технологій, а саме: технологію планування виконання замовлень, що забезпечує створення та коригування оперативно-календарних планів за рахунок використання модифікованих та комбінованих алгоритмів; підходи експертно-моделюючої системи рецептур виготовлення, що дає можливість підібрати у виробничих умовах заміну компонентів у рецептурі; технологію для удосконалення процесу аналізу та планування виконання замовлень на харчових підприємствах, яка, на відміну від відомих, ґрунтується на визначенні асортименту та коригуванні рецептур для зменшення собівартості та швидкого виконання замовлень.

При побудові програмної частини web-орієнтованої системи підтримки прийняття рішень використано класичну багатoshарову архітектуру з розподіленням на такі складові: шар контролерів, за допомогою яких забезпечується взаємодія з клієнтською частиною; адаптери, на рівні яких здійснюється перетворення даних із запитів клієнтів у основні бізнес-об'єкти, над якими здійснюється подальша обробка; класи, що здійснюють перевірки вхідних даних на відповідність встановленим правилам; бізнес-сервіси, які безпосередньо відповідають за виконання бізнес-логіки; допоміжні інструментальні класи, яким бізнес-сервіси делегують виконання окремих операцій; репозиторії, призначені для безпосередньої роботи з сховищем даних. Функціональні можливості серверної частини відповідають потребам користувачів web-орієнтованої системи підтримки прийняття рішень і у загальному вигляді розподілені на модулі, призначені для вирішення конкретних задач. Кожний модуль при виконанні обробки деякого запиту звертається для отримання даних до сховища даних або бази знань. Після

проведення обробки даних та необхідних підрахунків результат відображається клієнтською частиною у заданому вигляді. Кількість модулів не регламентується. Це забезпечує, у випадку необхідності, модернізацію існуючих та включення нових модулів для вирішення задач прийняття рішень. До основних модулів системи належать: модулі вирішення задач прийняття рішень та модулі вирішення задач із використанням інтелектуального аналізу даних, модулі підтримки функцій управління, які поєднують ведення статистичного обліку та створення звітів, модулі експертної системи.

Використання web-орієнтованої системи підтримки прийняття рішень, що буде включати запропоновану інформаційну технологію, забезпечить: оперативне формування планів виконання замовлень із мінімізацією витрат та направлених на максимізацію прибутку; оперативні реконфігурації існуючого календарного плану виконання замовлень, що забезпечить оптимальне використання технологічного обладнання і мінімізацію витрат при виникненні позаштатних ситуації під час появи термінових до виконання замовлень або позаштатних ситуацій; мінімізацію витрат на зберігання сировини та матеріалів; чітке розподілення усіх задач щодо виконання кожного замовлення між виробничими підрозділами, це дає можливість урахувати послідовність виконання та необхідні ресурси з прив'язкою у часі. Використання web-орієнтованої системи підтримки прийняття рішень, яка реалізує запропоновану інформаційну технологію, забезпечить підвищення техніко-економічних показників підприємства.

Формалізовано процес створення системи підтримки прийняття рішень із застосуванням спіральної моделі розробки програмного забезпечення, в якій, на додаток до традиційної, чітко визначено місце проведення рефакторингу, що дає можливість скорочення часу на розробку та підвищення ефективності розробки в цілому. Обґрунтовано застосування категорії «рефакторингу доступу», яка дає змогу впроваджувати зміни, пов'язані з доступом до даних, при адмініструванні сховища та «вітрин даних» без втрати працездатності системи. У процесі проєктування баз знань, вітрин та сховищ даних використано підходи щодо маркування та автоматизованого впровадження оновлення їх компонентів і елементів за рахунок

застосування категорії рефакторингу доступу та специфікації семантичного версіонування баз даних, що забезпечило підвищення якості та скорочення часу розробки.

Практичне значення роботи полягає в тому, що розроблені моделі, методи, алгоритми та інформаційна технологія формують основу для створення web-орієнтованих систем підтримки прийняття ефективних рішень при управлінні підприємствами харчової галузі, які дадуть змогу вивести весь процес управління на якісно новий рівень.

Результати дисертаційного дослідження пройшли апробацію та передані на ТОВ «Слобожанський бекон», ТОВ «Продеко», ТОВ «Козятинський м'ясокомбінат», ТОВ «Лайм Системс», ТОВ «Феракс», ПрАТ «Оболонь» для розв'язання задачі планування виконання замовлень, що підтверджено відповідними актами та довідками. За результатами промислової апробації керівництво підприємств надало рекомендації щодо впровадження інформаційної технології на підприємствах харчової галузі.

Результати дисертаційного дослідження впроваджено у навчальний процес Національного університету харчових технологій при викладанні дисципліни «Управління ІТ-проектами» для здобувачів освіти ступеня бакалавр за спеціальністю 122 – комп'ютерні науки освітньо-професійної програми «Комп'ютерні науки», дисципліни «Клієнт-серверні технології» для здобувачів освіти ступеня магістр за спеціальністю 122 – комп'ютерні науки освітньо-професійних програм «Інформаційні управляючі системи та технології» та «Комп'ютерний еколого-економічний моніторинг».

**Ключові слова:** управління харчовим підприємством, прийняття рішень, системи підтримки прийняття рішень, планування виконання замовлень, комбіновані алгоритми, комбіновані методи прийняття рішень, інформаційна технологія.

## ABSTRACT

*Hrybkov S.V.* Information technology of the decision making based on modified heuristic and evolutionary methods and algorithms for food companies. –Qualification scientific work as a manuscript.

Thesis for the degree of Doctor of Engineering Sciences in specialty 05.13.06 – information technology. – National University of Food Technology of the Ministry of Education and Science of Ukraine. – The Institute of Mathematical Machines and Systems Problems of the National Academy of Sciences of Ukraine, Kyiv, 2021.

The main result of the dissertation research is that there was found a solution to the scientific applied problem of increasing the efficiency of food company management processes through the development of an information technology to support decision making on the basis of modified heuristic and evolutionary methods and algorithms.

The thesis develops a mathematical model of the order execution planning process, which, in contrast to the traditional models, provides a breakdown by management levels and departments, as well as an assessment of the plan effectiveness, taking into account external and internal influences based on the proposed partial criteria and general constraints. The mathematical model takes into account all the conditions of the food company and the order execution planning process. These conditions are the following ones: the shelf life of raw materials and finished products; the possibility of occurrence and the need to process substandard products; features of execution of each order; features of using technological equipment, etc. An additive convolution of criteria is used to form a generalized objective function. The developed mathematical model allows to form an order execution plan, taking into account all operations of the technological process of products manufacturing. The created mathematical model allows to adjust and evaluate the efficiency of order execution depending on objective and subjective advantages of a person who makes decisions, and also provides both consideration and exclusion of certain partial criteria depending on the situation.

The paper proposes modifications of «ant colony» and «wolf pack» algorithms, the application of which allows reconfiguration of the plans while reducing the time necessary to find the optimum. Modification of the «ant colony» algorithm reduces time by using local optimization approaches, modification of the «wolf pack» algorithm – by selecting templates from the best alternative solutions to the problem.

A research of the application of the following algorithms aimed at solving the problem of order execution planning has been conducted: a «bee colony» algorithm, a «chaotic bat» algorithm, «a bat algorithm based on Levy flight search strategy», «a bat algorithm based on reduction factor», a «genetic algorithm», a «fish shoal» algorithm, a «wolf pack» algorithm, a modified «wolf pack» algorithm, «an ant colony algorithm using elite ants», a modified «ant colony» algorithm. When there are less than 25 orders, all algorithms have the same values and the best indicators of search time have standard versions of algorithms; when there are >50 orders, the best indicators have modified algorithms. The obtained results of the research indicated the need to use modifications of «ant colony» and «wolf pack» algorithms which allow to reconfigure the plans when reducing the search time for the optimum by almost 20 % compared to other versions of the algorithms.

A combined decision-making method based on the «fish shoal» and the modified «wolf pack» algorithms has been developed. This method works stably for small and medium-sized problems and reduces the search time to 35 %, taking into account 5-6 partial criteria. Due to the fact that the total number of iterations is distributed between different algorithms, there is a total reduction in the search time. That means that if each algorithm requires 100 operations, then all of them require 200 operations. As a result of the conducted experiments, it was found that the number of iterations between algorithms should be divided in the ratio 30:70. 30 iterations are allocated to the «fish shoal» algorithm, 70 – to the «wolf pack» algorithm. Such distribution was chosen experimentally. In multi-criteria cases, it provides search acceleration by 33,9 % compared to the modified «wolf pack» algorithm, by 35,1 % compared to the «ant colony» algorithm, by 33,8 % compared to the «fish shoal» algorithm. Therefore, the proposed combined algorithm should be used for companies with a narrow range of products and

with a minimum amount of process equipment, because it is more effective in finding limited-criteria alternative optimal plans (up to 5–6 partial optimality criteria). Such enterprises are the ones that use automated technological complexes that provide serial production.

A combined method based on «ant colony» and «wolf pack» algorithms, as well as on the «genetic algorithm» has been developed. This method stably works on multi-criteria problems of different size and has the search speed 10–45 % higher than other algorithms. A modified «ant colony» algorithm had been used with the aim of obtaining a primary plan. After that there was utilized a modified «wolf pack» algorithm where alpha, beta and delta wolves were used as the best options obtained in the result of the use of the «ant colony» algorithm. If the version of the plan does not improve for a given number of iterations, then the «genetic algorithm» is used. If such solution is found, it becomes current, and a new iteration begins. At each iteration of the use of mutation, crossover and crossbreeding operations, new modified versions of order execution plans are quickly obtained. In fact, the application of the «genetic algorithm» provides the formation of new alternative plans based on the best three ones. A repeated application of all the algorithms leads to halving the number of iterations for each algorithm. It continues until the specified number of iterations is performed. Thus appear new options, and the «ant colony» algorithm is applied to them. A combination of three algorithms in one modified method provides finding complex options for scheduling at multi-item food industry companies. It is confirmed by the results of tests conducted on retrospective data according to the order execution plans for different previous periods of food companies activity. According to the information obtained after conducting the experiments, there is a reason to believe that for multi-criteria problems the combined method provides a faster search than other algorithms. It is advisable to use for companies that manufacture a wide range of multicomponent products and require complex technological operations separated by certain areas of production.

An information technology for order execution planning, which provides the formation and operational reconfiguration of plans in a short period of time through a combination of data mining for pre-classification, as well as the use of modified

algorithms and methods based on a combination of algorithms, is developed in the thesis. The proposed technology makes it possible to select a set of algorithms and methods adequate for a particular subject area. The use of data mining provides an increase in the speed of finding solutions through data analysis for previous periods which allows to identify the best sequences of the use of specific technological equipment for the manufacture of products according to current orders. The application of this technology allows to find additional alternative optimal plans for the execution of orders, and the presentation of the results of their comparison in the form of a visual petal diagram helps the responsible person to make an effective decision quickly.

An architecture of a hybrid web-based decision support system has been developed. It implements the proposed information technology and is based on flexible integration in the data lake of information from statistical and expert sources. Data lake, data download tools, data warehouse, knowledge base, server part, web-user interface are the main components of the system. It is important to note that the user interface is displayed in the user's browser in the form of web-pages. Databases of information systems and the data lake are the sources of data for the data warehouse which ensure the adequacy and reliability of information support to support management decisions. The accumulation of information in the data lake and in the data warehouse is automatic due to the use of data download tools. It is proposed to choose a multi-level data warehouse architecture which provides integration and harmonization of data from the data lake and other sources first into a virtual data warehouse and then into the data showcase.

The proposed architecture makes it possible to combine the following information technologies: an order planning technology which provides the creation and adjustment of operational and calendar plans through the use of modified and combined algorithms; approaches of the expert-modeling system of manufacturing recipes that gives the chance to pick up in production conditions replacement of components in a compounding; a technology to improve the process of analysis and planning of orders at food companies which, in contrast to the known ones, is based on determining the range and adjusting recipes to reduce costs and provide faster execution of orders.

When building the software part of the web-oriented decision support system, the classic multilayer architecture is used with the division into the following components: a layer of controllers which provides interaction with the client part; adapters, at the level of which the data from customer requests is converted into the main business objects that are further processed; classes that check compliance of the input data with the established rules; business services that are directly responsible for the implementation of business logic; auxiliary tool classes to which business services delegate the execution of individual operations; repositories which are designed to work directly with the data warehouse. The functionality of the server part meets the needs of users of the web-oriented decision support system and is generally divided into modules designed to solve specific tasks. During the request processing, each module requests data to the data warehouse or to the knowledge base. When data processing and necessary calculations are carried out, the client part displays the result in the specified form. The number of modules is not regulated, and, if necessary, it provides the modernization of existing and the inclusion of new modules in order to solve decision-making problems. The main modules of the system are the following: decision-making modules and decision-making modules that use data mining, management functions support modules that combine statistical accounting and reporting, expert system modules.

The use of the web-oriented decision support system, which will include the proposed information technology, will provide the following issues: prompt formation of operational order execution plans aimed at cost minimization and profit maximization; prompt reconfiguration of the existing schedule of orders which will ensure optimal use of technological equipment and minimize costs in the event of emergency situations and subsequent urgent orders; minimization of costs for raw materials storage; a clear distribution of all tasks concerning the execution of each order between production units, which allows to take into account the sequence of execution and necessary resources with a timeline. The use of a web-based decision support system that implements the proposed information technology will improve the technical and economic performance of the enterprise.

The process of the creation of a decision support system using a spiral model of software development has been formalized. In addition to the traditional one, the place of refactoring in such a system is clearly defined. It generally allows to reduce development time and increase development efficiency. The use of the access refactoring category, which allows to implement changes related to data access in the administration of the repository and data showcases without loss of system performance, is justified in the thesis. In the process of designing knowledge bases, showcases and data warehouses, approaches to labeling and automated implementation of updating their components and elements through the use of access refactoring and specification of semantic versioning of databases were used. It resulted in quality improvement and reduction of development time.

The results of the dissertation research were tested and transferred to LLC «Slobozhansky Bacon», LLC «Prodeco», LLC «Kozyatyn Meat Processing Plant», LLC «Lime Systems», LLC «Ferax», PJSC «Obolon» with the aim of solving the problem of order execution planning. These acts are confirmed by the relevant documents and certificates. The executives of the stated companies took into account the results of industrial testing and provided recommendations for the implementation of the described information technology in food industry companies.

The results of the dissertation research are implemented into the educational process of the National University of Food Technologies in teaching the discipline «IT Project Management» for the students of bachelor programs in specialty 122 – computer science of the educational and professional program «Computer Science», discipline «Client-Server Technologies» for the applicants for a master's degree in specialty 122 – computer science of the educational and professional programs «Information control systems and technologies» and «Computer environmental and economic monitoring».

**Keywords:** food company management, decision making, decision support systems, order execution planning, combined algorithms, combined decision-making methods, information technology.

## СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

### Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Грибков С. В., Кононова В. О., Харкянен О. В. Оцінка засобів захисту інформаційних ресурсів. *Вісник Національного університету «Львівська політехніка». Комп'ютерні системи та мережі*. 2014. № 806. С. 99–105. Бази: *Google Scholar, Index Copernicus International*.

2. Грибков С. В., Загоровська Л. Г. Використання СА Erwin Model Manager для моделювання структури сховища даних. *Наукові праці НУХТ*. 2014. № 6, Т. 20. С. 125–130. Бази: *Index Copernicus, EBSCOhost, Google Scholar*.

3. Грибков С. В., Харкянен О. В., Логвин Т. В. Дослідження пакетів програмних модулів для ідентифікації динамічних об'єктів. *Технологічний аудит та резерви виробництва*. 2015. № 25 (5/2). С. 42–49. Бази: *EBSCO, Ifindr, Directory of Open Access Journals (DOAJ), EconBiz, IDEAS, ERIH PLUS (The European Reference Index for the Humanities and the Social Sciences), FSTA, OpenAIRE, Bielefeld Academic Search Engine (BASE), Index Copernicus Journals Master List, Наукова періодика України – проєкт Національної бібліотеки України імені В. І. Вернадського (НБУВ), Ulrich's Periodicals Directory, WorldCat, CrossRef*.

4. Грибков С. В., Литвинов В. В., Олійник Г. В. Задача планування виконання договорів та підходи до її ефективного вирішення. *Математичні машини і системи*. 2015. № 1. С. 61–70. Бази: *РИНЦ(eLIBRARY), CrossRef, Google Scholar, ВІНІТІ РАН*.

5. Hrybkov S. V., Oliiynk H. V. Modeling of the decision support system structure in the planning and controlling of contracts implementation. *Ukrainian Journal of Food Science*. 2015. N 3 (1). P. 123–130. Бази: *EBSCO, Chemical Abstracts Service Source Index (CASSI), Food Science and Technology Abstracts (FSTA), Index Copernicus International, Ulrichsweb, Directory of Open Access scholarly Resources (ROAD), Google Scholar*.

6. Грибков С. В., Харкянен О. В., Пархоменко І. І. Прогнозування стану технологічного обладнання харчового підприємства методом сингулярного спектрального аналізу. *Вісник інженерної академії України*. 2015. № 3. С. 87–91. *Бази: Innovative Journal Impact Factor (IJIF), Google Scholar*.

7. Грибков С. В., Харкянен О. В., М'якшило О. М. Моделювання процесу моніторингу та планування собівартості продукції багатоміноменклатурного харчового підприємства. *Наукові праці НУХТ*. 2015. № 6, Т. 21. С. 100–108. *Бази: Index Copernicus, EBSCOhost, Google Scholar*.

8. Грибков С. В., Загоровская Л. Г., Бондарь Н. П., Губеня В. А. Анализ и реинжиниринг процесса планирования закупок для предприятий ресторанного хозяйства. *Вестник Алмаатинского технологического университета*. 2015. №4 (109). С. 31-36. *Бази: CrossRef, РИНЦ(eLIBRARY)*.

9. Грибков С. В., Маковецька С. В., М'якшило О. М. Дослідження та математичне моделювання процесу постачання сировини на цукровий завод з врахуванням генетико-детермінованих властивостей цукрових буряків. *Наукові праці НУХТ*. 2016. № 6, Т. 22. С. 7–15. *Бази: Index Copernicus, EBSCOhost, Google Scholar*.

10. Грибков С. В., Литвинов В. А., Олійник Г. В. Обрання програмної платформи для побудови модуля безпеки web-орієнтованої системи підтримки прийняття рішень. *Вісник Національного університету «Львівська політехніка». Автоматика, вимірювання та керування*. 2016. № 852. С. 137–142. *Бази: РИНЦ(eLIBRARY), Google Scholar, Index Copernicus International*.

11. Грибков С. В., Павелко В. І., Заславський А. І., Дмитренко Д. С. Математичне моделювання процесу термічної обробки м'ясних виробів». *Харчова промисловість*. 2016. № 20. С. 115–120. *Бази: Google Scholar, Index Copernicus International*.

12. Грибков С. В., Олійник Г. В. Модифікований АСО алгоритм побудови календарного плану виконання договорів. *Математичне та комп'ютерне моделювання. Технічні науки*. 2017. № 15. С. 156–162. *Бази: Bielefeld Academic Search Engine (BASE), Citefactor, Cosmos Impact Factor, The Journals Impact Factor (JIF)*,

*General Impact Factor (GIF), Google Scholar, InfoBase Index, International Citation Index OF JOURNAL IMPACT FACTOR & INDEXING, Open Access Infrastructure for Research in Europe (OpenAIRE), Public Knowledge Project (PKP), ResearchBib, Scientific Indexing Services (SIS), WorldCat.*

13. Грибков С. В., Маковецька С. В. Функціональне моделювання організації та управління забезпечення сировиною цукрового заводу. *Науковий вісник Національного університету біоресурсів і природокористування України. Техніка та енергетика АПК.* 2017. № 286. С. 100–111. Бази: *Directory of Open Access Journals (DOAJ), CrossRef.*

14. Грибков С. В., Олійник Г. В. Використання JWT-маркерів для аутентифікації та авторизації користувачів у web-додатках. *Вісник Національного університету «Львівська політехніка». Автоматика, вимірювання та керування.* 2017. № 880. С 60–67. Бази: *РИНЦ(eLIBRARY), Google Scholar, Index Copernicus International.*

15. Hrybkov S. V., Breus N. M., Polischuk G. Ye. Hybrid expert system to model the ice cream recipes. *Ukrainian Journal of Food Science.* 2017. № 5 (2). P. 294–304. Бази: *EBSCO, Chemical Abstracts Service Source Index (CASSI), Food Science and Technology Abstracts (FSTA), Index Copernicus International, Ulrichsweb, Directory of Open Access scholarly Resources (ROAD), Google Scholar.*

16. Hrybkov S. V., Kharkianen O. V., Myakshylo O. M., Kostikov M. P. Development of information technology for supporting the process of adjustment of the food enterprise assortment. *Eastern-european journal of enterprise technologies.* 2018. № 3 (91). P. 77–87. Бази: *Scopus, CrossRef, Index Copernicus International, Applied Science & Technology Source, Chemical Abstracts Plus (CAplus), Computers & Applied Sciences Complete, Directory of Open Access Journals (DOAJ), Google Scholar, Directory of Open Access scholarly Resources (ROAD), MIAR, Ifindr, OpenAIRE (Open Access Infrastructure for Research in Europe), Polska Bibliografia Naukowa (PBN), WorldCat, Scientific Periodicals of Ukraine (SPU VNLU), Bielefeld Academic Search Engine (BASE), China National Knowledge Infrastructure (CNKI), Scilit, WorldWideScience.org, Zeitschriftendatenbank (ZDB), ResearchBib, EuroPub,*

*JournalTOCs, Dimensions, Scientific Indexing Services (SIS), The General Impact Factor (GIF), Genamics JournalSeek, SHERPA RoMEO, Ulrich's Periodicals Directory, CORE (COncnecting REpositories), Neliti, Socionet, Researchgate, ERIH PLUS, Lens, Korea Open Access Platform for Researchers (KOAR), Microsoft Academic, Paperity, Root Society, Wizdom.ai, KindCongress.*

17. Грибков С. В., Струзік В. А., Литвин А. О. Дослідження методів і підходів проведення рефакторингу баз даних. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2018. № 30. С. 151–155. Бази: РИИЦ(eLIBRARY), *Universal Impact Factor, Open Academic Journals Index.*

18. Hrybkov S. V., Litvinov V. V., Oliinyk H. V. Web-oriented decision support system for planning agreements execution. *Eastern-european journal of enterprise technologies*. 2018. № 2 (93). P. 13–24. Бази: *Scopus, CrossRef, Index Copernicus International, Applied Science & Technology Source, Chemical Abstracts Plus (CAplus), Computers & Applied Sciences Complete, Directory of Open Access Journals (DOAJ), Google Scholar, Directory of Open Access scholarly Resources (ROAD), MIAR, Ifindr, OpenAIRE (Open Access Infrastructure for Research in Europe), Polska Bibliografia Naukowa (PBN), WorldCat, Scientific Periodicals of Ukraine (SPU VNLU), Bielefeld Academic Search Engine (BASE), China National Knowledge Infrastructure (CNKI), Scilit, WorldWideScience.org, Zeitschriftendatenbank (ZDB), ResearchBib, EuroPub, JournalTOCs, Dimensions, Scientific Indexing Services (SIS), The General Impact Factor (GIF), Genamics JournalSeek, SHERPA RoMEO, Ulrich's Periodicals Directory, CORE (COncnecting REpositories), Neliti, Socionet, Researchgate, ERIH PLUS, Lens, Korea Open Access Platform for Researchers (KOAR), Microsoft Academic, Paperity, Root Society, Wizdom.ai, KindCongress.*

19. Грибков С. В., Бойко Р. О. Мережеві структури при керуванні складними організаційно-технічними (технологічними) системами. *Харчова промисловість*. 2019. № 25. С. 116–123. Бази: *Google Scholar, Index Copernicus International.*

20. Струзік В. А., Грибков С. В., Чобану В. В. Визначення місця рефакторингу в сучасних методологіях розробки інформаційних систем. *Вчені*

записки Таврійського національного університету імені В. І. Вернадського. Технічні науки. 2019. № 5, Т. 30 (69). С. 173–177. Бази: РИИЦ(eLIBRARY), Index Copernicus International.

21. Hrybkov S. V., Breus N. M., Seidykh O. L., Polischuk G. Ye. Development of mathematical apparatus of the expert system for modelling ice cream recipes with specified quality parameters. *Science and innovation*. 2019. Vol. 15 (5). P. 57–66. Бази: Web of Science, Scopus, Directory of Open Access Journals (DOAJ), CrossRef, EuroPub, Ulrich's Periodicals Directory, Index Copernicus International.

22. Hrybkov S. V., Kharkianen O. V., Ovcharuk V. O., Ovcharuk I. Development of information technology for planning order fulfillment at a food enterprise. *Eastern-european journal of enterprise technologies*. 2020. N 3 (103). P. 62–73. Бази: Scopus, CrossRef, Index Copernicus International, Applied Science & Technology Source, Chemical Abstracts Plus (CAplus), Computers & Applied Sciences Complete, Directory of Open Access Journals (DOAJ), Google Scholar, Directory of Open Access scholarly Resources (ROAD), MIAR, Ifindr, OpenAIRE (Open Access Infrastructure for Research in Europe), Polska Bibliografia Naukowa (PBN), WorldCat, Scientific Periodicals of Ukraine (SPU VNLU), Bielefeld Academic Search Engine (BASE), China National Knowledge Infrastructure (CNKI), Scilit, WorldWideScience.org, Zeitschriftendatenbank (ZDB), ResearchBib, EuroPub, JournalTOCs, Dimensions, Scientific Indexing Services (SIS), The General Impact Factor (GIF), Genamics JournalSeek, SHERPA RoMEO, Ulrich's Periodicals Directory, CORE (COncnecting REpositories), Neliti, Socionet, Researchgate, ERIH PLUS, Lens, Korea Open Access Platform for Researchers (KOAR), Microsoft Academic, Paperity, Root Society, Wizdom.ai, KindCongress.

23. Грибков С. В., Литвинов В. В., Олійник Г. В. Інструментальна модель веб-орієнтованої програмної реалізації підсистеми підтримки прийняття рішень у складі програмного комплексу ситуаційного центру. *Математичні машини і системи*. 2020. № 1. С. 73–81. Бази: РИИЦ(eLIBRARY), CrossRef, Google Scholar, ВІНІТИ РАН.

24. Грибков С. В., Струзік В. А., Чобану В. В. Категорія рефакторинг доступу. *Наукові праці НУХТ*. 2020. № 2, Т. 26. С. 31–49. Бази: *Index Copernicus, EBSCOhost, Google Scholar*.
25. Грибков С. В., Струзік В. А., Чобану В. В. Специфікація семантичного версіонування бази даних. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Технічні науки*. 2020. № 2, Т. 31 (70). С. 196–201. Бази: *РИНЦ(eLIBRARY), Index Copernicus International*.
26. Грибков С. В., Сєдих О. Л. Розробка СППР управління виробництвом. *Almanahul SWorld*. 2020. № 4 (1). С. 52–56. Бази: *Google Scholar, Index Copernicus International*.
27. Hrybkov S. V., Seidykh O. L. Manufacturing management methods at the expense of using bat algorithm. *Modern engineering and innovative technologies*. 2020. Issue 12 (3). P. 68–73. Бази: *Google Scholar, Index Copernicus International*.
28. Hrybkov S. V., Seidykh O. L. Hybrid algorithm based on fish school search and grey wolf optimizer algorithms for food enterprise management. *European science review*. Vienna: Premier Publishing s.r.o., 2020. N 7–8. P. 19–22. Бази: *РИНЦ(eLIBRARY), CrossRef, CyberLeninka, EBSCOhost, Google Scholar, Index Copernicus International*.
29. Hrybkov S. V., Struzik V. A., Chobanu V. V. Evolution of refactoring. *The Austrian Journal of Technical and Natural Sciences*. Vienna: Premier Publishing s.r.o., 2020. N 7–8 P. 11–16. Бази: *РИНЦ(eLIBRARY), CrossRef, CyberLeninka, EBSCOhost, Google Scholar, OpenAIRE, Ulrich's Periodicals Directory, WorldCat*.
30. Hrybkov S. V., Kharkianen O. V., Gladka Y. U. Finding the best versions of schedule for order fulfillment at food companies. *MIND journal Semiannual*. Poland : Bielsko-Biała. 2020. N 9. P. 1–12. Бази: *Index Copernicus International, BuzHul*.
31. Hrybkov S. V., Seidykh O. L. Solution of control tasks using bat algorithm modifications. *Modern engineering and innovative technologies*. 2020. Issue 13 (2). P. 60–64. Бази: *Google Scholar, Index Copernicus International*.

32. Грибков С. В., Сєдих О. Л. Розробка модифікованих ітераційних алгоритмів для розв'язання задачі формування оптимальних варіантів розкладу виконання замовлень. *Харчова промисловість*. 2020. № 27. С. 126–137. *Бази: Google Scholar, Index Copernicus International*.

### **Наукові праці, які засвідчують апробацію матеріалів дисертації:**

33. Грибков С. В., Олійник Г. В. Високопродуктивні пакети завантаження даних до сховища даних. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 81 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 23–24 квітня 2015 р.). Київ: НУХТ, 2015. Ч. 2. С. 356.

34. Грибков С. В., Нерадович М. С., Олійник Г. В. Використання методу бджолиного рою при формуванні оптимального розкладу виконання замовлень. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами*: матеріали II міжнар. наук.-техн. конф. Київ : НУХТ, 2015. С. 194.

35. Грибков С. В., Воловик О. О. Web-орієнтована інформаційна система підтримки контролю якості сировини та готової продукції. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами*: матеріали III міжнар. наук.-техн. конф. Київ: НУХТ, 2016. С. 219.

36. Hrybkov S. V., Ollinyk H. V., Litvinov V. V. Decision support system for the contracts execution planning based on genetic algorithms and collective mind algorithms. *Food Science for Well-being (CEFood 2016)*: Book of Abstracts of 8th Central European Congress on Food (Kyiv, 23–26 May 2016). Kyiv: NUFT, 2016. P. 58.

37. Грибков С. В., Буряченко М. М., Харкянен О. В. Використання еволюційних методів для формування виробничої програми харчового підприємства. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 83 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 5–6 квітня 2017 р.). Київ, 2017. С. 285.

38. Грибков С. В., Буряченко М. М. Використання середовища R для аналізу та виявлення причин виникнення некондиційної продукції у макаронному виробництві. *Обчислювальний інтелект (результати, проблеми, перспективи)*: матеріали 4 міжнар. наук.-практ. конф. (м. Київ, 16–18 травня 2017 р.). Київ-Черкаси: ВПЦ «Київський університет», 2017. С. 204–205.

39. Грибков С. В., Олійник Г. В. Розробка web-орієнтованої системи підтримки прийняття рішень при плануванні виконання договорів. *Автоматика–2017*: матеріали XXIV міжнар. конф. з автоматичного управління (м. Київ, 13–15 вересня 2017 р.). Київ: НУБіП, 2017. С. 192–193.

40. Грибков С. В., Олійник Г. В. Web-орієнтована система підтримки прийняття рішень планування виконання договорів. *Проблеми інформатизації*: міжнар. наук.-техн. конф.: зб. тез. Київ: ДУТ, 2017. С. 239.

41. Грибков С. В., Струзік В. А., Литвин А. О. Використання еволюційного підходу при рефакторингу баз даних. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами*: матеріали IV міжнар. наук.-техн. Internet-конф. (м. Київ, 22 листопада 2017 р.). Київ: НУХТ, 2017. С. 281.

42. Hrybkov S. V., Struzik V. A., Lytvyn A. O., Chobanu V. V. Study Of The Methods And Approaches Of Databases Refactoring. *Програмовані логічні інтегральні схеми та мікропроцесорна техніка в освіті і виробництві*: зб. тез міжнар. наук.-практ. семінару молодих вчених та студентів. Луцьк: Вежа-Друк, 2018. С. 12–13.

43. Грибков С. В., Ольшевська М. А. Підсистема підтримки діяльності начальника виробництва макаронного підприємства ТОВ «Продеко». *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 84 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 23–24 квітня 2018 р.). Київ: НУХТ, 2018. Ч. 2. С. 337.

44. Грибков С. В., Фурта О. О. Використання інтелектуального аналізу даних для планування виготовлення продукції на ТОВ «Слобожанський бекон». *Проблеми інформатизації*: матеріали XII міжнар. наук.-техн. конф. Київ: ДУТ,

НТУ; Полтава: ПНТУ; Катовице: КЕУ; Париж: Університет Париж VII Венсент-Сен-Дені; Вільнюс: ВДТУ; Харків: ХНДІТМ; Білорусь: БДАЗ, 2018. С. 28–29.

45. Грибков С. В., Олійник Г. В., Чернишов І. Використання модифікованих методів мурашиного алгоритму для розв'язку задачі формування розкладу. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті: матеріали 85 Ювілейної міжнар. наук. конф. молодих учених, аспірантів і студентів, присвяченої 135-річчю НУХТ (м. Київ, 11–12 квітня 2019 р.)*. Київ : НУХТ, 2019. Ч. 2. С. 435.

46. Грибков С. В., Ольшевська М. А. Створення інформаційної системи підтримки управління виробництва макаронних виробів. *LXXV наук. конф. професорсько-викладацького складу, аспірантів, студентів та співробітників відокремлених структурних підрозділів університету: зб. тез*. Київ: НТУ, НУХТ, 2019. Т. 2. С. 462.

47. Грибков С. В., Фурта О. О. Підсистема підтримки прийняття рішення головного технологу ТОВ «Слобожанський бекон». *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами: матеріали VI міжнар. наук.-техн. Internet-конф. (м. Київ, 20 листопада 2019 р.)*. Київ: НУХТ, 2019. С. 176.

48. Грибков С. В., Чорнобай К. Ю. Дослідження та проектування підсистеми управління молочним балансом молокозаводу. *Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій: наук. праці Другої міжнар. наук.-практ. конф. (м. Київ, 19 грудня 2019 р.)*. Київ: НУХТ, 2019. С. 298–302.

49. Грибков С. В., Сєдих О. Л., Ольшевська М. А. Дослідження та розроблення СППР управління виробництвом ТОВ «Продеко». *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті: матеріали 86 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 2–3 квітня 2020 р.)*. Київ: НУХТ, 2020. Ч. 2. С. 403.

50. Hrybkov S. V., Seidykh O. L. Combined algorithm based on algorithms for koryake fish and gray wolves for solving complex problems. *Sworld-Us conference proceedings: conf. proceed. of Global scien. and educ. in the modern real.* Washington, USA: «ISE&E» & SWorld in conjunction with KindleDP Seattle, 2020. P. 54–57.

**Наукові праці, які додатково відображають наукові результати дисертації:**

51. Грибков С. В., Балашева А. М. Дослідження і створення веб-орієнтованої СППР при управлінні розвитку компанії ТОВ «Доктор Філін». *Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій: матеріали I міжнар. наук.-практ. конф. (м. Краків, 7–8 лютого 2019 р.)*. Краків, Польща, 2019. С. 120–123.

52. Грибков С. В., Литвинов В. В., Олійник Г. В. До проблеми типізації проектних рішень у сфері програмної реалізації підсистем ситуаційного центру. *Цифрова економіка та інформаційні технології: матеріали Міжнар. наук.-практ. конф. (м. Київ, 15–16 квітня 2020 р.)*. Київ: ДУІТ, 2020. С. 37–38.

53. Грибков С. В., Литвинов В. В., Олійник Г. В. Инструментальная модель подсистемы поддержки принятия решений. *Математичне та імітаційне моделювання систем: матеріали XV міжнар. наук.-практ. конф. (м. Чернігів, 29 червня – 01 липня 2020 р.)*. Чернігів, 2020. С. 212–214.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....</b>	<b>4</b>
<b>ВСТУП.....</b>	<b>5</b>
<b>РОЗДІЛ 1 АНАЛІЗ СВІТОВОГО ДОСВІДУ СТОСОВНО УДОСКОНАЛЕННЯ ПІДХОДІВ ТА МЕТОДІВ УПРАВЛІННЯ ХАРЧОВИМ ПІДПРИЄМСТВОМ..</b>	<b>19</b>
1.1 Дослідження сучасних підходів удосконалення управління харчовим підприємством.....	19
1.2 Дослідження інформаційних технологій в управлінні харчових підприємств....	35
1.2.1 Огляд основних типів інформаційних систем для підтримки управління підприємством.....	38
1.2.2 Системи підтримки прийняття рішень.....	46
1.3 Огляд сучасних методів для оперативного планування замовлень.....	50
1.4 Постановка завдань досліджень і розробок.....	61
1.5 Висновки до розділу 1.....	63
<b>РОЗДІЛ 2 ФОРМАЛІЗАЦІЯ ЗАДАЧ УПРАВЛІННЯ.....</b>	<b>66</b>
2.1 Створення математичної моделі планування виконання замовлень.....	66
2.2 Створення математичної моделі планування постачання сировини на прикладі цукрового заводу.....	85
2.3 Створення математичної моделі складання розкладу виконання замовлень та забезпечення балансу сировини й готової продукції для молокозаводу.....	91
2.4 Висновки до розділу 2.....	93
<b>РОЗДІЛ 3 ДОСЛІДЖЕННЯ МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ ТА РОЗРОБКА МОДИФІКОВАНИХ АЛГОРИТМІВ.....</b>	<b>96</b>
3.1 Алгоритм «бджолиної колонії».....	96
3.2 Алгоритм «кажанів» та його модифікації.....	101
3.2.1 Алгоритм «хаотичного кажана».....	105
3.2.2 Алгоритм «кажана на основі стратегії пошуку польоту Леві».....	106
3.2.3 Алгоритм «кажана на основі коефіцієнта скорочення».....	107

3.3 «Генетичний алгоритм».....	107
3.4 Алгоритм пошуку «косяку риб».....	114
3.5 Алгоритм «зграї вовків».....	118
3.5.1 Класичний алгоритм «зграї вовків».....	118
3.5.2 Модифікований алгоритм «зграї вовків».....	124
3.6 Алгоритм «мурашиної колонії».....	129
3.6.1 Алгоритм «мурашиної колонії» з використанням елітних мурах.....	129
3.6.2 Модифікований алгоритм «мурашиної колонії».....	134
3.7. Порівняння обраних методів та рекомендації щодо їх застосування.....	146
3.8 Висновки до розділу 3.....	154
<b>РОЗДІЛ 4 СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПЛАНУВАННЯ ВИКОНАННЯ ЗАМОВЛЕНЬ НА ОСНОВІ КОМБІНОВАНИХ МЕТОДІВ ТА АЛГОРИТМІВ.....</b>	<b>157</b>
4.1 Комбінований метод прийняття рішень на основі алгоритмів «косяку риб» і модифікованого «зграї вовків».....	157
4.2 Комбінований метод прийняття рішень на основі «мурашиної колонії», «зграї вовків» та «генетичного алгоритму».....	164
4.3 Розробка інформаційної технології для планування виконання замовлень.....	169
4.4. Висновки до розділу 4.....	197
<b>РОЗДІЛ 5 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ.....</b>	<b>200</b>
5.1 Проектування архітектури гібридної web-орієнтованої системи підтримки прийняття рішень.....	200
5.2 Обґрунтування підходів розробки СППР та використання рефакторингу.....	210
5.3 Проектування та створення елементів баз знань та сховищ даних.....	220
5.4 Обрання та обґрунтування технологій розробки.....	228
5.5 Забезпечення аутентифікації та авторизації користувачів до СППР.....	233
5.6. Результати впровадження досліджень.....	248
5.7 Висновки до розділу 5.....	261
<b>ВИСНОВКИ.....</b>	<b>264</b>

<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>268</b>
<b>ДОДАТКИ.....</b>	<b>303</b>
Додаток А Акти впровадження.....	304
Додаток Б Список публікацій здобувача за темою дисертації.....	315
Додаток В Функціональні моделі.....	325
Додаток Д Структура СППР.....	343
Додаток Е Математична модель процесу термічної обробки м'ясних виробів....	344
Додаток Ж Моделі «вітрин даних».....	351
Додаток З Фрагменти SQL-кодів для створення «вітрин даних» .....	358

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СППР – система підтримки прийняття рішень.

ІТ – інформаційні технології.

ІС – інформаційна система.

ОПР – особа, що приймає рішення.

СУБД – система управління базами даних.

СД – сховище даних.

БД – база даних.

БЗ – база знань.

СД – сховище даних.

ЗПР – задача прийняття рішень.

ІАД – інтелектуальний аналіз даних.

ГА – «генетичний алгоритм».

OLAP – On-Line Analytical Process.

ABC – Artificial Bee Colony – алгоритм «бджолоїної колонії».

ACO – Ant Colony Optimization – алгоритм «мурашиної колонії».

FSS – Fish School Search – алгоритм на основі «косяку риб».

GWO – Grey Wolf Optimizer – алгоритм «зграї вовків»/«сірих вовків».

EMC – експертно-моделююча система.

MES – Manufacturing Execution Systems.

APS – Advanced Planning & Scheduling.

SCM – Supply Chain Management.

## ВСТУП

**Актуальність теми.** Харчова промисловість є однією із стратегічних галузей у будь-якій країні, адже вона забезпечує її продовольчу незалежність. Ринок харчової промисловості характеризується дуже високою конкуренцією, зокрема, між вітчизняними та зарубіжними виробниками. В умовах жорсткої конкуренції перед підприємствами харчової галузі, які намагаються працювати за довгостроковими договорами, гостро стоїть задача збереження споживача, що можливо тільки за рахунок виготовлення якісної продукції за прийнятною ціною та надання послуг за короткий термін. Для цього необхідно забезпечити закупку та зберігання сировини, яка має обмежені терміни використання, виготовити продукцію з дотриманням усіх необхідних технологічних вимог, забезпечити зберігання виготовленої продукції. В цих умовах управління підприємством повинно забезпечувати гнучке та швидко реагуюче виробництво з максимальним урахуванням зовнішніх та внутрішніх факторів і обмежень при плануванні виготовлення продукції і прийнятті оперативних рішень.

Вагомий внесок у розвиток та вдосконалення алгоритмізації процесів прийняття рішень та управління виробництвом зробили вітчизняні та зарубіжні вчені, зокрема: Сергієнко І. В., Ладанюк А. П., Павлов О. А., Згуровський М. З., Зайченко Ю. П., Мельников О. В., Мисюра Е. Б., Щербатенко О. В., Конвей Р. В., Максвелл В. Л., Міллер Л. В., Domschke W., Scholl A., Vob S., Pinedo M. L та багато інших.

Більшість технологічних і виробничих процесів у харчовій промисловості складно описувати кількісно, що у сукупності з багатокритеріальністю задач планування і управління ускладнює використання традиційних детермінованих математичних методів для прийняття відповідних рішень. Це викликає необхідність застосування нових інтелектуальних методів для вирішення задач управління з урахуванням оперативної інформації від фахівців-експертів відносно надання якихось переваг альтернативам у процесі вибору рішень.

Застосуванням відповідних евристичних та еволюційних методів для вирішення багатокритеріальних оптимізаційних задач займалися Dorigo M., Лагунова А. Д., Mirjalili S., Lewis A., Сагун А. В., Хайдуров В. В., Кунченко-Харченко В. И., Stutzle Th., Hoos Н. Н., Khan L., Ullah I., Saeed T., Filho B., Neto L., Holland J. H., Кажаров А. А., Курейчик В. М., Karaboga D., Yang X. S., Wang G., Guo L., Штовба С. Д., Сперкач М. О., Желдак Т. А. та ін. Більшість авторів у своїх роботах зазначає переваги і порівняльну ефективність застосування евристичних і еволюційних методів для розв'язання багатокритеріальних задач.

Опубліковані результати в цій сфері носять більшою частиною загальний характер і не прив'язані до оптимізації процесів прийняття рішень, пов'язаних з особливостями управління саме об'єктами харчової промисловості.

Важливо відзначити, що галузева проблема підвищення ефективності управління харчовими підприємствами вимагає комплексного рішення, що в сукупності має включати проблемно-орієнтовані моделі, методи і алгоритми, управлінські технології і відповідне інформаційно-програмне забезпечення.

Отже, актуальною науково-прикладною проблемою є розробка і реалізація інформаційних технологій прийняття рішень при управлінні харчовим підприємством на основі модифікованих евристичних і еволюційних методів та алгоритмів із метою забезпечення скорочення часу на прийняття рішень, зменшення впливу людського фактора та зниження витрат і часу на виконання замовлень щодо виготовлення продукції харчовим підприємством.

**Зв'язок роботи з науковими програмами, планами, темами.** У Національному університеті харчових технологій відповідно до державних програм і планів НДР: НДР «Наукове обґрунтування та розроблення активних пакувальних систем харчових продуктів» (Проблемна науково-дослідна лабораторія НУХТ, № ДР 0118U003558, 2019 р.); НДР «Дослідження та впровадження інформаційних технологій у галузях харчової промисловості та освіти» (кафедра інформаційних систем, № ДР 0117U003475, 2017–2022 рр.); НДР «Математичні методи аналізу комп'ютеризованих систем» (Національний університет харчових технологій, кафедра інформатики, № ДР 0117U003477, 2017–2020 рр.); НДР «Дослідження та

використання сучасних інформаційних технологій для виконання функцій та завдань виробничого і організаційного управління підприємств харчової галузі» (Національний університет харчових технологій, кафедра інформатики, № ДР 0120U105386, 2020–2025 рр.). Робота виконувалась згідно із планом науково-дослідних робіт кафедри інформаційних систем Національного університету харчових технологій «Розроблення нових інформаційних технологій прийняття рішень у виробництві та освітній діяльності» 2011–2015 рр.; «Інформаційні технології прогнозування погоди для сільського господарства», 2017 р.; «Дослідження та впровадження інформаційних технологій у галузях харчової промисловості та освіти» протягом 2015–2020 рр.

**Мета і завдання дослідження.** Метою дисертаційної роботи є підвищення ефективності процесів управління харчовим підприємством шляхом розробки інформаційних технологій підтримки прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів.

Для досягнення мети вирішувалися такі основні завдання:

- провести аналіз існуючих підходів, методів та інформаційних технологій для управління харчовим підприємством;
- здійснити математичне моделювання задачі планування виконання замовлень з урахуванням зовнішніх та внутрішніх впливів;
- розробити модифіковані методи та алгоритми для вирішення задачі управління харчовим підприємством;
- розробити інформаційні технології на основі модифікованих евристичних і еволюційних методів та алгоритмів для здійснення гнучкого управління харчовим виробництвом;
- розробити архітектуру гібридної web-орієнтованої системи підтримки прийняття управлінських рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів;
- розробити методичні рекомендації щодо практичної реалізації web-орієнтованої системи підтримки прийняття управлінських рішень;

– здійснити апробацію створеної інформаційної технології у промислових умовах на діючих підприємствах, обраних як базові.

*Об'єктом дослідження* є процеси прийняття рішень при управлінні харчовим підприємством в умовах невизначеності та ризиків.

*Предметом дослідження* є методи, технології та інструментальні засоби удосконалення процесу прийняття рішень при управлінні харчовим підприємством на основі модифікованих евристичних і еволюційних методів та алгоритмів.

**Методи дослідження.** Основними методами дослідження в цій роботі є дотримання загальнонаукових методологій і принципів системного підходу.

Для розв'язання поставлених задач використовувались такі методи:

– методи математичної статистики для обробки та оцінки ефекту застосування алгоритмів, методів та інформаційних технологій;

– методи функціонального та об'єктно-орієнтованого аналізу для дослідження предметних областей;

– методи інтелектуального аналізу даних для аналізу та класифікації замовлень;

– методи об'єктно-орієнтованого проєктування, використані для створення інформаційної технології та елементів системи;

– теорія реляційних та багатовимірних моделей даних при проєктуванні баз знань, вітрин та сховища даних;

– методи прийняття рішень, методи багатокритеріального аналізу та прийняття рішень, еволюційні методи і евристичні алгоритми для розроблення інформаційної технології прийняття рішень;

– методи, засоби і технології сучасного прикладного програмування для побудови практичних реалізацій.

Таким чином, дисертаційна робота припускає використання як теоретичного, так і емпіричного методу дослідження. Теоретичною основою роботи є наукові роботи провідних вітчизняних і зарубіжних учених в області прийняття рішень при управлінні харчовими підприємствами, програмної інженерії, імітаційного моделювання, проєктування систем. Емпіричною основою роботи стали методи

порівняння, вимірювання, спостереження, експерименту, аналізу, направлені на модифікацію існуючих методів та алгоритмів і які було покладено в інформаційну технологію.

**Наукова новизна отриманих результатів.** Винесені на захист наукові результати є теоретичною основою для розв'язання важливої науково-прикладної проблеми розроблення та практичного застосування методів і моделей прийняття рішень при управлінні харчовим виробництвом в умовах зміни режимів роботи, характеристик сировини та функцій оперативного персоналу для забезпечення регламентованих характеристик продуктів та підвищення ефективності багатоасортиментного виробництва. Науково-практичне значення роботи полягає у теоретичному обґрунтуванні, розробленні та практичному застосуванні нової інформаційної технології на основі модифікованих евристичних і еволюційних методів та алгоритмів.

Основні результати, які визначають наукову новизну роботи:

Вперше:

1) розроблено математичну модель процесу планування виконання замовлень, яка, на відміну від традиційних, що передбачають розбиття її за рівнями управління та підрозділами виконання, забезпечує комплексну оцінку ефективності сформованого варіанта плану виконання замовлень з урахуванням зовнішніх та внутрішніх впливів на основі запропонованих часткових критеріїв та зазначених обмежень;

2) розроблено новий метод прийняття рішень для розв'язання задачі планування виконання замовлень на основі комбінування алгоритмів «косяку риб» та «зграї вовків», що, на відміну від кожного з оригінальних алгоритмів, дає можливість уникнути локальних оптимумів та скоротити час пошуку обмежено критеріальних альтернативних оптимальних планів (до 5–6 часткових критеріїв оптимальності), час виконання замовлень та/або витрати на виконання; його доцільно застосовувати для підприємств з невеликим асортиментом продукції, що виготовляється на автоматизованій лінії;

3) розроблено новий метод прийняття рішень для розв'язання задачі

планування виконання замовлень на основі комбінування алгоритмів «мурашиної колонії», «зграї вовків» та «генетичного алгоритму», що, на відміну від кожного з оригінальних алгоритмів, дає можливість уникнути локальних оптимумів і поліпшити характеристики планування при пошуку багатокритеріальних альтернативних оптимальних планів; його доцільно застосовувати для підприємств, які виготовляють великий асортимент багатокомпонентних видів продукції і потребують складних технологічних операцій, розрізнених за певними ділянками виробництва;

4) розроблено інформаційну технологію для розв'язання задачі планування виконання замовлень за заданою сукупністю критеріїв оптимальності, яка ґрунтується на запропонованих комбінованих методах і формуванні початкових альтернативних стартових значень із використанням методів інтелектуального аналізу даних, що забезпечує усунення невизначеності та зменшення ризику при прийнятті оптимальних рішень;

5) розроблено архітектуру гібридної web-орієнтованої системи підтримки прийняття рішень, яка реалізує запропоновану інформаційну технологію і базується на гнучкій інтеграції в «озері даних» інформації із статистичних та експертних джерел.

Вдосконалено:

6) алгоритм «мурашиної колонії», який, на відміну від його класичної форми, при застосуванні для розв'язання задачі планування виконання замовлень дозволяє скоротити час за рахунок використання підходів до локальної оптимізації;

7) алгоритм «зграї вовків», який, на відміну від його класичної форми, при застосуванні для розв'язання задачі планування виконання замовлень дозволяє скоротити час за рахунок використання виділення шаблонів із найкращих альтернативних рішень задачі.

Отримало подальший розвиток:

8) підхід створення систем підтримки прийняття рішень із використанням спіральної моделі розробки програмного забезпечення, в якій, додатково до традиційної, чітко визначено місце проведення рефакторингу, що дає можливість

скоротити час на розробку.

**Практичне значення отриманих результатів** полягає в тому, що розроблені методи, алгоритми та моделі є основою для створення гібридних web-орієнтованих систем підтримки прийняття рішень для вирішення задач управління харчовими підприємствами, що дає змогу вивести весь процес управління на якісно новий рівень. Запропонована інформаційна технологія забезпечує виконання замовлень із такими перевагами: оперативно формує оперативно-календарний план виконання замовлень із мінімізацією витрат, що направлений на максимізацію прибутку; дозволяє оперативно коригувати існуючий календарний план замовлень, що дає можливість реагувати на замовлення в реальному часі та забезпечувати оптимальне використання технологічного обладнання; значно збільшує ефективність використання сировини та матеріалів, а також забезпечує мінімізацію витрат на їх зберігання; забезпечує швидке реагування при виникненні негативних та позаштатних ситуацій шляхом внесення відповідних змін до поточного плану виконання замовлень; чітко розподіляє усі задачі щодо виконання кожного замовлення між виробничими підрозділами, що дає можливість врахувати послідовність виконання та необхідні ресурси із прив'язкою в часі; дозволяє оптимізувати використання виробничих потужностей.

Результати дисертаційного дослідження впроваджено:

- у ТОВ «Слобожанський бекон» для оптимізації формування календарно-оперативних планів виконання замовлень (акт впровадження 16.12.2019);
- у ТОВ «Продеко» для удосконалення методів управління в умовах ризику та невизначеності (акт впровадження 10.12.2019);
- у ТОВ «Козятинський м'ясокомбінат» для прийняття рішень при розв'язанні задачі планування виконання замовлень (акт апробації 20.10.2020);
- у ТОВ «Лайм Системс» для розв'язання задачі планування виконання замовлень (акт апробації 18.01.2021);
- у ТОВ «Феракс» для прийняття рішень на основі модифікованих методів та алгоритмів при розв'язанні задачі планування виконання замовлень (акт апробації 27.01.2021);

– у ПрАТ «Оболонь» для планування виконання замовлень та при вирішенні задач управління підприємством (акт впровадження 18.02.2021);

– у Національному університеті харчових технологій при виконанні НДР «Наукове обґрунтування та розроблення активних пакувальних систем харчових продуктів» (№ ДР 0118U003558, 2019 р.) проблемною науково-дослідною лабораторією НУХТ, у навчальному процесі при підготовці магістрів за спеціальністю 122 – комп’ютерні науки (акт впровадження 30.06.2019);

– у Національному університеті харчових технологій при виконанні НДР «Дослідження та впровадження інформаційних технологій у галузях харчової промисловості та освіти» (№ ДР 0117U003475, 2017–2021 рр.) кафедрою інформаційних систем, у навчальному процесі при підготовці бакалаврів та магістрів спеціальності 122 – комп’ютерні науки (акт впровадження 29.12.2020).

Інформаційна технологія прийняття рішень та відповідний програмний продукт передані у Київський міський Радіоклуб Ліги Радіоаматорів України і використовуються для вирішення організаційно-управлінських задач при підготовці та проведенні змагань (довідка 26.09.2020).

Акти та довідки про апробацію і впровадження представлені у додатку до дисертації.

**Особистий внесок здобувача.** До дисертації увійшли наукові результати, отримані здобувачем особисто. З наукових праць, опублікованих у співавторстві, в дисертації використано лише ті ідеї та положення, які є результатом особистої роботи здобувача.

У спільних публікаціях автору належать такі результати:

[1] аналіз особливостей захисту інформаційних ресурсів у корпоративних мережах та системах, а також описаний підхід щодо їх оцінки;

[2] запропонований підхід швидкої розробки моделі корпоративного сховища даних із використанням СА Erwin Model Manager;

[3] досліджена та запропонована структура пакета програмного модуля для ідентифікації динамічних об’єктів, а також визначено його місце при перевірці на стаціонарність динамічного об’єкта ідентифікації;

[4, 18, 48] розроблені математичні моделі управління;

[5, 26, 35, 36, 39, 40, 43, 47, 49, 51] спроектовані й розроблені структура, програмні елементи і модулі системи підтримки прийняття рішень та проведено їх апробацію;

[6] запропонований метод прогнозування стану технологічного обладнання харчового підприємства на основі методу сингулярного спектрального аналізу;

[7] розробка та аналіз окремих діаграм моделі процесу моніторингу і планування собівартості продукції багатоміноменклатурного харчового підприємства;

[8] дослідження процесу закупок;

[9] розроблення основних елементів математичної моделі постачання сировини на цукровий завод з урахуванням генетико-детермінованих властивостей та запропоновані підходи до вирішення задачі складання послідовності переробки;

[10] проведення дослідження основних проблем захисту web-орієнтованої системи підтримки прийняття рішень під час планування виконання договорів для підприємств, а також здійснений порівняльний аналіз програмних платформ для реалізації модуля захисту таких систем;

[11] аналіз етапів процесу виготовлення м'ясних виробів та розроблена математична модель даного процесу;

[12] проведений аналіз та обробка даних при застосуванні модифікованого АСО алгоритму для вирішення задачі побудови календарного плану виконання договорів для підприємств, діяльність яких пов'язана з наданням послуг;

[13] проведене дослідження процесу оперативного управління постачанням сировини на цукровому заводі та виділено ключову задачу управління формування розкладу постачання сировини з урахуванням оптимізації процесів прийому, зберігання та переробки сировини;

[14] дослідження особливостей механізмів автентифікації та авторизації у web-орієнтованих системах, а також аналіз особливостей та переваг використання JWT-маркерів;

[15] розроблена структура експертної системи для моделювання рецептур багатокomпонентних харчових продуктів для застосування її при плануванні виготовлення продукції;

[16] аналіз та апробація інформаційної технології, яка включає в себе адаптовані методи інтелектуального аналізу даних для підтримки процесу коригування асортименту продукції багатомноменклатурного харчового підприємства;

[17] проведено дослідження та аналіз методів і підходів щодо проведення рефакторингу;

[18, 23, 52, 53] запропонована сукупність програмно-технологічних рішень, що входять до структури web-орієнтованої СППР;

[19] проведено аналіз та апробацію розроблених нечітких ситуаційних мереж для систем підтримки прийняття рішень;

[20] аналіз застосування рефакторингу в популярних моделях життєвого циклу програмного забезпечення;

[21] проведений аналіз застосування математичного апарата пошуку оптимальної рецептури багатокomпонентних харчових виробів із збереженням їх харчової цінності та можливості застосування їх у виробничих умовах;

[22] запропонована інформаційна технологія, яка ґрунтується на комбінуванні алгоритмів «мурашиної колонії», «сірих вовків» та «генетичного алгоритму» для розв'язання задачі планування виконання замовлень щодо виготовлення продукції на харчових підприємствах в умовах невизначеності та ризику; розроблена математична модель виконання замовлень для харчового підприємства;

[24] проведений аналіз застосування категорії рефакторингу баз даних для створення корпоративних інформаційних систем;

[25] аналіз необхідності використання специфікації семантичного версіонування баз даних, проведена класифікація баз даних за способом доступу до даних;

[27] проведений аналіз застосування різних модифікацій алгоритму кажанів для розв'язання задач управління на харчових підприємствах;

[28, 37, 45, 50] розроблені гібридні та модифіковані алгоритми для вирішення задач управління;

[29] формалізація процесу створення програмних продуктів, а також рефакторингу;

[30] проведено дослідження щодо використання модифікацій алгоритму «кажанів» для розв'язання задач управління харчовим підприємством, проведений аналіз результатів;

[31] проведено дослідження алгоритмів «кажанів» та їх модифікацій з адаптацією для розв'язання багатокритеріальної задачі формування виробничих завдань та оперативно-календарних планів на харчових підприємствах, обґрунтування обрання алгоритму «кажанів» на основі стратегії пошуку польотів Levy;

[32] розроблена математична модель багатокритеріальної задачі формування виробничих завдань та оперативно-календарних планів із виробництва ковбасних і м'ясних виробів, а також удосконалені ітераційні алгоритми на основі алгоритму «сірих вовків» та алгоритму «мурашиної колонії» для розв'язання задачі формування оптимальних варіантів розкладу виконання замовлень;

[33] розроблені алгоритми та методи завантаження інформації до сховища даних із різнорідних джерел;

[34] розроблений підхід до формування оптимального розкладу виконання замовлення з використанням методу «бджолиного рою»;

[38] розроблені модулі аналізу та виявлення причин виникнення некондиційної продукції при виготовленні макаронних виробів;

[41] аналіз використання еволюційного підходу при створенні баз даних;

[42] дослідження існуючих операцій рефакторингу баз даних;

[44] розроблено сховище даних і пакети Data Mining для визначення асортименту та планування виготовлення обсягів м'ясної продукції;

[46] запропоновано використання методики BSP (Business Systems Planning – системи організаційного планування) та виконання її запровадження в

інформаційній системі підтримки управлінських рішень при виробництві макаронних виробів.

**Апробація результатів дисертації.** Результати досліджень та розробок, викладених у дисертаційній роботі, доповідалися та обговорювалися на наукових конференціях і семінарах:

Міжнародній науковій конференції «Нові ідеї в харчовій науці – нові продукти харчової промисловості» (м. Київ, 2014 р.);

- 81-й міжнародній науковій конференції молодих учених, аспірантів і студентів «Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті» (м. Київ, 2015 р.);

- II-й міжнародній науково-технічній конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами» (м. Київ, 2015 р.);

- III-й міжнародній науково-технічній конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами» (м. Київ, 2016 р.);

- 8th Central European Congress on Food 2016 – Food Science for Well-being (CEFood, 2016) (м. Київ, 2016 р.);

- 83-й міжнародній науковій конференції молодих учених, аспірантів і студентів «Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті» (м. Київ, 2017 р.);

- IV-й міжнародній науково-практичній конференції «Обчислювальний інтелект (результати, проблеми, перспективи)» (м. Київ, 2017 р.);

- VI-й міжнародній науково-технічній конференції «Захист інформації й безпека інформаційних систем» (м. Львів, 2017 р.);

- XXIV-й міжнародній конференції з автоматичного управління «Автоматика – 2017» (м. Київ, 2017 р.);

- Восьмій міжнародній науково-технічній конференції «Проблеми інформатизації» (м. Київ, 2017 р.);

- IV-й міжнародній науково-технічній Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (м. Київ, 2017 р.);
- Міжнародному науково-практичному семінарі молодих учених та студентів «Програмовані логічні інтегральні схеми та мікропроцесорна техніка в освіті і виробництві» (м. Луцьк, 2018 р.);
- 84-й міжнародній науковій конференції молодих учених, аспірантів і студентів «Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті» (м. Київ, 2018 р.);
- XII-й міжнародній науково-технічній конференції «Проблеми інформатизації» (м. Київ, Полтава, Катовице, Париж, Вільнюс, Харків, 2018 р.);
- Міжнародній науково-практичній конференції «Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій» (м. Краків, Польща, 2019 р.);
- Міжнародній науковій конференції молодих учених, аспірантів і студентів «Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті» (м. Київ, 2019 р.);
- LXXV-й науковій конференції професорсько-викладацького складу, аспірантів, студентів та співробітників відокремлених структурних підрозділів університету (м. Київ, 2019 р.);
- VI-й міжнародній науково-технічній Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (м. Київ, 2019 р.);
- Другій міжнародній науково-практичній конференції «Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій» (м. Київ, 2019 р.);
- 86-й міжнародній науковій конференції молодих учених, аспірантів і студентів «Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті» (м. Київ, 2020 р.);
- Міжнародній науково-практичній конференції «Цифрова економіка та інформаційні технології» (м. Київ, 2020 р.);

- XV-ій міжнародній науково-практичній конференції «Математичне та імітаційне моделювання систем» (м. Чернігів, 2020 р.);
- Global science and education in the modern realities, Conference proceedings «SWorld-Us conference proceedings», «ISE&E» & SWorld in conjunction with KindleDP (Seattle, Washington, USA, 2020).

**Публікації.** Основні теоретичні та експериментальні результати наукових досліджень висвітлено у 53 наукових працях, із них: 32 статті у наукових іноземних виданнях та у виданнях України, які включені до міжнародних наукометричних баз [1–32] (з яких 4 індексовані у наукометричній базі Scopus [16, 18, 21, 22], 1 – Web of Science [21], 7 статей у зарубіжних періодичних виданнях [8, 26–31], 25 статей у наукових фахових виданнях України [1–7, 9–25, 32]); 21 публікація у збірниках праць конференцій [33–53].

**Структура та обсяг роботи.** Дисертаційна робота складається зі вступу, п'яти розділів, висновків, списку використаних джерел із 361 найменувань та 7 додатків. Загальний обсяг дисертації 395 сторінок, у тому числі 267 сторінок основного тексту. Дисертація містить 38 рисунків та 11 таблиць.

## РОЗДІЛ 1

### АНАЛІЗ СВІТОВОГО ДОСВІДУ СТОСОВНО УДОСКОНАЛЕННЯ ПІДХОДІВ ТА МЕТОДІВ УПРАВЛІННЯ ХАРЧОВИМ ПІДПРИЄМСТВОМ

#### 1.1 Дослідження сучасних підходів удосконалення управління харчовим підприємством

Сучасний виробник повинен постійно удосконалювати свої методи управління, адже втрата свого споживача на ринку збуту рівносильна банкрутству підприємства. Споживач постійно змінює свої потреби під впливом економічних, соціальних та політичних обставин. Виробник не тільки повинен забезпечити виготовлення якісної продукції, а також задовольнити вимоги споживача щодо продукції, при цьому забезпечивши мінімально можливу собівартість продукції. Отже сучасні виробники постійно працюють в умовах ризику та невизначеності.

Сьогодні на ринку залишаються тільки ті підприємства, які орієнтовані на якісну роботу в умовах постійної конкуренції та нестабільної економічної ситуації. Такі підприємства повинні швидко реагувати на запити ринку, які виникають за рахунок ефективного управління налагодженими процесами проектування, виробництва, постачання та підтримки свого продукту на ринку. Це досягається грамотним застосуванням різних інструментів сучасного менеджменту та сучасних інформаційних технологій.

Ефективність управління полягає у прийнятті управлінських рішень для подальшої реалізації передбачених цими рішеннями керуючих впливів на певний об'єкт управління [54–55].

До основних завдань відносяться: ефективна організація управління, більш повне використання всіх видів ресурсів, задіяння методів стимулювання праці. Вирішенню цих завдань може допомогти досвід освоєння передових методів управління виробництвом розвинених промислових країн.

Бережливе виробництво – напрям менеджменту, що забезпечує конкурентоспроможність підприємства за рахунок випуску продукції (надання

послуг) в кількості, необхідній замовнику, з високою якістю, мінімальними витратами ресурсів і низькою собівартістю [56, 57].

Концепція бережливого виробництва почала формуватися в Японії. Засновником концепції вважається Таїті Воно, який почав роботу в компанії Toyota Motor Corporation у 1943 р. Опанувавши досвід передових світових промислових країн, він у середині 1950-х рр. почав вибудовувати систему організації виробництва, що отримала назву «Виробнича система Toyota» (TPS – Toyota Production System). У західних країнах концепція TPS отримала назву Lean Production. Термін Lean Production запропонований Джоном Крафчиком, який був науковим співробітником Массачусетського інституту. Фахівцями в області організації виробництва використовуються терміни Лін-технології (Lean-технології), економне виробництво та ін. На даний час все більше застосування знаходить термін «Ощадливе виробництво» [56–62].

Впровадження і застосування концепції «ощадливого виробництва» дозволяє [57]:

- знизити вартість продукції на 50%;
- скоротити тривалість виробничого циклу на 50%;
- скоротити трудовитрати на 50% при одночасному збереженні або підвищенні продуктивності;
- збільшити виробничі потужності на 50% на тих же площах;
- скоротити складські запаси на 80%;
- підвищити якість продукції;
- збільшити прибуток;
- створити гнучку виробничу систему, що дозволяє швидко реагувати на зміну запитів споживачів.

Цілі «ощадливого виробництва» досягаються за рахунок зниження або усунення втрат у процесі виробництва товарів.

Втратами вважаються всі дії, які не створюють цінності для споживача. При виготовленні продукції цінність для споживача створюється тільки безпосередньо

при опрацюванні та складанні виробів, всі інші дії, наприклад, зберігання, транспортування та інші, знижують цінність [59].

На японській мові втрати називаються словом «муда» (muda). Таїті Оно (Taiichi Ohno) виділив сім видів втрат муда [63, 64]:

1. Втрати через перевиробництво – виробництво товарів, які не користуються попитом; виробництво продукції в більшому обсязі раніше або швидше, ніж це потрібно на наступному етапі процесу.

2. Втрати часу через очікування – перерви в роботі, пов'язані з очікуванням людей, матеріалів, устаткування або інформації.

3. Втрати через зайві обробки – додаткова обробка виробу через низьку якість інструменту, помилки проектування та ін.

4. Втрати через зайві рухи при виконанні операцій – переміщення людей, інструменту або обладнання, що не додає цінності кінцевому продукту.

5. Втрати через зайві запаси – надмірне надходження продукції в виробничий процес, чи то сировина, напівфабрикат або готовий продукт.

6. Втрати при транспортуванні – непотрібні переміщення або переміщення на великі відстані матеріалів, деталей, продукції.

7. Втрати через випуск дефектної продукції - продукції, що вимагає перевірки, сортування, утилізації, заміни або доопрацювання.

У даний час додають ще один вид втрат: втрати через невикористаний потенціал персоналу – втрати часу, ідей, навичок, можливостей вдосконалення і набуття досвіду співробітниками.

Стратегія «Кайдзен» вимагає безперервного вживання заходів щодо вдосконалення за участю всіх працівників даної організації – в рівній мірі і менеджерів, і робітників. Поняття «кайдзен» з'явилося в Японії. Воно утворено двома словами: «кай» – зміна і «дзен» – на краще. Безперервна зміна малими кроками, які не вимагають значних вкладень – ось сенс, який містить у собі поняття «кайдзен». Термін «кайдзен» запропонував Масаакі Імаї (Masaaki Imai) – засновник концепції безперервного вдосконалення. Різні автори виділяють різну кількість

ключових принципів, на яких ґрунтується «кайдзен». При цьому зазвичай в їх число включають такі принципи [65, 66]:

- фокусування на клієнтах – для компанії, що використовує «кайдзен», найбільш важливо, щоб їх продукція (послуги) задовольняла потреби клієнтів;
- безперервні зміни – принцип, що характеризує саму суть «кайдзен», тобто безперервні малі зміни у всіх сферах організації: постачанні, виробництві, збуті, взаємовідносинах тощо;
- відкрите визнання проблем – всі проблеми відкрито виносяться на обговорення;
- пропаганда відкритості – мала ступінь відокремленості між відділами і робочими місцями;
- утворення команд робочих – усі працівники стають членами конкретної робочої команди та відповідного гуртка якості;
- управління проектами за допомогою міжфункціональних команд – жодна команда не буде працювати ефективно, якщо вона діє тільки в одній функціональній групі. З цим принципом тісно пов'язана притаманна японському менеджменту ротація персоналу;
- формування «підтримки взаємовідносин» – для організації є важливими не тільки фінансові результати, а й залученість працівників до її діяльності та толерантні взаємини між працівниками, оскільки це завжди (нехай і не в певному звітному періоді) приведе організацію до високих результатів;
- удосконалення самодисципліни – уміння поважати та контролювати як себе, так і інших працівників та всю організацію;
- інформування кожного співробітника – весь персонал повинен бути повністю інформаційно обізнаний про свою компанію.

Особливість стратегії «Кайдзен» полягає в тому, що діяльність щодо поліпшення планується і виконується безпосередньо на робочих місцях. У зв'язку з цим «кайдзен» слугує інструментом залучення персоналу в діяльність щодо поступової зміни вигляду виробництва [66].

Ефективність впровадження технологій бережливого виробництва залежить від активної участі всіх працівників підприємства, починаючи від вищого керівництва і закінчуючи безпосередньо виконавцями на робочих місцях.

Система швидкого переналагодження обладнання – це інструмент бережливого виробництва, представляє собою набір теоретичних і практичних методів, які дозволяють скоротити час операцій налагодження і переналагодження обладнання. Слід підкреслити, що у переважній більшості випадків застосування даної системи дозволяє значно скоротити час переналагодження, однак вона не може гарантувати скорочення часу всіх процесів налагодження до десяти хвилин і менше. У свою чергу, скорочення часу переналагодження дає підприємству безліч переваг. Швидке переналагодження обладнання відповідає терміну SMED (Single Minute Exchange of Dies – швидка заміна штампів). Початково ця система була розроблена для того, щоб оптимізувати операції заміни штампів і переналагодження відповідного обладнання, проте принципи швидкого переналагодження можна застосовувати до усіх типів процесів [56–58].

Швидке переналагодження демонструє дійсно новий погляд на процес переналагодження. Система SMED ґрунтується як на теорії, так і на багаторічній експериментальній практиці. Вона являє собою науковий підхід до скорочення часу переналагодження, який можна застосувати на будь-якому підприємстві і будь-якому обладнанні. Переналагодження – процес підготовки обладнання до переходу від виробництва одного виду продукції до іншого (наприклад, штампувального преса, машини для лиття або конвеєра) шляхом заміни деталей, прес-форм, матриць, затискних пристосувань тощо. Час переналагодження – це проміжок між завершенням виробництва останнього виробу попередньої партії до виходу з виробництва першого придатного виробу після переналагодження. Існують два фундаментально різні типи операцій переналагодження: внутрішні та зовнішні [56, 61, 65, 66].

Внутрішні операції по переналагодженню – це операції, які можна робити тільки на відключеному обладнанні (установка і зняття штампів). Зовнішні операції

по переналагодженню – це дії, які можна виконувати без відключення обладнання (доставка нових штампів до пресу, підготовка елементів кріплення тощо).

Для успішного впровадження системи SMED необхідно на попередньому етапі створити робочу групу із впровадження, у яку повинні входити представники всіх зацікавлених служб підприємства, а на завершальному етапі необхідно провести стандартизацію отриманих результатів, розробити стандартні операційні процедури для повсюдного впровадження даного інструмента на підприємстві.

Поява і вдосконалення технології швидкого переналагодження нерозривно пов'язана з розвитком концепції «бережливого виробництва», а точніше виробничої системи компанії Toyota. Саме на заводах цієї компанії в 1969 р. були зроблені перші радикальні кроки щодо скорочення часу переналагодження обладнання. Для скорочення часу переналагодження необхідно не тільки розділити процедури на внутрішні і зовнішні, але й спробувати перетворити якомога більше робіт по внутрішній переналадці в зовнішню. Саме це рішення дозволяє істотно підвищити ефективний час використання обладнання шляхом скорочення його зупинок на переналагодження [63, 64].

Сутність технології швидкого переналагодження набагато глибше, ніж чисто технічне рішення по швидкій зміні інструмента або оснастки, так як саме вона лежить в основі високої гнучкості і клієнтської орієнтованості організації, що відповідає принципам «бережливого виробництва» і здатна без втрат виробляти продукцію малими партіями. Зі зменшенням питомих витрат на переналагодження знижуються витрати, пов'язані із зберіганням запасів, і обсяг партії виробів.

Освоєння технології швидкого переналагодження має такі переваги [57, 62]:

1. Дозволяє стерти грані між дрібносерійним і крупносерійним виробництвом (з точки зору витрат часу на переналагодження в розрахунку на один виріб).
2. Забезпечує гнучкість і орієнтованість виробництва на замовлення клієнта.
3. Скорочує такі втрати, як перевиробництво (запаси напівфабрикатів і готової продукції); транспортування і переміщення надлишкових запасів; очікування (простої устаткування і операторів при переналадці); випуск браку (при переналадці і регулюванні обладнання).

4. Дає можливість підвищити ефективність виробництва за рахунок скорочення втрат; випуску продукції дрібними партіями, тобто «під замовлення»; скорочення потреби в висококваліфікованих працівниках.

Одним напрямом удосконалення управління є теорія обмежень системи, який включає підхід до аналізу витрат, що базується на обліку пропускнуої здатності системи, а не на традиційному обліку витрат [67–69]. Практичні аспекти перетворень на будь-якому рівні організації, що дозволяють як виявляти явні і приховані проблеми з використанням аналітичних інструментів теорії обмежень, так і усувати ці проблеми за допомогою проривних рішень, описано в [69–71].

Основними перевагами теорії обмежень є [71]:

- швидкий ефект;
- відсутність потреби у значних інвестиціях;
- зниження обсягу запасів і коштів, заморожених у них;
- підвищення рівня логістичного обслуговування;
- підвищення гнучкості підприємства та його еластичності;
- системний підхід, що дозволяє досягати позитивних ефектів одночасно в декількох сферах;
- скорочення виробничих циклів;
- покращення фінансових результатів підприємства;
- зниження ймовірності збоїв;
- здобуття тривалих конкурентних переваг;
- знаходження принципово нових шляхів розв'язання;
- управління логістичними ризиками, зокрема, можливість передбачення небажаних явищ ще до їх появи та розробки заходів щодо їх запобігання або мінімізації;
- універсальність – унікальність розроблених рішень та їх адаптація під конкретні умови тощо.

Варто зазначити, що досягнення таких значних результатів за допомогою теорії обмежень системи є можливим за рахунок [71] концентрації зусиль на тих сферах, де є можливим отримання швидких ефектів; забезпечення ефективних

технік, що уможливають постійний процес вдосконалення системи. Ефективність застосування теорії обмежень систем на практиці підтверджується тим, що більшість великих компаній уже протягом багатьох років застосовують підходи щодо теорії обмежень системи у своїй діяльності. Варто наголосити, що теорія обмежень системи є джерелом здобуття підприємством довготермінових конкурентних переваг, а це створює передумови для постійного розвитку підприємства в майбутньому [71].

Харчові підприємства України тільки частково використовують світовий досвід з удосконалення управління. Основними напрямками удосконалення є впровадження системи якості НАССР (Hazard Analysis and Critical Control Points). Використання НАССР забезпечує захист підприємства та його торгової марки при просуванні на ринку харчових продуктів та захист виробничих процесів від біологічних (мікробіологічних), хімічних, фізичних та інших ризиків забруднення. Впровадження системи НАССР здійснюється на державному рівні й вимагає дотримання усіх вимог якості до готового продукту, а також усіх етапів його виготовлення. В Україні вимоги щодо розробки та впровадження систем управління безпечністю харчової продукції за принципами НАССР задекларовані ДСТУ 4161-2003 «Система управління безпечністю харчових продуктів. Вимоги» та ДСТУ ISO 22000:2007 «Системи управління безпечністю харчових продуктів. Вимоги до будь-яких організацій харчового ланцюга» [72, 73].

Діяльність будь-якого промислового підприємства можна умовно розділити на дві частини: перша – це безпосередньо виробничий процес, друга – фінансово-економічна діяльність підприємства. Вимоги до інформаційних систем із фінансово-економічної діяльності мають свою специфіку. Виробнича діяльність великого підприємства містить безліч технологічних циклів. При цьому споживаються різні види сировини (як вихідний, так і проміжний), що вимагає контроль технологічних ланцюжків на всіх етапах виробництва. Виникнення збою в технологічному циклі може призвести до тяжких фінансових збитків, а також призводити до виходу з ладу обладнання. Відповідно контроль повинен здійснюватися в реальному часі і безперервно, що висуває вимоги до продуктивності автоматизованих інформаційно-

керуючих систем, гарантії якості послуг і їх надійності. Втім, надійність і захищеність систем не в меншій мірі потрібні і для фінансово-економічної діяльності, так як обсяг вхідних і вихідних фінансових потоків, а також циркулюючих в середині підприємства, дуже великий [74–76].

Ієрархія складної автоматизованої системи управління визначається необхідністю структурування управління у складній системі з метою отримання обмеженого конструктивного рішення з ряду можливих рішень, з яких і обирається найкраще. Саме це рішення потім реалізується в системі у вигляді децентралізованих і узгоджених управлінь різного рівня відповідальності.

Необхідно відзначити, що ієрархічність як властивість системи призводить до наступних принципових особливостей системи. По-перше, система складається з сукупності субпідрядних підсистем різних рівнів ієрархії. По-друге, підсистеми більш високого рівня ієрархії використовують при прийнятті рішень агреговані (узагальнені) координати, які є функціями вихідних координат підсистем нижчого рівня ієрархії і формують директивні управління для цих підсистем. Недоступність повного вектора стану підсистем нижнього рівня ієрархії для підсистеми верхнього рівня є принциповою особливістю ієрархічної системи управління. Важливим тут є те, що верхній рівень ієрархії формулює рішення в термінах агрегованих координат для нижнього рівня. Рішення завдання на верхньому рівні не повністю визначає стан системи, адже воно сформульовано в термінах агрегованих координат.

Для того, щоб визначити вектор стану вихідної системи повністю, використовується нижній рівень управління. Мета управління на нижньому рівні формується в термінах вихідних змінних, але управління залишається тим, що визначено на верхньому рівні в термінах узагальнених координат. Це означає, що рішення підсистеми верхнього рівня обов'язкові для виконання на нижньому рівні ієрархії. Внаслідок цього, ієрархічна архітектура системи управління завжди звужує клас допустимих управлінь як за рахунок роботи з агрегованими координатами, так і за рахунок структурних обмежень. Як виняток частина допустимих управлінь є принциповою особливістю ієрархічної організації системи управління [77–80].

Розподіленість як властивість складної системи забезпечує найкраще узгодження топології системи управління із принципами організаційно-технологічного управління територіально і функціонально розподілених об'єктів управління і виключає в системі циркуляцію надлишкової інформації при її паралельній асинхронній обробці в реальному масштабі часу. При цьому найбільш раціонально забезпечується доступність для кожного споживача попередньо підготовленої і відформатованої інформації.

Розподілені системи управління є ергатичними системами з різним функціональним призначенням.

Участь людини у процесі управління призводить до безлічі особливостей системи і вимагає (на стадії проєктування системи) вирішення завдань технічної ергономіки з метою створення найбільш комфортних умов для виконання людиною – оператором функціональних завдань в умовах інформаційних і психофізіологічних перевантажень при дефіциті часу на прийняття рішень. В умовах функціонування ергатична система управління повинна постійно ідентифікувати фізіологічний стан людини-оператора і його здатність вирішувати поставлені функціональні завдання. При тимчасовому перевантаженні людини-оператора і, як наслідок, нездатністю людиною виконувати функціональні завдання в повному обсязі ергатична система повинна передавати частину функціональних завдань на рішення засобам автоматки з метою узгодження поведінкових, технологічних і організаційно-економічних аспектів управління. Перерозподіл функціональних завдань у процесі управління виникає при необхідності збереження керованості системи під час інформаційних і психофізіологічних перевантаженнях людини і може призводити до деякого зниження якості управління [81–84].

Участь людини у процесі управління вимагає необхідної кваліфікації і навичок управління обслуговуючого персоналу. Завдання підтримки необхідної кваліфікації персоналу вирішується в ергатичних системах введенням у систему управління професійних тестів і контрольних завдань, що імітують аварійні і позаштатні ситуації в системі. Аналіз професійних дій людини – оператора, що виконує ці завдання, здійснюється системою управління і документується. На основі

цього аналізу для кожного учасника управління формуються нові тести і контрольні завдання, які враховують помилки, допущені саме цим оператором [81–84].

Розглянутий клас складних систем управління функціонує, як правило, в умовах неповноти і недостовірності інформації про координати і параметри системи, невизначеності деяких оцінок і показників. Це вимагає розробки системи управління у вигляді інтелектуальної системи, в якій поєднуються інтелект професіонала-оператора і штучний інтелект експертної системи, яка є частиною інтелектуальної системи. Поєднання інтелектуальних елементів в одній системі приводить до необхідності вирішення проблеми: кому у відповідній ситуації, що склалася, слід віддати перевагу – експертній системі або оператору [84, 85]. Вирішення цієї проблеми пов'язано з оцінкою кваліфікації оператора, що здійснює управління системою. Якщо кваліфікація виконавця висока, то експертна система видає рекомендації, які враховуються оператором при ухваленні рішення щодо управління системою. Система управління виступає у цьому випадку як система прийняття рішень, в іншому випадку прийняття рішення здійснює експертна система. З огляду на те, що в системах управління працюють професіонали високої кваліфікації, системи управління проєктуються і виконуються як системи прийняття рішень.

Розглянуті системи управління відносяться до безперервно-дискретних систем, вектор стану яких може змінюватися «стрибком» у дискретні моменти часу. Миттєва зміна вектора стану системи може бути викликана як появою дискретної події, так і при виконанні певних умов при взаємодії безперервних координат. Зміна вектора стану в залежності від деякого параметра, що визначає характеристики системи, визначається як процес. Для розглянутого класу систем (динамічних безперервно-дискретних систем) як такий параметр обирається час. З огляду на це зауваження, процес управління визначається як орієнтована в часі кінцева або нескінченна послідовність дискретних подій, розділена на неперервні проміжки часу. Процес завжди асоціюється з будь-яким об'єктом, що входить у систему, і виступає як засіб функціональної декомпозиції системи.

Для аналізу діяльності підприємства харчової галузі та визначення вузьких місць використано системний аналіз, який є фундаментальним при вивченні складних систем [86–88].

При прийнятті управлінських рішень і прогнозуванні можливих результатів особа, що приймає рішення, зазвичай наштовхується на складну систему взаємопов'язаних компонент (ресурси, бажані наслідки або цілі), яку необхідно проаналізувати. І чим більше ОПР вникає в суть цієї складності, тим краще будуть прогнози чи прийняте рішення.

На діяльність підсистем будь-якого рівня впливають підсистеми вищих рівнів, частіше всього наступного старшого рівня. Узгодження усіх рівнів управління є запорукою оптимального управління всього підприємства, а також має прямий вплив на техніко-економічні показники всього підприємства, якість та конкурентоздатність продукції та її собівартість [78, 88].

Необхідно відмітити роботи [77, 89–109], які присвячені багатокритеріальній оптимізації та прийняттю рішень на складних системах.

Важливо зазначити, що виникнення можливих відхилень та недоліків, внесення певних корективів, виникнення пропозицій на кожному з етапів, представлених на схемі, індивідуальне для кожного окремого випадку, залежить від конкретної поточної ситуації і потребує для подальшого прийняття рішень оперативної оцінки значної кількості інформації, а також професійності та досвіду особи, що приймає рішення [78].

Робота з прийняття управлінських рішень вимагає високого рівня компетентності, значних затрат часу, енергії, досвіду. Вона передбачає і високий рівень відповідальності, оскільки для втілення прийнятих рішень у життя необхідно забезпечити ефективну та злагоджену роботу всього організаційного механізму.

Проведено системний аналіз бізнес-процесів планування та розподілення готової продукції з метою дослідження предметної області. Для дослідження використано CASE-засіб ERwin Process Modeler, призначений для функціонального моделювання предметної області, в основі якого покладено методологію SADT (structured analysis and design technique), що дає можливості для структурного

аналізу і проєктування. Засіб представляє собою інтегроване середовище для зручного моделювання функцій, процесів, відображення діяльності та їх взаємодії, має досить зручний інтерфейс і велику кількість варіантів для представлення звітів. В основі даного CASE-засобу покладена методологія структурного аналізу IDEF0 (Integration Definition for Function Modeling) [110–114].

Були розроблені функціональні моделі з необхідними ступенями деталізації, що дозволило виявити усі бізнес-процеси та інформаційні потоки, які їх забезпечують, а саме (Додаток В):

- «Організація та управління макаронним виробництвом на ВАТ «Макаронна фабрика»;
- «Організація виготовлення та розподілення готової продукції на молокопереробних підприємствах»;
- «Організація та контроль виготовлення продукції на ТОВ «Продеко»»;
- «Організація виконання замовлень на ТОВ «Слобожанський бекон».

Кожна функціональна модель розроблена з необхідним ступенем деталізації функцій системи, являє собою ієрархію взаємопов'язаних діаграм, кожна із яких представляє підсистему або її окрему компоненту. Вершина цієї структури містить загальний опис системи, який деталізується на наступних рівнях декомпозиції. Функціональна модель представляється у вигляді графічного зображення та опису системи, що має єдину мету, одну точку зору та область визначення (межі моделювання) [78, 110–114]. Побудована модель містить ієрархічно організований комплект діаграм, опис даних, словник та різномірні звіти.

В процесі аналізу системи управління сформульовано сукупність основних питань, які спрямували процес моделювання та дали змогу отримати відповіді на них за допомогою побудованих функціональних моделей [78, 110–114]. Основними питаннями при моделюванні були:

- Які основні бізнес-процеси виконуються для виконання замовлень?
- Хто є виконавцем бізнес-процесів?
- Хто є відповідальним за виконання бізнес-процесів?

- Як пов'язані між собою бізнес-процеси?
- Які існують зв'язки між бізнес-процесами?
- Які інформаційні та матеріально-технічні потоки необхідні для виконання кожного бізнес-процесу?
- Які основні задачі прийняття управлінських рішень виникають при управлінні макаронним виробництвом?

При побудові моделі обрана точка зору (Viewpoint) директора підприємства, що забезпечило адекватність відображення системи управління виробництвом. Необхідно зазначити, що на різних підприємствах за організацію і контроль дотримання виконання договорів, пов'язаних із виготовленням продукції, здійснюють працівники різних посад. Як правило, таку функцію виконує комерційний директор.

На певних харчових підприємствах за це відповідає головний технолог, адже він є особою, яка приймає рішення щодо формування виробничих програм щодо виготовлення продукції з дотриманням усіх технологічних вимог. Такий підхід притаманний багатоміністерним підприємствам. Наприклад, на більшості підприємств макаронної галузі відповідальним за виконання замовлень є начальник виробництва.

Для концентрації уваги безпосередньо на системі управління встановлено обмеження на об'єкт моделювання, що дало змогу виключити з розгляду сторонні об'єкти, які знаходяться поза його межами [78, 110–114].

Діяльністю вищого рівня в ієрархії діаграм функціональної моделі є контекстна діаграма, що містить один блок діяльності й пов'язані з ним стрілки. Зв'язки з зовнішнім середовищем представлені у вигляді стрілок входів та виходів по чотирьох сторонах блока діяльності на контекстній діаграмі. В залежності від розташування та напрямку стрілок визначається їх призначення [78, 110–114].

Стрілки з лівої сторони є входами системи (input), вони визначають інформаційні та матеріальні ресурси, що надходять у систему і перетворюються до відповідного результату. Верхні стрілки відповідають входам по керуванню

(control), тобто різним керуючим діям, командам, стратегіям поведінки, процедурам, документам, що регламентують виконання роботи, тощо. Вони не змінюються під час виконання функції, а слугують лише для керування. Стрілки з правої сторони відповідають виходам системи (output) і є результатом її діяльності за рахунок перетворення вхідних ресурсів. Стрілки, що входять до блока діяльності з нижньої сторони, відповідають механізмам (mechanism), за допомогою яких виконуються вказані у блоці діяльності функції. Контекстна діаграма деталізується діаграмою декомпозиції першого рівня. Нумерація діаграм у моделі починається з контекстної діаграми, що має номер А-0, а діаграма її декомпозиції – А0. На діаграмі декомпозиції зображено блоки діяльності, що представляють функції, які є складовими батьківської діяльності у процесі детального аналізу [78, 110–114].

Перший рівень декомпозиції представляє укрупнено окремі функції системи. Кожен функціональний блок відповідає окремій функції. Кожен функціональний блок, якому відповідає окрема функція, зображується прямокутником. Сторони функціональних блоків мають таке ж призначення, що й розглянуті вище сторони контекстної діаграми. Між окремими функціональними блоками встановлено зв'язки, що відповідають логіці функціонування системи. Стрілки зв'язків відображують передачу від блока до блока якогось певного об'єкта (предмета, речовини, документа), а інколи і усного розпорядження чи їх сукупності [78, 110–114].

Робота [8] присвячена аналізу процесу планування закупівлі сировини для підприємства ресторанного господарства з використанням функціонального моделювання. Робота [13] присвячена моделюванню організації та управління забезпеченням сировиною цукрового заводу з використанням функціонального моделювання. В результаті є можливість стверджувати, що усі підприємства харчової галузі повністю залежать від сировини.

У результаті аналізу робіт [115–121] та проведеного системного аналізу різних предметних областей [7–9, 13, 114] є можливість стверджувати, що усі підприємства харчової галузі повністю залежать від сировини.

Зв'язки з зовнішнім середовищем представлені у вигляді стрілок входів та виходів по чотирьох сторонах блока діяльності на контекстній діаграмі. В залежності від розташування та напрямку стрілок визначається їх призначення.

У результаті аналізу було встановлено, що планування виконання замовлень є ключовою функцією – головною та важливою на усіх підприємствах харчової галузі.

Планування виконання замовлень для підприємств харчової галузі здійснюється одночасно на всіх прошарках багаторівневої ієрархічної організаційної структури. Причина полягає в тому, що жодне планове рішення не є незалежним від інших і потребує повного розуміння поставлених задач усіма пов'язаними між собою ланками управлінського та виконавчого апарату. Планування виконання замовлень поєднує цілі як підприємства загалом, так і його підрозділів зокрема з засобами їх досягнення. Крім цього, планування виконання замовлень водночас є інструментом контролю, оскільки не лише встановлює цілі, нормативи та еталони діяльності, а й визначає межі відхилень від норм, порушення яких обумовлює прийняття додаткових координуючих рішень [78].

Результат процесу планування повинен мати такі характеристики [78]:

- єдність – завдання кожного відділу підприємства повинні бути пов'язані одне з одним і не бути суперечливими;
- нерозривність – виражається у проведенні як короткострокового, так і довгострокового планування;
- гнучкість – адаптація плану до зміни обставин;
- точність – виявлення найбільш імовірного розвитку подій.

Враховуючи основні напрями діяльності підприємств та ієрархічність структури управління, є можливість виділити таку послідовність цілей від загальної до конкретної в ієрархії системи управління виробництвом: максимізація прибутку; максимізація задоволення потреб за мінімальні проміжки часу; раціональне використання основного технологічного обладнання; мінімізація витрат на виробництво, зберігання та відвантаження виробів.

## 1.2 Дослідження інформаційних технологій в управлінні харчових підприємств

Управління підприємством має ієрархічну структуру. У загальній структурі управління виробництвом виділяють кілька ієрархічних рівнів, які відображені у вигляді піраміди на рис. 1.1. Автоматизація управління на різних рівнях реалізується за допомогою автоматизованих систем управління (АСУ) [122].

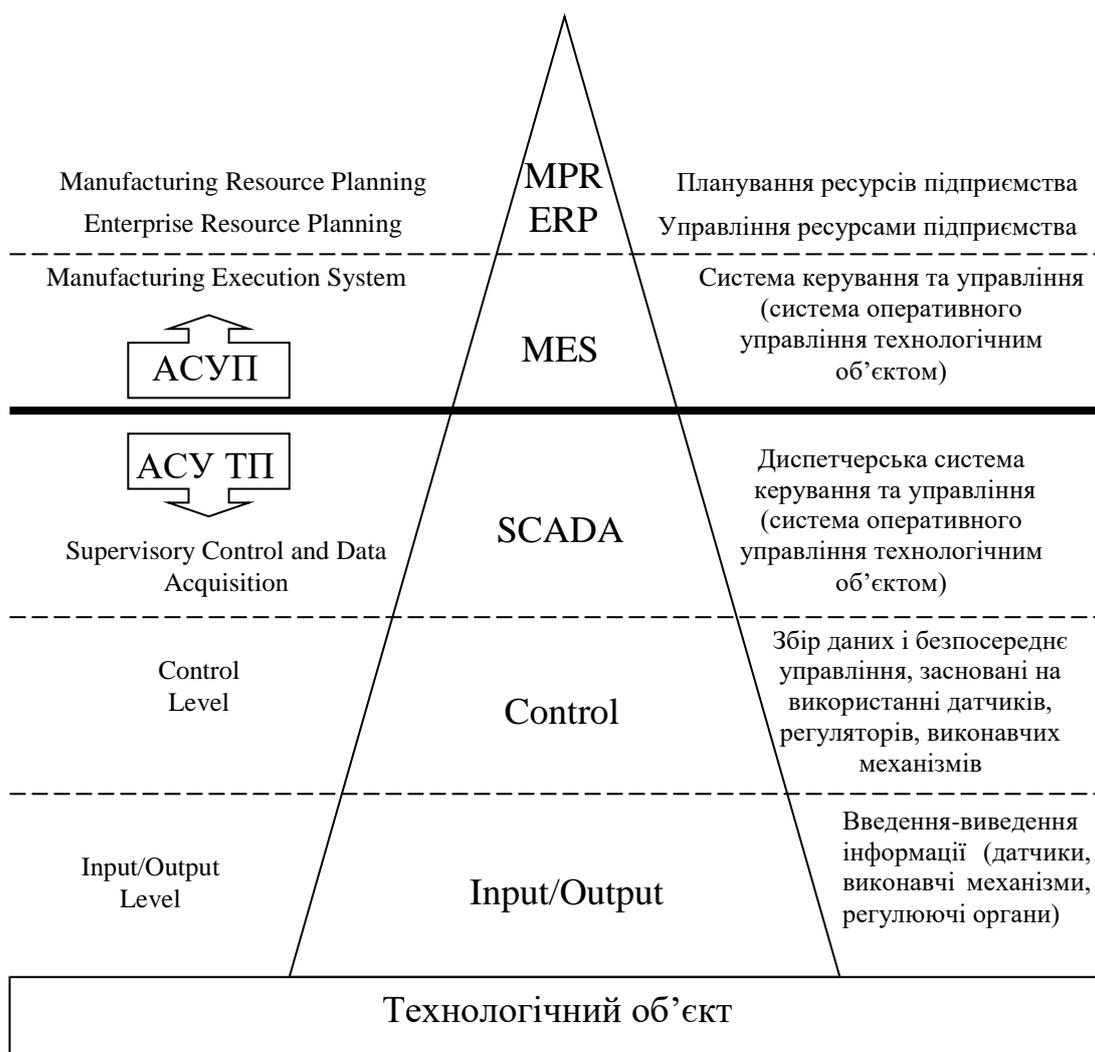


Рисунок 1.1 – Загальна структура управління виробництвом

Інформаційна підтримка виробництва продукції здійснюється автоматизованими системами управління підприємством (АСУП) і автоматизованими системами управління технологічними процесами (АСУ ТП).

До АСУП відносяться системи планування виробництва і вимог до матеріалів MRP (Manufacturing Requirement Planning) та системи планування і управління підприємством ERP (Enterprise Resource Planning). Найбільш розвинені системи ERP виконують різні бізнес-функції, пов'язані з плануванням виробництва, закупками сировини та матеріалів, збутом продукції, аналізом перспектив маркетингу, управлінням фінансами, персоналом, складським господарством, урахуванням основних фондів та ін. Системи MRP орієнтовані, головним чином, на бізнес-функції, безпосередньо пов'язані з виробництвом [122–124].

До АСУП також відносять виробничу виконавчу систему MES (Manufacturing Execution Systems), призначену для вирішення оперативних завдань управління виробництвом [125–128].

До складу АСУТП входить система SCADA (Supervisory Control and Data Acquisition), яка виконує диспетчерські функції (збір і обробка даних про стан обладнання та технологічних процесів) і допомагає розробляти ПЗ для вбудованого обладнання. Безпосереднє програмне керування технологічним обладнанням здійснюється рівнем Control level за допомогою системи CNC (Computer Numerical Control) на базі контролерів (спеціалізованих комп'ютерів, які називаються промисловими), вбудованих у технологічне обладнання. Системи CNC називають також вбудованими комп'ютерними системами. Найнижчий рівень призначений для організації зв'язку між технологічним об'єктом і керуючими пристроями (комп'ютерами) [122, 128].

Роль інформаційного циклу в системі управління представлена на рис. 1.2. Подія генерує інформацію, яка потім використовується із проміжною обробкою або без неї, що викликає керуючий вплив.

Інформаційна система являє собою інформаційне середовище, яке дозволяє визначити, де, коли, за яких обставин сталася певна подія з об'єктом управління;

яким чином її сприймати; задокументувати інформацію про стан об'єкта управління, де, коли і як слід застосувати керуючі впливи на нього.

Інформаційна система в загальному випадку призначена здійснювати такі операції (рис. 1.2.):

- накопичення, первинна обробка та оцінка достовірності інформації;
- перетворення інформації для приведення її до єдиного виду, якщо її формати чи представлення відрізняються від потрібної;
- запис та зберігання інформації;
- формування звітної документації;
- забезпечення виконання запитів на отримання інформації користувачами;
- розв'язання задач підтримки функцій управління шляхом оброблення інформації та проведення певних розрахунків;
- забезпечення підтримки прийняття рішень.

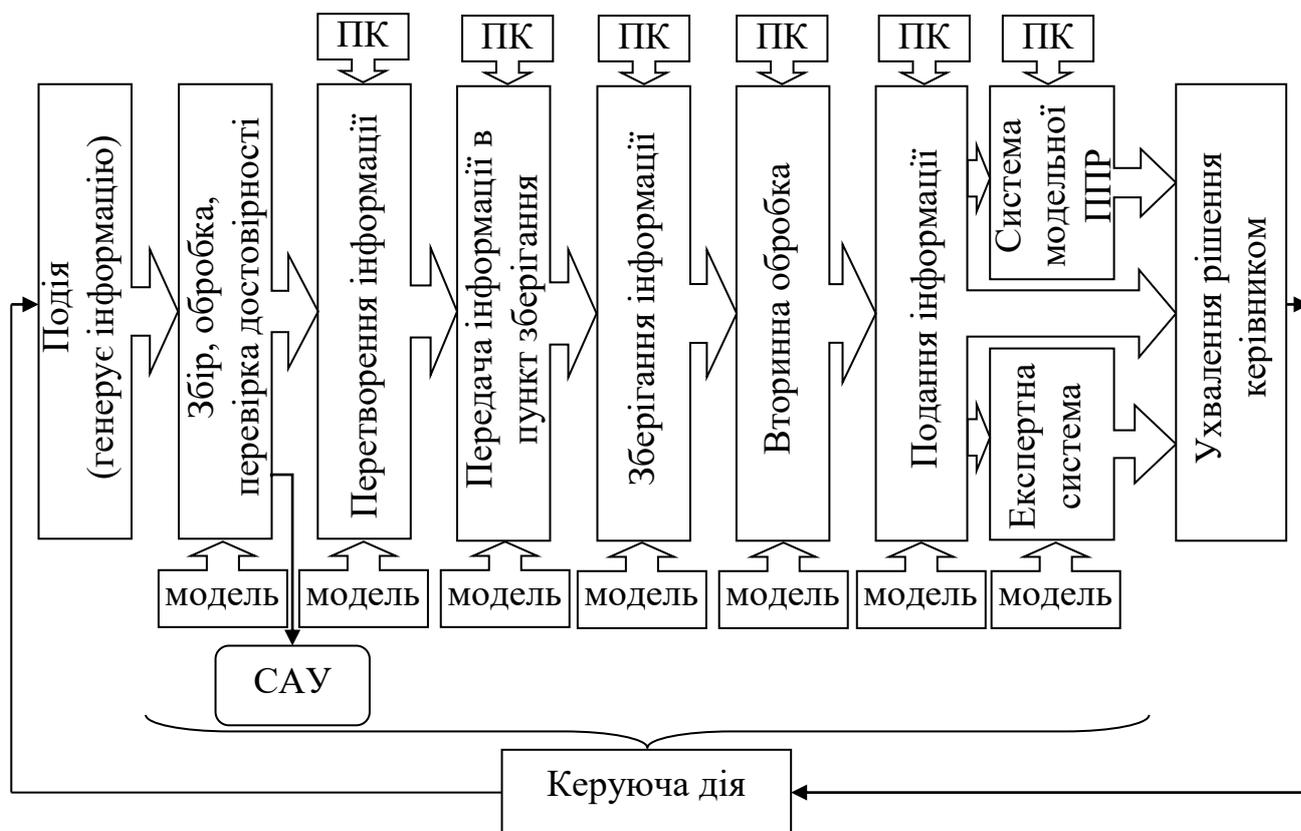


Рисунок 1.2 – Основні операції інформаційної системи управління

Слід відзначити особливу роль моделей різного класу і виду на всіх етапах. Досвід показує, що тільки використання адекватних моделей різного класу (математичних моделей технологічних процесів, моделей знань, моделей даних тощо) може забезпечити успішність і ефективність функціонування автоматизованих інформаційних систем.

### **1.2.1 Огляд основних типів інформаційних систем для підтримки управління підприємством**

Система управління виробничими процесами MES (Manufacturing execution system) – прикладне програмне забезпечення направлене на вирішення завдань управління виробничою діяльністю з виготовлення продукції. MES-системи є промисловими програмними комплексами, направленими на оптимізацію складних виробничих процесів, пов'язаних з виготовленням дрібносерійного та потокового продукту. Вони використовуються на підприємствах різних галузей, у тому числі харчової [125–129].

MES можуть забезпечувати підтримку різних функцій управління, а саме: управління формуванням техніко-економічних показників нового виробу протягом його життєвого циклу; планування ресурсів для забезпечення усього виробничого циклу та формування замовлень на них; аналіз виробничої діяльності та управління простоями виробництва для визначення коефіцієнта загальної ефективності обладнання; здійснення аналізу якості продукції; відстеження та контроль за походженням сировини. Система MES забезпечує накопичення детальної інформації та формування різних звітів про поставку й використання сировини та матеріалів, про процес виробництва та показники готової продукції. Такий підхід направлений на ретроспективне накопичення детальної інформації, що може забезпечити підтвердження дотримання усіх нормативно-правових вимог до

сировини та процесу виготовлення, що особливо важливо у харчовій та фармацевтичній галузях [125–127].

MES-системи дозволяють оптимізувати виробництво та зробити його більш рентабельним за рахунок швидкої реакції на позаштатні ситуації з метою компенсації відхилень від планових завдань. За рахунок того, що MES-системи забезпечують накопичення та узагальнення інформації в реальному часі, починаючи від формування замовлення та закінчуючи відвантаженням готової продукції, відбувається підвищення ефективності усіх бізнес-процесів та техніко-економічних показників у цілому [127–129].

Сучасні MES-системи дають можливість вдвічі пришвидшити обробку замовлень на виробництво. Застосування MES дає можливість своєчасного формування та коригування календарно-оперативних виробничих розкладів для визначення фактичної собівартості виготовлення як напівфабрикатів, так і усієї продукції в цілому [125–127].

У наш час системи MES частіше використовують як систему управління цехами та виробничими підрозділами, але на середніх підприємствах їх можливо використовувати для інтегрованого регулювання на виробництві. Використання MES із великою ймовірністю дозволить збільшити прибуток навіть без додаткових вкладень у сировину [126–128].

Впровадження MES-системи доцільно на підприємстві зі складним технологічним процесом виробництва, а для досягнення максимальної корисності необхідно здійснювати інтеграцію з існуючими ERP-системами цього підприємства [124, 128–130].

MES-система, в першу чергу, забезпечує підприємству вигоду за рахунок оптимізації процесів виробництва, а також економію часу, матеріалу і засобів.

Основною перевагою використання MES-систем є скорочення тривалості виробничого циклу виконання замовлення, що впливає на збільшення прибутку і конкурентоспроможності підприємства за рахунок зменшення витрат підприємства; підвищення виробничої потужності; збільшення обороту підприємства [125–128].

Серед основних завдань MES виділяють [127–130]:

- використання виробничих потужностей на основі детального операційного планування виробництва;
- відстеження виробничих потужностей;
- збір інформації, пов'язаної з виробництвом за допомогою систем автоматизації виробничого процесу, датчиків (систем контролю) та інших ІС-підприємств;
- відстеження і контроль параметрів якості;
- інформування персоналу та забезпечення вхідними даними обладнання на початок процесу виробництва;
- встановлення зв'язків між персоналом та обладнанням у рамках виробництва.
- встановлення зв'язків з усіма підрозділами підприємства, а також з партнерами, постачальниками та клієнтами;
- реагування на вимоги щодо номенклатури виробництва, зміни компонентів, сировини і напівфабрикатів, які застосовуються у процесі виробництва;
- реагування на необхідність зміни рецептури виготовлення продукції;
- гарантування відповідності нормативно-правовим документам та стандартам процесу виготовлення й готової продукції;
- відповідність індустріальним стандартам.

Ефективне впровадження MES системи вимагає дотримання такого:

- грамотного аналізу бізнес-функцій від відповідних фахівців;
- придбання і використання повністю уніфікованого обладнання;
- визначення чітких завдань і рамок проєкту;
- впроваджувальна система повинна повністю інтегруватися з існуючими ІС-підприємствами та мати можливості легкого масштабування.

До основних недоліків впровадження MES-систем відносять [126–128]:

- надзвичайну затратність, адже сюди відноситься вартість інтеграції та впровадження, програмне забезпечення, навчання персоналу, технічне обслуговування та підтримка;

- відсутність інформаційно-обчислювального комплексу на відповідному рівні, а також демотивацію персоналу;
- високу складність інтеграції та вартість (близько 35% від загальної вартості придбання та впровадження) MES-системи.

Впровадження та використання MES-системи недоцільно на підприємствах з невисокими техніко-економічними показниками (невеликі та малі підприємства), до яких відносять невисокий чистий прибуток; невеликий обсяг виробництва; недостатній рівень автоматизації технологічних процесів. На невеликих підприємствах із менш складним технологічним процесом, як правило, достатньо застосування ERP-системи [124–127].

Системи оперативного планування APS (Advanced Planning & Scheduling – вдосконалене планування та складання розкладу) – програмне забезпечення для виробничого планування, головною особливістю якого є можливість побудови розкладу роботи обладнання в рамках усього підприємства. Отримані таким чином приватні розклади виробничих підрозділів є взаємопов'язаними з точки зору виробу і його операцій. Вимоги SCM (Supply Chain Management – управління ланцюгами поставок) у даному випадку можуть дотримуватися як у межах підприємства (міжцеховий розклад), так і по відношенню до зовнішніх поставок на підприємство [126–130].

На відміну від традиційних систем календарного планування виробництва та оперативного планування виробництва, які використовують покрокову процедуру для розподілу матеріальних та виробничих потужностей, APS планує і формує розклад виробництва на основі наявних матеріалів, робочої сили та потужностей заводу [131–137].

APS зазвичай застосовується, коли є одна або більше з таких умов:

- виготовлення продукції здійснюється на замовлення без здійснення запасу на довгий період;
- капіталомісткі виробничі процеси з обмеженою виробничою потужністю;

- багатомноменклатурне виробництво, для якого використовується одне технологічне обладнання, що призводить до його переналагодження для виготовлення іншого виду продукту;
- у технологічному процесі виготовлення продукції задіяна велика кількість операцій або компонентів/сировини;
- виробництво потребує частих змін розкладу, які неможливо передбачити.

APS здійснюють синхронне планування необхідних до закупівлі і виробництва виробів, виконуючи це з урахуванням існуючих (обмежених) потужностей. Модель даних APS-системи передбачає можливість обліку при плануванні детальних характеристик конкретних одиниць обладнання, штату, транспортних засобів, технологічних маршрутів тощо [132–137].

Деталізації і точності моделі виробництва і ланцюжків поставок знаходяться на високому рівні. Точність плану залежить від вірно встановлених обмежень виробництва і ланцюжків поставок.

Створення планів із використанням APS-системи відбувається дуже швидко за рахунок того, що вся модель даних зберігається в оперативній пам'яті сервера. Все це сприяє швидкому реагуванню на різні зміни в ланцюзі постачань.

Широко використовується колективна робота зовнішніх і внутрішніх учасників в єдиному багатокористувацькому середовищі. Оскільки APS-додатки web-орієнтовані, то забезпечується можливість співпраці користувачів системи і узгодження планів між різними підрозділами в середині підприємства і з іншими учасниками процесу за його межами [132, 137].

Системи APS дають користувачам зручні засоби аналізу планової інформації – різні графіки, діаграми (наприклад, інтерактивна діаграма Ганта, графіки завантаження машин і ресурсів, складських запасів, обсягу незавершеного виробництва тощо), широкий набір вбудованих звітів, а також можливість створювати призначені для користувача звіти.

Основні можливості APS-систем [134, 137–142]:

- планування з точністю до секунди;

- пряме (всі виробничі замовлення і замовлення на поставку намагаються виконатися якомога раніше) і зворотне планування (замовлення на виробництво повинні завершитися до певної дати, замовлення на закупівлю повинні виконатися до того моменту, коли вони будуть використані у виробництві);
- багаторівнева агрегація/хвильові згладжування, що забезпечує прогнозування зверху-вниз, знизу-вгору і від середини (в обидві сторони);
- коригування прогнозу може здійснюватися як на рівні редагування числових даних, так і за допомогою миші у графічному поданні;
- може враховувати характер життєвого циклу продукту, при цьому беруться до уваги періоди освоєння нових продуктів і зняття з виробництва старих;
- необмежене число визначених користувачем одиниць вимірювання;
- безліч алгоритмів і параметрів розрахунку поповнення запасів;
- прогнозування і відстеження матеріально-виробничих запасів;
- контроль над виконанням і здійснення повідомлення у графічному форматі, а також у вигляді звітів.

Найбільшу ефективність від впровадження APS-системи можливо отримати при спільному використанні існуючих інформаційних систем підприємства у вигляді ERP-APS-MES, адже тоді буде забезпечено інформаційну підтримку на усіх рівнях управління всього підприємства [136–137].

При цьому APS використовує інформацію, що міститься у транзакційній частині ERP (історія продажів, інформація про фактичні замовлення клієнтів, залишки товарів на складах тощо). Архітектура самих ERP-рішень, орієнтована на транзакційну роботу з СУБД, не розрахована на обробку великих даних. Як джерело інформації про стан виробничих замовлень, що виконуються, можуть виступати MES-системи. По завершенні процесу планування APS-система передає відповідні результати, такі, як замовлення на виробництво, закупівлю і переміщення, прогнози тощо, в ERP-систему [128–131].

Оптимізація в системах APS базується на евристичних та складному математичному апараті, що створюються для конкретної галузі та підприємства (наприклад, металургія, прокат-оптимізація змін розміру листів). При цьому

налаштування алгоритмів може здійснюватися безпосередньо самими користувачами [131–133].

APS складається із трьох основних компонентів: Sales and Demand Forecasting – прогнозування збуту і попиту; Master Production Scheduling & Rough-Cut Capacity Planning – формування основного виробничого плану і загальне планування завантаження виробничих потужностей; Finite Capacity Scheduling – детальне планування завантаження виробничих потужностей [132–134].

Sales and Demand Forecasting (SDF) – модуль оперативного прогнозування та управління запасами, який узагальнює прогнози, згенеровані на основі «історії» попиту статистичної системи, і очікувані користувачем зміни умов ринку по ланцюжку постачальників. За допомогою модуля SDF менеджери можуть відстежувати стан запасів товарно-матеріальних цінностей і параметри поповнення запасів для кожного продукту в місцях його складування. Планування забезпечення сировиною та матеріалами здійснюється одночасно з урахуванням можливостей використання основного технологічного обладнання і параметрів складських приміщень для зберігання сировини та матеріалів.

Master Production Scheduling дає можливість проаналізувати різні сценарії, щоб розробити виробничий план, який відповідає потребам як існуючого портфеля замовлень, так і прогнозованого обсягу продажів з урахуванням доступних ресурсів компанії [131–133].

Rough Cut Capacity Planning – модуль загального планування завантаження потужностей надає можливість формування узагальнених планів завантаження виробничих потужностей на підставі інформації, отриманої від компонента Master Production Scheduling, а також переглядати заплановані потреби в ресурсах і порівнювати їх з існуючими обмеженнями. Це дозволяє виявити всі групи критичних ресурсів, включаючи робочу силу, устаткування, енергоресурси, матеріали та складські приміщення, і порівняти альтернативні моделі календарних графіків, щоб досягти необхідного рівня використання критичних ресурсів при реалізації моделюючого плану. На відміну від MRP II, APS-системи підтримують розподілене планування, при якому кілька користувачів можуть планувати

одночасно, але кожен із них несе відповідальність за певну виробничу ділянку [123–125].

Production Planning & Finite Capacity Scheduling – модуль планування виробництва і детального планування завантаження технологічного обладнання дозволяє враховувати динаміку і реальний стан справ, щоб формувати календарні графіки відповідно до стану обладнання, наявної робочої сили, сховища, джерел енергії, основних матеріалів [122].

Необхідно відмітити, що не всі системи APS підтримують функції диспетчеризації. В APS передбачено перепланування тільки при появі нових замовлень (зворотний зв'язок у режимі реального часу для APS вважається надлишковим), на відміну від MES, які виконують цю операцію набагато частіше (для завдання значно меншої розмірності, коригуючи плани лише окремих ділянок), оскільки реагують на будь-яку зміну ходу технологічного процесу. Постійні корекції планів виробництва – це типове явище для виробництв дрібносерійного і одиничного типів. Для створення більш точного контуру зворотного зв'язку з виробництвом необхідно використовувати інтеграцію з MES-системами [127–131].

Тривалість горизонту планування в APS-системах визначається як різниця в часі між моментами видачі найбільш далеких замовлень з усього портфеля замовлень підприємства і поточною датою, оскільки при появі нового замовлення і відповідного перерахунку всього розкладу, треба визначити не тільки терміни його виготовлення, а й можливість дотримання термінів виконання вже запущених замовлень.

APS-системи дозволяють швидко здійснити реконфігурацію оперативних планів при виникненні позаштатних ситуацій, які можуть негативно вплинути на виконання поточного плану, а саме непередбачені зміни в зовнішньому середовищі (наприклад, скасування замовлень); на виробництві (відмова обладнання) [132].

Окремі APS-системи підтримують розподілене планування, за допомогою якого кілька людей можуть планувати синхронно, але кожен із них несе відповідальність за певну виробничу ділянку.

## 1.2.2 Системи підтримки прийняття рішень

Найбільш придатними для розв'язання багатьох задач управління є СППР. Необхідно відзначити, що СППР призначені для підтримки багатокритеріальних рішень у складних середовищах, які характеризуються великими обсягами даних, а обробка яких неможлива без використання сучасної обчислювальної техніки [143].

При цьому результат прийняття рішень оцінюються за сукупністю багатьох критеріїв, які необхідно розглядати одночасно, використовуючи один із наступних підходів: пошук оптимального рішення за рахунок обрання найкращого рішення з усіх можливих варіантів; ранжування можливих рішень за заданими перевагами.

Сучасні СППР поділяються на [144–147]:

- моделі-орієнтовані, що працюють на основі обробки статистичних, фінансових або інших моделей;
- засновані на комунікаціях між користувачами, що займаються спільною справою;
- орієнтовані на обробку внутрішніх та зовнішніх даних на основі часових рядів;
- орієнтовані на маніпуляції з неструктурованою інформацією, представленою у різних інформаційних системах;
- орієнтовані на обробку знань і надають спеціалізовані рішення проблем, заснованих на фактах.

Незважаючи на те, що за останні десятиріччя було створено багато СППР, постійні зміни в умовах господарювання з метою підвищення ефективності управління підприємств вимагають створення нових СППР [5, 78, 146, 147].

Сучасні СППР являють собою результат багатьох досліджень у сфері інформаційних технологій, а конкретніше: баз даних і баз знань, штучного інтелекту, інтерактивних комп'ютерних систем, методів імітаційного моделювання.

Для СППР характерними є такі особливості:

- надання допомоги керівнику у процесі прийняття рішень і забезпечення підтримки для всього діапазону розв'язуваних задач, причому думка людини щодо даної задачі та інформація, представлена комп'ютером, являють собою єдине ціле для прийняття управлінських рішень;

- підтримка і посилення, але не зміна, міркування та оцінки керівника, оскільки ситуація контролюється людиною, яка приймає остаточне рішення, а система їй лише в цьому допомагає;

- у СППР акцент робиться на максимальній ефективності процесу прийняття рішень;

- виконання інтеграції математичних, статистичних, імітаційних та інших моделей і аналітичних методів зі стандартним доступом до даних та вибіркою з них;

- СППР є дружніми для користувачів і не потребують глибоких знань про обчислювальну техніку, вони забезпечують роботу в діалоговому режимі, просту навігацію у системі, а також наявність ряду контекстних підказок для усіх елементів.

- побудова за принципом інтерактивного розв'язання задач, коли користувач має можливість підтримувати активний діалог із СППР, а не обмежується поданням окремих команд із подальшим очікуванням результату кожної з них;

- гнучкість і адаптивність для пристосування до змін середовища або модифікації підходів до розв'язання задач;

- СППР не нав'язує визначений процес прийняття рішень, а користувач повинен мати набір можливостей у формі і послідовності, що відповідають стилю його пізнавальної діяльності.

СППР розширюють можливості людини в маніпулюванні даними під час пошуку ефективних варіантів рішень, надають можливість виявити різні підходи до вирішення проблеми, а також є гнучким інструментом опрацювання альтернатив.

Важливим є питання вибору тієї чи іншої технології з метою прийняття рішень. В основі кожної з СППР лежить той чи інший математичний метод вирішення задачі. Таким чином, розгляд інформаційних технологій у складі СППР неможливий без висвітлення використовуваних у них методах. У СППР частіше

використовують методи штучного інтелекту для формування варіантів прийняття рішень, а саме:

- пошук інформації;
- пошук знань у базах даних;
- інтелектуальний аналіз даних;
- імітаційне моделювання;
- ситуаційний аналіз;
- нейронні мережі;
- еволюційні обчислення та генетичні алгоритми.

СППР, які базуються на методах штучного інтелекту, називаються інтелектуальними СППР (ІСППР). Найближчими до СППР є експертні системи або автоматизовані системи управління.

У роботах [19, 80, 148, 149] запропонована та обґрунтована структура нечіткої ситуаційної мережі для системи підтримки прийняття рішень з урахуванням слабо структурованих або неструктурованих проблем та відсутністю точних моделей для опису проблемних ситуацій підвищення ефективності керування організаційно-технічними (технологічними) системами на прикладі цукрового заводу, що працює на значних часових інтервалах.

Більшість систем СППР мають функціонал, направлений на прогнозування виникнення подій. Як правило для прогнозування необхідно здійснити складне дослідження об'єкта [3]. Більшість СППР використовують інтелектуальний аналіз даних для розв'язання складних задач прогнозування. У роботах [6, 7, 19, 35, 37, 38, 78, 150-159] наведено різні методи та підходи, які забезпечують якісне прогнозування.

СППР забезпечує формування та вибір оптимальних варіантів рішень, які повинні виконуватися з використанням набутого досвіду й адаптації існуючих алгоритмів на основі аналізу попередніх ситуацій. При цьому повинен забезпечуватися інтерактивний зв'язок із користувачами за допомогою розуміння комп'ютером природної мови, візуалізації отриманих результатів у вигляді рисунків,

багатовимірних графіків, діаграм, графів, інших формальних та неформальних зображень [78, 126–147, 158–162].

У роботах [49, 78, 163] запропонована структура СППР для підтримки прийняття рішень на підприємствах макаронної галузі, яка наведена у Додатку Д.

Така структура забезпечує: за допомогою засобів завантаження даних відбувається накопичення інформації у сховищі даних із джерел даних; OLAP-сервери формують тематичні підмножини даних та забезпечують виконання OLAP-операцій над ними; інструментарій користувача забезпечує діалоговий режим роботи та виконання покладених на нього задач і функцій [16, 33, 78, 163].

Необхідно відмітити, що більшість СППР створюються як веб-орієнтовані, що обумовлено багатьма факторами, а саме:

- доступність через мережу Інтернет, що забезпечує можливість доступу до системи з будь-якої точки світу через браузер;
- підтримка приватних осіб / клієнтів / співробітників / менеджерів / груп у процесі прийняття рішень незалежно від їх фізичного місця розташування або часу;
- використання даних, баз знань, документів і моделей, які мають можливість звернутися до величезного розмаїття великих груп користувачів, тощо;
- мінімізацію витрат на встановлення та обслуговування системи.

Web-орієнтовані системи підтримки прийняття рішень працюють через мережу Інтернет, а також не поступаються функціональністю мережевим системам.

Переваги використання web-орієнтованих СППР мають вагомі причини, а саме:

- web-доступ до СППР економить витрати на встановлення програмного забезпечення, адже воно встановлено на певному порталі централізовано, доступ забезпечується з декількох місць, а web-сервери замінюють ще й мережеві сервери для цих систем;
- web-доступ знижує витрати на технічне обслуговування системи, оновлення компонентів, структур, форми збереження інформації, заміну сервісів без залучення користувачів, а, головне, без втрати функціональності та цілісності даних, які можуть виникнути в системі, що розвивається з часом;

- особи, які приймають рішення, і споживачі мають більш широкий доступ до системи, тому що вона доступна з будь-якого комп'ютера в будь-який час.

Таким чином, web-доступ здатний зменшити витрати при встановленні, експлуатації, обслуговуванні та оновленні системи. Це, у свою чергу, збільшує доступ до даних, а також забезпечує спільне вирішення поставлених завдань.

### **1.3 Огляд сучасних методів для оперативного планування замовлень**

Виникнення задач теорії розкладу пов'язане з необхідністю вибору деякої визначеної послідовності виконання робіт за встановлений період часу з використанням певних ресурсів. Вибір варіанта виконання, а саме визначення часу та порядку виконання кожної роботи, значним чином впливає на величину витрат на виконання, а також на загальний час, необхідний для завершення всіх робіт планового періоду.

У загальному вигляді задача визначення розкладу виконання може бути сформульована так: виконати у визначений період часу задану сукупність робіт, використовуючи деяку множину ресурсів. При цьому задача додатково характеризується обмеженнями на послідовність виконання робіт, вартістю виконання робіт при використанні різних ресурсів, наявними директивними термінами здійснення окремої роботи тощо.

Коло вирішуваних у рамках теорії розкладів задач можна розділити на такі групи [164–171]:

- задачі розподілу завдань, обрання пристроїв чи ресурсів для їх виконання;
- задачі упорядкування набору завдань на конкретному технологічному обладнанні;
- задачі упорядкування набору завдань для виконання за визначеним часом;
- задачі обрання складу та розміщення обладнання, визначення кількості;
- побудова маршруту руху для транспорту на виробництві;

- побудова розкладів проведення навчальних занять або руху транспортних засобів.

Для вирішення задач побудови плану виконання замовлень існує значна кількість методів та підходів, які відносяться до різних категорій, мають свої області застосування та характеристики.

Необхідно відмітити ряд робіт, направлених на розв'язання багатокритеріальних задач управління [172–177], присвячених дослідженням моделей і методів математичної оптимізації, автоматизації процесів обробки і аналізу даних математичними методами.

У залежності від виробництва можуть висуватися різні критерії оптимальності розкладу виконання замовлень [164–171], а саме:

- мінімальний середній та середньозважений час очікування виконання всіх завдань;
- мінімальний середній та середньозважений час виконання всіх завдань;
- мінімальний середній або середньозважений час завершення виконання всіх завдань;
- мінімальний час виконання розкладу;
- мінімальне фактичне та середньозважене число невиконаних у директивний термін завдань;
- мінімальний час відхилення від директивного терміну завершення виконання серед усієї множини завдань;
- сумарний, середній та середньозважений час очікування пристроями початку виконання завдань;
- сумарний, середній та середньозважений час роботи пристроїв, необхідний для виконання всіх завдань розкладу.

Теорія розкладів є одним із розділів дослідження операцій – напряму, що бере свій початок із роботи Г. Ганта [178], в якій було запропоновано візуальне представлення послідовності операцій у вигляді діаграм Ганта.

Для розв'язання задач теорії розкладів застосовуються такі види математичних методів [164–171]:

- математичне цілочислове програмування;
- побудова допустимих екстремальних шляхів на графі.

Однією з найперших робіт у теорії розкладів була робота С. Джонсона [179], в якій запропоновано оптимальний алгоритм складання розкладу для випадку двох машин. Також були запропоновані інші ефективні поліноміальні методи для задач розмірності  $2 \times m$ ,  $n \times 2$  (для випадку, коли потрібно виконати не більше двох операцій для одного завдання) і  $n \times 2$  (для випадку, коли всі операції мають одиничний час виконання).

Термін «теорія розкладів» був запропонований Р. Беллманом [180].

Активне теоретичне дослідження задач теорії розкладів проводилося у роботах Дж. Джексона [181], В. Сміта [182], Р. Конвея [183] В. Танаєва [165]. Однією з головних проблем, яка висвітлювалася у даних роботах, була класифікація задач складання розкладів та встановлення їх складності. Найпоширеніша на сьогоднішній день класифікація запропонована Р. Грехемом [164, 184].

Умовно математичні методи вирішення задач планування можна поділити на дві великі групи: точні та наближені методи.

До точних методів належать методи цілочислового програмування; послідовні алгоритми оптимізації (метод гілок та меж, методи аналізу і відсіву варіантів, методи теорії графів).

Наближені алгоритми включають генетичні алгоритми та еволюційні стратегії; евристичні методи; метаевристичні методи; імітаційне моделювання.

В основі більшості точних алгоритмів лежить метод гілок і меж [164, 185–188]. Як вершини в дереві пошуку розглядаються частинні розклади. Процес розгалуження, як правило, полягає в побудові частинного розкладу за допомогою різних стратегій. Використання таких прийомів, як правила домінування, нижні оцінки, безпосередній відбір дозволяє в більшості випадків зменшити дерево пошуку. Залежно від конкретного методу використовуються різні схеми розгалуження і способи відсікання. Для скорочення перебору обчислюються нижні оцінки цільової функції та використовуються комбінаторні властивості задач.

Публікація [189] присвячена формуванню графіків виробництва. Розглядається двоступенева модель нечіткого стохастичного програмування з нечітким періодом обробки, де одночасно оцінюються усі елементи. Для вирішення задачі планування методами нечіткої логіки запропоновано використання підходу подієво-керованого програмування. На основі алгоритму гілок та меж розроблено алгоритм з ефективною нижньою межею. Для задач великої розмірності запропоновано спосіб удосконалення з використанням генетичного алгоритму.

Для вирішення задач теорії розкладів знайшли широке застосування методи динамічного програмування [187, 190–193].

У застосуванні до окремих предметних областей задачі теорії розкладів можуть бути сформульовані як задачі цілочислового лінійного програмування [164, 181–184].

Важливо зазначити, що застосування точних алгоритмів є доцільним лише в тому випадку, коли розмірність задачі відносно невелика (до 50 робіт), оскільки занадто високі обчислювальні вимоги не дозволяють застосовувати їх до задач більшої розмірності. У задачах великої розмірності застосування наближених методів залишається єдиним ефективним способом їх вирішення, хоча вони і не гарантують однозначного знаходження оптимального рішення [164, 194, 195].

Серед наближених алгоритмів доцільно виділити алгоритми, що гарантують відносну погрішність [164, 196, 197], і алгоритми, що гарантують абсолютну погрішність.

Декодуючі процедури є основою більшості евристичних алгоритмів для вирішення задачі календарного планування з обмеженими ресурсами. Вони являють собою процес покрокової побудови розкладу. Залежно від способу включення в розклад чергової роботи, розрізняють послідовну, Т-пізню і паралельну декодуючу процедуру.

Послідовна процедура складається з  $n+2$  кроків. На кожному кроці розглядається множина робіт, вже включених до розкладу, і залишковий обсяг наявних ресурсів у кожен момент часу. Кожний крок полягає в додаванні до

розкладу чергової роботи з загальної множини. Послідовна ДП по черзі розглядає роботи проекту і для кожної з них підбирає момент її початку [164].

T-пізня процедура є аналогом послідовної і будує розклад у зворотній послідовності.

Паралельна процедура, на відміну від послідовної, на кожному кроці розглядає момент часу і по ньому будує множину робіт, які будуть з нього починатися.

Підґрунтям методів пріоритетних правил, як правило, є одна з вищеописаних декодуючих процедур. Пріоритетне правило ставить у відповідність кожній роботі деяке число. Вибір чергової роботи для включення в розклад здійснюється, виходячи з розстановки пріоритетів серед робіт допустимої множини [164].

До класу однокрокових методів відноситься більшість детермінованих «жадібних» алгоритмів. Їх основна ідея полягає в побудові одного розкладу, використовуючи послідовну або паралельну декодуючу процедуру відповідно до задалегідь обраного певного правила пріоритету.

Основна ідея багатокрокових евристик полягає в побудові декількох розкладів із наступним вибором найкращого з них. При цьому черговий розклад може будуватися як незалежно від побудованих раніше, так і використовуючи їх. Різноманітність розкладів може досягатися використанням декількох пріоритетних правил і декількох декодуючих процедур [164].

Метаевристики за своєю суттю являють собою загальні схеми побудови алгоритмів, які можуть бути реалізовані для більшості задач дискретної оптимізації. Всі метаевристики є ітераційними процедурами, і для багатьох з них встановлена асимптотична збіжність найкращого знайденого рішення до глобального оптимуму. До цього класу відносяться алгоритми імітації відпалу, пошук із заборонами, генетичні алгоритми, алгоритми колективного інтелекту та ін [164].

Алгоритм методу імітації відпалу відноситься до класу порогових алгоритмів локального пошуку. На кожному кроці цього алгоритму в околі поточного рішення обирається деяке нове рішення, і якщо різниця за значенням цільової функції між

новим і поточним рішенням не перевершує заданого порогу, то нове рішення замінює поточне. В іншому випадку вибирається нове сусіднє рішення [164].

Проблема планування для паралельних непов'язаних машин розглядається у публікації [198]. Для вирішення даної проблеми запропоновано багатоцільовий алгоритм багатопозиційного моделювання відпалу з мінімізацією загального часу роботи. Оцінка отриманих із використанням запропонованого алгоритму результатів проводилася з застосуванням чотирьох показників ефективності, а також було виконане їх порівняння з декількома еталонними значеннями. За результатами виконаних обчислень було визначено, що запропонований евристичний алгоритм помітно перевершив еталонні евристики за усіма чотирма показниками.

У роботі [199] розглядається проблема планування для паралельних машин із мінімізацією часу виконання за умови, коли зменшувальний ефект залежить від послідовності виконання робіт. Для вирішення такої проблеми авторами був розроблений алгоритм побудови плану виконання робіт на основі методу імітації відпалу.

Метод пошуку з заборонами дозволяє алгоритму локального спуску не зупинятися в точці локального оптимуму, а переходити від одного локального оптимуму до іншого з метою знаходження серед них глобального оптимуму. Основним механізмом, що дозволяє алгоритму оминати локальний оптимум, є список заборон. Він будується на основі попередньої історії пошуку, тобто за декількома останніми рішеннями і забороняє частину околів поточного рішення [164].

«Генетичний алгоритм» використовує ітераційний підхід поліпшення результатів. Тобто на кожній ітерації відбувається пошук найкращого рішення в околі даного. Якщо таке рішення знайдено, то воно стає поточним, і починається нова ітерація. Це продовжується до тих пір, поки приріст цільової функції не зменшиться практично до нуля або не буде виконана задана кількість ітерацій. Для підвищення ймовірності знаходження глобального оптимуму використовується множинний експеримент із різними початковими точками, що істотно збільшує час пошуку.

Теоретичні основи застосування генетичних алгоритмів для різних сфер діяльності людини та їх застосування у складних системах розглянуто у фундаментальній роботі [200]. Особливу увагу приділено поняттю адаптації, яка у своїй найвідомішій формі є біологічним процесом, за допомогою якого організми розвиваються шляхом перегрупування генетичного матеріалу для виживання в навколишньому середовищі. У роботі представлено математичну модель, яка описує нелінійність таких складних взаємодій, а також демонструється універсальність моделі та її застосування у галузях економіки, психології, теорії ігор та штучного інтелекту. Крім цього, виділено спосіб, у якому розглянутий підхід змінює традиційні погляди математичної генетики [200].

Авторами [201, 202] представлено теоретичне підґрунтя та детально розглянуті особливості генетичних алгоритмів з демонстрацією у вигляді значного числа практичних прикладів.

У статті [203] авторами було проведене експериментальне дослідження роботи генетичного алгоритму та виконане оцінювання результатів із використанням контрольних функцій.

У роботі [204] запропоновано підвищення ефективності генетичного алгоритму шляхом використання механізму вибіркової ініціалізації. Розглянуто проблему залежності швидкості знаходження результату та його близькості до оптимального від вибору первинної популяції алгоритму. Запропоновано новий підхід до підвищення ефективності генетичного алгоритму за допомогою селективної ініціалізації, яка спрямована на знаходження найбільш придатних «хромосом» та їх використання як початкових для генетичного алгоритму.

Авторами праці [205] розглядаються можливості практичного застосування генетичних алгоритмів для вирішення задач оптимізації управління та планування промислових процесів, побудови фінансових стратегій, планування, шифрування даних, використання у галузях робототехніки, медицини, автомобілебудування, розробки комп'ютерних ігор, побудови фінансових стратегій.

У роботах [78, 206–212] представлені існуючі підходи і методи застосування генетичних алгоритмів для вирішення задач багатокритеріальної оптимізації.

Запропоновано математичні моделі та алгоритми вирішення багатокритеріальних задач вибору стратегії розвитку виробничих систем.

Роботи [164, 213–225] присвячені використанню ройових алгоритмів для розв'язання задач планування та багатокритеріальної оптимізації. Характерною рисою ройового інтелекту є використання великої кількості агентів, здатних взаємодіяти між собою і навколишнім середовищем локально.

Загальна схема ройових алгоритмів включає в себе такі етапи:

1. «Ініціалізація популяції агентів» – створюємо певну кількість початкових наближень до рішення задачі в області пошуку.

2. «Здійснюємо пересування агентів популяції» в області пошуку з використанням певної низки правил, притаманних певному ройовому алгоритму, для наближення до шуканого екстремуму функції, що оптимізується.

3. «Завершення пошуку» відбувається за умови закінчення ітерацій, а як розв'язок обирається найкращий із знайдених варіантів розташування агентів популяції. Якщо зазначені умови не виконано, то повертаємося до виконання етапу 2.

Найважливішим поняттям ройових алгоритмів є поняття фітнес-функції (fitness-function). Часто цю функцію називають функцією придатності, функцією корисності, функцією пристосованості тощо. Важливість функції обумовлена тією обставиною, що з її допомогою оцінюють якість агентів популяції. Стратегічно, у процесі міграції агенти рухаються таким чином, щоб наблизитися до глобального екстремуму фітнес-функції.

Алгоритми ройового інтелекту можуть бути адаптовані до вирішення багатокритеріальної задачі. Але за своєю природою найбільш підходить для таких завдань алгоритм «рою світлячків», оскільки кожна частка виконує черговий крок, орієнтуючись на всі частинки, що займають кращі позиції. Це дозволяє алгоритму краще дослідити варіанти в області простору пошуку рішень у багатокритеріальних задачах. Алгоритм пошуку косяком риб є універсальним методом для знаходження оптимуму функцій будь-якої складності.

У роботі [226] проведено дослідження алгоритму «зграї вовків» на прикладі оптимізаційної задачі. Було вивчено вплив на ефективність алгоритму таких параметрів, як розмір популяції, кількість ітерацій, величина коефіцієнтів атаки і вистежування. Для дослідження ефективності і порівняльного аналізу алгоритму був розроблений програмний модуль, в якому також реалізовані такі методи, як повний перебір, метод сіток, а також їх паралельні реалізації.

У роботі [221] розглядається класична задача комівояжера. Рішення завдання проводиться алгоритмом «зграї вовків» і його модифікацією. Зроблено порівняльний аналіз розглянутих алгоритмів із різними ефективними алгоритмами вирішення даного типу завдань.

У роботі [219] розглянуто задачу планування виконання замовлень металургійними виробництвами з урахуванням чинника втрати часу на переналадження обладнання, який раніше не розглядався. Отримано зведений критерій оптимізації та цільову функцію мінімізації втрат від послідовного виконання комплексу завдань. Для задачі запропонований алгоритм розв'язання в реальному часі на основі моделювання «мурашиної колонії».

У роботі [227] запропоновано новий гібридний алгоритм на основі багатоцільового генетичного алгоритму з використанням імітаційного відпалу для паралельних машин із залежними послідовностями виконання, різними термінами виконання та встановленими пріоритетами.

Авторами публікації [228] для вирішення задачі планування представлено підхід на основі алгоритму штучної «бджолиної колонії» з використанням методу локального пошуку для посилення експлуатаційної спроможності базового алгоритму, а також з введенням додаткового оператора сусідства на основі «жадібною» конструктивно-деструктивною процедури. Крім цього, проведено порівняння запропонованого алгоритму з такими алгоритмами, як пошук із заборонами та генетичним алгоритмом.

У роботі [220] запропоновано гібридний алгоритм планування на основі алгоритму «рою часток» (Particle Swarm Optimization) та алгоритму «мурашиної колонії» (Ant Colony Optimization). Запропонований алгоритм визначає найкращі

послідовності для різних комбінацій затримок і попереджень виконання для заданого набору робіт. Після цього, для визначення оптимального варіанта, який більшою мірою задовольняє задану цільову функцію, використовується нечітка логіка. Крім цього, у роботі проведено порівняння запропонованого алгоритму з різними модифікаціями генетичного алгоритму, такими як «генетичний алгоритм» із частково відображеним оператором кроссовера, «генетичний алгоритм» із багатокомпонентним уніфікованим послідовним генератором, нечітким генетичним алгоритмом тощо.

У 1975 році Холланд опублікував теорему схем у своїй книзі [229], яка є аналітичним обґрунтуванням збіжності генетичних алгоритмів із фіксованим набором параметрів і запропонував основну ідею генетичних операторів і репродуктивний план, а Д. Гольдберг у праці [230] вперше використав термін «генетичний алгоритм», виклав теорію і можливі сфери застосування генетичних алгоритмів, тобто здійснив значний внесок у розвиток теорії Джона Холланда [164].

Метод оптимізації «зграєю» або «роєм» частинок у 1995 році був запропонований вченими Дж. Кеннеді і Д.К. Еберхарт [231]. Алгоритм «рою часток», як і «генетичний» алгоритм, моделює деякі закономірності, що існують у природі, наприклад, поведінка групи птахів або риб у пошуках корму. Алгоритм рою часток принципово відрізняється від генетичного алгоритму відсутністю базових генетичних операторів (селекція, мутація, схрещування).

З основних алгоритмів, які імітують поведінку живих біологічних істот, доцільно виділити такі алгоритми: алгоритм «зграї вовків» (Grey Wolf Optimizer, GWO), що імітує їх полювання [221, 225]; алгоритм «світлячків» (Firefly Algorithm, FFA), що імітує їх політ; алгоритм «пошуку зозулі» (Cuckoo Search Algorithm, CSA), що базується на процесі відкладання яєць; алгоритм «кажанів» (Bat Algorithm, BA), що імітує їх полювання [222, 223]; «косяк риб», що імітує їх пошук та дії при годуванні з використанням методів зчитування інформації боковою лінією [224, 232, 233]; «мурашина колонія», імітує їх поведінку збору їжі та їх здатність формувати оптимальні шляхи [219, 220]; «бджолоїної колонії», імітує їх поведінку пошуку та збору їжі [234–236].

Авторами [237] запропоновано комбінований підхід нелінійного цілочислового програмування для вирішення задачі планування роботи паралельних машин із попередженням та затримкою, а також з одночасним впливом навчального та лінійного спуску, які залежать від послідовностей виконання робіт та загального терміну роботи. Запропоновано підхід, за якого час виконання роботи визначався зростом функції початку часу її виконання та положенням у черзі виконання. Зокрема, розроблену модель застосовано до проблеми планування із двома машинами та одинадцятьма роботами.

Однією з основних проблем конструювання ройових алгоритмів є забезпечення балансу між інтенсивністю пошуку (швидкістю збіжності алгоритму) і шириною пошуку (диверсифікацією пошуку) [164].

Основними критеріями ефективності ройових алгоритмів є надійність алгоритму – оцінка ймовірності локалізації глобального екстремуму, а також швидкість його збіжності – оцінка математичного очікування необхідного числа випробувань (обчислень значення функції, що оптимізується) [164].

Особливостями задач глобальної оптимізації, що виникають у даний час у багатьох додатках, є [164]:

- нелінійність;
- недиференційованість;
- багатокритеріальність;
- наявність локальних мінімумів;
- відсутність аналітичного виразу (погана формалізація) і висока обчислювальна складність функцій оптимальності;
- висока розмірність простору пошуку;
- складна топологія області допустимих значень тощо.

Розрізняють траєкторні і популяційні алгоритми пошукової глобальної оптимізації. Траєкторні алгоритми (single-state algorithms) припускають оновлення на кожній ітерації положення лише одного кандидата у вирішенні завдання. При цьому загальне число кандидатів може бути більшим одиниці, і на різних ітераціях можуть переміщатися різні кандидати.

## 1.4 Постановка завдань досліджень і розробок

Важливість задач прийняття рішень та їх складність на різних рівнях управління обумовлює необхідність створення інформаційної технології прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів. Створення та використання інформаційних технологій прийняття рішень забезпечить вирішення більшості задач управління й підвищить їх якість, що відрізняється від теперішніх можливостей запобігати виникненню позаштатних ситуацій, зменшити неефективне використання технологічного обладнання, витрати основних ресурсів, мінімізувати витрати на постачання та зберігання матеріально-технічних ресурсів, також готової продукції, зменшити простой технологічного обладнання, підтримувати ефективний режим роботи та зменшити ймовірність виникнення критичних ситуацій.

Усе вищезазначене обумовило вибір теми дисертації, встановлення її мети та завдань. Планування виконання замовлень на основі використання класичних, евристичних та еволюційних методів розглядалося різними науковцями, але задача досі залишається актуальною.

У сучасних умовах жорсткого ринку необхідно утримати споживача, що можливо тільки за рахунок виготовлення відповідної продукції або надання послуги за мінімальний час, а також з максимальною мінімізацією витрат на це. Така задача гостро стоїть перед підприємствами харчової галузі, тому що для виготовлення високоякісної продукції із вчасним виконанням замовлення необхідно забезпечити закупку та використання якісної сировини; у разі необхідності забезпечити якісне зберігання сировини, яка має обмежені терміни використання; виготовити вчасно якісну продукцію з дотриманням усіх необхідних технологічних вимог; забезпечити зберігання виготовленої продукції, а також якісне постачання. Кожен із наведених етапів при збільшенні часу на його виконання призводить до збільшення витрат, які

можуть миттєво змінюватись в умовах невизначеності, але вартість кінцевого продукту чи послуги частіше є фіксованою величиною, що, у свою чергу, може призвести до банкрутства. Більшість підприємств, що відносяться до малих чи середніх, намагаються зменшити обсяги виготовлення, змінити терміни виконання, працювати за довгостроковими договорами, а також намагаються працювати в режимі початкового отримання певної кількості замовлень, виконання яких буде незбитковим, а потім їх виконувати. Все це призводить до втрати клієнтів-споживачів, які знаходять альтернативу продукції чи послугі, яка буде виконана вчасно й має відповідну якість.

Здійснення управління усіма ланками підприємства повинно бути узгодженим, що неможливо досягти, використовуючи стандартні методи управління. Їх застосування призводить до втрат часових ресурсів, що не відповідає потребі управління в реальному часі. Все це обумовлює розробку та створення інтелектуальних web-орієнтованих інформаційних систем нового покоління, які використовують евристичні, еволюційні та мультиагентні методи й підходи до інтелектуальної оптимізації, в основі яких лежить моделювання колективного інтелекту суспільних тварин, комах та інших живих істот. За рахунок використання сучасних інформаційних систем такі методи дають швидкі результати у розв'язанні різних задач оптимізації, що свідчить про перспективність цього напрямку. Особливо важливим є напрям розробки інформаційних технологій на основі модифікованих алгоритмів та методів.

Таким чином, вдосконалення інформаційних технологій необхідно доцільно здійснювати у двох напрямках:

- 1) розроблення нових модифікованих алгоритмів, що дозволить вирішувати складні багатокритеріальні задачі прийняття рішення;
- 2) розроблення нових підходів та методологій щодо створення гібридних систем підтримки прийняття рішень.

У зв'язку з тим, що реалізація інформаційних технологій не може бути відокремленою без інформаційної системи, а також, враховуючи поставлену задачу,

необхідно її реалізувати через побудову web-орієнтованої СППР для управління харчовим виробництвом із такими характеристиками:

- ієрархічність планування – система повинна вирішувати задачу планування в комплексі, а саме мати можливість застосування для попереднього (стратегічного), узгодженого (тактичного) та точного (оперативного) планування для забезпечення єдності прийнятих на різних рівнях управління рішень;

- багатокритеріальність – наявність та можливість вибору різних критеріїв оптимізації плану, таких як максимізація прибутку, мінімізація штрафів за запізнення виконання, мінімізація загального часу виконання замовлень у межах директивних термінів з урахуванням доступності ресурсів;

- можливість простого та доступного переходу від одного критерію до іншого, введення додаткових обмежень, адаптація до різних систем управління. Для ефективного функціонування системи та комплексного вирішення поставлених задач необхідне використання комплексу алгоритмів на основі єдиної логіки вирішення за різними критеріями оптимальності;

- адекватність реальному виробництву – відображення обмеженості ресурсів, фактичного завантаження виробничих підрозділів, взаємозв'язку між виконанням різних етапів робіт, нерівномірності замовлень у планових періодах та неодночасності надходження замовлень.

## **1.5 Висновки до розділу 1**

1. Проведено аналіз сучасних методів та підходів щодо управління виробництвом. Детально розглянуті методи та підходи щодо управління виробництвом: «ощадливого виробництва»; стратегія «Кайдзен»; розробка потоку створення цінності; карта потоку створення цінності; система швидкого переналадження; система «Канбан»; система бездефектного виготовлення товарів.

Проведено аналіз існуючої інформаційної системи управління підприємством, а саме розглянута роль систем SCADA, MES, APS, СППР.

2. Використовуючи системний аналіз та методології структурного аналізу й проєктування SADT, проведено дослідження усіх бізнес-процесів та інформаційних потоків, що їх забезпечують, а саме були розроблені такі функціональні моделі: «Організація та управління макаронним виробництвом для ВАТ «Макаронна фабрика»», «Організація роботи відділу планування компанії «Danone»», «Організація та контроль виготовлення продукції на ТОВ «Продеко»», «Організація виконання замовлень на ТОВ «Слобожанський бекон»». Кожен рівень управління підприємством потребує врахування складного ієрархічного впливу по вертикалях рівнів управління. Проведений аналіз наукових напрямів оптимізації процесів управління харчовим підприємством дає підстави стверджувати про актуальність створення інформаційної технології прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів.

3. На основі проведеного аналізу літературних джерел були сформульовані такі висновки. Здійснення управління усіма ланками підприємства повинно бути узгодженим, що неможливо досягти, використовуючи стандартні методи управління. Використання стандартних методів призводить до витрат часових ресурсів, що не відповідає потребі управління в реальному часі. Все це обумовлює розробку та створення інтелектуальних web-орієнтованих інформаційних систем нового покоління, які використовують евристичні, еволюційні та мультиагентні методи й підходи до інтелектуальної оптимізації, в основі яких лежить моделювання колективного інтелекту суспільних тварин, комах та інших живих істот. За рахунок використання сучасних інформаційних систем такі методи дають швидкі результати у розв'язанні різних задач оптимізації, що свідчить про перспективність цього напрямку. Особливо важливим є напрям розробки інформаційних технологій на основі модифікованих алгоритмів та методів.

4. Визначена така актуальна науково-прикладна проблема: розробка інформаційних технологій прийняття рішень при управлінні харчовим підприємством на основі модифікованих евристичних і еволюційних методів та

алгоритмів із метою забезпечення скорочення часу на прийняття рішень, зменшення впливу людського фактора та зниження витрат та часу на виконання замовлень щодо виготовлення продукції харчовим підприємством.

5. Для досягнення мети вирішувалися такі основні завдання:

- провести аналіз існуючих підходів, методів та інформаційних технологій для управління харчовим підприємством;
- здійснити математичне моделювання системи управління виробництвом з урахуванням зовнішніх та внутрішніх впливів, а також розробити системи їх моніторингу та сформулювати правила адаптивного керування;
- розробити модифіковані методи та алгоритми для вирішення задачі управління харчовим підприємством;
- розробити інформаційні технології на основі модифікованих евристичних і еволюційних методів та алгоритмів для здійснення гнучкого управління харчовим виробництвом;
- розробити структуру гібридної web-орієнтованої системи підтримки прийняття рішень;
- розробити рекомендації щодо створення web-орієнтованої системи підтримки прийняття управлінських рішень на базі модифікованих евристичних і еволюційних методів та алгоритмів;
- визначити ефективність функціонування створеної інформаційної технології шляхом її апробації на діючих підприємствах.

Основні результати розділу опубліковані автором у працях [3, 6, 7, 8, 9, 13, 19].

## РОЗДІЛ 2

### ФОРМАЛІЗАЦІЯ ЗАДАЧ УПРАВЛІННЯ

#### 2.1 Створення математичної моделі планування виконання замовлень

Основною задачею управління, що потребує врахування завдання та обмежень усіх рівнів, є оперативне коригування плану виготовлення продукції для виконання замовлень. Як правило, вирішення такої задачі зводилось декомпозицією до потрібного рівня. При цьому враховувалися лише необхідні критерії та функції з обмеженнями, але втрачався зв'язок усіх критеріїв впливу.

План виробництва продукції є оптимальним, якщо його виконання забезпечує отримання максимального прибутку за заданий проміжок часу. Оптимальний план виконання замовлень не порушує загальний стратегічний план діяльності підприємства, мінімізує змінні витрати, дозволяє максимально використати виробничо-технологічне обладнання. Оперативна реконфігурація плану виготовлення продукції виникає у таких випадках:

- необхідність включення у поточний план виготовлення замовлення, яке тільки надійшло і має високий пріоритет виконання;
- виникнення позаштатних ситуацій, пов'язаних з порушенням графіка надходження сировини та матеріалів, експлуатації та стану технологічного обладнання, з порушенням умов договорів постачальниками;
- соціально-економічні потреби, пов'язані з сезонними та політичними вподобаннями споживачів;
- зміни у стратегічному напрямі діяльності підприємства.

Дана задача відноситься до класу ієрархічних багатокритеріальних оптимізаційних задач, включає різні часткові критерії, які в сукупності впливають на вибір оптимального рішення і використання яких визначається конкретними умовами [4, 12, 18, 22, 48].

Для підприємств харчової галузі виділяють такі часткові критерії:

- максимізація прибутку від виконання усіх замовлень;

- мінімізація часу на виготовлення продукції;
- максимізація запасу часу при виконанні замовлень;
- мінімізація сумарних штрафів за невчасне виконання замовлення;
- мінімізація сумарного часу простою усіх одиниць технологічного обладнання та відділів;
- мінімізація сумарних витрат при виробництві;
- мінімізація сумарних витрат від простоїв при невикористанні обладнання, коли деякі технологічні контури, не зважаючи на невикористання, потребують електро- та тепло-споживання;
- мінімізація сумарних витрат на переробку та утилізацію отриманої некондиційної продукції при виконанні усіх замовлень;
- мінімізація сумарних витрат на зберігання готової продукції, а іноді і на транспортування;
- мінімізація сумарних витрат на зберігання сировини та матеріалів.

У залежності від соціальної та економічної ситуації, а також особливостей менеджера, який відповідає за складання оперативно-календарного плану виконання замовлення, задача може вирішуватися за різними варіантами, а саме [18, 22]:

- в оцінці ефективності оперативно-календарного плану виконання замовлення враховуються усі критерії, а також вони ранжуються;
- задача спрощується до вибору певних часткових критеріїв.

Для критеріїв, які оцінюються у грошових одиницях, застосовується адитивна згортка із встановленням пріоритетів для кожного з них у відповідності за конкретним випадком [22, 30, 32].

Загальна оціночна функція буде представлена адитивною згорткою усіх критеріїв (2.1):

$$F'_0 = \lambda_1 F_1 - \sum_{\gamma=5}^{10} \lambda_\gamma F_\gamma \rightarrow \max, \quad (2.1)$$

де  $\lambda_\gamma$  – коефіцієнт важливості критерію  $\lambda_\gamma \in (0,1)$ ,  $\sum \lambda_\gamma = 1$ .

У залежності від соціальної та економічної ситуації, а також досвіду менеджера, який відповідає за складання оперативного плану виконання замовлення, задача може вирішуватись за різними варіантами.

Отримання максимального прибутку від виконання замовлень на виготовлення продукції за заданий часовий проміжок є першим частковим критерієм оцінки ефективності варіанта плану (2.2) на плановий період  $(t + \Delta t)$  [22].

$$F_1(t + \Delta t) = \sum_{i=1}^n (\theta_i * (sd_i(t + \Delta t) - (vc_i(t + \Delta t) + vz_i(t + \Delta t))) * op_i(t + \Delta t)) \rightarrow \max, \quad (2.2)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – замовлення, що знаходиться на черзі, регламентує один вид продукції, яку необхідно випустити у період  $(t + \Delta t)$ , а також регламентує вид упаковки та фасування;

$n$  – загальна кількість замовлень, яку необхідно виконати за період  $(t + \Delta t)$ ;

$\theta_i$  – параметр, який приймає значення  $\{0,1\}$  ( $\theta_i = 1$ , якщо  $i$ -е замовлення виконується за період  $(t + \Delta t)$ ;  $\theta_i = 0$  в іншому випадку);

$sd_i(t + \Delta t)$  – вартість одиниці продукції для  $i$ -го замовлення за період  $(t + \Delta t)$ ;

$vc_i(t + \Delta t)$  – постійні витрати на виготовлення одиниці продукції для  $i$ -го замовлення за період  $(t + \Delta t)$ ;

$vz_i(t + \Delta t)$  – змінні витрати на виготовлення одиниці продукції для  $i$ -го замовлення за період  $(t + \Delta t)$ ;

$op_i(t + \Delta t)$  – обсяг готової продукції, яку необхідно виготовити за  $i$ -м замовленням у період  $(t + \Delta t)$ .

Мінімізація часу на виготовлення продукції кожного  $i$ -го – замовлення за період  $(t + \Delta t)$  є частковим критерієм і впливає на загальну ефективність варіанта плану (2.3), а обмеження (2.4) регламентує виготовлення продукції за період  $(t + \Delta t)$ , обмеження (2.5) обумовлює закінчення виготовлення продукції не пізніше визначеного терміну.

$$F_{2i}(t + \Delta t) = \sum_{j=1}^{\omega_i} \sum_{l=1}^{\sigma_i} \left( o_{ijl} * (pt_{ijl} + t_{ijl} + \eta t_{ijl} + to_{ij-1}) \right) \rightarrow \min, \quad (2.3)$$

$$t \leq t_i + F_{2i}(t + \Delta t) \leq t + \Delta t, \quad (2.4)$$

$$t \leq t_i + F_{2i}(t + \Delta t) \leq dt_i, \quad (2.5)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – замовлення, що знаходиться на черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$j$  – номер етапу із множини етапів ( $j \in \omega_i$ ) для  $i$ -го замовлення;

$\omega_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення;

$l$  – номер обладнання із множини обладнання ( $l \in \sigma_i$ ) для  $i$ -го замовлення;

$\sigma_i$  – кількість задіяного обладнання для виконання усіх етапів при виготовленні  $i$ -го замовлення;

$t_i$  – час початку виконання продукції за  $i$ -м замовленням;

$dt_i$  – час, на який необхідно виготовити продукцію за  $i$ -м замовленням;

$\Delta t_{ijl}$  – час виконання здійснення  $j$ -го етапу на  $l$ -му обладнанні для виготовлення продукції за  $i$ -м замовленням;

$pt_{ijl}$  – час, необхідний для підготовки  $l$ -го обладнання для здійснення  $j$ -го етапу при виготовленні продукції за  $i$ -м замовленням, може приймати нуль, якщо підготовка не потрібна;

$to_{ij-1}$  – час переходу/очікування між виконанням  $j$ -го етапу до  $(j-1)$  етапу;

$\eta t_{ijl}$  – час для очищення обладнання після  $j$ -го етапу виготовлення продукції за  $i$ -м замовленням на  $l$ -му обладнанні;

$o_{ijl}$  – параметр, що приймає значення  $\{0,1\}$  ( $o_{ijl} = 1$ , якщо  $j$ -й етап можливо виконати на  $l$ -му обладнанні для виготовлення продукції за  $i$ -м замовленням;  $o_{ijl} = 0$  – в іншому випадку).

Максимізація запасу часу при виконанні кожного  $i$ -го замовлення за період  $(t + \Delta t)$  є частковим критерієм для певного замовлення і впливає на загальну ефективність варіанта плану (2.6).

$$F_{3i}(t + \Delta t) = \begin{cases} dt_i - (t_i + F_{2i}(t + \Delta t)), & \text{якщо } (t_i + F_{2i}(t + \Delta t)) < dt_i \\ 0, & \text{в іншому випадку} \end{cases} \rightarrow \max, \quad (2.6)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – номер замовлення, що знаходиться на черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$j$  – номер етапу з множини етапів ( $j \in \omega_i$ ) для  $i$ -го замовлення;

$\omega_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення;

$t_i$  – час початку виконання продукції за  $i$ -м замовленням;

$dt_i$  – час, на який необхідно виготовити продукцію за  $i$ -м замовленням.

Мінімізація сумарного часу невикористання (простою) усіх одиниць технологічного обладнання та відділів у цілому, який визначається за формулою (2.7) [4, 18, 22]:

$$F_4(t + \Delta t) = \sum_{i=1}^n \sum_{j=1}^{\omega_i} \sum_{l=1}^{\sigma_i} (o_{ijl} * (t z l_{i(j-1),l} - t p l_{ij,l})) \rightarrow \min, \quad (2.7)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – замовлення, що знаходиться на черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$j$  – номер етапу із множини етапів ( $j \in \omega_i$ ) для  $i$ -го замовлення;

$\omega_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення;

$l$  – номер обладнання із множини обладнання ( $l \in \sigma_i$ ) для  $i$ -го замовлення;

$\sigma_i$  – кількість задіяного обладнання для виконання усіх етапів при виготовленні  $i$ -го замовлення;

$n$  – загальна кількість замовлень, які необхідно виконати за період  $(t + \Delta t)$ ;

$t z l_{i(j-1),l}$  – час завершення виконання  $(j-1)_l$ -го етапу на  $l$ -му обладнанні при виготовленні продукції за  $i$ -м замовленням;

$t p l_{ij,l}$  – час початку виконання  $j_l$ -го етапу на  $l$ -му обладнанні при виготовленні продукції за  $i$ -м замовленням;

$o_{ijl}$  – параметр, який приймає значення  $\{0,1\}$  ( $o_{ijl} = 1$ , якщо  $j$ -й етап виконується на  $l$ -му обладнанні при виготовленні продукції за  $i$ -м замовленням;  $o_{ijl} = 0$  в іншому випадку).

Сумарний розмір штрафів за невчасне виконання замовлення у плановому періоді, який визначається за формулою (2.8):

$$F_5(t + \Delta t) = \sum_{i=1}^n (g_i * \Psi_i * hk_i) \rightarrow \min, \quad (2.8)$$

де  $(t + \Delta t)$  – плановий період на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – замовлення, що знаходиться на черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$g_i$  – розмір штрафу, прописаний у договорі, який необхідно відшкодувати замовнику, якщо буде порушено  $dt_i$  - термін виконання замовлення;

$\Psi_i$  – коефіцієнт розраховується за формулою (2.9) й визначає необхідність відшкодування замовнику, якщо буде порушено термін виконання  $dt_i$ .

$$\Psi_i = \begin{cases} 1, & \text{якщо } (t_i + F_{2i}(t + \Delta t)) < dt_i, \\ 0, & \text{в іншому випадку,} \end{cases} \quad (2.9)$$

де  $t_i$  – час початку виконання виготовлення продукції за  $i$ -м замовленням;

$hk_i$  – коефіцієнт, який враховує необхідність сплати штрафу за запізнення за кожну добу і розраховується за формулою (9):

$$hk_i = \begin{cases} 1, & \text{якщо розмір штрафу фіксований і не залежить від кількості діб} \\ \left\lceil \frac{F_{2i}(t + \Delta t)}{24} \right\rceil, & \text{в іншому випадку визначаємо кількість діб} \end{cases} \quad (2.10)$$

Шостим критерієм є значення сумарних витрат, які неминучі при виконанні  $j$ -го етапу на  $l$ -му обладнанні при виготовленні продукції за  $i$ -м замовленням за одиницю часу, адже якщо технологічна ділянка виступає як окремий цех, тоді

враховується увесь час експлуатації, що розраховується за формулою (2.11) [22, 30, 32]:

$$F_6(t + \Delta t) = \sum_{i=1}^n \left( \sum_{j=1}^{\omega_i} \sum_{l=1}^{\sigma_i} (o_{ijl} * (pt_{ijl} + \Delta t_{ijl} + \eta t_{ijl})) * c_{ijl} \right) \rightarrow \min, \quad (2.11)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – номер замовлення, що знаходиться на черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$j$  – номер етапу із множини етапів ( $j \in \omega_i$ ) для  $i$ -го замовлення,

$\omega_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення;

$l$  – номер обладнання із множини обладнання ( $l \in \sigma_i$ ) для  $i$ -го замовлення;

$\sigma_i$  – кількість задіяного обладнання для виконання усіх етапів при виготовленні  $i$ -го замовлення;

$\Delta t_{ijl}$  – час виконання здійснення  $j$ -го етапу на  $l$ -му обладнанні для виготовлення продукції за  $i$ -м замовленням;

$pt_{ijl}$  – час, необхідний для підготовки  $l$ -го обладнання для здійснення  $j$ -го етапу при виготовленні продукції за  $i$ -м замовленням, може приймати нуль, якщо підготовка не потрібна;

$c_{ijl}$  – витрати за одну годину при здійсненні  $j$ -го етапу на  $l$ -му обладнанні для виготовлення продукції за  $i$ -м замовленням;

$\eta t_{ijl}$  – час для очищення обладнання після  $j$ -го етапу виготовлення продукції за  $i$ -м замовленням на  $l$ -му обладнанні, якщо воно не потрібно або це необхідно перед проведенням ремонтно-профілактичних робіт;

$o_{ijl}$  – параметр, який приймає значення  $\{0,1\}$  ( $o_{ijl} = 1$ , якщо  $j$ -й етап можливо виконати на  $l$ -му обладнанні для виготовлення продукції за  $i$ -м замовленням;  $o_{ijl} = 0$  в іншому випадку).

Сьомим критерієм є мінімізація витрат на обладнання, яке простоює, але у випадку, коли в обладнанні є контури, що потребують постійного включення, він визначається за формулою (2.12) [22, 30, 32]:

$$F_7(t + \Delta t) = \sum_{i=1}^n \sum_{j=1}^{\omega_i} \sum_{l=1}^{\sigma_i} (NOT(o_{ijl})) * (tzl_{i(j-1)l} - tpl_{ijl}) * cp_l * vk_l \rightarrow \min, \quad (2.12)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – номер замовлення, що знаходиться на черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$j$  – номер етапу з множини етапів ( $j \in \omega_i$ ) для  $i$ -го замовлення,  $\omega_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення;

$l$  – номер обладнання із множини обладнання ( $l \in \sigma_i$ ) для  $i$ -го замовлення,  $\sigma_i$  – кількість задіяного обладнання для виконання усіх етапів при виготовленні  $i$ -го замовлення;

$n$  – загальна кількість замовлень, які необхідно виконати за період  $(t + \Delta t)$ ;

$tzl_{i(j-1)l}$  – час завершення виконання  $(j-1)$ -го етапу на  $l$ -у обладнанні при виготовленні продукції за  $i$ -м замовленням;

$tpl_{i(j-1)l}$  – час початку виконання  $j$ -го етапу на  $l$ -у обладнанні при виготовленні продукції за  $i$ -м замовленням;

$vk_l$  – параметр, що приймає значення  $\{0,1\}$  ( $vk_l = 1$ , якщо  $l$ -е обладнання при простій має постійно включені елементи;  $vk_l = 0$  в іншому випадку).

$cp_l$  – витрати при простої  $l$ -го обладнання;

$o_{ijl}$  – параметр, який приймає значення  $\{0,1\}$  ( $o_{ijl} = 1$ , якщо  $j$ -й етап можливо виконати на  $l$ -у обладнанні для виготовлення продукції за  $i$ -м замовленням;  $o_{ijl} = 0$  – в іншому випадку).

Восьмим критерієм є мінімізація витрат на переробку та утилізацію отриманої некондиційної продукції при виконанні усіх замовлень, він визначається за формулою (2.13):

$$F_8(t + \Delta t) = \sum_{i=1}^n \sum_{j=1}^{\omega_i} \sum_{l=1}^{\sigma_i} (o_{ijl} * cn_{ijl} * vnk_{ijl} * op_i(t + \Delta t)) \rightarrow \min, \quad (2.13)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – номер замовлення, що знаходиться у черзі на виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$j$  – номер етапу із множини етапів ( $j \in \omega_i$ ) для  $i$ -го замовлення;

$\omega_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення;

$l$  – номер обладнання із множини обладнання ( $j \in \sigma_i$ ) для  $i$ -го замовлення;

$\sigma_i$  – кількість задіяного обладнання для виконання усіх етапів при виготовленні  $i$ -го замовлення;

$n$  – загальна кількість замовлень, які необхідно виконати за період  $(t + \Delta t)$ ;

$cn_{ijl}$  – вартість перероблення чи утилізації одиниці отриманої некондиційної продукції при виконанні  $i$ -го замовлення на  $l$ -му технологічному обладнанні;

$op_i(t + \Delta t)$  – обсяг готової продукції, яку необхідно виготовити за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$vnk_{ijl}$  – загальна кількість некондиційної продукції, отриманої на кожну одиницю продукції під час виконання  $j$ -го етапу  $i$ -го замовлення на  $l$ -му технологічному обладнанні;

$o_{ijl}$  – параметр, який приймає значення  $\{0,1\}$  ( $o_{ijl} = 1$ , якщо  $j$ -й етап можливо виконати на  $l$ -му обладнанні для виготовлення продукції за  $i$ -м замовленням;  $o_{ijl} = 0$  – в іншому випадку).

Дев'ятим критерієм є мінімізація витрат на зберігання готової продукції, який визначається за формулою (2.14):

$$F_9(t + \Delta t) = \sum_{i=1}^n (vz_i * op_i(t + \Delta t) * \max(0, dt_i - (t_i + F_{2i}(t + \Delta t)))) \rightarrow \min, \quad (2.14)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – номер замовлення, що знаходиться у черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$op_i(t + \Delta t)$  – обсяг готової продукції, яку необхідно виготовити за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$vz_i$  – вартість зберігання продукції за  $i$ -м замовленням;

$dt_i$  – час, на який необхідно виготовити продукцію за  $i$ -м замовленням.

У випадку, коли підприємство здійснює постачання до замовників самостійно або здійснює зберігання готової продукції на проміжних-регіональних складах, критерій 9 перетворюється на (2.15), що описує сумарні витрати на зберігання, перевезення продукції від заводу до регіональних складів, а потім до замовників для кожного  $i$ -го виду продукції, а загальні сумарні витрати за період будуть мати вигляд (2.16) [22, 48]:

$$F_{9i}(t + \Delta t) = \sum_{z_i=1}^{Z_i} (vzgt_{itz} * zgt_{itz}) + \sum_{z_i=1}^{Z_i} \sum_{r=1}^R (cgt_{itzr} * v cgt_{itzr}) + \sum_{r=1}^R (vzgt_{itr} * zgt_{itr}) + \sum_{r=1}^R \sum_{b=1}^B (cgt_{irb} * v cgt_{irb}) \rightarrow \min, \quad (2.15)$$

$$F_9(t + \Delta t) = \sum_{i=1}^n F_{9i}(t + \Delta t) \rightarrow \min, \quad (2.16)$$

де  $z_i$  – завод, на якому виготовляється  $i$ -й вид продукції;

$Z_i$  – кількість заводів, де виготовляється  $i$ -й вид продукції;

$r$  – регіональний склад;

$R$  – кількість регіональних складів;

$b_i$  – замовник, який замовив  $i$ -й вид продукції;

$B_i$  – кількість замовників, які замовили  $i$ -й вид продукції;

$zgt_{itz}$  – вартість зберігання одиниці  $i$ -го виду продукції на заводі  $z$  за момент часу  $t$ ;

$vzgt_{itz}$  – обсяг  $i$ -го виду продукції, що зберігається на заводі  $z$  за момент часу  $t$ ;

$cgt_{itzr}$  – вартість перевезення одиниці  $i$ -го виду продукції з заводу  $z$  до регіонального складу  $r$  за момент часу  $t$ ;

$v cgt_{itzr}$  – обсяг  $i$ -го виду продукції, що транспортується з заводу  $z$  до регіонального складу  $r$  за момент часу  $t$ ;

$zgt_{itr}$  – вартість зберігання одиниці  $i$ -го виду продукції на регіональному складі  $r$  за момент часу  $t$ ;

$vzgt_{itr}$  – обсяг  $i$ -го виду продукції, що зберігається на регіональному складі  $r$  за момент часу  $t$ ;

$cgt_{irb}$  – вартість перевезення одиниці  $i$ -го виду продукції з регіонального складу  $r$  до замовника  $b$  за момент часу  $t$ ;

$v cgt_{irb}$  – обсяг  $i$ -го виду продукції, що транспортується з регіонального складу  $r$  до замовника  $b$  за момент часу  $t$ .

Обмеження на вміст складу заводу  $z$  для  $i$ -го виду продукції на момент часу  $t$  визначається за формулою (2.17):

$$\sum_{i=1}^I v_z g_{itz} \leq V_{zi}, \quad (2.17)$$

де  $i$  – номер замовлення, що регламентує один вид продукції;

$r$  – регіональний склад;

$v_z g_{itz}$  – обсяг  $i$ -го виду продукції, що зберігається/надходить у момент часу

$t$ ;

$I$  – загальна кількість замовлень однакової продукції;

$V_{zi}$  – загально допустимий обсяг продукції  $i$ -го виду, що може зберігатися на складі заводу  $z$  одночасно.

Обмеження на вміст регіонального складу  $r$  на момент часу  $t$  визначається за формулою (2.18):

$$\sum_{i=1}^I v_r g_{itr} \leq V_{ri}, \quad (2.18)$$

де  $i$ -й вид продукції;

$v_r g_{itr}$  – обсяг  $i$ -го виду продукції, що зберігається/надходить на момент часу

$t$ ;

$I$  – загальна кількість замовлень однакової продукції;

$V_{ri}$  – загально допустимий обсяг продукції  $i$ -ого виду, що може зберігатися на регіональному складі  $r$  на момент часу  $t$ .

Десятим критерієм є мінімізація витрат на зберігання сировини та матеріалів, що необхідні для виготовлення продукції, який визначається за формулою (2.19) [22, 48]:

$$F_{10}(t + \Delta t) = \sum_{i=1}^n \sum_{k=1}^{r_i} (vsr_{ik} * rk_{ik} * op_i(t + \Delta t) * \max(0, t_i - rt_{ik}) \rightarrow \min, \quad (2.19)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$t$  – час початку виконання плану, який позначається датою та часом до хвилини;

$\Delta t$  – час, за який необхідно виконати усі замовлення у хвилинах;

$i$  – замовлення, що знаходиться у черзі виконання, регламентує один вид продукції, яку необхідно випустити за період  $(t + \Delta t)$ ;

$k$  – вид ресурсу, який необхідний для виконання  $i$ -го замовлення;

$r_i$  – перелік необхідних ресурсів для виготовлення продукції за  $i$ -м замовленням;

$vsr_{ik}$  – вартість зберігання  $k$ -го компонента для виконання  $i$ -го замовлення;

$rk_{ik}$  – обсяг сировини, необхідний для виготовлення одиниці продукції за  $i$ -м замовленням;

$op_i(t + \Delta t)$  – обсяг готової продукції, яку необхідно виготовити за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$t_i$  – час початку виготовлення продукції за  $i$ -м замовленням;

$rt_{ik}$  – час надходження  $k$ -го компонента для виконання  $i$ -го замовлення.

Для представленої математичної моделі висувуються додаткові умови та обмеження, які враховують усі особливості задачі:

1. Кожне  $l$ -е технологічне обладнання або виробничий підрозділ може виконувати не більше одного  $j$ -го етапу  $i$ -го замовлення, що описується умовою в формулі (2.20):

$$\sum_{j=1}^{\sigma_i} o_{ijt} = 1. \quad (2.20)$$

2. Для того, щоб отримати готову продукцію в повному обсязі та асортименті згідно з плановими замовленнями, усі етапи повинні бути виконані з використанням потрібного технологічного обладнання за виразом (2.21):

$$\sum_{j=1}^J \sum_{l=1}^L o_{ijl} = \omega_i, \quad (2.21)$$

де  $l$  – технологічне обладнання, що належить до загальної множини технологічного обладнання  $L$ ;

$j$  – технологічний етап, що входить до загальної множини етапів  $J$ .

3. Для кожного  $i$ -го замовлення необхідно враховувати те, що кожен етап технологічного процесу може розпочатися лише після завершення усіх попередніх етапів. Якщо  $o_{i(j-1)(l-1)} = 1$  та  $o_{ijl} = 1$ , тоді час початку виконання  $l$ -го етапу визначається виразом (2.22):

$$\tau_{i(j-1)(l-1)} + \Delta\tau_{i(j-1)(l-1)} \leq \tau_{ijl}, \quad (2.22)$$

де  $\tau_{i(j-1)(l-1)}$  – час початку виконання  $(j-1)$  попереднього етапу  $i$ -го замовлення на  $(l-1)$  попередньому технологічному обладнанні;

$\Delta\tau_{i(j-1)(l-1)}$  – сумарний час експлуатації  $(l-1)$  попереднього технологічного обладнання для виконання  $(j-1)$  попереднього етапу  $i$ -го замовлення;

$\tau_{ijl}$  – час початку виконання  $j$ -го поточного етапу  $i$ -го замовлення на  $l$ -му технологічному обладнанні.

4. Обмеження (2.4) регламентує виготовлення продукції за період  $(t + \Delta t)$  і не може виходити за ці межі.

5. Обмеження (2.5) обумовлює закінчення виготовлення продукції не пізніше визначеного терміну встановленого замовником.

6. Час початку виконання  $i$ -го замовлення на  $l$ -му технологічному обладнанні може бути задано константою чи проміжком, що визначається показниками роботи конкретної бригади при фасуванні/пакуванні або при виготовленні специфічного виду продукції і описується виразом (2.23):

$$o_{ijl} = const \quad \text{або} \quad o_{ijl} \in [a, b], \quad a, b - const. \quad (2.23)$$

Наприклад, виготовлення  $i$ -го замовлення повинно припадати тільки на другу зміну, тобто :  $o_{ijl} \in [16:15..23:55]$ .

7. У випадку, якщо на плановий період  $(t + \Delta t)$  передбачається виконання ремонтно-профілактичних робіт на певному обладнанні, тоді плановий період розбивається на два проміжки: до та після виконання таких робіт.

8. Усі види ресурсів для виконання  $i$ -го замовлення повинні бути у повному обсязі, а їх надходження повинно бути не пізніше початку етапу для виконання  $i$ -го замовлення, що визначається виразом (2.24):

$$\sum_{k=1}^{r_i} rk_{ik} * op_i(t + \Delta t) \geq rkz_k, \quad (2.24)$$

$$trp_{ik} \leq \tau_{ijl},$$

де  $\tau_{ijl}$  – час початку виконання  $j$ -го етапу  $i$ -го замовлення на  $l$ -му технологічному обладнанні;

$k$  – вид ресурсу необхідного для виконання  $i$ -го замовлення;

$r_i$  – перелік ресурсів, необхідних для виготовлення продукції за  $i$ -м замовленням;

$op_i(t + \Delta t)$  – обсяг готової продукції, яку необхідно виготовити за  $i$ -м замовленням у період  $(t + \Delta t)$ .

$rk_{ik}$  – обсяг сировини, необхідний для виготовлення одиниці продукції за  $i$ -м замовленням;

$rkz_k$  – обсяг сировини  $k$ -го виду ресурсу, що надійшов у момент часу  $trp_{ik}$ ;

$trp_{ik}$  – час надходження  $k$ -го виду ресурсу до початку виконання  $j$ -го поточного етапу  $i$ -го замовлення на  $l$ -му технологічному обладнанні.

9. Бувають випадки, коли необхідно зберегти повну або часткову пріоритетність виконання замовлення, тому можуть вводитися часткове або повне ранжування замовлень, а також чітка послідовність у вигляді нерівності, наприклад,  $u < p$ , де  $u$  та  $p$  – номери замовлень.

10. Обсяг виготовлення продукції не може перевищувати максимально можливу потужність підприємства в цілому і задається формулою (2.25):

$$MinOp_i \leq op_i(t + \Delta t) \leq MaxOp_i, \quad (2.25)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$op_i(t + \Delta t)$  – обсяг готової продукції, яку необхідно виготовити за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$MinOp_i$  – мінімально можливий обсяг виготовлення продукції за  $i$ -м замовленням у період  $(t + \Delta t)$ , який буде рентабельним;

$MaxOp_i$  – максимально можливий обсяг виготовлення продукції за  $i$ -м замовленням у період  $(t + \Delta t)$ .

11. Термін зберігання готової продукції не може перевищувати термін придатності до вживання та термін переробки і задається формулою (2.26):

$$\begin{aligned} tvug_i(t + \Delta t) &\leq tvug_i(t + \Delta t) + tper_i(t + \Delta t) \leq tvug_i(t + \Delta t) + tvidg_i(t + \Delta t) \leq \\ &\leq tvug_i(t + \Delta t) + tprid_i(t + \Delta t) \end{aligned} \quad (2.26)$$

де  $(t + \Delta t)$  – плановий період, на який розраховується виробничий план;

$tvug_i(t + \Delta t)$  – час виготовлення продукції, яку виготовили за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$per_i(t + \Delta t)$  – час, за який можливо здійснити переробку продукції, яку виготовили за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$tvidg_i(t + \Delta t)$  – час, через який можливо здійснити відвантаження готової продукції, що була виготовлена за  $i$ -м замовленням у період  $(t + \Delta t)$ ;

$tprid_i(t + \Delta t)$  – час придатності готової продукції, яку виготовили за  $i$ -м замовленням у період  $(t + \Delta t)$ .

12. Термін зберігання сировини не може перевищувати термін можливого використання і задається формулою (2.27):

$$tr_k \leq tr_k + tvug_k, \quad (2.27)$$

де  $tr_k$  – час надходження сировини;

$tvug_k$  – час, який може зберігатися сировина, яка надійшла у час  $tr_k$ .

13. Зберігання виготовленої продукції обмежено можливим обсягом зберігання (2.28):

$$pg_i(t) \leq Maxpg_i(t), \quad (2.28)$$

де  $pg_i(t)$  – обсяг  $i$ -го виду продукції, який виготовлено, але не відвантажено на момент часу  $t$ ;

$Maxpg_i(t)$  – максимальний обсяг  $i$ -го виду продукції, який може зберігатися в момент часу  $t$ .

14. Зберігання сировини обмежено можливим обсягом зберігання (2.29):

$$pr_k(t) \leq Maxr_k(t), \quad (2.29)$$

де  $pr_k(t)$  – обсяг  $k$ -го виду сировини, який присутній на момент часу  $t$ ;

$Maxr_k(t)$  – максимальний обсяг  $k$ -го виду сировини, який може зберігатися в момент часу  $t$ .

Розроблена математична модель дає змогу оцінити та побудувати оперативно-календарний план виконання замовлень, а дана задача належить до класу багатокритеріальних NP-складних комбінаторних задач [4, 18, 22, 30, 32].

Складність розв'язання такої задачі зростає з кількістю замовлень, а також зі збільшенням етапів різних варіантів виконання на різних технологічних ланках виробництва.

Задача може бути спрощена тільки на підприємствах, де технологічний процес виготовлення продукції відбувається на одному автоматизованому технологічному комплексі, що має конвеєрний безперервний цикл виробництва. Прикладом може бути макаронне виробництво, де виготовлення певного виду продукції здійснюється на одній автоматизованій технологічній лінії.

Необхідно відмітити, що частина обладнання, яка використовується на вітчизняних харчових підприємствах, потребує додаткових зусиль налаштування при впровадженні нових видів продукції. В такому випадку додаються витрати на налаштування та регулювання технологічного обладнання у відповідності з кожним процесом виготовлення продукції. Наприклад, процес приготування варених ковбасних виробів складається з підсушування, обсмажування, варіння. Усі зазначені етапи залежать від діаметра ковбасного виробу [11] й виконуються на одному технологічному обладнанні.

Більш детально математична модель процесу термічної обробки варених ковбасних виробів наведена у Додатку Е.

Запропонована математична модель може бути адаптована для вирішення задач планування виконання замовлень не тільки для харчових підприємств, а також для підприємств, діяльність яких пов'язана з наданням послуг чи виготовленням промислових товарів.

## 2.2 Створення математичної моделі планування постачання сировини на прикладі цукрового заводу

Частина підприємств переробної та харчової промисловості прив'язана до сировини, яка має свої властивості з обмеженнями на термін їх використання. Прикладом такого підприємства є цукрове виробництво. Особливістю цукрової промисловості є залежність техніко-економічних показників роботи від якості сировини. Як сировина можуть виступати цукровий буряк, кавуни та цукрова тростина. Планування перероблення закладається на рівні планування посіву культур, адже визрівання сировини відбувається нерівномірно й залежить від сировинної зони, якості сорту та гібридів цукрового насіння, термінів посіву, погодно-кліматичних умов вегетаційного періоду, збирання, транспортування та зберігання. Кожен вид гібридів відмінний за своїми генетико-детермінованими властивостями, термінами технологічного дозрівання, показниками врожайності, цукристості та лежкості. За рахунок постійного розвитку та впровадження нових гібридів необхідно постійно здійснювати перепланування послідовності постачання та переробки сировини [9].

Автори роботи [9] запропонували математичну модель виготовлення цукру з урахуванням показників сировини.

Основну мету функціонування виробничого процесу цукрового заводу доцільно визначити цільовою функцією (2.30):

$$f = \sum_{j=1}^{t_i} \sum_{i=1}^k (x_{ij} - x_{ij} \cdot Z_i) \cdot c_{ij} - \sum_{i=1}^k \bar{x}_i \cdot \bar{C}_i - \sum_{j=1}^{t_i} Tr_i \rightarrow \max, \quad (2.30)$$

де  $x_{ij}$  – обсяг постачання цукрових буряків з  $i$ -ої сировинної зони ( $i \in 1 \dots k$ ), за кожний  $j$ -й день надходження за весь період роботи цукрового заводу ( $j \in [1 \dots t_i]$ , від першого до останнього ( $t_i$ ) дня роботи надходження з  $i$ -ої сировинної зони);

$k$  – кількість сировинних зон, з яких надходить цукровий буряк;

$Z_i$  – коефіцієнт забрудненості коренеплодів цукрових буряків з  $i$ -ої сировинної зони;

$c_{ij}$  – вміст цукру в коренеплодах цукрового буряку, що надходить з  $i$ -ої сировинної зони, за кожний  $j$ -й день;

$\bar{x}_i$  – втрати маси цукрових буряків з  $i$ -ої сировинної зони під час зберігання в заводських кагатах;

$\bar{C}_i$  – втрата цукру в коренеплодах з  $i$ -ої сировинної зони під час зберігання в заводських кагатах;

$Tr_i$  – витрати на транспортування цукрових буряків з  $i$ -ої сировинної зони на приймальний пункт цукрового заводу [9].

Цільова функція (2.30) направлена на забезпечення максимального виходу цукру з сировини, що надійшла на цукровий завод. Одним з основних часткових критеріїв оптимізації є максимізація вилучення цукрози з цукрового буряка, що надійшов з  $i$ -ої сировинної зони ( $i \in 1 \dots k$ ) за кожний  $j$ -й день надходження за весь період роботи цукрового заводу ( $j \in [1 \dots t_i]$ , від першого до останнього ( $t_i$ ) дня роботи надходження з  $i$ -ої сировинної зони) (2.31) [9]:

$$C_i = \sum_{j=1}^{t_i} c_{ij} \rightarrow \max, \quad (2.31)$$

де  $c_{ij}$  – цукристість цукрового буряку, що надійшов з  $i$ -ої сировинної зони за кожний  $j$ -й день надходження за весь період роботи цукрового заводу (від першого до останнього ( $t_i$ ) дня роботи надходження з  $i$ -ої сировинної зони).

Прогнозований виробіток цукрози з 1 га посівів цукрового буряку за результатами обстежень (2.32) [9]:

$$C_{i,np} = \frac{\left( x_i - \frac{x_i \cdot P_m}{100} \right)}{100 * S_{i,цб}} V_C, \quad (2.32)$$

де  $P_m$  – прогнозовані внутрішньозаводські втрати маси цукрових буряків відповідного гібрида при зберіганні та транспортуванні, прийнятих на від  $i$ -го господарства, у відсотках до маси прийнятих буряків;

$V_C$  – прогнозований вихід цукрози у відсотках до маси перероблених буряків;

$S_{i,цб}$  – площа сировинної зони посівів відповідного гібрида цукрового буряка, га.

Прогнозований вихід цукру у відсотках до маси прийнятих коренеплодів визначаємо за формулою (2.33).

$$V_C = C_{np} - SB - C_M, \quad (2.33)$$

де  $C_{np}$  – цукристість гібрида буряка за результатами визначення цукристості коренеплодів на пробних ділянках зони бурякосіяння цукрового заводу у відсотках до маси буряків;

$SB$  – нормативна сума втрат цукрози від приймання до одержання готової продукції % до маси заготовлених (прийнятих) буряків;

$C_M$  – нормативний вміст цукрози в мелясі буряків у відсотках до маси буряків, усі показники враховуються для  $i$ -ої сировинної зони.

У виробничих умовах встановлено, що при однакових обсягах партій одного виду цукрового буряку та з однаковою цукристістю коренеплодів обсяги виготовленої продукції неоднакові. В технологічному процесі виготовлення необхідно відмітити різницю вмісту цукрози в мелясі під час переробки свіжозібраних коренеплодів і після зберігання їх у кагатах, що також впливає на обсяги кінцевого продукту [9].

З  $i$ -ої сировинної зони сировина має забрудненість, а коефіцієнт забрудненості для сировини з такої зони буде визначатися за формулою (2.34):

$$Z_i = \frac{m_1 - m_2}{m_1} \cdot 100, \quad (2.34)$$

де  $m_1$  – вага сировини з забрудненням, кг;

$m_2$  – вага сировини без домішок, кг.

Для кожної сировинної зони необхідно оптимізувати транспортні витрати на доставку сировини [10] за формулою (2.35):

$$Tr_i = \sum_{j=1}^{t_i} ((x_i - Z_m) \cdot v_i \cdot r) \rightarrow \min, \quad (2.35)$$

де  $Z_m$  – вага забрудненості та засміченості цукрового буряку, яка визначена на ньому у приймальних пунктах;

$v_i$  – відстань від  $i$ -ої сировинної зони до заводу;

$r$  (ц/т км) – нормативний тариф на автоперевезення, виражений у кількості цукру в центнерах на один тонно-кілометр [9].

Для максимально ефективного використання сировини для виробництва цукру необхідно враховувати біологічне дозрівання кожного виду гібрида, що дає необхідність враховувати терміни завезення сировини на цукровий завод розраховується як загальна вага завезеного цукрового буряку усіх гібридів (2.36).

$$R_1 + R_2 + R_3 \rightarrow \max, \quad (2.36)$$

де  $R_1 = \sum_{\tau=1}^9 C_y X_y$  – завезення на цукровий завод ранньостиглих гібридів;

$R_2 = \sum_{\tau=2}^9 C_y X_y$  – завезення на цукровий завод середньостиглих гібридів;

$R_3 = \sum_{\tau=3}^9 C_y X_y$  – завезення на цукровий завод гібридів нормального типу;

$C_y$  – цукристість відповідного гібрида цукрового буряку;

$R_h$  – обсяг цукрових буряків даного типу ( $h = 1, 2, 3$ );

$\tau$  – декади дозрівання гібридів цукрового буряку (в межах від 1 декади вересня до 9 – останньої декади листопада);

$X_y$  – сумарний обсяг постачання буряку відповідного гібрида;

$y$  – відповідає виду гібрида цукрового буряку  $y \in (1 \dots d)$ ;

$d$  – загальна кількість гібридів [9, 13].

Виробничий сезон  $T$  у цукровій промисловості (кількість діб цукроваріння) один із найважливіших показників, так як він суттєво впливає на кінцевий результат виробничої діяльності підприємства і залежить від величини добової потужності цукрового заводу (2.37).

$$T = \frac{\sum_{y=1}^d X_y}{P \cdot k_p}, \quad (2.37)$$

де  $X_y$  – сумарний обсяг постачання цукрового буряку відповідного гібриду;

$k_p$  – встановлений коефіцієнт використання потужності;

$P$  – добова потужність цукрового заводу.

Для забезпечення оптимальної роботи цукрового заводу на початку сезону необхідно сформувати запас сировини на період від  $N$  до  $M$  діб (2.38).

$$N \cdot P \leq \sum_{j=1}^{t_i} \sum_{i=1}^k \sum_{y=1}^d (x_{ijy}) \leq M \cdot P, \quad (2.38)$$

де  $N$  – 9-денний запас;

$M$  – 15-денний запас цукрового буряку;

$y$  – вид гібрида цукрового буряку  $y \in (1 \dots d)$ ;

$d$  – загальна кількість гібридів.

У залежності від характеристик заводу значення параметрів періоду запасів цукрового буряку може бути змінено.

Прогнозований сумарний обсяг постачання цукрового буряку відповідного гібрида  $X_y$  за весь період на цукровий завод визначаємо за формулою (2.39):

$$X_y = \sum_{j=1}^{t_i} \sum_{i=1}^k x_{ijy}, \quad (2.39)$$

де  $i$  – сировинна зона,  $i \in 1 \dots k$ ,  $k$  – кількість сировинних зон, з яких надходить цукровий буряк;

$j$  – номер дня поставки за весь період роботи цукрового заводу ( $j = 1 \dots t_i$ , від першого до останнього ( $t_i$ ) дня роботи надходження з  $i$ -ї сировинної зони).

Для забезпечення оптимального завантаження цукрового заводу необхідно дотримуватися тієї умови, що прогнозований обсяг поставки цукрового буряку повинен бути не менше фактичної врожайності (2.40).

$$X_{iy} \leq S_i \cdot U_y, \quad (2.40)$$

де  $S_i$  – площа  $i$ -ї сировинної зони, га;

$U_y$  – урожайність коренеплодів за результатами визначення густоти насаджень та маси коренеплодів на пробних ділянках  $i$ -ої сировинної зони.

Наведена математична модель була детальніше описана у роботі [9] і належить до класу багатокритеріальних NP-складних комбінаторних задач. Для її розв'язання необхідно використовувати комбінаторні методи.

### 2.3 Створення математичної моделі складання розкладу виконання замовлень та забезпечення балансу сировини й готової продукції для молокозаводу

Більшість підприємств харчової галузі залежать від якості сировини, яку необхідно вчасно поставити на підприємство для початку виготовлення у певному обсязі. Управлінням постачання сировиною займаються цілі відділи, які здійснюють постійний моніторинг постачання сировини. Для забезпечення потреб замовників більшість виробників закладають додаткові витрати на вчасне постачання сировини у вартість готової продукції. Адже такий підхід обумовлений тим, що у випадку неможливості постачання необхідного виду сировини від певного постачальника здійснюється постачання більш дорогої сировини з віддалених районів, що збільшує вартість сировини. Тобто, у витрати на постачання сировини закладена вартість постачання з інших регіонів. Все це впливає на вартість кінцевого продукту [48].

У ситуаціях, коли сировина надходить невчасно, виникає потреба коригування плану виготовлення продукції, що впливає вже на відносини між замовниками.

Особливо гостро проблема постачання стоїть для підприємств, діяльність яких пов'язана з виготовленням молочних виробів, тому що молоко, яке є основною сировиною, може втратити свої якості за години. Тому виникає необхідність у введенні додаткових обмежень у математичну модель, яка представлена у розділі 2.1.

На момент часу  $t$  необхідно забезпечити достатню кількість сировини, необхідної для виготовлення  $x_{it}$  обсягу готової продукції (2.41):

$$\sum_{p=1}^P v s_{jp} \geq V_{jt}, \quad (2.41)$$

де  $v s_{jp}$  – обсяг  $j$ -го виду сировини та матеріалів від  $p$ -го постачальника;

$V_{jt}$  – мінімальний обсяг  $j$ -го виду сировини, необхідний для виготовлення усієї продукції на момент часу  $t$ , який розраховується за формулою (2.42):

$$V_{jt} = \sum_{i=1}^J (x_{it} * v_{sn_{ij}}), \quad (2.42)$$

де  $v_{sn_{ij}}$  – необхідний обсяг  $j$ -го виду сировини для виготовлення одиниці  $i$ -го виду продукції;

$x_{it}$  – обсяг готової продукції, яку необхідно виготовити на момент часу  $t$ .

Необхідно врахувати, що окремі постачальники можуть на певний період часу не мати можливості забезпечити придбання необхідної кількості певного виду матеріалів, тоді доведеться докуповувати цей вид ресурсу у іншого постачальника за його цінами. Така умова описується формулою (2.43):

$$v_{s_{jpt}} \leq V_{p_{jpt}}, \quad (2.43)$$

де  $v_{s_{jpt}}$  – обсяг  $j$ -го виду сировини та матеріалів від  $p$ -го постачальника на момент часу  $t$ ;

$V_{p_{jpt}}$  – максимальний доступний обсяг  $j$ -го виду сировини та матеріалів у  $p$ -го постачальника на момент часу  $t$ .

Обов'язково враховується територіальне розміщення постачальника та транспортні витрати на доставку ресурсів. Якщо такі витрати перевищують певну визначену величину, то підприємству доцільніше придбати сировину та матеріали у інших постачальників за вищими цінами, але з меншими транспортними витратами. Така умова описується обмеженням (2.44) [48]:

$$\sum_{j=1}^J cts_{jpt} * v_{s_{jpt}} \leq CT_{jt}, \quad (2.44)$$

де  $vs_{jpt}$  – обсяг  $j$ -го виду сировини та матеріалів від  $p$ -го постачальника на момент часу  $t$ ;

$cts_{jpt}$  – вартість транспортування одиниці  $j$ -го виду сировини та матеріалів від  $p$ -го постачальника на момент часу  $t$ ;

$CT_{jt}$  – максимальний доступний обсяг  $j$ -го виду сировини та матеріалів у  $p$ -го постачальника на момент часу  $t$ .

Обсяги поставок не можуть мати від'ємні значення та мають таке обмеження (2.45):

$$0 \leq vs_{jpt}, j = 1..J, p = 1..P. \quad (2.45)$$

Підприємство випускає  $I$  видів продукції, використовуючи  $J$  видів ресурсів. При цьому відомі запаси кожного  $j$ -го виду ресурсу ( $B_j$ ), витрати кожного виду ресурсу на випуск одиниці кожного  $i$ -го виду продукції ( $vsn_{ij}$ ).

Не зважаючи на достатню кількість робіт, присвячених управлінню запасів, сучасні умови вимагають оперативного рішення, що ускладнює розв'язання задачі складання оперативного плану виконання замовлень.

## 2.4 Висновки до розділу 2

Розроблено математичну модель планування виконання замовлень із урахуванням основних особливостей діяльності підприємств харчової галузі. Математична модель враховує усі умови діяльності харчового підприємства та процесу планування виконання замовлень: терміни зберігання сировини та готової продукції; можливість виникнення та необхідність переробки некондиційної

продукції; особливості виконання кожного окремого замовлення; особливості використання технологічного обладнання тощо. Для формування узагальненої цільової функції використовується адитивна згортка критеріїв. Розроблена математична модель дозволяє сформулювати план виконання замовлень із урахуванням усіх операцій технологічного процесу при виготовленні продукції. Створена математична модель дозволяє коригувати та оцінювати ефективність виконання замовлень у залежності від об'єктивних та суб'єктивних переваг, наданих ОПР, а також забезпечує як врахування, так і виключення певних часткових критеріїв у залежності від певної ситуації.

На основі формалізованих процесів планування виконання замовлень для різних підприємств харчової галузі необхідно виділити основні ознаки даного процесу: кожне підприємство є унікальним, адже для кожного підприємства будуть обиратися свої часткові критерії з обмеженнями. Навіть подібні підприємства можуть мати різні часткові критерії при виконанні замовлень. На прийняття рішення керівників впливає: економічно-соціальний стан у країні; в якому стані знаходяться активи підприємства та чи є певний грошовий резерв; в якому стані технологічне обладнання, а також вартість його утримання та ремонту; яка ситуація на ринках сировини та збуту; чи укладені довготривалі договори і які умови на виготовлення продукції. Але головною особливістю харчових підприємств є динаміка різновиду сировини для виготовлення продукції в залежності від попиту та об'єму замовлень, що спонукає весь час коригувати виробничі плани, змінні завдання, рецептуру виготовлення без порушень ДСТУ, складність прогнозування виникнення позаштатних ситуацій, визначення коректного технологічного процесу виготовлення продукції та постійно здійснювати моніторинг процесу виробництва з метою зниження некондиційної продукції.

Крім того, більшість підприємств харчової промисловості мають унікальні технології щодо виготовлення кожного виду готової продукції, що накладає вимоги до дотримання усіх норм якості, адже готова продукція є стратегічним товаром та необхідна для споживачів внутрішнього та зовнішнього ринків збуту.

Розв'язання задачі планування в сучасних умовах господарювання потребує від керівників швидкого реагування на ситуацію на ринках збуту та сировини. Помилка та відкладення навіть на 15 хвилин може призвести до отримання некондиційного продукту, який буде необхідно перероблювати або утилізувати. У будь-якому випадку це все додаткові витрати.

Більшість підприємств вирішують задачу оперативного планування за рахунок постійного диспетчерського контролю та використання цілої низки керуючих впливів на різних ланках управління. Все це збільшує людських фактор при прийнятті управлінських рішень.

Чим більше людей задіяно у процесі, тим більший час на прийняття рішення, а іноді втрачений час може перевищувати ризик прийняття рішення, адже отримана за цей час некондиційна продукція потребує утилізації та не підлягає переробці; втрата частково або повністю партії замовлення; простоювання обладнання так само призводить до витрат.

Отже, для підприємств харчової галузі вкрай необхідно мати методи та технології для швидкого та ефективного прийняття рішення при формуванні плану виконання замовлення, які забезпечать формування оперативних планів виконання замовлень із мінімізацією витрат та направлені на максимізацію прибутку; оперативну реконфігурацію існуючих календарних планів виконання замовлень, що забезпечить оптимальне використання технологічного обладнання та мінімізацію витрат при виникненні позаштатних ситуацій під час появи термінових до виконання замовлень або позаштатних ситуацій; мінімізацію витрат на зберігання сировини та матеріалів; чітке розподілення усіх задач щодо виконання кожного замовлення між виробничими підрозділами, що дає можливість врахувати послідовність виконання та необхідні ресурси з прив'язкою в часі.

Основні результати розділу опубліковані автором у працях [4, 9, 11, 12, 18, 22, 30, 32, 48].

## РОЗДІЛ 3

### ДОСЛІДЖЕННЯ МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ ТА РОЗРОБКА МОДИФІКОВАНИХ АЛГОРИТМІВ

#### 3.1 Алгоритм «бджолої колонії»

Класичний алгоритм «бджолої колонії» (Artificial Bee Colony (ABC) Algorithm, Bees Algorithm (BA)) відноситься до алгоритмів ройового інтелекту. Він базується на імітації поведінки бджіл, а саме на поведінці взаємодії комах у рої та на здатності виконувати різні добре організовані складні дії окремою бджолою. Ефективність використання бджолоного алгоритму показано в роботах [34, 213, 234-236, 238-245].

Для збору нектару у «бджолої колонії» застосовуються бджоли-розвідувачі і бджоли-робітники. Бджоли-розвідувачі здійснюють пошук нектару на території навколо вулика, а після закінчення роботи передають іншим бджолам інформацію про кількість нектару, відстань та орієнтири розташування. Після отримання інформації робочі бджоли вилітають у місця, збагачені нектаром. Чим краще місце, тим більше бджіл вилітає до нього. Процес збору нектару має інформативну складову про кількість залишку нектару в певному місці на момент його відвідування [240, 242].

Бджоли-розвідувачі на початку пошуку здійснюють рух у випадково обраних напрямках і в радіусі льоту визначають ділянки з найбільшою кількістю квітів як нові джерела нектару. У природі інформацію бджола передає під час «бджолоного танцю» при поверненні до вулика. Кожна бджола перед вильотом обирає напрямок руху та місце збору, використовуючи інформацію від інших бджіл та свій досвід знаходження ділянки з найбільшою щільністю квітів та обсягом зібраного нектару. Остаточний вибір бджоли визначається в залежності від того, що в даний момент часу має більший вплив на її рішення, а саме персональний спогад або соціальний рефлекс. У результаті виконання описаних дій робочі бджоли рою концентрують свою роботу навколо місця з найбільшим об'ємом нектару [238, 240].

Замість поля із квітами розглядається область рішень. Замість нектару використовуються критерії задачі оптимізації та цільова функція.

На кожній ітерації обираються  $n^b$  області із кращими значеннями цільової функції, які називаються кращими, а з інших обирають  $n^s$  кращих, що називаються перспективними. Можна задати певну мінімальну відстань між двома сусідніми ділянками. У цьому випадку при виникненні суміщення ділянка з гіршим значенням цільової функції замінюється альтернативою. Дані ділянки запам'ятовуються і при наступній ітерації до них надсилається певна кількість бджіл [240].

Робота алгоритму складається із двох етапів: ініціалізація та локальний пошук.

При ініціалізації для  $n$  розвідувачів генеруються початкові положення, у найпростішому випадку використовується метод випадкового перебору [238].

Локальний пошук складається з таких кроків:

- після формування списків кращих і перспективних областей у ці місця відправляється певна кількість робочих бджіл, яка визначається в залежності від якості області, з точки зору цільової функції або переліку часткових критеріїв, а координата відправлення таких бджіл визначається за формулою (3.1):

$$X_{(i-1)c^b+kj} = N_{ij-1}^b + Rnd * rad, \quad i = 1 \dots n^b, \quad k = 1 \dots c^b, \quad (3.1)$$

де  $i$  – номер області;

$k$  – номер робочої бджоли;

$c^b$  – кількість робочих бджіл;

$n^b$  – кількість областей із кращими значеннями цільової функції;

$N_{ij-1}^b$  – центр координат області, куди відправляється бджола;

$rand$  – функція, що генерує випадкову величину з  $[-1, 1]$ ;

$Rnd$  – радіус, в який може потрапити бджола.

- при кожній ітерації розвідувачі відправляються у нові області за формулою (3.2):

$$X_{n^b c^b + (i-1)c^b + kj} = N_{ij-1}^g + Rnd * rad, \quad i = 1 \dots n^g, \quad k = 1 \dots c^g, \quad (3.2)$$

де  $i$  – номер області;

$k$  – номер бджоли-розвідника;

$c^g$  – кількість бджіл-розвідників;

$c^b$  – кількість робочих бджіл;

$n^g$  – кількість перспективних областей;

$n^b$  – кількість областей із кращими значеннями цільової функції;

$N_{ij-1}^g$  – центр координат області, куди відправляється бджола;

$rand$  – функція, що генерує випадкову величину з  $[-1, 1]$ ;

$Rnd$  – радіус, в який може потрапити бджола.

Для реалізації алгоритму та забезпечення пошуку якісного рішення за мінімальний проміжок часу необхідно підібрати такі параметри: кількість бджіл-розвідувачів; кількість кращих і перспективних ділянок; радіус локальної розвідки; кількість бджіл для кожного класу області; мінімально можлива відстань між сусідніми областями.

Кожна бджола в рої розглядається як окрема особа, яка діє індивідуально, відповідно до керуючого принципу: пошук відбувається у напрямі найкращої персональної та спільної позиції, постійно перевіряючи значення поточної позиції [244, 245]. Якщо прийняти  $w_i^j$  як оцінку танцю бджоли  $i$  на кроці часу  $t = j$ , тоді ймовірність того, що бджола-розвідник полетить згідно з танцюючою бджолою на корм, може бути визначена за формулою (3.3) [242, 244, 245]:

$$\rho_i = \frac{w_i^j}{\sum_{i=1}^{n_f} w_i^j}, \quad (3.3)$$

де  $n_f$  – кількість бджіл задіяних в алгоритмі;

$t$  – псевдочас або час на пошук їжі.

Кількість бджіл-спостерігачів  $n_f \in N$ , де  $N$  – загальна кількість бджіл. Як альтернативу можливо використовувати формулу (3.4) [242, 244, 245]:

$$\rho_e = 1 - \rho_i = e^{-\frac{w_i^2}{2\sigma^2}}, \quad (3.4)$$

де  $\sigma$  – коефіцієнт, який відповідає за розвідку та різноманітність місць пошуку.

Якщо місця з їжею не знайдено, тоді  $w_i \rightarrow 0$ , і  $\rho_e \rightarrow 1$ , тоді усі бджоли обирають шлях випадково.

Для задачі планування робіт бджоли-робітники виконують передачу інформації тривалістю  $\tau = \alpha * f_\rho$ , де  $f_\rho$  – оцінка ефективності обраного варіанта, а  $\alpha$  – фактор масштабування. Оцінка ефективності пов'язана з цільовою функцією, крім того, рейтинг кожного маршруту класифікується, і шлях із найбільшою кількістю бджіл стає кращим шляхом [238–240].

Ймовірність вибору маршруту між будь-якими двома вузлами визначається за формулою (3.5):

$$\rho_{ij} = \frac{w_{ij}^\alpha \cdot d_{ij}^\beta}{\sum_{i=1}^n \sum_{j=1}^n w_{ij}^\alpha \cdot d_{ij}^\beta}. \quad (3.5)$$

Модифікації даного алгоритму полягають у підборі значень параметрів та зменшенні залежності між ними, а також сприяють збільшенню швидкості пошуку якісного рішення.

Для пояснення роботи алгоритму використовують такі визначення позиції та придатності.

Позиція – розташування бджоли на площині з осями x-y або у  $N$ -мірному просторі, що визначається задачею.  $N$ -мірний простір є областю рішень для оптимізованого завдання, де кожен набір координат представляє рішення.

Придатність – за прикладом бджолиного рою функція придатності буде щільністю квітів: чим більше щільність, тим краще позиція. Функція придатності слугує засобом зв'язку між фізичною проблемою і алгоритмом оптимізації.

Персональна найкраща позиція (ПНП) – кожна бджола пам'ятає позицію, де вона сама виявила найбільшу кількість квітів. У кожній точці вздовж шляху руху бджола порівнює значення придатності поточної позиції зі значенням ПНП. Якщо поточна позиція має значення придатності вище, значення ПНП замінюється на значення поточної позиції [242, 244, 245].

Глобальна найкраща позиція (ГНП) – кожна бджола оновлює інформацію про знаходження найкращої ділянки з найбільшою концентрацією квітів, яка визначена всім роєм і є глобальною найкращою позицією. В кожній точці вздовж шляху руху бджола порівнює значення придатності поточної позиції зі значенням ГНП. Як тільки бджола виявить позицію з більш високою придатністю, ніж ГНП, відбудеться заміна його значення на поточну позицію бджоли. Метод рою бджіл доцільно розподілити на декілька паралельних процесів, у результаті чого значно збільшується швидкість його роботи [242, 244, 245].

Виділяють окремо «алгоритм медоносних бджіл» або його називають «віртуальних бджіл» Virtual Bees Algorithm (VBA). Його робота починається з формування «зграї віртуальних бджіл», кожна бджола випадковим чином відправляється в n-вимірний пошуковий простір. Основні кроки алгоритму «віртуальних бджіл» для оптимізації функцій має такі кроки [216, 236, 238]:

- 1) створення популяції «віртуальних бджіл», кожна з яких пов'язана з банком пам'яті;
- 2) кодування цілей функції оптимізації і перетворення її у віртуальну їжу;
- 3) визначення критеріїв напрямку і відстані до віртуальної їжі;
- 4) перехід та оновлення популяції на нові позиції відповідно до віртуальної їжі та поширення інформації про її розташування за допомогою віртуального танцю;
- 5) після певного часу еволюції більшість віртуальних бджіл обирають найкращу позицію, що визначається інтенсивністю відвідування такої ділянки;
- 6) декодування результатів для отримання рішення задачі.

Модифікована позиція кожної бджоли розраховується за формулами (3.6–3.7):

$$x_k^{i+1} = x_k^i * (1 - \beta) + x_{best} * \beta + \alpha * (Rand(i) - 0.5), \quad (3.6)$$

$$y_k^{i+1} = y_k^i * (1 - \beta) + y_{best} * \beta + \alpha * (Rand(i) - 0.5), \quad (3.7)$$

де  $\alpha$ ,  $\beta$  – дві додатні константи, які є амплітудою випадковості та швидкості сходження відповідно;

$x_{best}$ ,  $y_{best}$  – найкращі параметри  $i$ -ої ітерації;

$Rand(i)$  – випадкове число в межах  $[0, 1]$ .

Кожна бджола в рої представлена  $k$ -вимірним вектором, де  $x_k = (x_1, x_1 \dots x_n)$  та  $y_k = (y_1, y_2 \dots y_n)$  – координати.

Поточна позиція (точка пошуку у просторі рішень) обраховується за формулою (3.8):

$$S_k^{i+1} = S_k^i + S_{best} + S_{rand}, \quad (3.8)$$

де  $S_k^{i+1} = (x_k^{i+1}, y_k^{i+1})$  – поточна позиція у просторі бджоли;

$S_k^i = (x_k^i, y_k^i)$  – попередня позиція у просторі бджоли;

$S_{best} = (x_{best}, y_{best})$  – поточна позиція у просторі бджоли  $i$ -ої ітерації;

$S_{rand} = (x_{rand}, y_{rand})$  – випадкова позиція у просторі бджоли  $i$ -ої ітерації.

### 3.2 Алгоритм «кажанів» та його модифікації

Алгоритм кажанів (bat algorithm(BA)) базується на поведінці кажанів під час полювання. Кажани використовують ехолокацію для пошуку їжі та перешкод на їх

шляху руху [30, 31, 218, 222, 223, 246]. Завдяки ехолокації кажан визначає розташування жертви і здійснює напад або наступні рухи.

Алгоритм «кажанів» дотримується таких правил:

- усі кажани на основі використання ехолокації аналізують відстань до певного об'єкта, а також визначають, є він їжею чи перешкодою;
- кажани рухаються випадково зі швидкістю  $v_i$  у позиції  $x_i$  із фіксованою частотою  $f_{\min}$ , змінною довжиною хвилі  $\lambda$  і гучністю  $A_0$ , щоб знайти здобич;
- кажани можуть змінювати довжину хвилі (або частоту, оскільки частота=1/довжина хвилі), випускання імпульсу і швидкість імпульсу  $r_i \in [0,1]$ , що залежать від відстані до певного об'єкта;
- показник гучності змінюється від більш позитивного  $A_0$  до меншого постійного значення  $A_{\min}$ .

Враховуючи, що швидкість звуку в повітрі приблизно дорівнює 300 м/с, довжина хвилі для звуку з постійною частотою визначається формулою (3.9):

$$\lambda = \frac{u}{f}. \quad (3.9)$$

Кажани використовують затримку від випромінювання сигналу до виявлення відлуння, різницю в часі виявлення відлуння у двох вухах для того, щоб побудувати тривимірну модель оточуючого простору. Використовуючи побудовану модель, вони виявляють відстань до оточуючих предметів, відстань до цілі, тип здобичі, швидкість її руху. При цьому вони використовують ефект Доплера, який полягає у зміні частот і довжини хвилі випромінювання через рух джерела випромінювання хвилі. Якщо джерело хвиль рухається в середовищі і при цьому випромінює хвилі, то відстань між хвилями залежить від швидкості та напрямку руху джерела і приймача. Якщо джерело рухається в напрямку до приймача, тобто наздоганяє хвилі, то довжина хвилі зменшується, і навпаки, якщо рухається в напрямку, протилежному джерелу, то довжина збільшується за формулою (3.10) [222, 223, 246]:

$$\lambda = \frac{2\pi(c - v)}{w_0}, \quad (3.10)$$

де  $w_0$  – кутова частота хвилі;

$c$  – швидкість поширення хвиль у середовищі;

$v$  – швидкість джерела звуку відносно середовища (зі знаком «+», якщо джерело наближується до приймача, і зі знаком «-», якщо віддаляється).

Для роботи алгоритму використовуються такі значення [30, 31, 222, 223, 246]:

- частоти хвилі знаходяться у заданому діапазоні  $[f_{\min}, f_{\max}]$ ;
- відповідають діапазону довжини хвиль  $[\lambda_{\min}, \lambda_{\max}]$ .

Рух одного описується формулами (3.11–3.13):

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \quad (3.11)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i, \quad (3.12)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (3.13)$$

де  $\beta \in [0,1]$  – випадкова величина;

$x_*$  – найкращий поточний розв’язок.

На кожній ітерації алгоритму відбувається оновлення амплітуди імпульсу і його емісія. Коли кажан наближається до цілі, гучність його імпульсів зменшується (3.14), а частота імпульсів збільшується (3.15).

$$A_i^{t+1} = \alpha A_i^t, \quad A_i^t \rightarrow 0, \quad (3.14)$$

$$r_i^{t+1} = r_i^0 \left[ 1 - \frac{1}{e^{-\gamma}} \right], \quad r_i^t \rightarrow r_i^0, \quad t \rightarrow \infty, \quad (3.15)$$

де  $\alpha$  – константа,  $0 < \alpha < 1$ ;

$\gamma$  – константа,  $\gamma > 0$ .

У загальному випадку псевдокод алгоритму кажана можливо представити так [222, 223, 246]:

1. Визначаємо цільову функцію  $f(x)$ .
2. Ініціалізуємо популяцію кажанів  $x_i (i = 1 \dots n)$  та  $v_i$ .
3. Визначаємо частоту пульсу  $f_i$  для  $x_i$ .
4. Ініціалізуємо частоту імпульсу  $r_i$  та гучність  $A_i$ .
5. While ( $t < \text{Max}$  кількість ітерацій).
  - 5.1. Створюємо нові "рішення", регулюючи частоту.
  - 5.2. Оновлюємо швидкість та локацію рішень.
  - 5.3. If ( $\text{rand} > r_i$ ).
    - 5.3.1. Вибір рішення серед найкращих рішень.
    - 5.3.2. Створюємо локальне рішення серед обраного найкращого рішення.
  - 5.4. Кінець If.
  - 5.5. If ( $\text{rand} < A_i \ \&\& \ f(x_i) < f(x^*)$ ).
    - 5.5.1. Приймаємо нові рішення.
    - 5.5.2. Збільшуємо  $r_i$  та зменшуємо  $A_i$ .
  - 5.6. Закінчення If.
  - 5.7. Ранжування кажанів та знаходження найкращого  $x$ .
6. Закінчення while.
7. Виведення результатів.

При використанні даного алгоритму необхідно відмітити, що його складність залежить від розмірності задачі; кількості функцій, що є цільовими; кількості агентів, що використовуються в алгоритмі [222, 223, 246].

### 3.2.1 Алгоритм «хаотичного кажана»

Алгоритм хаотичного кажана (chaotic bat algorithm CBA) використовує у своїй основі теорію хаосу з використанням частково хаотичних карт [30, 31, 231, 250–252].

Хаотичні системи є дуже поширеними у природних та соціальних системах, які мають складну, випадкову та точну характеристику.

Хаотичний рух – це дуже нестабільний рух у детермінованій системі, обмежений простором кінцевих фаз. Хаос – це форма аперіодичного руху, яка є унікальною й загальною у нелінійних системах.

Хаотичний рух відбувається за рівнянням (3.16). Воно відоме як найбільш типове для хаотичних систем [352].

$$S_{k+1} = \mu S_{k+1} (1 - S_k), \quad (3.16)$$

де  $\mu$  – коефіцієнт переходу, що є константою;

$S_k$  – значення коефіцієнта формується за правилом хаотичної карти,  $S \in (0,1)$ . Послідовність  $S_0, S_1, \dots, S_n$  є детермінованою, яка сформована за певним правилом хаотичної карти.

Існують 10 типових видів хаотичної картографії. Оптимальний вигляд хаотичної карти обирається шляхом проведення імітаційних експериментів на тестових функціях.

У роботі [249] було проаналізовано застосування різних хаотичних карт і запропоновано обрання як найкращої частково хаотичної карти.

При цьому запропоновано початкове значення 0,7, що забезпечує швидкість знаходження оптимального рішення.

### 3.2.2 Алгоритм «кажана на основі стратегії польоту Леві»

В основі даного методу знаходиться ефект поведінки польоту Леві (Levy Flight Search Strategy Bat Algorithm (LBA)), який базується на траєкторії руху живих істот під час пошуку їжі у невизначеному та непередбачуваному середовищі. Польоти Леві складаються з коротких швидких рухів та періодичних довгих повільних пересувань.

Польоти Леві відображають траєкторію польоту альбатроса, бджіл та фруктових дрозодів. Поведінка польоту Леві є найкращою стратегією пошуку для  $N$  незалежних дослідників, коли відбувається пошук об'єкта, що має випадкове розташування у просторі, а також неможливо математично змодельовати простір [27, 30, 31, 231, 249-251].

Рух кажана в даному алгоритмі відповідає негауссовському стохастичному процесу, здійснює велику кількість стрибків у просторі і багаторазово змінює напрямок, що дозволяє розширити простір пошуку.

У поєднанні з функцією ехолокації кажана це допомагає значно та ефективніше покращити роботу «алгоритму кажана» [231, 249-251]. Тому рівняння руху вдосконалений алгоритм замінює рівняння (3.13) рівнянням (3.17):

$$x_i^t = x_i^{t-1} + levyx(x_i^{t-1} - x_*) + v_i^t. \quad (3.17)$$

Політ Леві використовується окремим кажаном при пошуку оптимального локального положення, що забезпечує збільшення ітерацій перевірок та оптимізацію у глобальному процесі пошуку. Все це направлено на уникнення потрапляння в локальний оптимум та прискорення знаходження оптимального рішення.

### 3.2.3 Алгоритм «кажана на основі коефіцієнта скорочення»

Алгоритм «кажана на основі коефіцієнта скорочення» (shrink factor bat algorithm (SBA)) є модифікацією класичного алгоритму. Для алгоритму кажана існує проблема попадання у локальний оптимум, що може запобігати знаходженню глобального оптимуму. Для уникнення такої проблеми пропонується використовувати коефіцієнт скорочення, який забезпечить різноманітність популяції та підвищить ефективність пошуку оптимуму. Для застосування такого підходу формула (3.12) перетворюється на формулу (3.18) [249–251]:

$$v_i^t = v_i^{t-1} + (x_i^t - x_*) f_i * \frac{2}{|2 - d - \sqrt{d - 4d}|}, \quad (3.18)$$

де  $d$  – константа, що підбирається експериментально.

За результатами порівняння розглянутих алгоритмів пропонується використання алгоритму кажана на основі коефіцієнта скорочення.

Алгоритм «кажана на основі коефіцієнта скорочення» підтверджує свою ефективність як метод випадкового пошуку для задач з мінімальним набором додаткових часткових критеріїв.

До недоліків даного методу слід віднести велике число вільних параметрів, від значення яких часто залежить результат, з іншого боку, відсутні підстави для вибору цих значень [31].

### 3.3 «Генетичний алгоритм»

За своєю природою «генетичні алгоритми» відносяться до класу процедур випадкового пошуку, який не зводиться до безладного блукання в пошуковому

просторі допустимих рішень завдяки можливості ефективного використання досвіду, набутого кожною популяцією в визначенні нової області пошуку рішень, в якій передбачається поліпшення значення цільової функції [78, 200, 205, 206, 208-211, 253, 254].

Можна виділити два основні підходи до вирішення задач багатокритеріальної оптимізації на основі еволюційно-генетичних алгоритмів [206, 208].

Перший підхід заснований на перетворенні розв'язуваної багатокритеріальної задачі в задачу скалярної оптимізації. При цьому можуть бути використані як згортка локальних критеріїв у загальному інтегральному показнику, так і зведення розв'язуваних багатокритеріальних задач до послідовності спеціальним чином сформульованих завдань однокритеріальної оптимізації. Для розв'язання багатокритеріальних задач використовуються класичні нееволюційні процедури. Даний підхід спрямований, як правило, на отримання єдиного рішення. Однак у багатьох практичних завданнях багатокритеріальної оптимізації більш кращим є отримання набору рішень [206].

Другий підхід до алгоритмізації задач багатокритеріальної оптимізації заснований на використанні концепції Парето-домінування [206]. При цьому визначається не одичне рішення, а множина всіх ефективних точок (множина Парето) і їх образ в області пошуку – фронт Парето. Отримана множина надалі досліджується для виявлення компромісів. Множина рішень Парето може бути різних розмірів, але зазвичай розмір збільшується зі збільшенням числа критеріїв [206].

Механізм кожного «генетичного алгоритму» завжди складається із трьох основних операторів [200, 206]:

- Репродукція – процес, в якому хромосоми обираються із кращим значенням цільової функції. Оператор репродукції є штучною версією натуральної селекції, тобто виживанням найсильніших.

- Кроссовер – схрещування батьківських пар, генерація нащадків.

- Мутація – дія випадкових чинників.

Загальна послідовність «генетичного алгоритму» має такий вигляд [200, 210, 353]:

1. Обрання необхідної кількості особин у популяції ( $K$ ). Задаємо відлік часу  $t = 0$ .
2. Формування початкової популяції  $P_0$ , що складається з  $K$  особин випадковим чином.
3. Визначення для кожної особини функції пристосованості  $F(x)=fitness(x)$ ,  $i=1\dots k$  і популяції в цілому  $F_t(x)=fitness(P_0)$ .
4. Перевірка умови припинення роботи, яка визначена виконанням усіх обмежень та знаходженням такого рішення, що повторюється з популяції в популяцію, якщо виконується, то КІНЕЦЬ.
5. Обирання з певною ймовірністю особин із високою пристосованістю з попереднього покоління для схрещування. Запис у банк генів  $GP$ .
6. Випадкова селекція половини батьківських пар типу  $A$  і  $B$  із банку генів  $GP$  та з певною ймовірністю  $P_x$ , яка залежить від  $f(x)$  і розташування їх у тимчасовій популяції  $TP$ .
7. Схрещення обраних батьківських пар типу  $A$  та  $B$  із банку генів  $GP$  з ймовірністю схрещування  $P_c$  для отримати нащадків типу  $C$  і  $D$ .
8. Розміщення половини нащадків типу  $C$  і  $D$  у тимчасову популяцію  $TP$ .
9. З певною ймовірністю  $P_m$  виконати мутацію особин із тимчасової популяції  $TP$ , при чому  $P_m \ll P_c$ .
10. Розміщення отриманої хромосоми в новій популяції  $P = TP$ .
11. Виконання операції, починаючи з пункту 3,  $k$  раз.
12. Збільшення номера поточного часу  $t = t + 1$ .
13. Якщо виконалась умова зупинки, то завершити роботу, а інакше здійснити перехід на крок 3.
14. КІНЕЦЬ.

Блок-схема класичного «генетичного алгоритму» наведена на рис. 3.1.

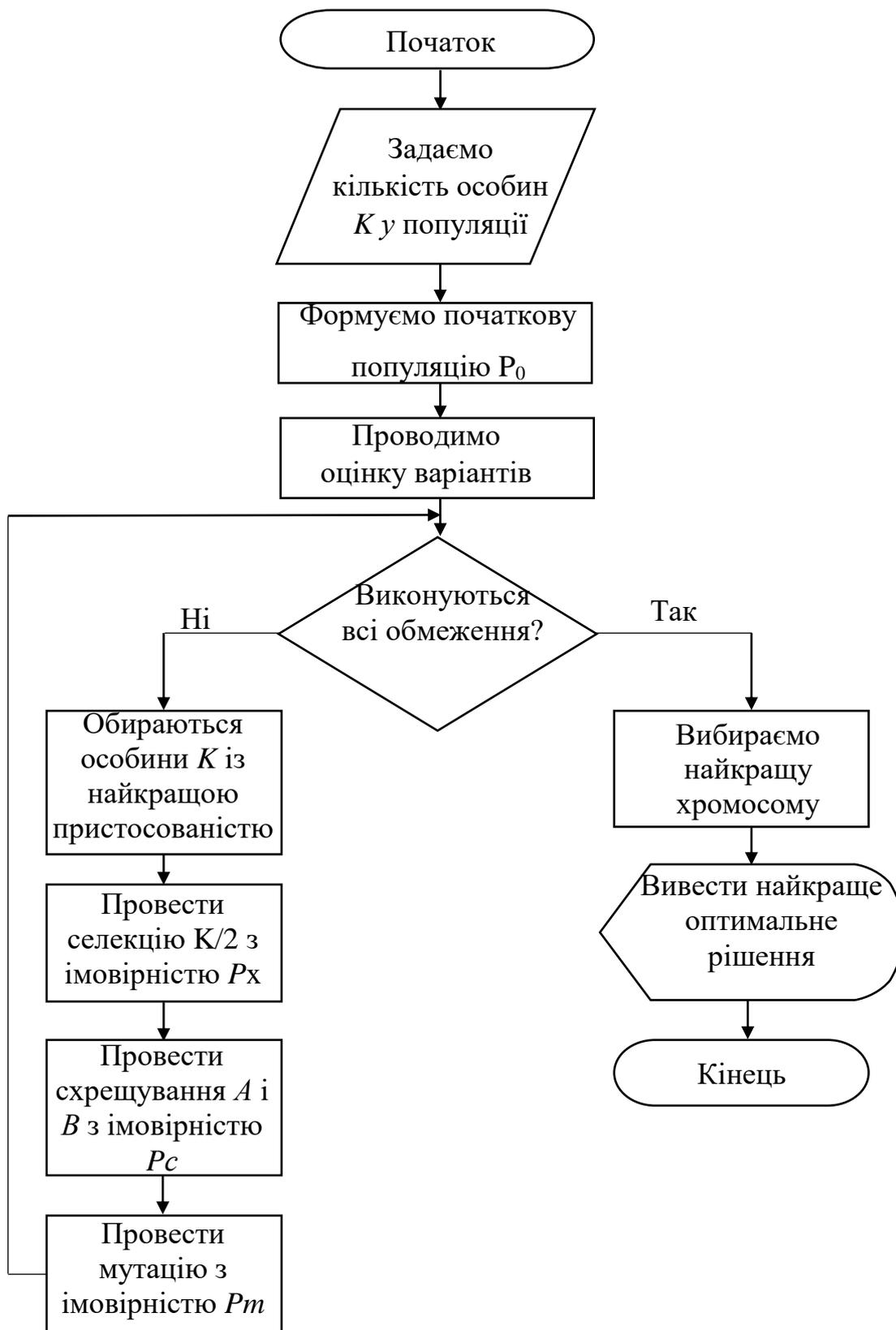


Рисунок 3.1 – Блок-схема класичного «генетичного алгоритму»

До найбільш поширених модифікацій «генетичних алгоритмів», що дозволяють розв’язувати задачі багатокритеріальної оптимізації, є такі [206, 208]:

- SPEA – Strength Pareto Evolutionary Algorithm;
- VEGA – Vector Evaluated Genetic Algorithm;
- FFGA – Fonseca and Fleming's Multiobjective Genetic Algorithm;
- NPGA – Niche Pareto Genetic Algorithm.

У методі SPEA створюється зовнішня множина (архів), призначена для зберігання недомінуючих рішень.

Кількість індивідів, що включаються в архів, регулюється процедурою кластеризації. Визначення значень придатності індивідів у методі SPEA здійснюється тільки щодо індивідів зовнішньої множини, які беруть участь у селекції на рівні з іншими членами популяції [208–211].

Загальну роботу методу SPEA доцільно представити так:

1. Сформувати первісну популяцію  $P_0$  і порожню зовнішню множину (архів)  $A = \emptyset$ ,  $t = 0$ .

2. Модернізувати зовнішню множину.

2.1. Визначити проміжну зовнішню множину  $A' = A_t$ .

2.2. Скопіювати рішення, що не домінують відносно  $A_t$  у множині  $A'$ , і вираховуються таким виразом:

$$A' : A = A \cup \{x / x \in A_t \cap x \in p(A_t)\}.$$

2.3. Видалити із множини  $A'$  особи, які слабо домінують відносно  $A'$ . При цьому, якщо існує пара осіб  $x, y \in A'$  і  $x \succ y$ , тоді  $A' = A' - \{y\}$ .

2.4. За допомогою процедури кластеризації зменшити кількість осіб, які зберігаються у зовнішній множині, і помістити результат зменшеної множини в  $P_{t+1}$ .

3. Вирахувати значення придатності індивідів  $P_t$  та  $A'$ .

4. Селекція. Покласти  $A' = \emptyset$  і виконати для  $s=1, N$  такі дії.

4.1. Випадково вибрати двох індивідів  $x, y \in A' \cup P_t$ .

4.2. Якщо  $F(x) < F(y)$  при рішенні задачі мінімалізму, то  $A' = A' \cup \{x\}$ , в іншому випадку  $A' = A' \cup \{y\}$ .

5. Рекомбінація.

6. Мутація.

7. Закінчення. Результат  $P_{t+1} = A'$  та  $t = t + 1$ .

Якщо  $t \geq T$ , де  $T$  – максимальна кількість поколінь, то з останніх популяцій відбираються недомінантні особи. Якщо  $t < T$ , тоді потрібно повернутись до кроку 2. Етапи 5 і 6 (рекомбінація і мутація) проводяться згідно з загальною схемою «генетичного алгоритму» [200, 205, 206]. Етап призначення придатності відрізняється від подібного в загальному еволюційному алгоритмі і реалізується за такою схемою.

1. Кожному індивіду  $x \in A_t$  присвоюється значення  $S(x) \in [0,1]$ , що має назву «сила» (або вага) особини. Значення «сили» пропорційно кількості членів популяції  $y \in P_t$ , для яких  $x > y$  (3.19).

$$S(x) = \frac{|\{y : y \in P_t \cap x > y\}|}{N + 1}. \quad (3.19)$$

Таким чином, придатність особи  $x$ , що належить архіву, дорівнює її силі:  
 $F(x) = S(x)$ .

2. Придатність особи  $y \in P_t$  визначається таким чином (3.20):

$$F(y) = 1 + \sum_{x \in A', x > y} S(x). \quad (3.20)$$

При цьому, чим менша пристосованість, тим більше шансів у особи перейти в наступне покоління.

У генетичному алгоритмі VEGA створюються часткові популяції з однаковою чисельністю особин. Кожному  $n$ -у критерію  $F_n$  відповідає підмножина часткової популяції хромосом  $P_n$  цілої популяції  $P$ .

У кожній частковій популяції  $P_n$  виконується незалежна селекція згідно з критерієм  $F_n$ . Однак асоціація і схрещування особин виконуються для всієї популяції [206, 208, 211].

У цьому алгоритмі селекція виконується турнірним методом. Причому найкраща особина в кожній частковій популяції обирається на основі своєї функції пристосованості.

Найкраща особина з кожної підпопуляції змішується з іншими особинами, і всі інші генетичні операції проводяться аналогічно алгоритму при оптимізації однієї функції. Недоліком методу VEGA є те, що в ньому не враховуються проміжні рішення, оптимальні в сенсі Парето, які є допустимими для всіх критеріїв, але не задовольняють будь-якого із скалярних критеріїв [206].

FFGA – це метод, який використовує ранжування особин, засноване на Парето-домінуванні. Рангом у даному методі називається число особин, що домінують над заданою особиною. В методі використовується процедура придатності, яку можливо описати таким чином [206].

1. Для кожної особини  $S_j$ ,  $j=1..N$  з популяції  $P_0$ , де  $N$  – розмір популяції, розраховується ранг  $r_j=1+i$ ,  $i \in P_0$ ,  $i>j$ .

2. Популяція сортується за значенням рангу  $r$ , потім кожній особині  $S_j$ ,  $j=1..N$  призначається так звана неповна придатність  $F_j'$  за допомогою інтерполяції. При цьому опорними точками для інтерполяції обираються точки гіршої  $F_j'=1$  для особини  $S_j$ , ранг якої максимальний, і кращої  $F_j'=N$  для особини  $S_j$ , ранг якої мінімальний.

3. Обчислюється підсумкова придатність за допомогою усереднення «сирої» придатності особин  $S_r$  з однаковими рангами (3.21).

$$F_{S_r} = \sum_{s=1}^n \frac{F'(S_r)}{n}, \quad (3.21)$$

де  $n$  – кількість особин з однаковим рангом.

У методі NPGA етап призначення придатності замінюється модифікованою схемою розподілу придатності з використанням поняття ніші, яка визначається для індивідів у просторі альтернатив або цільових функцій і забезпечує можливість підтримки різноманітності, дозволяючи отримати представницьку множину Парето [206, 211].

### 3.4 Алгоритм пошуку «косяку риб»

Алгоритм пошуку «косяку риб» (Fish School Search або Fish shoal) був запропонований у роботі [224, 232, 233]. Відомо більше 25 000 видів риб, що більше загальної кількості всіх видів хребетних, а з них приблизно 50–80% риб плавають у «зграях», які називають косяками. «Косяком риб» називають агрегацію риб, які плавають приблизно з однаковою швидкістю та у напрямку, підтримуючи приблизно постійну відстань між собою. Зазвичай косяк містить риб, розміри яких відрізняються не більше ніж на 30%. Доведено, що будь-якого роду об'єднання риб відіграють важливу роль у підвищенні ефективності пошуку ними їжі, захисту від хижаків, а також у зменшенні енергетичних витрат у процесі плавання [232, 233].

В алгоритмі «косяку риб» риби плавають в акваріумі (область пошуку) в пошуках їжі (рішення задачі оптимізації).

Вага кожної риби формалізує її індивідуальний успіх в пошуку рішення і відіграє роль пам'яті риби.

Саме наявність ваги у агентів популяції є головною особливістю парадигми «косяку риб». Ця особливість алгоритму «косяку риб» дозволяє відмовитися від необхідності відшукувати і фіксувати глобально кращі рішення, як це робиться в інших алгоритмах [224, 232, 233].

Весь косяк рухається до набору найуспішніших риб. Колективно-вольовий рух збирає всіх особин у косяк, щоб рух основних риб косяку був ефективним. В іншому випадку косяк поширюється по всій запропонованій території. Вага показує,

наскільки ефективний косяк риб: якщо вага збільшується, косяк успішний. Якщо вага зменшується, то косяк не успішний [224, 232, 233].

Рух кожного агента описується формулою (3.22):

$$x_i(t+1) = x_i(t) + \text{rand}(-1,1) * v_i^{\text{ind}}, \quad (3.22)$$

де  $i$  – номер агента;

$x_i(t+1)$  – нова позиція на момент часу  $(t+1)$ ;

$x_i(t)$  – попередня позиція на момент часу  $(t)$ ;

$\text{rand}(-1,1)$  – генератор напрямку руху  $[-1, 1]$ ;

$v_i^{\text{ind}}$  – розмір індивідуального кроку.

Вважаємо, що вирішується завдання глобальної умовної максимізації в області  $D$  і що всюди в цій області фітнес-функція приймає невід'ємні значення.

Оператори алгоритму «косяку риб» об'єднані у дві групи [224, 232, 233]:

- оператор годування формалізує успішність дослідження агентами тих чи інших областей акваріума;
- оператори плавання реалізують алгоритми міграції агентів косяку риб.

*Оператор годування (feeding).* Позначимо  $w_i$  – поточна вага агента  $s_i$ . Збільшення ваги агента на даній ітерації  $t$  пропорційно нормалізованій різниці значень фітнес-функції на ітераціях  $t$ ,  $(t-1)$ , тобто вага кожного агента визначається за формулою (3.23) [224, 232, 233]:

$$w_i' = w_i + \frac{\varphi_i - \varphi_i(t-1)}{\max(\varphi_i, \varphi_i(t-1))}. \quad (3.23)$$

Алгоритм обмежує максимально допустиму вагу агентів величиною  $w_{\max} > 0$ , так що у всіх випадках вага агента  $w_i$  задовольняє умові  $w_i, i \in [1:|S|]$ , де  $w_{\max}$  –

вільний параметр алгоритму. При ініціалізації популяції всім агентам як вага присвоюємо значення  $w_{\max} / 2$  [224, 232, 233].

*Оператор плавання (swimming).* В алгоритмі FSS розрізняють три види плавання: індивідуальне, інстинктивно-колективне і колективно-вольове. Ці види плавань виконуються на інтервалах  $(t; \tau], (\tau; \theta], (\theta; t']$  основного ітераційного інтервалу  $(t; t']$  відповідно. Тут  $t < \tau < \theta < t'$ ,  $t' = t + 1$ .

*Індивідуальне (individual) плавання.* Напрямок переміщення агента у цьому випадку випадково рівновірогідний. Якщо переміщення агента в обраному напрямку виводить агента за межі області допустимих значень  $D$ , а також порушує задані обмеження, тоді переміщення не виконується. Аналогічно, переміщення агента  $s_i$  не виконується, якщо має місце нерівність  $\varphi_i^r < \varphi_i^t$ , тобто, якщо в новій точці  $x_i^r$  значення фітнес-функції не вище її значення в попередній точці  $x_i^t$ .

Величину кроку переміщення  $v_i^{ind}$  вважаємо випадковою величиною, рівномірно розподіленою в інтервалі  $[0; v_{\max}^{ind}]$  (3.24).

$$v_i^{ind} = U_1(0,1)v_{\max}^{ind}. \quad (3.24)$$

Для поступового переходу від диверсифікації пошуку на початкових ітераціях до його інтенсифікації на завершальних ітераціях лінійно зменшуємо величину  $v_{\max}^{ind}$  (вільний параметр алгоритму) з ростом числа ітерацій [224, 232, 233].

Індивідуальне плавання агента може включати в себе не одну ітерацію, як у розглянутій схемі, а їх деяке фіксоване число. Плавання можна інтерпретувати як локальний пошук в околиці розміщення місця розташування [224, 232, 233].

Інстинктивно-колективне (collective-instinct) плавання реалізуємо після завершення усіма  $N$  агентами індивідуальних плавань за формулою (3.25):

$$x_i^\theta = x_i^\tau + \left( \frac{\sum_{j=1}^N v_j^{ind}(\tau)(\varphi_j^\tau - \varphi_j^t)}{\sum_{j=1}^N (\varphi_j^\tau - \varphi_j^t)} \right), \quad (3.25)$$

де  $v_{\max}^{ind}(\tau) = x_j^c - x_j^t$  – переміщення агента  $s_j$  у процесі його індивідуального плавання. Загальний для всіх агентів крок міграції, який представляє собою зважену суму індивідуальних переміщень агентів. У процесі інстинктивно-колективного плавання на кожного з агентів впливають всі інші агенти популяції, і цей вплив пропорційний індивідуальним успіхам агентів [28, 224, 232, 233].

Колективно-вольове (collective volition) плавання виконуємо слідом за інстинктивно-колективним плаванням. Даний вид плавання полягає у зміщенні всіх агентів у напрямку поточного центра ваги популяції, якщо сумарна вага косяка в результаті індивідуального і інстинктивно-колективного плавання збільшилася, і в протилежному напрямку, якщо ця вага зменшилася. Іншими словами, в разі успіху зазначених плавань популяція стягується до свого центра тяжіння, тобто підвищує інтенсивність пошуку. В іншому випадку популяція розширюється від того ж центра, підвищуючи свої диверсифікаційні властивості [28, 224, 232, 233].

Координати центра ваги  $x_c^\theta$  косяка після завершення усіма його агентами інстинктивно-колективних плавань визначаємо за формулою (3.26):

$$x_c^\theta = \frac{\sum_{i=1}^N w_i^\theta x_i^\theta}{w_\Sigma^\theta}, \quad (3.26)$$

де  $w_\Sigma^\theta = \sum_{i=1}^N w_i^\theta$  – поточна сумарна вага популяції.

Колективно-вольове плавання виконуємо за правилом (3.27):

$$x_i' = x_i^\theta \pm v^{vol}(x_i^\theta - x_c^\theta), \quad (3.27)$$

де знак плюс використовуємо в разі  $w_{\Sigma}^{\theta} > w_{\Sigma}^{\theta-1}$ , а знак мінус у протилежному випадку. Тут  $w_{\Sigma}^{\theta-1}$  – сумарна вага популяції після завершення її агентами інстинктивно-колективних плавань на попередній ітерації. Змінна  $v^{vol}$  у формулі (3.27) визначає розмір кроку переміщень агентів і є випадковою величиною (3.28):

$$v^{vol} = v_{\max}^{vol} U_1(0; 1). \quad (3.28)$$

Тут  $v_{\max}^{vol} > 0$  – задана максимально допустима довжина кроку (вільне володіння параметрами алгоритму). Автори алгоритму рекомендують лінійне зменшення величини  $v_{\max}^{vol}$  зі збільшенням числа ітерацій.

Ініціалізацію популяції здійснюємо шляхом випадкового рівномірного розподілу агентів в області пошуку.

Як умова завершення ітерацій канонічний алгоритм «косяку риб» використовує умову досягнення апріорі заданого числа ітерацій [28, 53, 224, 232, 233].

### 3.5 Алгоритм «зграї вовків»

#### 3.5.1 Класичний алгоритм «зграї вовків»

Алгоритм «зграї вовків» або «сірих вовків» (Grey Wolf Optimizer Algorithm (GWO)) – метаевристичний стохастичний алгоритм ройового інтелекту, розроблений у 2014 році, ідея якого побудована на основі моделі полювання зграї вовків [225, 255]. В англійській літературі він зустрічається під назвою «wolf pack».

Виділяють чотири типи вовків: «альфа», «бета», «дельта» і «омега». «Альфа» має найбільшу «вагу» у прийнятті рішень та управлінні зграєю. Далі йдуть «бета» і

потім «дельта», які підкоряються «альфа» і мають владу над іншими вовками. Вовк «омега» завжди підпорядковується іншим домінуючим вовкам [221, 225, 226, 255].

У математичній моделі ієрархії вовків «альфа» вважається найбільш сприятливим рішенням. Друге і третє кращі рішення, відповідно це «бета» і «дельта». Решта рішень вважаються «омега».

Вважають, що до самих пристосованих вовків («альфа», «бета» і «дельта»), тобто, які були найближче до жертви, будуть наближатися інші вовки. Після кожного наближення визначається, хто на даному етапі «альфа», «бета» і «дельта», а потім вовки знову перебудовуються.

Перестроювання відбувається до тих пір, поки вовки не зберуться у зграю, що буде оптимальним напрямком для атаки з мінімальною відстанню [225, 255].

У ході алгоритму виконуються три основні етапи, в яких вовки шукають жертву, оточують і атакують її.

У процесі пошуку виявляються «альфа», «бета» і «дельта» – вовки, які знаходяться ближче до жертви. Решта вовків, підкоряючись домінуючим, можуть почати оточувати жертву або продовжити довільне переміщення в пошуках кращого варіанта [221, 225, 226, 255].

Модель оточення вовками жертви описується такими формулами (3.29–3.32):

$$\vec{D} = |\vec{C} * \vec{X}_p(t) - \vec{X}(t)|, \quad (3.29)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} * \vec{D}, \quad (3.30)$$

$$\vec{A} = 2\vec{\alpha} * \vec{r}_1 - \vec{\alpha}, \quad (3.31)$$

$$\vec{C} = 2\vec{r}_2, \quad (3.32)$$

де  $t$  позначає поточну ітерацію;

$\vec{A}$  та  $\vec{C}$  – вектори-коефіцієнти;

$\vec{X}_p$  – вектор положення жертви;

$\vec{X}$  – вектор положення вовка;

$\vec{D}$  – вектор напрямку (від вовка до жертви);

$\vec{\alpha}$  – параметр лінійно зменшується від 2 до 0 в кожній ітерації;

$\vec{r}_1$  та  $\vec{r}_2$  – випадкові вектори з інтервалу  $[0,1]$ , які дозволяють моделювати

переміщення вовків.

Простір пошуку може бути  $N$ -вимірним. Це означає, що вовки будуть переміщуватися навколо жертви в гіперпаралелепіпеді [221, 225, 226, 255].

Самим полюванням, як правило, займається «альфа» (іноді за участю «бети» і «дельти»). Однак в абстрактному просторі пошуку ми не можемо навіть припустити, де знаходиться «оптимум» (жертва).

Для побудови математичної моделі полювання припустимо, що «альфа», «бета» і «дельта» мають найбільш чітке уявлення про місцезнаходження жертви. Отже, необхідно зберігати перші три кращі рішення і направляти «омега»-вовків ближче до цих рішень.

Вовки закінчують полювання атакою на жертву, коли жертва перестає тікати. Щоб математично змоделювати це явище, необхідно зменшувати значення  $\vec{\alpha}$ . Діапазон коливань коефіцієнта атаки  $\vec{A}$  також знизиться на  $\vec{\alpha}$ . Тобто,  $\vec{A}$  – це випадкове число в інтервалі  $[-2a, 2a]$ , де  $a$  - число, яке послідовно зменшується від 2 до 0.

Якщо  $\vec{A}$  знаходиться від -1 до 1, то наступна позиція даного вовка буде знаходитися десь між поточною позицією і потенційним положенням жертви [225, 255].

Повертаючись до процесу пошуку жертви, необхідно звернути увагу на періодичне віддалення «омег» від «альфи», «бети» і «дельти» заради пошуку більш сприятливого рішення. Щоб математично сформулювати цю розбіжність, можна змінити інтервал для вектора  $\vec{A}$ .

Використовуючи значення більше 1 або менше -1, ми примушуємо «омегу» віддалятися від потенційної здобичі, що дозволяє алгоритму проводити пошук оптимуму на більшій території. Іншим важливим параметром пошуку є коефіцієнт вистежування  $\vec{C}$ , що є випадковим на інтервалі  $[0, 2]$ . Даний параметр забезпечує випадкові ваги для здобичі, збільшуючи ( $C > 1$ ) або зменшуючи ( $C < 1$ ) вплив (запах) жертви при визначенні відстані для переміщення. Це дозволяє алгоритму «зграї вовків» демонструвати більш випадкову поведінку у процесі оптимізації, сприяючи пошуку і уникнення локальних оптимумів. Параметр  $\vec{C}$  не зменшується послідовно на відміну від параметра  $\vec{A}$ . Параметр  $\vec{C}$  завжди є випадковим як на початкових ітераціях, так і на кінцевих. Ця властивість дуже корисна в разі попадання у локальний оптимум, особливо на останніх ітераціях [225, 255].

У загальному вигляді реалізується такий алгоритм [335–337]:

1. Установка розміру популяції  $n$  і максимальної кількості ітерацій  $iter$ .
2. Ініціалізувати позиції «зграї вовків»  $X_i$  ( $i = 1..n$ ).
3. Ініціалізувати  $\vec{\alpha}$ ,  $\vec{A}$  та  $\vec{C}$ .
4. Обчислити значення цільової функції для кожного вовка.
5. Визначити  $X_a$ ,  $X_b$ ,  $X_d$  – «альфу», «бету» і «дельту».
6. Поки  $t < iter$ :
  - 6.1. Перерахувати позиції для кожного вовка.
  - 6.2. Оновити  $\vec{\alpha}$ ,  $\vec{A}$  та  $\vec{C}$ .
  - 6.3. Обчислити значення цільової функції для кожного вовка.
  - 6.4. Оновити  $X_a$ ,  $X_b$ ,  $X_d$ .
  - 6.5.  $t = t + 1$ .
  - 6.6. Перевірити умову, чи відбувається знаходження одного і того самого плану за задану кількість ітерацій. Фактично оптимальний план не змінюється протягом заданої кількості ітерацій.
7. Визначити  $X_a$ .

Блок-схема класичного алгоритму «зграї вовків» наведена на рис. 3.2.

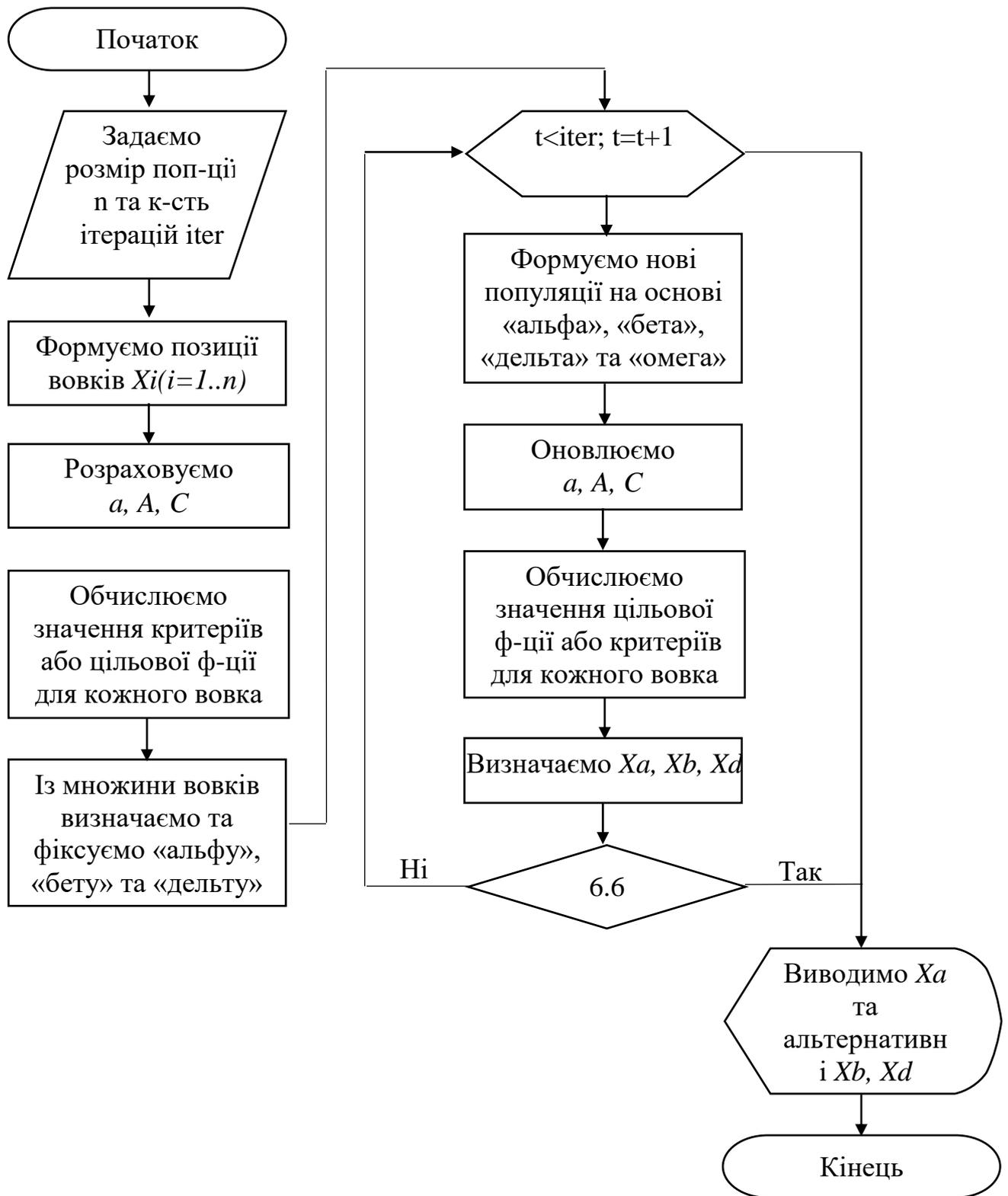


Рисунок 3.2 – Блок-схема класичного алгоритму «зграї вовків»

Припустимо, що місцевість, на якій вовки полюють, це область пошуку в сенсі оптимізації, а зграя – це вовки. Нехай спочатку згенеровано  $N$  «вовків» в

евклідовому просторі розмірності  $D$ , тобто кожен вовк представлений у вигляді вектора  $x_i = (x_{i1}, \dots, x_{iD})$ , що визначає його координати у просторі [225].

На даному етапі оціночна функція для «ватажка» та мети пошуку, «жертви» повинна мати найменшу різницю. Тобто для усіх інших вовків ватажок є метою, до якої вони рухаються. Тоді «вовки» зграї переміщуються в напрямку ватажка з заданим кроком  $step$ , причому  $d$ -я координата  $i$ -го «вовка» на  $(k+1)$ -й ітерації обчислюється за формулою (3.33):

$$x_{iD}^{(k+1)} = x_{iD}^{(k)} + step \frac{G_{best}^{(k+1)} - x_{iD}^{(k)}}{\|G_{best}^{(k+1)} - x_{iD}^{(k)}\|}. \quad (3.33)$$

Знаходимо спільні частини у «альфа», «бета» та «дельта».

З формули і опису алгоритму видно, що в методі пошуку «зграєю вовків» оновлюються тільки координати «вовків» без урахування швидкості їх переміщення у просторі.

Для роботи алгоритму «зграї вовків» досить підібрати два параметри: розмір популяції  $N$  та крок  $step$ , з яким переміщуються «вовки» за напрямком до ватажка і жертви [225, 255].

Таким чином, у алгоритмі «зграя вовків» є безліч потенційних рішень, координати яких оновлюються на кожній ітерації, поки не буде знайдено оптимальне рішення, або не буде виконано максимально задану кількість обчислень цільової функції. Тоді функція  $f(x)$ , що характеризує, наскільки сильно відчувається запах жертви вовками, є цільовою, а координати самої жертви – оптимальною точкою [225].

Відстань між двома вовками  $p$  і  $q$  описується метрикою виду  $L(p, q)$ . Алгоритми «зграї вовків» здійснюють пошук оптимальної точки-жертви. Вони діляться на групи, які переміщуються в різних напрямках і обмінюються інформацією між собою.

Сам алгоритм пошуку (спільного полювання) можна охарактеризувати за допомогою таких правил [28]:

- «вовк» із кращим значенням цільової функції на даній ітерації є ватажком. Якщо на наступній ітерації знайдеться інша «особина» із кращим значенням цільової функції, ніж у ватажка, то, відповідно, зграя знайде нового лідера;
- інші вовки досліджують місцевість на наявність жертви. Причому функція  $f(x_i)$  характеризує, як сильно відчувається запах жертви  $i$ -м «вовком». Тоді величина  $G_{Best}$  характеризує, як сильно відчувається запах жертви ватажком зграї;
- якщо  $f(x_i) > G_{Best}$ , тоді  $i$ -й «вовк» знаходиться ближче до жертви, ніж ватажок зграї, отже,  $i$ -й «вовк» стає ватажком на даному етапі  $f(x_i) = G_{Best}$ . Якщо ж  $f(x_i) < G_{Best}$ , тоді «вовк» переміщується у просторі з деяким наперед заданим кроком  $step$ ;
- ватажок зграї повідомить іншим «вовкам» в зграї про своє місцезнаходження, як про найближче на даний момент до жертви, для того, щоб вони перемістилися в його напрямку.

Після виконання алгоритму отримуємо три варіанти оптимальних рішень, що відповідають «альфа», «бета» і «дельта» вовкам останньої популяції.

### 3.5.2 Модифікований алгоритм «зграї вовків»

Відповідно до нашої задачі, зграя полює за жертвою, яка відповідає оптимальному оперативному плану виконання замовлень.

Кожен «вовк зграї» відповідає альтернативному оперативному плану  $L_{it}$ ,  $i$ -му замовленню на ітерації  $t$ .

Роботу модифікованого алгоритму «зграї вовків» представлено на рис. 3.3.

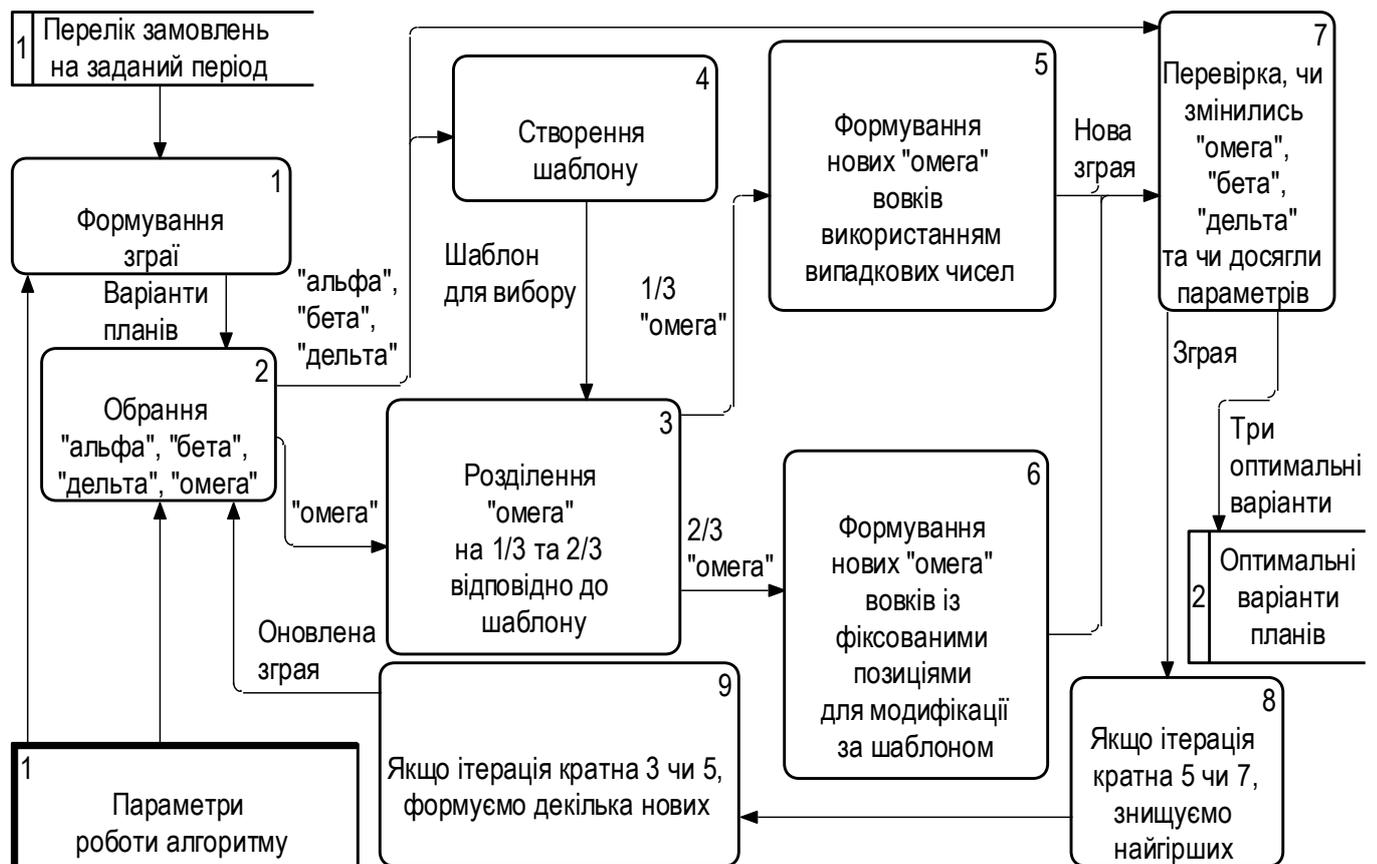


Рисунок 3.3 – Робота модифікованого алгоритму «зграї вовків»

На першій ітерації кожному вовку популяції присвоюється одне замовлення, без повторення, а потім для кожного вовка формується своя популяція з використанням генератора випадкових чисел, що виключає повторення номерів замовлень, якщо такі було вже згенеровані для цього агента.

Для формування певних членів популяції можна залучити особу, яка приймає рішення і сформує, згідно зі своїми знаннями та міркуваннями, наближені варіанти рішення.

Для формування першої популяції доцільно використати результати, отримані з використанням інших алгоритмів чи методів.

Після виконання кожної ітерації для кожного вовка розраховується значення його альтернативного оперативного плану, використовуючи часткові критерії або оціночну функцію  $F'_0$ . За значенням оцінки кожного вовка у зграї їх розділяють на чотири типи: «альфа» – вовк ватажок, оцінка якого має найоптимальніше рішення за

частковими критеріями або оціночною функцією  $F'_0$ ; «бета» та «дельта» – це вовки, які заганяють жертву, оцінка яких займає друге та третє місце серед найкращих; «омега» – усі інші.

Перші три типи вовків фіксуються на ітераціях до тих пір, поки не буде знайдено нові альтернативні варіанти рішення, що будуть кращі поточних, або буде вичерпано задану кількість ітерацій. Вовки «альфа», «бета», «дельта» впливають на формування «омега» вовків.

Багато зберігати значення «альфа», «бета», «дельта» від 3 до 5 ітерацій підряд, що даватиме можливість знаходження глобального оптимуму.

Кількість ітерацій, які необхідно утримувати (значення «альфа», «бета», «дельта» вовків), задаються користувачем. Кількість впливає на швидкість пошуку оптимального значення.

Фактично на кожній ітерації вовки типу «омега» намагаються наблизитися до вовків «альфа», «бета», «дельта», що відповідає оточенню та захвату жертви вовками зграї.

Прийнято, що усі вовки здійснюють переміщення з однаковою швидкістю, це забезпечує на кожній ітерації їх зміщення на певну відстань.

На кінці кожної ітерації  $t$  проводимо аналіз «альфа», «бета», «дельта» вовків та виділяємо шаблон  $W_s$ , а саме однакові позиції з однаковими значеннями в усіх трьох варіантах.  $W_s$  буде використаний для формування «омега вовків» на наступній ітерації.

Усі інші потрапляють у шаблон  $W_z$ . Шаблони  $W_s$  та  $W_z$  представляються як одновимірні вектори, перелік індексів елементів, які не будуть змінюватися, позначаємо  $J_{W_s}$ , а перелік індексів елементів, які будуть змінюватися, позначаємо  $J_{W_z}$ .

На рис. 3.4 наведено приклад рішень для «альфа», «бета», «дельта» вовків і виділений шаблон, в якому фіксується 2, 5 та 6 елемент зі значеннями 5, 7, 6 відповідно.

«альфа»	9	5	4	1	7	6	2	3	8
«бета»	1	5	2	9	7	6	3	4	8
«дельта»	2	5	8	3	7	6	9	1	4
шаблон $W_s$		5			7	6			
шаблон $W_z$		*			*	*			

Рисунок 3.4 – Приклад виділення шаблону

Усі позиції, які не зафіксовані у шаблоні  $W_s$ , будуть змінюватися, тобто вони будуть внесені у шаблон  $W_z$ , а саме ті елементи, що мають індекси 1, 3, 4, 7, 8, 9.

Необхідно відмітити, що зміни «омега» вовків відбуваються за правилом: усі вовки мають однакову швидкість, що дозволяє на кожній ітерації поміняти місцями тільки два замовлення у послідовності виконання з шаблоном  $W_z$ .

Усі «омега» вовки розділяються на 3 частини. У 2/3 частини «омега» вовків виділяють тих, хто повністю або частково співпадає з шаблоном  $W_s$ . Вовки, що повністю співпадають із шаблоном, модифікуються за такими правилами (3.34–3.36).

$$v = W_z(j + r * \alpha), \quad (3.34)$$

$$W_z(j + r * \alpha) = W_z(j), \quad (3.35)$$

$$v = W_z(j + r * \alpha), \quad (3.36)$$

де  $j$  – індекс елемента, значення якого буде замінено на значення елемента з індексом  $j + r * \alpha$ , а його значення генерується випадково  $j = rnd(J_{W_z})$  з елементів множини  $J_{W_z}$  ( $j \in J_{W_z}, j + r * \alpha \in J_{W_z}, 0 < j \leq n, 0 < j + r * \alpha \leq n$ , де  $n$  – кількість замовлень);

$r$  – коефіцієнт, що визначає напрямок руху вовка й визначається з використанням генератора випадкових чисел  $r = rnd(-1;1)$ ;

$a$  – коефіцієнт розміру кроку вовка визначається за допомогою генератора випадкових чисел  $\alpha = rnd(1;2)$ , тобто буде прийнято один із варіантів: 1 чи 2.

Вовки, що тільки частково співпадають із шаблоном, змінюються за таким алгоритмом:

- відбувається порівняння поточного  $W_i$  з шаблоном  $W_s$  і відшукується  $j_{ai}$  – індекс елемента в поточному, який не співпадає за значенням з елементом у шаблоні  $W_i(j_{ai}) \neq W_s(j_{ai})$ ;
- знаходимо  $j_{bi}$  – індекс елемента у  $W_i$ , що дорівнює значенню  $W_s(j_{ai})$ , тобто  $W_i(j_{bi}) = W_s(j_{ai})$ , при цьому  $j_{ai} \neq j_{bi}$ ;
- здійснюємо заміну (3.37):

$$\begin{aligned} W_i(j_{bi}) &= W_i(j_{ai}) \\ W_i(j_{ai}) &= W_s(j_{bi}) \end{aligned} \quad (3.37)$$

У 1/3 частині «омега» вовків виділяють усі ті, хто не увійшов до першої 2/3 частини. Вовки, які знаходяться у 1/3 частині, змінюються за правилами (3.34–3.36), але індекс  $j$  генерується випадково  $j = rnd(1 \dots n)$ .

Пропонується здійснювати розширення популяції вовків на кожній 3-ій та 5-ій ітерації алгоритму при формуванні усіх нових вовків за правилом (3.34–3.36), але, створюючи двох нових, використовуємо для одного  $r = -1$ , для іншого  $r = 1$ . При цьому необхідно відслідковувати загальну кількість вовків популяції, яка не повинна перевищувати  $2n$ . Також при застосуванні цього підходу на 5-ій та 7-ій ітерації здійснювати знищення агентів, які мають найгірше значення відповідно до набору критеріїв або загальної оціночної функції. Для будь-якої модифікації алгоритму «зграї вовків» достатньо контролювати два основних параметри, а саме розмір популяції і крок, з яким переміщаються вовки.

### **3.6 Алгоритм «мурашиної колонії»**

#### **3.6.1 Алгоритм «мурашиної колонії» з використанням елітних мурах**

Враховуючи складність задачі планування виконання замовлень та її велику розмірність, застосування алгоритму «мурашиної колонії» в первинному вигляді не дозволяє досить швидко знайти оптимальне або близьке до нього рішення. Отже, необхідним є визначення можливих шляхів покращення алгоритму, орієнтуючись на досліджувану предметну область.

Основа алгоритму «мурашиної колонії» полягає в реалізації принципу колективного розуму на прикладі мурашиної колонії. Для пошуку екстремальних значень цільової функції на кожній ітерації використовується визначена кількість агентів (штучних мурах), які будують таку ж кількість допустимих рішень задачі. Серед цих рішень вибирається частина найкращих за цільовою функцією. Отримана інформація накопичується в загальнодоступному банку даних і використовується агентами незалежно один від одного. Для наступних ітерацій дані компоненти будуть мати більшу ймовірність увійти до остаточного рішення, ніж це було на попередніх ітераціях. Кожен агент діє за правилами ймовірнісного алгоритму та при виборі напрямку орієнтується не тільки на прирощення цільової функції, а й на статистичну інформацію, що відображає попередню історію колективного пошуку [4, 12, 18, 45, 219, 256–275].

Однією з можливостей ефективного вдосконалення алгоритму «мурашиної колонії» для розв'язання задачі планування є його поєднання з генетичним алгоритмом, завдяки чому досягається динамічне налаштування параметрів під різні варіанти вхідних даних. Як наслідок, зменшується кількість необхідних для знаходження розв'язку ітерацій, а загальний час роботи зменшується.

За основу для подальшого вдосконалення береться модифікований алгоритм «мурашиної колонії», який представлений у роботі [12]. Комбінований алгоритм для розв'язання задачі планування виконання замовлень має такий вигляд [4, 12, 36, 219].

Схематичне представлення алгоритму «мурашиної колонії» наведено на рис.

3.5.

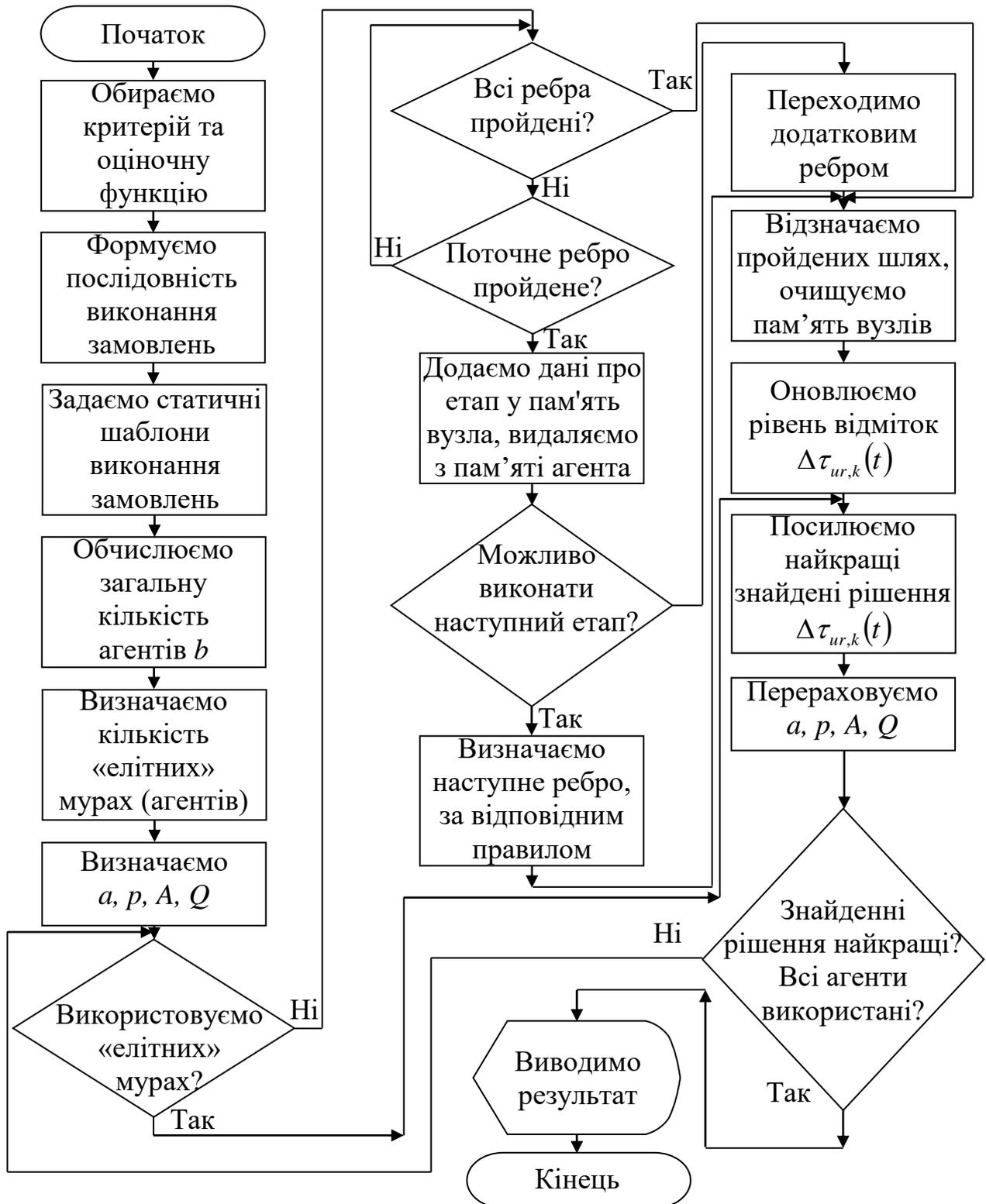


Рисунок 3.5 – Узагальнена блок-схема алгоритму «мурашиної колонії»

Комбінований алгоритм для розв'язання задачі планування виконання замовлень має наступні кроки.

1. Формуємо послідовності виконання етапів для кожного замовлення згідно із встановленими індивідуальними вимогами до їх здійснення, враховуючи при цьому можливості паралельної роботи над деякими окремими етапами.

2. Задаємо статичні шаблони виконання для замовлень, які необхідно виконувати разом або у певній послідовності, або на певний час. Таким чином, шаблон фіксує перехід від певного замовлення до іншого або визначає значення часу, коли саме необхідно почати/закінчити виконання замовлень.

3. Обчислюємо загальну кількість агентів для застосування алгоритму «мурашиної колонії» за формулою (3.38):

$$b = m * \left( (g - 1) * \omega_n * \max_l \sum_{i=1}^n \sum_{j=1}^{J_i} y_{ijl} \right), \quad (3.38)$$

де  $m$  – кількість виконавців, задіяних у виконанні замовлень за плановий період;

$l$  – номер виконавця,  $l \in M$ ,  $M = \{1, 2, \dots, m\}$  – множина виконавців;

$g$  – кількість вузлів на площині графа;

$n$  – кількість замовлень, які необхідно виконати за плановий період;

$\omega_n$  – загальна кількість етапів для  $n$  замовлень;

$i$  – номер замовлення;

$J_i$  – множина етапів  $i$ -го замовлення;

$j$  – номер етапу із множини етапів,  $j \in J_i$ ;

$y_{ijl}$  – параметр, який приймає значення  $\{0,1\}$ :  $y_{ijl} = 1$ , якщо  $j$ -ий етап  $i$ -го замовлення виконується  $l$ -им виконавцем;  $y_{ijl} = 0$  в іншому випадку.

4. Визначаємо кількість «елітних» агентів, тобто таких, які рухаються тільки найкращими знайденими на попередніх ітераціях маршрутами (3.39).

$$b_l = \frac{b}{2 * e}, \quad (3.39)$$

де  $e \approx 2.718$  [12, 339].

Використання «елітних» агентів забезпечує швидкість знаходження оптимального рішення, що є дуже важливим, оскільки в реальних ситуаціях необхідно підтримувати баланс між якістю запропонованих рішень та часом роботи алгоритму. Збільшення кількості спеціальних агентів є досить ефективним способом, який дозволяє значно скоротити число ітерацій всього алгоритму. Проте занадто велика їх кількість призводить до зупинки алгоритму на неоптимальному рішенні [12, 18].

5. Встановлюємо випадкові допустимі значення коефіцієнтів алгоритму «мурашиної колонії», які впливають на ефективність його роботи, а саме:

$\alpha$  – вага феромону,  $0 \leq \alpha \leq 1$ , яка визначає відносну значимість впливу інтенсивності відміток на вибір шляху, при  $\alpha = 0$  буде обрано найкоротше ребро для переходу, при  $\alpha = 1$  буде вибране ребро, на якому рівень відміток найбільший;

$p$  – коефіцієнт оновлення рівня відміток,  $0 \leq p \leq 1$ , який визначає відносне його зниження в часі;

$Q$  – параметр, значення якого має бути близьким з довжиною оптимального для поточної ітерації маршруту;

$A$  – коефіцієнт важливості спеціальних агентів, який визначає міру підсилення варіанта шляху.

6. Визначаємо із співвідношення кількості звичайних та спеціальних агентів до номера поточної ітерації якого агента використати. Якщо обираємо звичайного, переходимо до кроку 7, спеціального – 13.

7. Перевіряємо, чи всі ребра вже пройдені. Якщо так, переходимо до кроку 11, якщо ні – до кроку 8. Причому при розрахунку пройденого шляху не враховуються ті пройдені ребра у кінці шляху, для яких на певних часових проміжках всі етапи для даного виконавця вже виконані. Враховується тільки довжина шляху, ребра у якому відповідають виконанню робіт та затримці між ними.

8. Перевіряємо, чи пройдено на даний момент часу поточне ребро (завершений виконуваний етап). Якщо так, то крок 9, інакше – крок 7.

9. Додаємо дані про здійснений етап у пам'ять вузла й видаляємо з пам'яті агента.

10. Перевіряємо можливість виконання наступного етапу, враховуючи необхідну послідовність виконання та дані у пам'яті вузла. Якщо така можливість є, визначаємо за модифікованим імовірно-пропорційним правилом або, дотримуючись наявних шаблонів, наступне ребро для переходу (етап для виконання). У випадку необхідності виконання етапу, що має пару у шаблоні, відразу, без розрахунку, визначається, яке ребро буде наступним. Якщо можливість відсутня, переходимо до додаткового ребра.

11. Відмічаємо пройдений агентом шлях і очищаємо пам'ять вузлів, готуючи граф для проходу наступним агентом.

12. Оновлюємо рівень відміток на всіх ребрах, пройдених агентом. Для цього розраховуємо значення узагальненого критерію  $F_0$  для поточного варіанта рішення, який представляє собою маршрут  $T_k(t)$ , пройдений  $k$ -тим агентом на ітерації  $t$ . Рівень відміток визначаємо за формулою (3.40):

$$\Delta\tau_{ur,k}(t) = \begin{cases} \frac{1}{F_0} * \frac{Q}{L_k(t)}, & \text{якщо } (u, r) \in T_k(t) \\ 0, & \text{якщо } (u, r) \notin T_k(t) \end{cases}, \quad (3.40)$$

де  $u$  та  $r$  – індекси пари вузлів, що поєднують ребро, яким пройшов агент;

$L_k(t)$  – довжина маршруту  $T_k(t)$ .

13. Посилуємо найкращі знайдені рішення за допомогою відмітки спеціальних агентів. Значення відміток на кожному ребрі, яке залишається  $k$ -тим спеціальним агентом, пропонується визначати за формулою (3.41):

$$\Delta\tau_{ur,k}(t) = A * \frac{1}{F_0} * \frac{Q}{L_k(t)}. \quad (3.41)$$

14. Перераховуємо значення коефіцієнтів  $\alpha$ ,  $p$ ,  $Q$ ,  $A$  з використанням генетичного алгоритму за такою схемою:

- використовуємо наявні їх значення для отримання початкового набору хромосом.

- обчислюємо коефіцієнт виживання – кожному рішенню (хромосомі) ставиться у відповідність певне числове значення, залежне від близькості цього рішення до кращого отриманого на даний момент варіанта за значенням загальної оціночної функції, посиленого «спеціальними» агентами;

- виконуємо відтворення – хромосоми, що мають великий коефіцієнт виживання (більше числове значення), потрапляють до нащадків із більшою ймовірністю, а потім здійснюються оператори схрещування та мутації;

- формуємо покоління, отримані значення параметрів якого буде використовувати наступний набір агентів при знаходженні шляху.

15. Порівнюємо поточні рішення з отриманими на попередніх ітераціях і перевіряємо їх на оптимальність за допомогою розрахунку узагальненого критерію. Якщо знайдені рішення найкращі або вже використані усі звичайні та спеціальні агенти, тоді переходимо до кроку 16, інакше – крок 6.

16. Виводимо отримані результати, які є варіантом сформованого плану виконання замовлень. Отриманий варіант рішення включає в себе послідовності виконання замовлень планового періоду, покладені на часову шкалу.

### 3.6.2 Модифікований алгоритм «мурашиної колонії»

В основі алгоритму «мурашиної колонії» покладено принцип колективного розуму на прикладі їх поведінки при знаходженні оптимальних маршрутів для

пошуку їжі. У задачі складання оперативного плану виконання замовлень їжею виступає найкращий план, що є оптимальним за заданими критеріями та обмеженнями. Кожен варіант розкладу відповідає одній мураші, що здійснює під час алгоритму побудову власного маршруту, який відповідає послідовності виконання замовлень. Серед отриманих варіантів альтернативних рішень обираються найкращі варіанти за значенням цільової функції або заданих критеріїв. Отримана інформація накопичується і використовується мурахами на наступних ітераціях. Кожна мураха здійснює свої дії за правилами ймовірнісного алгоритму, при обранні наступного кроку використовує статистичну інформацію, яка відображає попередню історію колективного пошуку, а не тільки базується на цільовій функції [12, 219].

Спочатку всі замовлення розбиваються за можливістю їх виконання на певному обладнанні, і починається оперативне планування для визначеного обладнання та замовлень. За основу береться багатосаровий граф, кожен із шарів відповідає за певний рівень декомпозиції задачі.

Перший рівень має представлення рівня замовлень, на якому вузол позначає номер замовлення; ребра мають напрямлення, що відображають послідовність виконання замовлень, а саме, перехід від одного замовлення до іншого. Час початку виконання наступного замовлення визначається в залежності від виконання першого етапу поточного замовлення згідно з технологічним шляхом виготовлення продукції [12, 219].

Перехід від  $a$  та  $b$  вершинами графа визначає час, необхідний, щоб витратити для проходження технологічних операцій виготовлення продукції за замовленням  $a$ . При цьому, щоб не було простою та накладання операцій при виконанні замовлення  $b$  [12, 18, 219].

Другий рівень має деталізацію, на якій відображаються усі можливі варіанти виконання замовлень, що надійшли від замовника у вигляді графа, в якому позначаємо ребро – відділ або технологічну лінію  $l$ , на якій може виконуватись певна технологічна операція  $j$ . Вузлом графа є проміжний стан, в якому знаходиться напівфабрикат при переході між технологічними операціями. Перехід між вершинами графа визначає час  $\Delta t_{ijl}$ , необхідний для проходження усієї партії

напівфабрикату для здійснення відповідної  $i$ -тої технологічної операції,  $j$ -го етапу, на  $l$ -му обладнанні [12, 219].

Третій рівень декомпозиції відображає граф послідовності виконання замовлень для кожного окремого  $l$ -го обладнання.

Застосування модифікованого алгоритму «мурашиної колонії» передбачає, в залежності від виробництва та умов, які визначає ОПР, здійснення пошуку на різних рівнях. У найпростішому варіанті застосовується тільки перший рівень, що передбачає використання тільки повної автоматизованої лінії, яка виконує усі технологічні операції щодо виготовлення продукції [12, 219].

За рахунок того, що відбувається багаторазовий ітераційний пошук, на кожній ітерації обирається найкращий варіант серед поточних та порівнюється з глобальним значенням. Також відбувається контролювання найкращої послідовності виконання замовлень для заданої кількості мурах. При цьому кожна мураха відповідає одному з варіантів розкладу виконання замовлень, тобто при виконанні однієї ітерації алгоритму саме він його формує. Позитивний зворотний зв'язок реалізується як імітація поведінки мурах типу «залишаємо сліди – переміщаємося по слідах». Чим більше слідів залишено на стежці – ребра графа, тим більше мурах буде рухатись по ній [12, 18, 219, 272].

При цьому на стежці (яка відповідає послідовності виконання замовлень) з'являються нові сліди, які залучають додаткових мурах. Позитивний зворотний зв'язок реалізується за таким стохастичним правилом: імовірність включення ребра графа у маршрут мурахи пропорційна кількості феромону на ньому. Це правило забезпечує випадковість формування нових варіантів. Кількість відкладених мурахою феромонів на ребрі графа обернено пропорційно ефективності розкладу. Чим ефективніше розклад, тим більше феромонів буде відкладено на відповідних ребрах графа і тим більше мурах буде використовувати їх при синтезі своїх маршрутів. Відкладений на ребрах феромон дозволяє ефективним маршрутам (послідовностям виконання) зберігатися у глобальній пам'яті мурашника. Такі маршрути можуть бути удосконалені на наступних ітераціях алгоритму [12, 18, 270–274].

Використання тільки позитивного зворотного зв'язку призводить до передчасного знаходження збіжного рішення, коли усі мурахи рухаються одним і тим самим субоптимальним маршрутом. Для уникнення цього використовується негативний зворотний зв'язок – випаровування феромону. Час випаровування не повинен бути занадто великим, адже може виникнути попередня ситуація зі знаходженням субоптимального маршруту. Дуже малий час випаровування призведе до втрати пам'яті колонії, таким чином не забезпечить кооперативну поведінку мурах.

Кооперативність є важливою для алгоритму: безліч ідентичних мурах одночасно досліджують різні точки простору рішень і передають свій досвід через зміни комірок глобальної пам'яті мурашника. Для кожної мурашки перехід з вузла  $a$  у вузол  $b$  залежить від трьох складових: пам'ять мурах, видимість, віртуальний слід феромонів. Для другого рівня виникає перелік операцій, які необхідно виконати [12, 18, 219, 265–267].

Пам'ять мурах – перелік вже включених до виконання замовлень, які не повинні розглядатися при формуванні розкладу. Використовуючи цей список, мураха гарантовано не обирає одне й те ж саме замовлення. На кожному кроці пам'ять мурахи зростає, а на початку кожної нової ітерації алгоритму пам'ять стає порожньою. Позначимо через  $Lt_{a,i}$  список вже включених замовлень до розкладу  $i$ -ої мурахи ( $Lt_{a,i} \in n$ , де  $n$  – усі замовлення) після включення  $a$ -го замовлення, а перелік, який необхідно включити у розклад –  $Ln_{a,i}$  ( $Ln_{a,i} \in n$ , де  $n$  – усі замовлення). Тобто  $Lt_{a,i} \cup Ln_{a,i} = n$ , при цьому  $Lt_{a,i} \cap Ln_{a,i} = \emptyset$ . Для другого та третього рівнів декомпозиції графа враховується перелік  $j$ -го номера етапу із множини етапів  $j \in w_i$  для  $i$ -го замовлення,  $w_i$  – кількість необхідних етапів виготовлення  $i$ -го замовлення, а також  $l$ -й номер обладнання із множини обладнання ( $j \in \sigma_i$ ) для  $i$ -го замовлення,  $\sigma_i$  – кількість задіяного обладнання для виконання усіх етапів при виготовленні  $i$ -го замовлення [219, 220, 275].

Видимість – величина, обернена оцінка ребра графа  $a$  та  $b$ , а саме  $\eta_{a,b}(t) = 1/D_{a,b}(t)$ , де  $D_{a,b}(t)$  – величина оцінки ребра графа (в залежності від рівня декомпозиції управління може бути представлена у вартісному або часовому вигляді) між вершинами  $a$  та  $b$  (чим оптимальніша величина, тим бажаніше обрати цей перехід) на ітерації  $l$  [12, 18, 219, 265–270].

Віртуальний слід феромону на ребрі  $(a, b)$  відображає бажання відвідати вершину  $b$  з вершини  $a$  на основі мурашиного досвіду. На відміну від видимості, слід феромону є більш глобальною і динамічною інформацією. Вона змінюється після кожної ітерації алгоритму, відображаючи накопичений мурахами досвід [340]. Кількість віртуального феромону на ребрі  $(a, b)$  на ітерації  $l$ , що залишає  $i$ -а комаха, позначимо через  $\xi_{a,b}(l, i)$ . На першій ітерації  $l=0$  роботи алгоритму кількість феромону для кожної  $i$ -ої мурахи приймається рівним невеликому позитивному числу  $0 < \xi_{0,b}(0, i) < 1$ .

Імовірність переходу  $i$ -ої мурахи у вершину  $b$  з вершини  $a$  на  $l$ -й ітерації визначається за формулою (3.42) [275]:

$$P_{a,b}(l, i) = \begin{cases} \frac{[\xi_{a,b}(l, i)]^\alpha [\eta_{a,b}(l, i)]^\beta}{\sum_{b=0}^{Ln_{a,i}} ([\xi_{a,b}(l, i)]^\alpha [\eta_{a,b}(l, i)]^\beta)},, \text{ якщо } b \in Ln_{a,i} \\ 0, \text{ якщо } b \notin Ln_{a,i} \end{cases}, \quad (3.42)$$

де  $\alpha$  – коефіцієнт ваги феромону,  $0 \leq \alpha \leq 1$ , який визначає відносну значимість впливу інтенсивності відміток на вибір шляху. При  $\alpha = 0$  буде обрано найкоротше ребро (за оцінкою задачі – часу, витрат) для переходу, що відповідає жадібному алгоритму, при  $\alpha = 1$  буде вибране ребро, на якому рівень відміток найбільший;

$\beta$  – коефіцієнт видимості при виборі маршруту,  $0 \leq \beta \leq 1$ , яка визначає відносну значимість впливу видимості на вибір шляху, при  $\beta = 0$  буде працювати тільки посилення феромонів, що приведе до знаходження локального субоптимального рішення;

$\alpha + \beta = 1$ , при цьому коефіцієнт  $\alpha$  визначає жадібність алгоритму, а  $\beta$  стабільність.

Формула (3.51) визначає лише ймовірність обрання наступного замовлення, для включення його до розкладу виконання, що буде відповідати на графі обрання наступної вершини. Отримане значення за формулою (3.51) не змінюється в ході ітерації алгоритму, але у двох різних мурах значення ймовірності переходу будуть відрізнятися, оскільки вони мають різний перелік відвіданих вершин [269–275].

Після завершення маршруту кожна мураха відкладає на кожному ребрі кількість феромонів, що включено до його маршруту й відповідає розкладу, які розраховуються за формулою (3.43):

$$\Delta \xi_{a,b}(t,i) = \begin{cases} \frac{1}{F'_0} * \frac{Q}{S_i(t)}, & \text{якщо } (a,b) \in Lt_i(t) \\ 0, & \text{якщо } (a,b) \notin Lt_i(t) \end{cases}, \quad (3.43)$$

де  $a$  та  $b$  – індекси пари вузлів, що поєднують ребро, яким пройшов агент;

$Lt_i(t)$  – сформований маршрут на  $t$ -й ітерації  $i$ -ою мурахою;

$S_i(t)$  – довжина маршруту  $Lt_i(t)$ , яка виражена у часі або у вартості;

$Q$  – регульований параметр, наближений до оптимального маршруту, що заданий чи розрахований за попередні ітерації;

$F'_0$  – оцінка маршруту згідно з обраним частковим критерієм або оціночною функцією.

Для того, щоб кількість відкладених феромонів не було надлишковою, проводиться їх оновлення згідно з формулою (3.44) із урахуванням коефіцієнта оновлення рівня феромонів  $0 \leq \rho \leq 1$ , який визначає відносне його зниження в часі.

$$\Delta \xi_{a,b}(t+1) = (1 - \rho) * \Delta \xi_{a,b}(t) + \sum_{i=1}^n \Delta \xi_{a,b}(t,i), \quad (3.44)$$

де  $n$  – кількість мурах, що відповідає кількості замовлень;

$\rho \in [0,1]$  – коефіцієнт випаровування феромону.

У вітчизняній та зарубіжній літературі виділяється модифікований алгоритм Min-Max Ant System (MMAS), який базується на підвищенні концентрації феромонів тільки на кращих шляхах [275].

Для захисту від передчасного знаходження субоптимального рішення пропонується введення обмежень на концентрацію феромонів на ребрах  $\Delta\xi_{\min} \leq \Delta\xi_{a,b} \leq \Delta\xi_{\max}$ .

Формула оновлення кількості феромонів на ребрах графа перетворюється з виразу (3.44) на формулу (3.45) [220, 275].

$$\Delta\xi_{a,b}(t+1) = (1-\rho) * \xi_{a,b}(t) + \Delta\xi_{a,b,best} = (1-\rho) * \xi_{a,b}(t) + \frac{1}{Lt_{best}(t)}, \quad (3.45)$$

де  $Lt_{best}(t)$  – найкращий сформований маршрут на  $t$ -й ітерації.

Після кожної ітерації алгоритму залишає за собою слід лише одна мураха на вибір із двох варіантів: краща на поточній ітерації  $Lt_{best}(t) = Lt_{t\_best}(t)$ ; краща за весь час роботи алгоритму  $Lt_{best}(t) = Lt_{gl\_best}(t)$ . Доцільно на перших ітераціях використовувати  $Lt_{t\_best}(t)$ , а на останніх ітераціях  $Lt_{gl\_best}(t)$ .

При цьому на кожній ітерації виконується коригування кількості феромону на кожному ребрі графа за формулою (3.46):

$$\xi_{a,b}(t) = \begin{cases} \xi_{\min}, & \text{якщо } \xi_{a,b}(t) < \xi_{\min} \\ \xi_{a,b}(t), & \text{якщо } \xi_{\min} \leq \xi_{a,b}(t) \leq \xi_{\max} \\ \xi_{\max}, & \text{якщо } \xi_{\max} < \xi_{a,b}(t) \end{cases} \quad (3.46)$$

Замість емпіричного правила вибору кращої мурахи пропонується використовувати формулу (3.47) на основі закону розподілу ймовірностей Коші, що дозволяє плавно здійснювати перехід від технології глобального пошуку, на

початкових кроках, до технології обрання найкращих глобального пошуку за весь час роботи алгоритму на заключних етапах.

$$P_{term}(t) = \frac{1}{\pi} \arctg \left( \frac{t - \frac{l_{max}}{2}}{t - \frac{l_{max}}{10}} \right) + 0.5, \quad (3.47)$$

де  $l_{max}$  – обрана кількість поколінь життя колонії.

Кращою мурахою за весь час роботи алгоритму обирається  $Lt_{gl\_best}(t)$  або найкращий на поточній ітерації  $Lt_{t\_best}(t)$  на основі порівняння їх значень.

Використання операцій диверсифікації здійснюється, якщо за час від останньої ініціалізації не сталося покращення глобального рішення на кожні  $\frac{l_{max}}{2}$  або  $\frac{l_{max}}{10}$  ітерації.

Під час операції диверсифікації відбувається повторна ініціалізація кількості феромону на всіх ребрах. Визначення максимального значення феромонів відбувається за формулою (3.48) шляхом присвоєння їм значення, оберненого до коефіцієнта випаровування і помноженого на оцінку глобального оптимального шляху [270-274].

$$\xi_{max} = \frac{1}{\rho * Lt_{gl\_best}(t)}. \quad (3.48)$$

Мінімальне значення концентрації феромону на ребрі обчислюється через (3.49):

$$\xi_{min} = \frac{\xi_{max}}{2 * n}, \quad (3.49)$$

де  $n$  – кількість мурах, що відповідає кількості замовлень.

Після кожного знаходження кращого рішення значення величин  $\xi_{\max}$  та  $\xi_{\min}$  повинні бути перераховані за формулами (3.58) і (3.59) відповідно [275].

При побудові маршруту за алгоритмом MMAS обрання переходу з  $a$ -го вузла здійснюються на основі правила (3.48), але для обрання наступного вузла використовується не весь перелік вузлів  $L_{n_{a,i}}$  ( $L_{n_{a,i}} \in n$ , де  $n$  – усі замовлення), які ще необхідно відвідати  $i$ -й мурасі, а тільки список найближчих вершин –  $nlist$ . Список  $nlist$  – це матриця розміром  $n \times n'$ , де  $n$  – кількість вершин графа,  $n'$  – кількість найближчих вершин.

Кожний  $i$ -ий рядок цієї матриці містить номери найближчих вершин, впорядкованих за віддаленістю від  $a$ -ї вершини. За рахунок того, що список  $nlist$  розраховується й формується до початку роботи алгоритму за відомою матрицею відстаней  $D$ , значно прискорюється побудова маршруту на етапі вибору наступної вершини. Якщо усі вершини множини  $nlist$  вичерпані, обрання вершини  $b$  буде визначатися з урахуванням відстані і рівня феромонів ребра за формулою (3.50) [270–275]:

$$b = \max_{b \in L_{n_{a,i}}} \{ [\xi_{a,b}(t,i)]^\alpha [n_{a,b}(t)]^\beta \}. \quad (3.50)$$

Отже, згідно з урахуванням усіх модифікацій, алгоритм розв'язку буде мати такий вигляд [340].

1. Обрання періоду планування. Згідно з заданим періодом, обираються замовлення, які необхідно виконати за цей період на певному обладнанні.
2. Обрання оціночної функції та часткових критеріїв вибору для розв'язання задачі.
3. Формування багат шарового графа, який описує кожен із шарів, що відповідає за певний рівень декомпозиції задачі.
4. Ініціалізація параметрів алгоритму  $\alpha$ ,  $\beta$ ,  $\rho$  та  $Q$ .

5. Обрання правил розрахунку параметрів видимості  $\eta_{a,b}$  та концентрації феромону  $\xi_{a,b}$ .

6. Усі замовлення сортуються за параметром часу  $dt_i$ , на який необхідно виготовити продукцію за  $i$ -м замовленням. У результаті отримуємо відсортований список замовлень  $Ln_{a,i}$  ( $Ln_{a,i} \in n$ , де  $N$  – усі замовлення, які необхідно виконати за період).

7. Здійснюємо побудову початкових маршрутів одночасно у прямому та у зворотному напрямку. Побудова прямого напрямку передбачає послідовність, отриману після сортування. Фактично ми приймаємо за таку послідовність відсортований список замовлень  $Ln_{a,i}$  ( $Ln_{a,i} \in n$ , де  $N$  – усі замовлення). А побудова зворотного напрямку передбачає формування від останнього до першого замовлення. При побудові у зворотному порядку додатковою умовою обрання наступного замовлення є час  $dt_i$ , на який необхідно виготовити продукцію за  $i$ -м замовленням. Але потім здійснюємо реверс. Таким чином, ми отримуємо значення  $Lt_{gl\_best}(t)$  глобального та  $Lt_{l\_best}(t)$  локального (на поточній ітерації) оптимуму.

8. Формуємо популяцію мурах, кожна з яких на першій вершині відповідає певному замовленню, що сформоване з використанням генератора випадкових чисел. Фактично кожна мураха на початку свого шляху повинна пройти шлях на графі, що відповідає певному замовленню.

9. Виконуємо цикл за часом життя колонії  $t_{max} \Leftarrow 1..t$ .

9.1. Виконуємо цикл по всіх мурашках  $i \Leftarrow 1..n$ .

9.1.1. Будується маршрут для кожної нової мурахи за (3.53), і розраховується довжина  $Lt_i(t)$ .

9.1.2. Застосувати до маршруту локальний пошук 2-орт та/або 3-орт.

9.1.3. Проводимо виділення феромонів за (3.45).

9.2. Проводимо оцінку кожного з маршрутів та порівнюємо їх з локальним і глобальним оптимальним значенням, для оновлення інших.

9.3. Виконуємо оновлення феромону на усіх ребрах графа (3.44).

10. Виведення найкращого локального та глобального оптимуму.

### 11. Завершення алгоритму.

Після завершення алгоритму отримуємо оптимальний план виконання замовлень у вигляді локального та глобального оптимального маршруту.

Як видно з наведеного алгоритму, після того, як був побудований маршрут, над отриманим рішенням здійснюється додаткове покращення одним із методів локальної оптимізації, а саме методами 2-opt та 3-opt [275, 276].

Модифікований алгоритм «мурашиної колонії» у вигляді блок-схеми наведений на рис. 3.6.



Рисунок 3.6 – Модифікований алгоритм «мурашиної колонії»

На початку застосування алгоритму «мурашиної колонії» необхідно задати, чи буде автоматично обиратися один із методів 2-opt або 3-opt, чи необхідно використати обидва методи та обрати один із найкращих.

Локальна оптимізація за методом 2-opt (рис. 3.7) полягає в тому, що необхідно обрати дві пари елементів розкладу, що не є суміжними, і здійснити їх обмін між собою, потім провести оцінку і залишити найкращий варіант. Якщо після заміни отримуємо найкращий варіант, то залишаємо його [276], що й наведено на рис. 3.7.

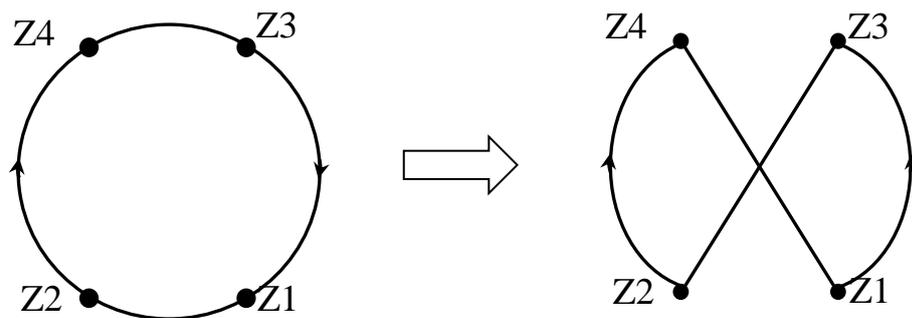


Рисунок 3.7 – Схематичне використання методу 2-opt

Застосування локальної оптимізація за методом 2-opt необхідно проводити для кожного варіанта плану, що в алгоритмі мурах відповідає маршруту мурахи. У варіанті плану здійснюється пошук двох несуміжних послідовностей, обмін яких забезпечить максимальний ефект згідно з оціночною функцією або набором критеріїв. Приклад застосування методу 2-opt наведено на рис. 3.8.

Варіант плану до застосування методу 2-opt

1	2	5	6	4	3
---	---	---	---	---	---

Варіант плану після застосування методу 2-opt

1	3	5	6	2	4
---	---	---	---	---	---

Рисунок 3.8 – Схематичне використання методу 2-opt

При застосуванні методу 3-opt (рис. 3.9) використовується підхід до методу 2-opt, але обираються 3 пари не суміжних елементів розкладу для заміни їх між собою [276].

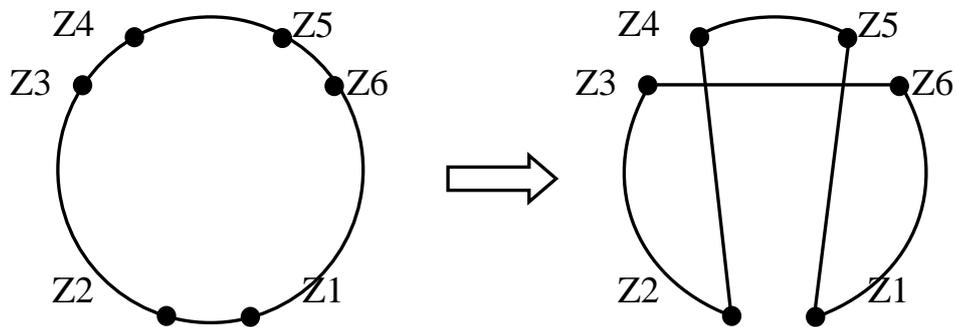


Рисунок 3.9 – Приклад використання методу 3-opt

Таким чином відбувається покращення знайденого маршруту на кожній ітерації, що забезпечить скоротити загальну кількість ітерацій алгоритму, а також загальний час роботи алгоритму.

### 3.7. Порівняння обраних методів та рекомендації щодо їх застосування

Розроблена математична модель у розділі 2 дає змогу оцінити оперативно-календарний план виконання замовлень. Задача полягає у тому, що потрібно розподілити виконання замовлень між існуючими виробничими цехами і визначити послідовність їх виконання на існуючому технологічному обладнанні таким чином, щоб мінімізувати витрати виконання замовлень.

Необхідно відмітити, що поставлена задача повинна вирішуватися з урахуванням зовнішніх та внутрішніх впливів. У залежності від поточної ситуації особа, яка приймає рішення, повинна врахувати усі часткові критерії, проранжувавши їх, або обрати необхідний набір критеріїв.

Математична модель враховує такі основні напрями діяльності харчового підприємства та процесу планування виконання замовлень:

- максимізація прибутку від виконання усіх замовлень; мінімізація часу на виготовлення продукції;
- максимізація запасу часу при виконанні замовлень; мінімізація сумарних штрафів за невчасне виконання замовлення;
- мінімізація сумарного часу простою усіх одиниць технологічного обладнання та відділів; мінімізація сумарних витрат при виробництві;
- мінімізація сумарних витрат від простоїв при невикористанні обладнання, коли деякі технологічні контури, не зважаючи на невикористання, потребують електро- та теплоспоживання;
- мінімізація сумарних витрат на переробку та утилізацію отриманої некондиційної продукції при виконанні усіх замовлень;
- мінімізація сумарних витрат на зберігання готової продукції, а іноді й на транспортування; мінімізація сумарних витрат на зберігання сировини та матеріалів.

Кожне підприємство має набір критеріїв, які відповідають його потужності та особливостям виготовлення продукції. Разом з тим харчова промисловість має спільні характеристики, але підприємства відрізняються між собою певними особливостями.

Для побудови плану виконання замовлень необхідно здійснити повний перебір варіантів послідовності їх виконання на відповідному технологічному обладнанні. Складність збільшується, коли є можливість виконання замовлень на технологічному обладнанні, що може виготовляти однакову продукцію з різною потужністю. Існують різні варіанти її виготовлення.

Фактично остаточний план виконання замовлень пов'язується з технологічним обладнанням, на якому продукція буде виготовлятися. Необхідно враховувати, що в певних випадках виникають прив'язки до надходження відповідної сировини.

Отже, дана задача належить до класу багатокритеріальних NP-складних комбінаторних задач. Складність розв'язання такої задачі зростає з кількістю

замовлень, а також зі збільшенням етапів різних варіантів виконання на різних технологічних ланках виробництва.

Задача спрощується, якщо на підприємстві технологічний процес виготовлення продукції відбувається на одному автоматизованому технологічному комплексі, що має конвеєрний безперервний цикл виробництва.

Прикладом може бути макаронне виробництво, де виготовлення певного виду продукції здійснюється на одній автоматизованій технологічній лінії. Такий варіант найпростіший, тому що план формується для одного виду технологічного обладнання, що вимагає підібрати тільки послідовність виконання замовлень.

Зі збільшенням кількості різного обладнання на підприємстві збільшується і кількість варіантів виконання плану, адже вони будуть виконуватися паралельно.

Проведено порівняння таких алгоритмів:

- алгоритм «бджолої колонії»;
- алгоритм «хаотичного кажана»;
- алгоритм «кажана на основі стратегії пошуку польоту Леві»;
- алгоритм «кажана на основі коефіцієнта скорочення»;
- «генетичний алгоритм»;
- алгоритм пошуку «косяку риб»;
- алгоритм «зграї вовків»;
- модифікований алгоритм «зграї вовків»;
- алгоритм «мурашиної колонії з використанням елітних мурах»;
- модифікований «алгоритм мурашиної колонії».

Апробація розглянутих алгоритмів проводилась на різних підприємствах харчової галузі. Для отримання достовірної інформації при застосуванні розглянутих алгоритмів обирались дані за планами виконання замовлень за різні попередні періоди діяльності підприємств. Випадковим чином за різні періоди часу було обрано 25, 50 та 100 замовлень. На початку роботи алгоритмів замовлення розташовувалися у тому ж порядку, як вони надходили на підприємство.

Для порівняння алгоритмів спочатку для кожного з них було здійснено підбір параметрів його роботи для обраної вибірки, а потім порівнювались результати роботи алгоритмів.

У зв'язку з тим, що неможливо розголошувати інформацію, яка складає комерційну таємницю, а саме відомості про асортимент та обсяги виготовлення продукції в замовленнях, терміни виконання замовлень, закупівельні ціни на сировину, собівартість продукції тощо, ефективність алгоритмів оцінювалась на основі таких показників:

- часу пошуку оптимального розкладу;
- ефективність знайденого плану у грошовому вимірі як різниця між оцінками за цільовою функцією розрахункового плану та фактичного плану;
- ефективність скорочення часу виконання замовлень (розраховується, як різниця між розрахованим планом та фактичним планом за часом виконання).

Відсоток ефективності розраховується за формулою (3.51):

$$\varepsilon = \frac{f - f_{opt}}{f_{opt}} 100\% , \quad (3.51)$$

де  $f$  – оцінка фактичного плану за цільовою функцією;

$f_{opt}$  – оцінка сформованого плану за цільовою функцією.

Ефективність знайденого плану розраховується за формулою (3.52):

$$f_E = f - f , \quad (3.52)$$

де  $f$  – оцінка фактичного плану за цільовою функцією;

$f_{opt}$  – оцінка сформованого плану за цільовою функцією.

У табл. 3.1–3.3 наведено порівняння застосування алгоритмів для різної кількості замовлень.

Таблиця 3.1 – Порівняння алгоритмів при 25 замовленнях

№ п/п	Назва алгоритму	Час пошуку плану, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
			у.о.	%	год	%
1	Алгоритм «бджолоїної колонії»	4	753	2	2	1
2	Алгоритм «хаотичного кажана»	4	753	2	2	1
3	Алгоритм «кажана на основі стратегії пошуку польоту Леві»	4	753	2	2	1
4	Алгоритм «кажана на основі коефіцієнта скорочення»	4	753	2	2	1
5	«Генетичний алгоритм»	4	753	2	2	1
6	Алгоритм пошуку «косяку риб»	4	753	2	2	1
7	Алгоритм «зграї вовків»	4	753	2	2	1
8	Модифікований алгоритм «зграї вовків»	3,9	753	2	2	1
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	4	753	2	2	1
10	Модифікований «алгоритм мурашиної колонії»	3,9	753	2	2	1

З результатів розрахунків, наведених у табл. 3.1, видно, що усі алгоритми знайшли однаковий план, який на 2% ефективніший від фактичного, і загальний час

його виконання на 2 години менший від фактичного. Швидкість пошуку в порівнянні з іншими алгоритмами на 2,56 % швидша.

Таблиця 3.2 – Порівняння алгоритмів при 50-ти замовленнях

№ п/п	Назва алгоритму	Час пошуку плану, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
			у.о.	%	год	%
1	Алгоритм «бджолоїної колонії»	5,9	1256	3	1,5	1
2	Алгоритм «хаотичного кажана»	5,9	1256	3	1,5	1
3	алгоритм «кажана на основі стратегії пошуку польоту Леві»	5,85	1256	3	1,5	1
4	Алгоритм «кажана на основі коефіцієнта скорочення»	5,8	1256	3	1,5	1
5	«Генетичний алгоритм»	5,7	1503	3,7	3	2
6	Алгоритм пошуку «косяку риб»	5,6	1756	4,3	3	2
7	Алгоритм «зграї вовків»	5,5	1756	4,3	3	2
8	Модифікований алгоритм «зграї вовків»	5,4	1756	4,3	3	2
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	5,55	1756	4,3	3	2

Продовження таблиці 3.2

10	Модифікований «алгоритм мурашиної колонії»	5,4	1756	4,3	3	2
----	--	-----	------	-----	---	---

З даних, наведених у табл. 3.2, видно, що результати пошуку для алгоритмів різні. Ефективність плану для алгоритмів «бджолоїної колонії», «хаотичного кажана», «кажана на основі стратегії пошуку польоту Леві» та «кажана на основі коефіцієнта скорочення» майже однакові і мінімальні в порівнянні з іншими алгоритмами.

Найкращий результат показали модифікований алгоритм «зграї вовків» та модифікований алгоритм «мурашиної колонії». Їх швидкість пошуку в порівнянні з першими 4-ма алгоритмами на 9,25 % швидша, а у порівнянні зі своїми немодифікованими версіями швидша на 1,8%.

Таблиця 3.3 – Порівняння алгоритмів при 100-та замовленнях

№ п/п	Назва алгоритму	Час пошуку плану, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
			у.о.	%	год	%
1	Алгоритм «бджолоїної колонії»	7,8	2156	4	3	2
2	Алгоритм «хаотичного кажана»	7,7	2364	4,39	3	2
3	Алгоритм «кажана на основі стратегії пошуку польоту Леві»	7,55	2345	4,35	3	2

Продовження таблиці 3.3

4	Алгоритм «кажана на основі коефіцієнта скорочення»	7,6	2163	4,01	3	2
5	«Генетичний алгоритм»	7	2420	4,49	6	4
6	Алгоритм пошуку «косяку риб»	6,8	3523	6,99	6	4
7	Алгоритм «зграї вовків»	6,8	3523	6,99	6	4
8	Модифікований алгоритм «зграї вовків»	6,51	3523	6,99	6	4
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	6,8	3523	6,99	6	4
10	Модифікований «алгоритм мурашиної колонії»	6,51	3523	6,99	6	4

З даних, наведених у табл. 3.3, видно, що результати пошуку для алгоритмів різні. Найкращий результат показали модифікований алгоритм «зграї вовків» та модифікований алгоритм «мурашиної колонії». Швидкість їх пошуку майже на 20 % більша, а у порівнянні зі своїми немодифікованими версіями швидша на 4,45%.

Також було проведено ряд досліджень, при яких бралися до уваги тільки грошові критерії математичної моделі, наведеної у розд. 2. В результаті досліджень встановлено, що алгоритм «кажана на основі стратегії пошуку польоту Леві» може дати більший ефект при порівнянні з іншими, але виникає необхідність у підборі значень вільних параметрів, від значення яких часто залежить результат, а також відсутність підстав для вибору цих значень. Час, витрачений на підбір таких параметрів, може бути більшим, ніж робота самого алгоритму. А їх підбір та

відповідність покладаються на людину. Крім того, алгоритм «кажана на основі стратегії пошуку польоту Леві» сильно залежить від початкового згенерованого варіанта послідовності виконання плану.

Необхідно відмітити, що час пошуку оптимального плану залежить від характеристик комп'ютерної техніки. Більшість випробувань проводилась на Intel Core i5-4690K (3.5ГГц) / RAM 16 ГБ / HDD 1 ТБ.

Ведення розрахунків із зберіганням великих обсягів даних без використання спеціальних засобів зберігання, таких, як СУБД, неможливо, адже необхідно постійно зберігати велику кількість додаткових розрахунків.

У зв'язку з тим, що модифіковані алгоритми «зграї вовків» та «мурашиної колонії», а також алгоритм «косяку риб» та «генетичний алгоритм» показали свою ефективність, виникає гіпотеза про їх комбіноване застосування у вигляді методу розв'язання задачі формування плану виконання замовлень.

При застосуванні розглянутих алгоритмів необхідно організувати діалог з особою, що приймає рішення, для забезпечення ефективності та комфортності їх використання.

### **3.8 Висновки до розділу 3**

Були досліджені такі алгоритми: алгоритм «бджолоїної колонії»; алгоритм «хаотичного кажана»; алгоритм «кажана на основі стратегії пошуку польоту Леві»; алгоритм «кажана на основі коефіцієнта скорочення»; «генетичний алгоритм»; алгоритм пошуку «косяку риб»; алгоритм «зграї вовків»; модифікований алгоритм «зграї вовків»; алгоритм «мурашиної колонії з використанням елітних мурах»; модифікований «алгоритм мурашиної колонії».

Усі розглянуті алгоритми в зарубіжній літературі мають у своїй назві слово «алгоритм», але за складністю застосування їх до розв'язання задачі формування розкладу виконання замовлення потребують підготовки та перетворення вхідних

даних до їх застосування. Таким чином, процедура застосування таких алгоритмів для отримання розв'язку оптимального рішення NP-повної задачі формування розкладу повинна розглядатися як метод на базі відповідного методу.

Модифікація та покращення окремих алгоритмів не завжди сприяє пришвидченню розв'язання задачі, що підтверджено експериментально. При кількості замовлень менше 25-ти усі алгоритми знаходять однакові значення, але за часом пошуку вони відрізняються. При кількості замовлень менше 25-ти кращі показники за часом мають стандартні версії алгоритмів. А при кількості замовлень 50 і більше кращі показники мають модифіковані алгоритми.

На основі проведених досліджень літературних джерел були запропоновані модифікації алгоритмів «мурашиної колонії» та «зграї вовків» для розв'язання задачі формування плану виконання замовлень, що дозволяє при їх застосуванні формувати новий план майже на 20% швидше у порівнянні з іншими алгоритмами.

Також необхідно відмітити, що при врахуванні тільки грошових критеріїв математичної моделі алгоритм «кажана на основі стратегії пошуку польоту Леві» може дати більший ефект при порівнянні з іншими, але виникає необхідність підбору значень вільних параметрів, від значення яких часто залежить результат, а також відсутність підстав для вибору цих значень. Витрати часу на підбір вільних параметрів не виправдані, тому що переважає час роботи самого алгоритму. Алгоритм «кажана на основі стратегії пошуку польоту Леві» сильно залежить від початкового згенерованого варіанта послідовності виконання плану, що є стартовою при генерації нових варіантів.

Без використання інформаційних технологій застосування розглянутих методів неможливе, адже виникає потреба у зберіганні великих обсягів даних як для вхідних даних, так і для проміжних. Необхідно відмітити, що застосування усіх алгоритмів повинно відбуватися у діалозі з особою, що приймає рішення, для забезпечення ефективності та комфортності їх використання. Кількість критеріїв оптимальності впливає на ефективність та час пошуку рішення.

Зі зменшенням кількості часткових критеріїв задача спрощується, і час розрахунків зменшується.

Неефективність використання алгоритму «бджолої колонії», алгоритму «хаотичного кажана», алгоритму «кажана на основі стратегії пошуку польоту Леві», алгоритму «кажана на основі коефіцієнта скорочення» обумовлюється складністю налаштування їх параметрів.

Ведення розрахунків із зберіганням великих обсягів даних без використання спеціальних засобів зберігання, таких як СУБД, неможливе, оскільки необхідно постійно зберігати велику кількість додаткових розрахунків.

У зв'язку з тим, що модифіковані алгоритми «зграї вовків» та «мурашиної колонії», а також алгоритм «косяку риб» та «генетичний алгоритм» показали свою ефективність, виникає гіпотеза про їх комбіноване застосування у вигляді методу розв'язання задачі формування плану виконання замовлень.

Основні результати розділу опубліковані автором у працях [3, 18, 22, 27, 28, 31, 32, 34, 37, 45, 50].

## РОЗДІЛ 4

### СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПЛАНУВАННЯ ВИКОНАННЯ ЗАМОВЛЕНЬ НА ОСНОВІ КОМБІНОВАНИХ МЕТОДІВ ТА АЛГОРИТМІВ

#### 4.1 Комбінований метод прийняття рішень на основі алгоритмів «косяку риб» і модифікованого «зграї вовків»

На основі проведених досліджень у розділі 3 було висунуто гіпотезу про те, що поєднання алгоритмів «косяку риб» і «зграї вовків» може дати скорочення часу пошуку.

Пропонується новий комбінований метод на основі алгоритмів «косяку риб» і модифікованого «зграї вовків» для вирішення задач управління та складання послідовності виконання замовлень.

У комбінованому методі алгоритмів «косяку риб» та «зграї вовків» доцільніше застосувати модифікований алгоритм «зграї вовків» замість алгоритму «зграї вовків», що копіює процес полювання зграї вовків у природі [28, 32, 50]. Кожен вовк зграї ототожнюється альтернативному оперативному плану на кожній ітерації. Після виконання кожної ітерації для кожного вовка, використовуючи цільову функцію, розраховується значення його альтернативного оперативного плану. За значенням оцінки кожного вовка в зграї їх поділяють на чотири типи: «альфа» – вовк-ватажок, оцінка якого є оптимальне рішення за частковими критеріями або оціночною функцією; «бета» і «дельта» – це вовки, які заганяють жертву, оцінка яких друге і третє місце серед кращих; «омега» – всі інші вовки [22]. Перші три типи вовків фіксуються на такі ітерації до тих пір, поки знайдуться нові альтернативні варіанти рішення, які будуть краще поточних або буде вичерпано задану кількість ітерацій. Вовки «альфа», «бета», «дельта» впливають на формування «омега» вовків [22]. Використовуючи алгоритм «зграї вовків», отримуємо варіанти планів виконання замовлень.

Алгоритм «косяку риб» заснований на харчовій поведінці косяку риб. Вони пересуваються з метою пошуку їжі, що є оптимальним рішенням у межах області пошуку. При харчуванні вага кожної риби збільшується та формалізує її індивідуальний успіх у пошуку рішення і відіграє роль пам'яті риби. При знаходженні ефективного місця харчування риби об'єднуються у косяк.

Загальна вага показує, наскільки ефективний косяк риб: якщо вага збільшується, косяк успішний, якщо зменшується, то неуспішний. Весь косяк рухається до розташування найуспішніших риб. Як вага виступає оціночна функція або сукупність результатів критеріїв, які були обрані особою, яка приймає рішення. В іншому випадку косяк поширюється на всій запропонованій території [28, 32, 50]. У проєкції на задачу виконання замовлень харчового підприємства фактичний рух відбувається по порядку виконання замовлень на виготовлення продукції, а найефективнішим є той, при якому за менший час виконано всі замовлення з мінімальними витратами і отримано максимальний прибуток.

Основними операторами алгоритму «косяку риб» є оператор годування і правило плавання (пересування агентів). Оператор годування визначає, як виконується приріст ваги агента на кожній ітерації. В алгоритмі «косяку риб» розрізняють три види плавання: індивідуальне, інстинктивно-колективне і колективно-вольове.

При індивідуальному плаванні відбувається випадкове переміщення агента з однаковою ймовірністю в будь-якому напрямку із заданою швидкістю або на задану відстань. У розглянутій задачі відстань переходу виступає як замовлення на виконання. Індивідуальне плавання включає в себе не одну ітерацію, воно направлено на пошук оптимального рішення.

Колективно-вольове плавання виконується після інстинктивно-колективного плавання. Даний вид плавання полягає у зміщенні всіх агентів у напрямку поточного центру ваги популяції, якщо сумарна вага косяку в результаті індивідуального та інстинктивно-колективного плавання збільшилась. В протилежному напрямку, якщо ця вага зменшилась. В іншому випадку популяція розширюється від того ж центру, підвищуючи свої диверсифікаційні властивості.

В комбінованому алгоритмі «косяку риб» та «зграї вовків» при колективно-вольовому плаванні доцільно застосувати алгоритм «зграї вовків», що копіює процес полювання зграї вовків в природі [28, 32, 50]. Згідно з нашим завданням, зграя полює за жертвою, що відповідає оптимальному оперативному плану виконання замовлень. Кожен вовк зграї відповідає альтернативному оперативному плану на кожній ітерації. Після виконання кожної ітерації для кожного вовка розраховується значення його альтернативного оперативного плану, використовуючи цільову функцію. За значенням оцінки кожного вовка в зграді їх поділяють на чотири типи: «альфа» – вовк-ватажок, оцінка якого є оптимальне рішення за частковими критеріями або оціночною функцією; «бета» і «дельта» – це вовки, які заганяють жертву, оцінка яких – друге і третє місце серед кращих; «омега» – всі інші вовки [22]. Перші три типи вовків фіксуються на наступні ітерації до тих пір, поки не будуть знайдені нові альтернативні варіанти рішення, краще поточних, або буде вичерпано задану кількість ітерацій. Вовки «альфа», «бета», «дельта» впливають на формування «омега» вовків [22]. Використовуючи алгоритм «зграї вовків», отримуємо варіанти планів виконання замовлень.

Комбінований метод на основі алгоритмів «косяку риб» та модифікованого «зграї вовків» представлено на рис. 4.1.

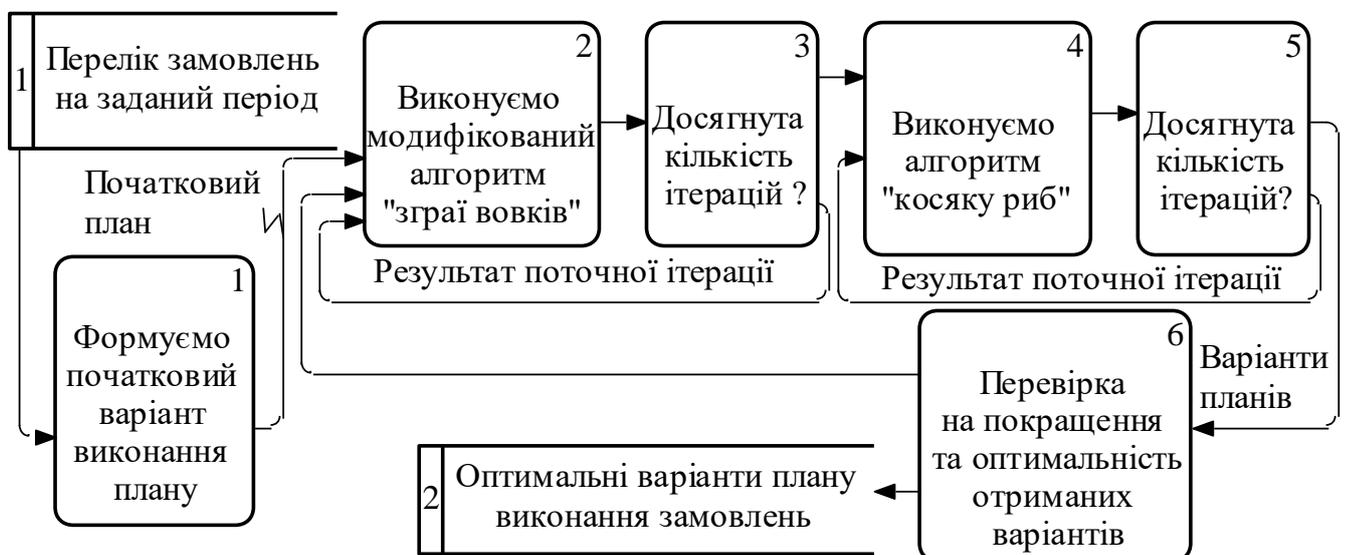


Рисунок 4.1 – Комбінований метод на основі алгоритмів «косяку риб» та модифікованого «зграї вовків»

Зупинка роботи алгоритму відбувається при вичерпанні заданої кількості ітерацій або коли оптимальне рішення повторюється протягом заданої кількості ітерацій.

За рахунок того, що загальна кількість ітерацій розподіляється між різними алгоритмами, відбувається загальне скорочення часу пошуку. Тобто, якщо для здійснення кожного алгоритму необхідно 100 операцій, то для усіх необхідно 200 операцій. На основі проведених експериментів було виявлено, що кількість ітерацій між алгоритмами доцільно розділити у пропорції 30/70. Для алгоритму «косяку риб» виділяється 30 ітерацій, а для алгоритму «зграї вовків» – 70.

Комбінований алгоритм працює за таким псевдокодом:

1) визначаються вхідні параметри для пошуку;

2) задаємо параметри роботи модифікованого алгоритму, а саме визначаємо пропорційність між кількістю ітерацій для кожного алгоритму. Наприклад, загальна кількість ітерацій 100, для «косяку риб» обрано 30 ітерацій, для «зграї вовків» – 70;

3) формуємо план  $P_S$  із використанням генератора випадкових чисел або задаємо власноруч;

4) загальний цикл:

- виконуємо алгоритм «зграї вовків» за заданими параметрами, використовуючи  $P_S$  «косяку риб» як вхідний;

- отримуємо наближений план  $P_{\text{«зграї вовків»}}$ ;

- виконуємо алгоритм «косяку риб» за заданими параметрами, використовуючи  $P_{\text{«зграї вовків»}}$  як вхідний;

- отримуємо наближений план  $P_{\text{«косяку риб»}}$ ;

- визначаємо найкращий варіант із  $P_{\text{«зграї вовків»}}$ ,  $P_{\text{«косяку риб»}}$  та присвоюємо  $P_i$ ;

5) перевірка виконання умов виконання загального циклу, головною умовою є досягнення оптимального рішення  $P_i$ , що визначається за рахунок змін не у бік покращення, на відміну від  $P_{i-1}$ ;

б) отримуємо пропозицію  $P_i$ .

Для апробації запропонованого комбінованого методу обирались дані за планами виконання замовлень за різні попередні періоди діяльності харчових підприємств. Апробація проводилась із різними наборами часткових критеріїв для перевірки впливу їхньої кількості на ефективність та швидкість знаходження оптимального рішення. В результаті проведених експериментів було виявлено, що на швидкість розв'язку впливають часові критерії, а також їх кількість. У табл. 4.1 наведені результати порівняння комбінованого методу з іншими алгоритмами при використанні усіх часткових критеріїв при 100 замовленнях.

Таблиця 4.1 – Таблиця порівняння комбінованого методу з іншими алгоритмами при використанні усіх часткових критеріїв при 100 замовленнях

№ п/п	Назва алгоритму/методу	Час пошуку плану, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
			у.о.	%	год	%
1	Алгоритм «бджолоїної колонії»	7,8	2156	4	3	2
2	Алгоритм «хаотичного кажана»	7,7	2364	4,39	3	2
3	Алгоритм «кажана на основі стратегії пошуку польоту Леві»	7,55	2345	4,35	3	2
4	Алгоритм «кажана на основі коефіцієнта скорочення»	7,6	2163	4,01	3	2
5	«Генетичний алгоритм»	7	2420	4,49	6	4
6	Алгоритм пошуку «косяку риб»	6,8	3523	6,99	6	4
7	Алгоритм «зграї вовків»	6,8	3523	6,99	6	4

Продовження таблиці 4.1

8	Модифікований алгоритм «зграї вовків»	6,51	3523	6,99	6	4
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	6,8	3523	6,99	6	4
10	Модифікований алгоритм «мурашиної колонії»	6,51	3523	6,99	6	4
11	Метод на основі комбінації модифікованого алгоритму «зграї вовків» та алгоритму пошуку «косяку риб»	6,18	3523	6,99	6	4

За наведеними результатами видно, що при застосуванні усіх часткових критеріїв модифікований метод забезпечує пошук на 5,2% швидше, ніж модифікований алгоритм «зграї вовків». При порівнянні роботи запропонованого методу з алгоритмом пошуку «косяку риб» видно, що швидкість пошуку збільшується на 9,9%.

У табл. 4.2 наведені результати порівняння алгоритму з іншими алгоритмами при використанні п'яти часткових критеріїв при 100 замовленнях.

Таблиця 4.2 – Таблиця порівняння комбінованого методу з іншими алгоритмами при використанні п'яти часткових критеріїв при 100 замовленнях

№ п/п	Назва алгоритму/методу	Час пошуку плану, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
			у.о.	%	год	%
1	Алгоритм «бджолоїної колонії»	6,24	1724,8	3	2,5	2

Продовження таблиці 4.2

2	Алгоритм «хаотичного кажана»	6,16	1891,2	3,29	2,5	2
3	Алгоритм «кажана на основі стратегії пошуку польоту Леві»	6,04	1876	3,26	2,5	2
4	Алгоритм «кажана на основі коефіцієнта скорочення»	6,08	1730,4	3,01	2,5	2
5	«Генетичний алгоритм»	5,60	1936	3,37	4,8	3,84
6	Алгоритм пошуку «косяку риб»	5,44	2818,4	4,90	5	4
7	Алгоритм «зграї вовків»	5,24	2818,4	4,90	5	4
8	Модифікований алгоритм «зграї вовків»	5,21	2818,4	4,90	5	4
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	5,44	2818,4	4,90	5	4
10	Модифікований алгоритм «мурашиної колонії»	5,26	2818,4	4,90	5	4
11	Метод на основі комбінації модифікованого алгоритму «зграї вовків» та алгоритму пошуку «косяку риб»	3,89	2818,4	4,90	5	4

За наведеними результатами видно, що при застосуванні усіх часткових критеріїв модифікований метод забезпечує пошук на 33,87% швидше, ніж модифікований алгоритм «зграї вовків». Якщо порівнювати роботу запропонованого методу з алгоритмом пошуку «мурашиної колонії», то видно, що швидкість пошуку збільшується на 35,15%. У порівнянні з алгоритмом пошуку «косяку риб» метод надає збільшення швидкості пошуку на 33,8%.

Як показали результати апробації, запропонований комбінований метод на основі «косяку риб» та «зграї вовків» дає зменшення часу пошуку до 40% при умові врахування максимум 5–6 часткових критеріїв, а в інших випадках не перебільшує швидкість пошуку у 5%.

Тому запропонований комбінований метод доцільно застосовувати для підприємств з невеликим асортиментом та з мінімальним парком технологічного обладнання, адже використання спрощеного варіанта моделі можливий у випадку використання автоматизованих технологічних комплексів, які забезпечують серійне виготовлення продукції.

#### 4.2 Комбінований метод прийняття рішень на основі «мурашиної колонії», «зграї вовків» та «генетичного алгоритму»

Пропонується новий комбінований метод прийняття рішень, заснований на алгоритмах «мурашиної колонії», «зграї вовків» та «генетичного алгоритму». Схема роботи методу представлена на рис. 4.2.

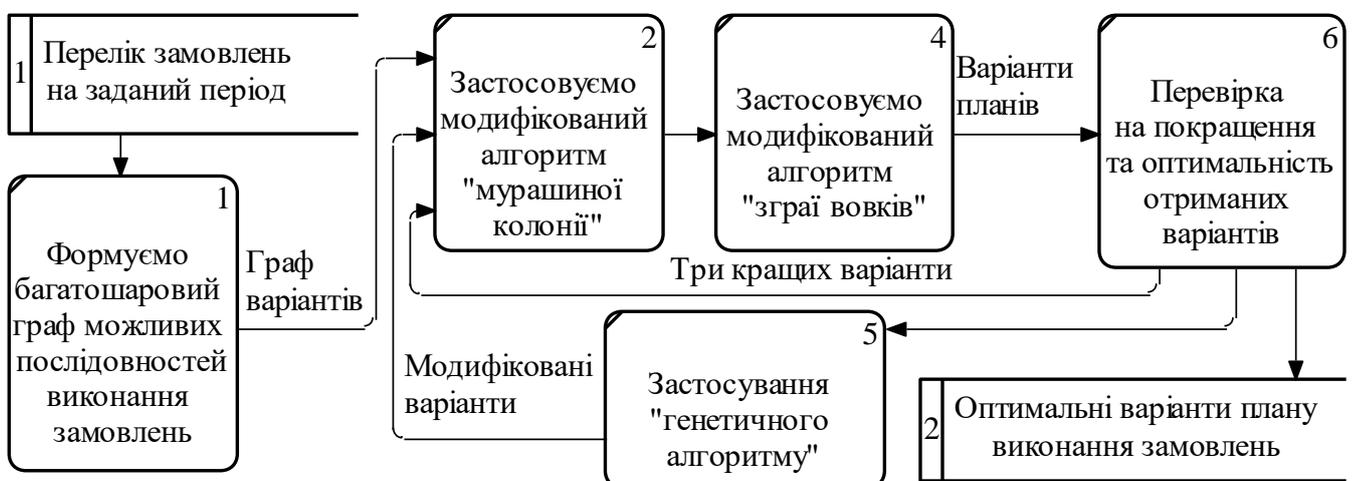


Рисунок 4.2 – Комбінований метод, заснований на алгоритмах «мурашиної колонії», «зграї вовків» та «генетичного алгоритму»

Метод базується на наведеній послідовності застосування алгоритмів, але особливістю є те, якщо ОПР прийме рішення про покращення плану, тоді застосовується «генетичний алгоритм» для модифікацій поточного плану. Такий підхід забезпечує формування альтернативних планів на основі результатів, отриманих у попередніх алгоритмах.

Комбінований метод працює за таким псевдокодом:

1) визначаються вхідні параметри для пошуку;

2) задаємо параметри роботи комбінованого алгоритму;

3) загальний цикл:

- виконуємо пошук за модифікованим алгоритмом «мурашиної колонії»;
- виконуємо пошук за модифікованим алгоритмом «зграї вовків»;
- якщо результат знайдено, то переходимо до останнього пункту;
- якщо результат не задовольняє, застосовуємо «генетичний алгоритм» та переходимо до пункту «а»;

4) перевірка виконання умов виконання загального циклу. Головною умовою є досягнення оптимального рішення  $P_i$ , що визначається за рахунок змін не у бік покращення, на відміну від  $P_{i-1}$ ;

5) отримуємо пропозицію  $P_i$ .

Спочатку особа, яка приймає рішення, обирає ряд критеріїв, які будуть враховуватись при розв'язанні та оцінці оптимальності плану виконання замовлень. Обрання критеріїв здійснюється згідно з соціально-економічною ситуацією, а також у залежності від стратегічних напрямів розвитку підприємства в цілому.

Далі будують граф варіантів. Вузли графа – це стан вибору та переходу від одного етапу до іншого. Ребра, які поєднують два вузли, відображають процес виконання певного етапу. Загальною тривалістю виконання певного етапу позначаємо вагу кожного ребра. Для усунення можливої затримки виконання певного етапу додається ще одне додаткове ребро між сусідніми вузлами. Це забезпечить перехід, якщо неможливо виконати жоден інший етап. Формування графа здійснюється для наочності та при застосуванні метаевристичних алгоритмів.

Один шар графа відповідає технологічному обладнанню. Кожен вузол відповідає певному моменту часу, а зупинка в ньому необов'язково вимагає необхідність переходу до наступного вузла іншим ребром. Це забезпечує узгодженість між виконаннями етапів на різному технологічному обладнанні [12, 22].

Для отримання перших наближених варіантів розкладу використовується модифікований алгоритм «мурашиної колонії». Умовою отримання варіанта плану виконання замовлень є відвідання усіх різних ребер, крім ребер, які відповідають затримці, кожного шару графа для формування одного повного шляху. На кожній ітерації згенеровані штучні мурахи здійснюють пошук шляху на графі згідно з обраною особою, яка приймає рішення часткових критеріїв або загальної згортки критеріїв. Для пошуку екстремальних значень цільової функції на кожній ітерації використовується визначена кількість штучних мурах, які будують відповідну кількість варіантів розкладу виконання замовлень. Використовуючи цільову функцію та часткові критерії математичної моделі складання плану виконання замовлень, здійснюється відбір найкращих. Відібрані варіанти зберігаються та використовується незалежно один від одного на наступних ітераціях. Кожен агент діє за правилами ймовірнісного алгоритму, при виборі напрямку орієнтується не тільки на прирощення цільової функції, а й на статистичну інформацію, що відображає попередню історію колективного пошуку. Якщо процес виготовлення готової продукції складається менше, ніж з трьох технологічних операцій на різному технологічному обладнанні, що може бути розташовано на різних виробничих ділянках, є можливість відразу перейти до оцінки отриманих варіантів.

Для уникнення отримання псевдоплану, що може бути тільки локально-оптимальним, застосовується алгоритм «зграї вовків», що моделює процес їх полювання в природі. На першій ітерації кожному вовку популяції присвоюється одне замовлення, без повторення, а потім для кожного вовка формується своя популяція з використанням генератора випадкових чисел, що виключає повторення номерів замовлень, якщо таке було вже згенеровано для цього агента. Як «альфа», «бета» та «дельта» на першій ітерації використовують найкращі варіанти, отримані при застосуванні алгоритму «мурашиної колонії».

Після застосування алгоритму отримуємо три найкращих варіанта плану виконання замовлень, що відповідають «альфа», «бета» та «дельта» агентам.

Якщо менеджер, який виступає в ролі особи, що приймає рішення, приймає та затверджує один із варіантів виконання замовлень, то воно приймається та передається до виробничих підрозділів. Якщо отримані варіанти виконання замовлень не задовольняють, тоді застосовують «генетичний алгоритм» для формування нових варіантів, що стануть початковими для алгоритму «мурашиної колонії».

Використання «генетичного алгоритму» обумовлено тим, що він використовує ітераційний підхід поліпшення результатів серед відібраних найкращих варіантів [22, 78]. Якщо таке рішення знайдено, воно стає поточним і починається нова ітерація. На кожній ітерації використання операцій мутації, кросовера та схрещування відбувається швидке отримання нових модифікованих варіантів планів виконання замовлень. Фактично застосування «генетичного алгоритму» забезпечує сформування нових альтернативних планів на основі трьох найкращих.

При повторному проході через усі алгоритми кількість ітерацій зменшується вдвічі для кожного алгоритму.

Це продовжується до тих пір, поки не буде виконана задана кількість ітерацій. Таким чином ми отримуємо нові варіанти, які подаються на етап із використанням алгоритму «мурашиної колонії».

Важливо зазначити, що результат роботи залежить не лише від кількості замовлень, а й від кількості етапів, які необхідно виконати для виготовлення продукції, а також від кількості варіантів технологічного обладнання.

Поєднання трьох алгоритмів в один модифікований метод забезпечує знаходження складних варіантів виконання розкладу на багатоміномієнатурних підприємствах харчової галузі, що підтверджено результатами випробування на ретроспективних даних за планами виконання замовлень за різні попередні періоди діяльності харчових підприємств.

У табл. 4.3 наведені результати порівняння комбінованого методу з іншими алгоритмами при використанні усіх часткових критеріїв при 100 замовленнях.

Таблиця 4.3 – Таблиця порівняння комбінованого методу з іншими алгоритмами при використанні усіх часткових критеріїв при 100 замовленнях

№ п/п	Назва алгоритму/методу	Час пошуку плану, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
			у.о.	%	год	%
1	Алгоритм «бджолої колонії»	7,80	2156,00	4,00	3,00	2,00
2	Алгоритм «хаотичного кажана»	7,70	2364,00	4,39	3,00	2,00
3	Алгоритм «кажана на основі стратегії пошуку польоту Леві»	7,55	2345,00	4,35	3,00	2,00
4	Алгоритм «кажана на основі коефіцієнта скорочення»	7,60	2163,00	4,01	3,00	2,00
5	«Генетичний алгоритм»	7,00	2420,00	4,49	6,00	4,00
6	Алгоритм пошуку «косяку риб»	6,80	3523,00	6,54	6,00	4,00
7	Алгоритм «зграї вовків»	6,80	3523,00	6,54	6,00	4,00
8	Модифікований алгоритм «зграї вовків»	6,51	3523,00	6,54	6,00	4,00
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	6,80	3523,00	6,54	6,00	4,00
10	Модифікований алгоритм «мурашиної колонії»	6,51	3523,00	6,54	6,00	4,00

Продовження таблиці 4.3

11	Комбінований метод на основі модифікованого алгоритму «зграї вовків» та алгоритму пошуку «косяку риб»	6,18	3523,00	6,54	6,00	4,00
12	Комбінований метод, заснований на алгоритмах «мурашиної колонії», «зграї вовків» та «генетичного алгоритму»	5,85	3587,00	6,65	6,00	4,00

За наведеними результатами видно, що при застосуванні усіх часткових критеріїв модифікований метод забезпечує пошук швидше, ніж усі інші алгоритми.

Крім цього, необхідно відмітити, не зважаючи на те, що при великих розмірностях задачі відбувається мінімальний виграш у часі, знайдені оптимальні варіанти більш ефективні у порівнянні з іншими. Якщо на початку виконання комбінованого алгоритму давати певні альтернативні плани, то швидкість пошуку збільшується.

#### **4.3 Розробка інформаційної технології для планування виконання замовлень**

Вирішення задачі оперативного планування виконання замовлень полягає у швидкому формуванні оптимального плану виконання замовлень. Необхідно відмітити, що сучасні вимоги ведення ділових відносин, особливо для підприємств харчової галузі, вимагають швидкого реагування на потреби замовника. Фактично замовник, який виступає як кінцевий продавець, є тим, хто керує ринком збуту, і

нехтувати його вимогами керівникам підприємства не бажано. Як правило, замовниками є фірми-посередники або представники великих торгових мереж типу «Ашан» та «Метро». У боротьбі за ринок такі фірми-посередники намагаються забезпечити ринок готовою продукцією за мінімальний час, а харчовому підприємству необхідно забезпечити швидке виконання замовлень для усіх їхніх замовників.

Для вирішення задачі оперативного планування виконання замовлень не достатньо мати алгоритми та методи пошуку оптимальних планів, адже їх реалізація можлива тільки у складі інформаційної технології.

У роботі запропоновано інформаційну технологію для розв'язання задачі планування виконання замовлень по виготовленню продукції на харчових підприємствах в умовах невизначеності та ризику. Інформаційна технологія забезпечує швидку реконфігурацію поточного плану виконання замовлень у разі виникнення позаштатних ситуацій або необхідності термінового виконання певного замовлення. Реконфігурація поточного плану виконання замовлень повинна проходити без порушень інтересів інших замовників, а також із мінімальними витратами для підприємства в цілому.

Для запропонованої інформаційної технології притаманні такі властивості:

- усі елементи чітко визначені та уніфіковані, що дозволяє точно її реалізовувати;
- повністю забезпечує формування/реконфігурацію плану виготовлення продукції;
- регулярність процесу й однозначність його фаз, що забезпечує можливість обліку, планування, диспетчеризації інформаційних процесів.

Основною особливістю даної технології є забезпечення вирішення задачі планування виконання замовлень різними підходами в діалоговому режимі, що гарантує високий рівень вирішення задачі. Діалоговий режим прийняття рішень передбачає, що прийняття рішення відбувається в результаті ітераційного процесу, в якому (рис. 4.3) як об'єкт управління виступає інформаційна технологія, що виконує обробку інформації та формує альтернативи рішень, а суб'єктом управління є

людина, яка задає вхідні дані, обирає оптимальний результат та приймає остаточне рішення.



Рисунок 4.3 – Ітераційний процес прийняття рішень із використанням інформаційної технології

Є суттєвою помилкою вважати, що задачі управління можуть повністю вирішуватися тільки використанням інформаційних технологій без участі людини, тому що навіть штучний інтелект потребує навчання та часу на його відлагодження, і тільки людина може відповідати за прийняте рішення.

При реалізації інформаційної технології дуже важливо організувати діалог з особою, яка приймає рішення, а також забезпечити надання усіх результатів у зручній наочній формі.

Запропонована інформаційна технологія поєднує в собі традиційні підходи і має такі риси [277, 278]:

- орієнтація на рішення комбінаторної задачі;
- обрання методу вирішення задачі;
- пошук оптимального рішення є ітераційним процесом;
- закінчення ітераційного процесу пошуку відбувається за рішенням людини;
- поєднання традиційних методів доступу і обробки даних із можливостями математичних моделей і методами вирішення завдань на їх основі;
- спрямованість на неспеціаліста з інформаційних технологій;
- висока адаптивність до особливостей наявного технічного і програмного забезпечення, а також вимог користувача.

Необхідно відмітити, що більшість підприємств харчової галузі використовують широкий спектр інформаційних систем, які дуже відрізняються одна від одної. Більшість інформаційних систем мають власні бази даних, що працюють під управлінням різних СУБД. Накопичена інформація за довгі роки може стати джерелом інформації для виявлення особливостей, пов'язаних з виробничим процесом на відповідному підприємстві. Виявлення таких особливостей дасть можливість формувати додаткові обмеження при формуванні плану виконання замовлень, адже для кожного підприємства існують особливості, притаманні тільки йому. Прикладом подібних особливостей можуть слугувати особливості експлуатації технологічного обладнання, які можуть призводити до виникнення некондиційної продукції або необхідності додаткового обслуговування [78].

Загальна структура використання інформаційної технології представлена на рис. 4.4.

Використання інформаційної технології базується на накопичених даних у базах даних (БД) підприємства та інших інформаційних джерелах (ІД). Для зменшення навантаження на мережу та отримання більш загальної картини на момент часу пропонується актуальну інформацію завантажувати у віртуальне сховище даних, яке вміщує тільки актуальну інформацію для певного проміжку часу. Проміжок часу визначає особа, яка приймає рішення. Обрання саме віртуальної моделі СД обумовлено тим, що кожен раз будуть завантажуватися дані за необхідний період, а також не буде потреби зберігати надлишок даних. Для кожної окремої задачі визначається «вітрина даних», яка вміщує необхідну інформацію. У разі необхідності часовий діапазон можливо змінити за рахунок зміни проміжку, за який буде отримуватися інформація з першоджерел. Завантаження здійснюється засобами завантаження даних, орієнтованих на СУБД, на базі якої буде побудовано віртуальне сховище даних. Після використання і за непотрібністю цієї інформації воно очищується. При завантаженні даних водночас усуваються їх надлишковість та повторення. Таке віртуальне сховище даних чимось

нагадує велику «вітрину даних», яка складається з окремих розділів для вирішення окремих вузькоспеціалізованих задач (рис. 4.5).

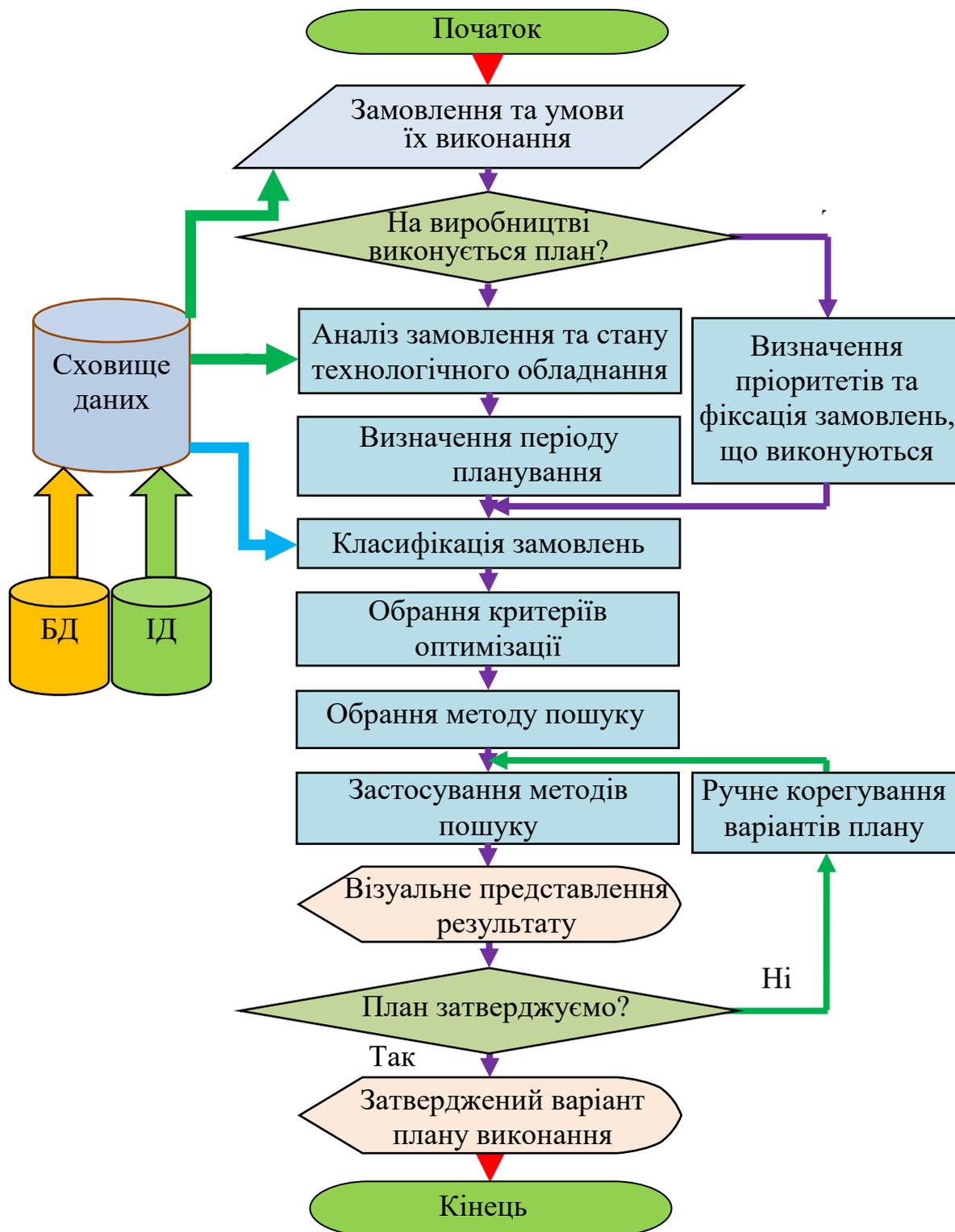


Рисунок 4.4 – Основна схема інформаційної технології планування виконання замовлення

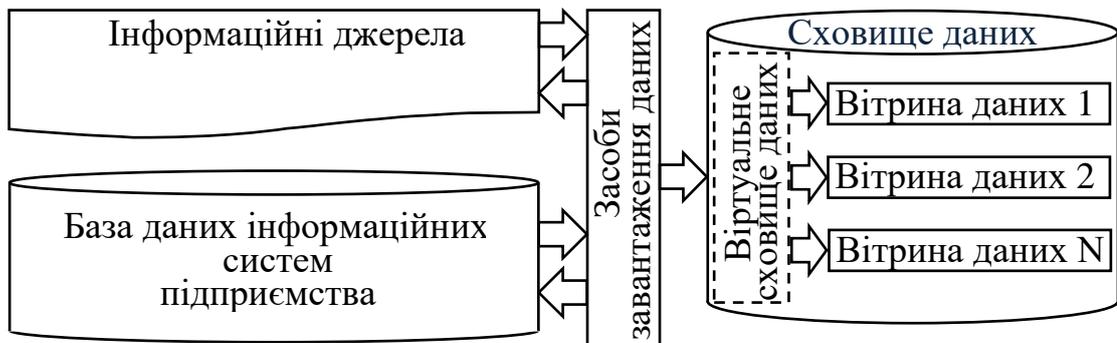


Рисунок 4.5 – Основна схема інформаційної технології планування виконання замовлення

Для інтеграції даних використовуються пакети завантаження даних, прикладом яких є засоби Integration Services (SSIS), що входять до СУБД MS SQL Server. Побудова кожного пакета включає в себе визначення операцій потоку управління, алгоритм оброблення потоку даних та налаштування його виконання. Потік управління визначає алгоритм виконання дій до оброблення потоку даних та після нього. Він забезпечує виконання системних команд, сценаріїв та операцій взаємодії з об'єктами й службами операційної системи, а також з елементами та сервісами обраної СУБД. Потік даних забезпечує алгоритм зчитування, перетворення та запис інформації. При створенні пакетів пов'язані операції потоку даних об'єднують у контейнери, а для забезпечення повторення певних операцій використовують цикли. Кожен алгоритм оброблення потоку даних визначає джерело та приймача інформації. Пакети завантаження даних забезпечують завантаження даних до усіх складових сховища даних [6, 33, 78].

У СД завантажується така інформація: про стан технологічного обладнання та його характеристики по експлуатації, наявна сировина, актуальний план надходження сировини на підприємство, умови актуальних договорів, усі необхідні деталі договорів (розміри штрафів при запізненні виконання, загальні умови) та замовлень (терміни, види продукції, обсяги, терміни), що надійшли на даний момент, особливості етапів технологічної послідовності виготовлення на існуючому обладнанні, техніко-економічні показники, повна інформації про стан виконання

поточного плану, детальна інформація замовлень, включених у поточний план, тощо.

Спочатку відбуваються отримання замовлення та аналіз умов його виконання. Якщо на даний момент не відбувається виконання ніяких планів, тоді здійснюється аналіз стану технологічного обладнання з метою з'ясування запланованих ремонтно-профілактичних робіт обладнання і можливостей їх відкладення чи перенесення. Чітко формуються планові дати початку та закінчення плану виконання замовлення. Все це виступає як обмеження при формуванні плану виконання замовлень. У разі необхідності та загальної домовленості є можливість скоригувати терміни проведення ремонтно-профілактичних робіт технологічного обладнання, якщо це не порушує паспортні вимоги щодо його експлуатації.

Сьогодні умови експлуатації виробничого обладнання багатьох підприємств харчової промисловості мають такі ознаки: значна частина обладнання вичерпала свій нормативний ресурс; через економічну ситуацію у країні у найближчі роки малоймовірна масова заміна зношеного обладнання; максимальне завантаження потужностей зумовлює підвищення ймовірності виходу обладнання з ладу; скорочуються обсяги планових ремонтів обладнання та збільшується тривалість міжремонтних періодів. Отже, характерною особливістю функціонування технологічного обладнання українських підприємств харчової промисловості є значне перевищення експлуатаційного терміну. Для такого обладнання звичайні підходи до оцінки ресурсів та планування ремонтних робіт є неефективними й потребують суттєвого доопрацювання. В цій ситуації питання продовження експлуатаційного терміну встановленого обладнання і забезпечення надійності його функціонування має першочергове значення. Вчасний аналіз його стану дозволить мінімізувати ризики виникнення аварійних ситуацій [6, 279].

Аналіз стану технологічного обладнання виконується спеціальними службами підприємства, але у разі виникнення термінових замовлень виникає необхідність додаткового аналізу його стану. Необхідно відмітити, що у процесі виробничого використання окремі частини технологічного обладнання зношуються і поступово втрачають свою здатність виконувати належні функції. Відновлення їх

працездатності та експлуатаційних властивостей досягається шляхом ремонту, планування якого повинно бути пов'язано з його раціональною організацією та доглядом за обладнанням. Основою для цього на промислових підприємствах є система технічного обслуговування та ремонту основних фондів [6].

Під системою технічного обслуговування і планових ремонтів (ТО і ПР) розуміють сукупність запланованих заходів щодо догляду, нагляду та ремонту обладнання. Система ТО і ПР включає технічне обслуговування, поточний ремонт (ПР) та капітальний ремонт (КР) [6].

Частіше на практиці, при експлуатації технологічного обладнання харчових підприємств, зустрічається система ремонтів за потребою. Це пов'язано з тим, що виробники продукції намагаються максимально використовувати свої потужності в умовах ринкової конкуренції. Особливо це характерно для малих та середніх виробництв, що можуть втратити свій сегмент ринку, якщо перестануть задовольняти потреби споживачів певного регіону. При цьому підприємства часто нехтують інструкціями з експлуатації, що надають заводи-виробники технологічного обладнання. Ремонт проводиться тільки тоді, коли цього вимагає його технічний стан, тобто подальша експлуатація обладнання стає неможливою через зношеність. Таке використання обладнання має ряд недоліків: відсутність продуманої системи планування ремонтів обладнання, відсутність профілактичних заходів, що попереджували б інтенсивний знос деталей, неминуче погіршення у процесі експлуатації стану обладнання та його технічної продуктивності, зменшення ступеня надійності та довговічності обладнання в результаті інтенсивного зношення деталей, невизначеність термінів зупинки обладнання, що не дає можливості планування ремонтних робіт [6, 279].

Неполадки і несправності було умовно розділено на чотири групи: технологічні відмови, відмови обладнання, відмови автоматизації, відмови енергогосподарства.

Успішне прогнозування стану обладнання є запорукою тривалого терміну його експлуатації, а також уникнення несподіваного виходу з ладу. Економічна діагностика є складовою комплексного аналізу діяльності підприємства і охоплює ту

частину аналітичної роботи, яка включає оцінку ефективності управлінських рішень, ситуаційний аналіз та оцінку техніко-економічних показників діяльності підприємства [6, 280].

Прогнозування з використанням моделей, побудованих на основі статистичних даних, є одним із популярних підходів до виявлення динаміки процесів у соціально-економічних, фінансових, технічних та інших системах. Значення показників стану обладнання змінюються в часі, тому пропонується будувати модель поведінки обладнання на основі аналізу ретроспективних даних із використанням часових рядів. Особливістю часових рядів, що відображають поведінку стану технологічного обладнання, є те, що їх характеристики (вібрація, шум, температура, тиск, кількість дрібних збоїв, обсяги виготовленої некондиційної продукції, показники індикаторів та контролерів тощо) формуються з декількох складових: тренд, сезонна складова і випадкова складова, які описують випадковий процес певного типу. Важливою особливістю сезонної складової, у свою чергу, є наявність періодичності зі змінним періодом і амплітудою [6].

Необхідно відмітити, що технологічне обладнання західних фірм обладнано власною системою діагностики, але такі системи вимагають постійне підключення до мережі Інтернет. Прикладом таких АСУТП можуть бути технологічні лінії Buhler, у яких забезпечені автоматичний контроль та підтримка параметрів технологічного процесу на високому рівні. Кожна складова такої лінії обладнана електронними датчиками й контролерами, що дозволяє здійснювати моніторинг усіх показників її вузлів (температура, кількість обертань двигунів, напруга та струму, вібрація, шум, тиск, кількість дрібних збоїв, обсяги виготовленої некондиційної продукції, показники індикаторів та контролерів тощо), а також вести збір даних про вихід із ладу кожного вузла. Інформація про кожний знятий параметр зберігається у відповідних файлах репозиторію АСУТП [6, 78].

Для аналізу стану обладнання доцільно використовувати засоби інтелектуального аналізу даних або запропоновану систему прогнозування в роботі [6], яка може швидко оцінити стан обладнання, проаналізувати зміни значень параметрів обладнання за обраний період, спрогнозувати зміну характеристик

обладнання. Необхідно відмітити, що рішення про відміну чи перенесення ремонтно-профілактичних робіт повинні приймати інженери та механіки, що відповідають за це, для уникнення позаштатних ситуацій, пов'язаних із відмовою вузлів, а також для забезпечення економії коштів на капітальних ремонтах.

Якщо на виробництво надійшло термінове замовлення або виникла позаштатна ситуація (перебої з сировиною, зупинка обладнання, анулювання певного замовлення тощо), то така ситуація вимагає реконфігурації плану виконання замовлень, і тоді менеджер визначає в існуючому плані замовлення, які мають високий пріоритет на виконання, тобто такі, що повинні бути виконані обов'язково. До таких замовлень можуть відноситись замовлення постійних клієнтів або такі, за невиконання яких необхідно сплачувати високі штрафи. Крім того, визначається група замовлень, що має певний запас виконання, тобто виконується з певним запасом часу та може бути відкладена. Таким чином, визначається можливість повної або часткової реконфігурації плану, що виконується на даний час. Звісно, ті замовлення, які вже почали виконувати, не враховуються у реконфігурації плану, адже переривати технологічний процес виготовлення, якщо не виникла позаштатна ситуація, не будуть. Однією із причин зупинення виконання виготовлення поточної продукції може бути випадок, коли замовлення скасовує сам замовник. Тоді відбувається перевірка наявності подібного наближеного замовлення за видом продукції і у разі виявлення такого замовлення, підбирається обсяг для нього.

Далі відбувається класифікація замовлень за групами з використанням методів інтелектуального аналізу даних. Для виробництв із невеликою номенклатурою пропонується алгоритм дерева рішень [22, 78], а для підприємств із великою номенклатурою пропонується комбінація методів інтелектуального аналізу даних [16, 78].

Вибір алгоритму дерева рішень обумовлений його можливістю на основі параметрів класифікації представляти об'єкти у вигляді ієрархії, в якій кожному елементу відповідає єдиний вузол прийняття рішення [22, 281, 282].

Перевагами алгоритму є швидкий процес навчання, генерація інтуїтивно зрозумілих правил, можливість опису непараметричних моделей, висока точність

прогнозу порівняно з іншими методами (статистика, нейронні мережі). Даний алгоритм ідеально підходить для опису непараметричних моделей [22, 281, 282].

Алгоритм дерева рішень забезпечує представлення результируючих даних у компактній формі з точним описом об'єктів за рахунок класифікації за певними ознаками [22].

Наприклад, для макаронного виробництва застосування алгоритму дерева рішень буде складатися з таких дій [22, 37, 78]:

- усі замовлення поділяються на коротко різані та довго різані макаронні вироби, адже для випуску кожного типу макаронів будуть використовуватися різні технологічні лінії, а замовлення будуть виконуватись незалежно;
- кожна група розбивається на підгрупи, які класифікуються за певними видами пакування;
- проводиться ранжування усіх замовлень за термінами, визначеними у замовленнях;
- за можливістю замовлення об'єднуються або розміщуються поряд;
- визначаються розміри штрафів при порушенні термінів виконання замовлень.

У випадку багатоміноменклатурного виробництва широкого профіля, наприклад, підприємство займається виготовленням різнопланової продукції, доцільно використовувати алгоритм нейронних мереж, що входить до методів та алгоритмів інтелектуального аналізу даних. Одним із гібридних методів нейронних мереж є Adaptive-Network-Based Fuzzy Inference System (ANFIS) – адаптивна мережа нечіткого висновку. ANFIS є одним із перших варіантів гібридних нейро-нечітких мереж – нейронної мережі прямого поширення сигналу особливого типу. Архітектура нейро-нечіткої мережі ізоморфна нечіткій базі знань. У нейро-нечітких мережах використовуються диференційовані реалізації трикутних норм (множення й імовірнісне АБО), а також гладкі функції належності. Це дозволяє застосовувати для налаштування нейро-нечітких мереж швидкі алгоритми навчання нейронних мереж, засновані на методі зворотного поширення помилки [19].

ANFIS реалізує систему нечіткого висновку у вигляді п'ятишарової нейронної мережі прямого поширення сигналу. Призначення шарів таке [19]:

- перший шар – терми вхідних змінних;
- другий шар – антецеденти (посилки) нечітких правил;
- третій шар – нормалізація ступенів виконання правил;
- четвертий шар – висновки правил;
- п'ятий шар – агрегування результату, отриманого за різними правилами.

Прикладом багатоасортиментного виробництва може слугувати фірма ТОВ «Продеко», яке займається такою діяльністю: виробництво хліба та хлібобулочних виробів; виробництво продуктів борошномельно-круп'яної промисловості; виробництво макаронних виробів і подібних борошняних виробів; виробництво тортів і тістечок, борошняних кондитерських виробів нетривалого зберігання; виготовлення та торгівля кормами для тварин; надання послуг перевезення, транспортування та використання вантажного автомобільного транспорту. Отже, якщо говорити про такий різноманітний асортимент виготовлення та діяльність, то необхідно відмітити, що напрями виробництва пов'язані між собою. Наприклад, замовлення виготовлення макаронних виробів із пшениці замовника. У випадку, якщо таких замовлень багато, виникає потреба аналізу етапів виготовлення між різними підрозділами підприємства. Тоді аналіз та розбиття замовлень відбуваються серед цехів, в яких є можливість виготовити продукцію.

Фактично здійснюється розбиття усіх замовлень у залежності від виду продукції та необхідності використання певних відділів чи обладнання для визначення кількості планів, які будуть виконуватися паралельно [22]. В подальшому відбувається ранжування замовлень у кожній групі. Таким чином здійснюється розбиття замовлень за обладнанням, яке буде використане для їх реалізації.

На наступному етапі особа, що приймає рішення, обирає ряд критеріїв, які будуть враховуватися при розв'язанні та оцінці оптимальності плану виконання замовлень. Особа, що приймає рішення, повинна визначити будь-який набір часткових критеріїв для врахування особливостей параметрів виконання замовлень.

Обрання критеріїв здійснюється згідно з соціально-економічною ситуацією, а також у залежності від стратегічних напрямів розвитку підприємства у такому переліку:

- максимізація прибутку від виконання усіх замовлень;
- мінімізація часу на виготовлення продукції;
- максимізація запасу часу при виконанні замовлень;
- мінімізація сумарних штрафів за невчасне виконання замовлення;
- мінімізація сумарного часу простою усіх одиниць технологічного обладнання та відділів;
- мінімізація сумарних витрат при виробництві;
- мінімізація сумарних витрат від простоїв при невикористанні обладнання, коли деякі технологічні контури, не зважаючи на невикористання, потребують електро- та теплоспоживання;
- мінімізація сумарних витрат на переробку та утилізацію отриманої некондиційної продукції при виконанні усіх замовлень;
- мінімізація сумарних витрат на зберігання готової продукції, а іноді і на транспортування;
- мінімізація сумарних витрат на зберігання сировини та матеріалів.

Усі критерії, а також обмеження та оціночна функція для визначення ефективності сформованого варіанта плану виконання замовлень описані у розд. 2.1. Особа, що приймає рішення, задає обмеження відповідно до конкретного підприємства.

Далі обирається метод пошуку з модулів, які реалізують різні алгоритми розв'язання задачі планування виконання замовлень. У залежності від обраного методу чи алгоритму задаються параметри його роботи.

Після закінчення розрахунків за обраним методом особа, що приймає рішення, отримує певну кількість найкращих альтернативних варіантів плану виконання замовлень для кожного окремого цеха або лінії. Для кожного альтернативного варіанта для наочності наводяться результати розрахунку за обраними критеріями. А для порівняння альтернативних варіантів виконання замовлень усі значення обраних критеріїв відображаються у вигляді таблиці та пелюсткової діаграми.

Приклад порівняння чотирьох критеріїв для сформованих п'яти варіантів альтернатив планів наведено у табл. 4.4.

Таблиця 4.4 – Порівняння альтернативних варіантів за чотирма критеріями

№	Частковий критерій	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5
1	Мінімізація сумарного часу простою	500	600	700	750	1500
2	Мінімізація сумарних штрафів за невиконання	600	550	406	756	550
3	Мінімізація витрат при використанні обладнання	4500	4200	4800	4500	5000
4	Мінімізація витрат на зберігання готової продукції	1500	1600	1800	1700	1600

Як видно з таблиці 4.4, швидко проаналізувати навіть незначну кількість варіантів виконання замовлень важко, тому доцільно це робити за допомогою пелюсткової діаграми або діаграми Ганта.

Приклад пелюсткової діаграми порівняння п'яти варіантів альтернативних планів наведено на рис. 4.6.



Рисунок 4.6 – Пелюсткова діаграма порівняння варіантів альтернатив варіантів за чотирма критеріями

Обрання пелюсткової діаграми обумовлено тим, що такий графічний спосіб відображення багатовимірних даних у вигляді двовимірної діаграми з необхідною кількістю змінних, представлених на осях зі спільним початком. Кожна вісь відповідає певному частковому критерію. Відносне положення та кут нахилу осей залежить від їх кількості та значень. Числові значення кожного критерію відображаються як довжини відрізків, які виходять із центру діаграми під різними кутами. Варіанти планів відображені різними кольорами. Такий тип діаграми дає можливість швидко зробити логічний висновок із великої кількості отриманих даних. Вибір такого виду діаграми обумовлений тим, що при оцінці сукупності критеріїв, які мають різні види та градацію шкал, на діаграмі зручно наочно представити та оцінити кожен варіант.

Враховуючи складність та обсяги даних, які необхідно аналізувати, відбувається заміна позначень назв критеріїв на діаграмі. Це відбувається у випадку, коли враховуються майже усі критерії. Приклад порівняння трьох варіантів із сьома критеріями наведено у табл. 4.5, а діаграма представлена на рис. 4.7.

Таблиця 4.5 – Порівняння альтернативних варіантів

№	Назва часткового критерію	Позн. крит.	Варіант 1	Варіант 2	Варіант 3
1	2	3	4	5	4
1	Мінімізація сумарного часу простою	F4	1500	0	500
2	Мінімізація сумарних штрафів за невиконання	F5	350	200	250
3	Мінімізація витрат при використанні обладнання	F6	4500	4200	4800
4	Мінімізація витрат на обладнання, що простоює	F7	1500	1000	1700
5	Мінімізація витрат на переробку та утилізацію отриманої некондиційної продукції	F8	5000	4500	4000
6	Мінімізація витрат на зберігання готової продукції	F9	1500	1600	1800
7	Мінімізація витрат на зберігання сировини та матеріалів	F10	5000	6000	3220

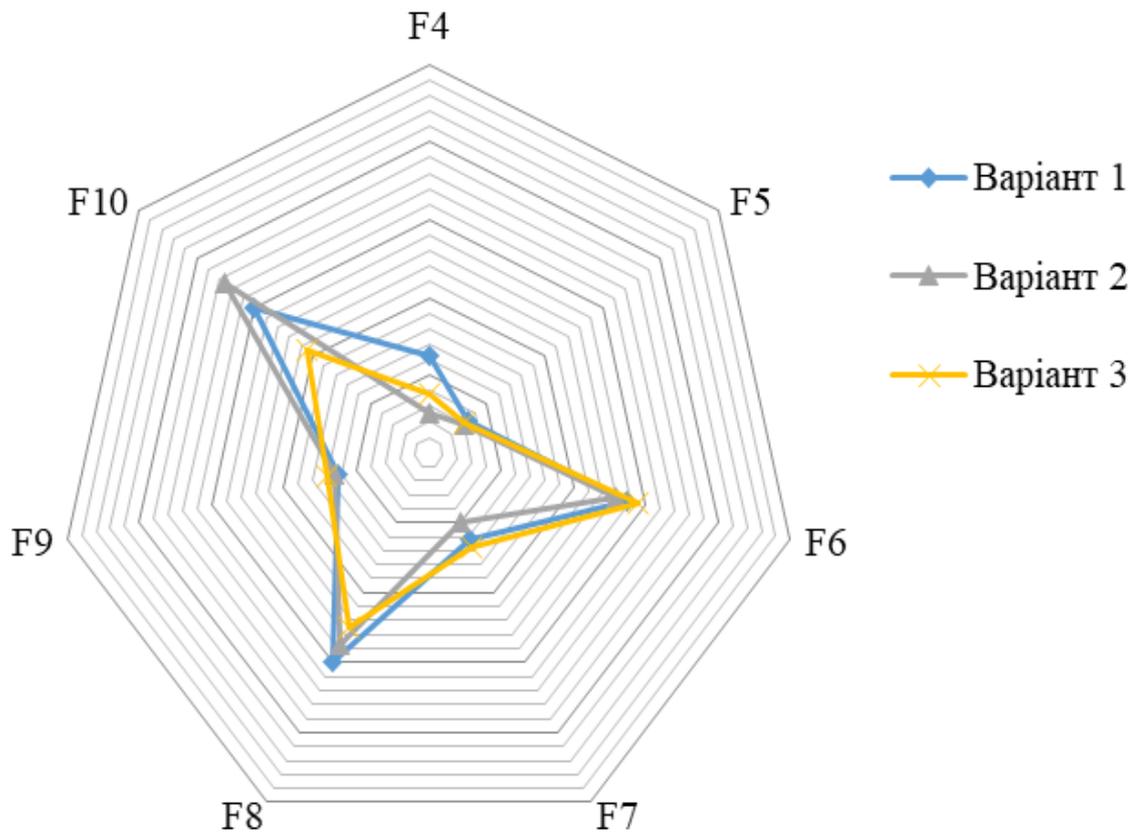


Рисунок 4.7 – Пелюсткова діаграма порівняння варіантів альтернатив варіантів за сімома критеріями

План виконання замовлень можливо переглянути в табличній формі та у вигляді діаграми Ганта, що дозволяє візуально оцінити послідовність виконання етапів. На рис. 4.8 відображено приклад сформованого плану щодо виконання трьох замовлень на ковбасні вироби.

Виробнича ділянка	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	0:00	1:00	2:00	3:00	4:00	
Кулінарний цех																							
Приготування фаршу	█		█	█			█	█															
Осадка багонів		█	█		█	█			█	█													
Варіння				█			█	█	█		█	█	█										
Копчення					█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Охолодження										█				█	█	█	█	█	█	█	█	█	█
Пакування											█						█					█	

Рисунок 4.8 – Приклад сформованої діаграми Ганта, яка відображає план виготовлення трьох видів ковбасних виробів

Якщо особа, що приймає рішення, затверджує один із варіантів планів виконання замовлень, то цей варіант приймається й передається до виробничих підрозділів, а прийнятий план записується у базу даних підприємства.

Якщо отримані варіанти виконання замовлень не задовольняють менеджера, тоді здійснюється фіксація певних частин у варіантах планів виконання замовлень або штучна заміна послідовності виконання замовлень. При необхідності додаються додаткові параметри роботи алгоритму, але як вхідні альтернативи обираються найкращі альтернативи варіантів виконання замовлень.

Важливо зазначити, що результат роботи залежить не лише від кількості договорів, а й від кількості етапів, які необхідно виконати для виготовлення продукції, а також від кількості варіантів технологічного обладнання.

На кожному підприємстві швидкість формування виконання замовлень буде різною, але необхідно намагатися максимально знизити цей час, адже при прийомі замовлення необхідно за лічені хвилини відповісти, на яку дату буде виконано те чи інше замовлення. Так само у випадку виникнення позаштатних ситуацій скоригувати існуючий план, щоб мінімізувати збитки.

За рахунок того, що перед початком застосування алгоритмів пошуку використовуються аналіз замовлень та можливість їх виконання, відшукується подібний, виконаний раніше план. На початку застосування алгоритмів не потрібно формувати наближений план, використовуючи генератор випадкових чисел.

Необхідно відмітити, що усі алгоритми та методи, розглянуті в розд. 3, та модифікований алгоритм на основі «косяку риб» і «зграї вовків» можуть бути застосовані в запропонованій технології.

При застосуванні комбінованого методу на основі алгоритмів «мурашиної колонії», «зграї вовків» та «генетичного алгоритму» схема 4.4 зазнає модифікацій і зміниться на схему інформаційної технології планування виконання замовлення, яка наведена на рис. 4.9.

При застосуванні комбінованого методу на основі алгоритмів «мурашиної колонії», «зграї вовків» та «генетичного алгоритму» спочатку для кожного плану



Для отримання перших наближених варіантів розкладу використовується модифікований алгоритм «мурашиної колонії». Після здійснення заданої кількості ітерацій обираємо найкращі варіанти сформованого плану на основі обраних часткових критеріїв математичної моделі. Обрані найкращі варіанти зберігаються та використовуються незалежно один від одного на наступних ітераціях. Необхідно відзначити, що у випадку, коли процес виготовлення має менше трьох технологічних операцій на різному технологічному обладнанні, то у цьому випадку відразу переходимо до оцінки отриманих варіантів.

З усіх отриманих варіантів обираються три найкращі, які називаються «альфа», «бета» та «дельта» агентами для застосування модифікованого алгоритму «зграї вовків». Відразу формуються «омега»-агенти на основі трьох найкращих. Далі застосовуємо модифікований алгоритм «зграї вовків» та в кінці отримуємо не менше трьох альтернативних варіантів.

Особа, що приймає рішення, аналізує отримані варіанти рішення, використовуючи пелюсткову діаграму, діаграму Ганта та детальний опис варіантів. Якщо вона приймає та затверджує один із варіантів виконання замовлень, то цей варіант приймається й передається до виробничих підрозділів, а прийнятий план записується у базу даних підприємства.

Якщо отримані варіанти виконання замовлень не задовольняють менеджера, тоді здійснюються частково або повністю такі дії:

- фіксація певних частин варіантів плану виконання замовлень або штучна заміна послідовності виконання замовлень;
- для формування нових варіантів застосовується «генетичний алгоритм».

Використання «генетичного алгоритму» забезпечує отримання нових модифікованих варіантів планів виконання замовлень за рахунок відповідних операцій мутації, кросовера та схрещування.

Він використовується до тих пір, поки не буде виконана задана кількість ітерацій, або отримані альтернативи не будуть кращими за вхідні.

Отримані варіанти стають вхідними варіантами для алгоритму «мурашиної колонії». Застосування комбінованого методу на основі алгоритмів «мурашиної

колонії», «зграї вовків» та «генетичного алгоритму» в інформаційній технології забезпечує знаходження складних варіантів виконання розкладу на багатоменклатурних підприємствах харчової галузі.

Необхідно відзначити, що ефективність застосування комбінованого методу полягає у тому, що як вхідні дані при виконанні кожного алгоритму подається не хаотичний набір, а оптимізована певним чином послідовність виконання замовлень. Такий підхід забезпечує високу швидкість пошуку глобального оптимуму, що є головною метою задачі.

Проведення апробації результатів впровадження запропонованої інформаційної технології проходило на реальних ретроспективних даних за попередні роки.

За вимогою представників підприємств, на яких проводилась апробація результатів дисертаційного дослідження, дані, що стосуються техніко-економічних показників підприємства, а саме потужності, обсяги виготовлення, вартісні та кількісні показники, не наводяться.

Але є можливість відобразити кількість замовлень, на яких проводилися дослідження. Адже всі ці замовлення мають показники виду продукції, її обсягу, термін виготовлення.

Для апробації обирались дані за планами виконання замовлень за різні попередні періоди діяльності харчових підприємств, що були обрані при апробаціях комбінованих алгоритмів для порівняння ефективності застосування запропонованої інформаційної технології.

Необхідно відмітити, що при застосуванні інформаційної технології були знайдені такі самі плани, як і без використання інформаційної технології з однаковою ефективністю, тому стовпчики «Ефективність знайденого плану» та «Скорочення часу виконання замовлень» наведено в табл. 4.6 у однині.

У табл. 4.6 представлено порівняння застосування різних методів та алгоритмів із використанням інформаційної технології та без неї для 100 замовлень.

Таблиця 4.6 – Порівняння застосування різних методів та алгоритмів із використанням інформаційної технології та без неї для 100 замовлень

№ п/п	Назва алгоритму/методу	Час пошуку плану без викори- стання ІТ, хв	Час пошуку плану без викори- стання ІТ, хв	Ефективність знайденого плану		Скорочення часу виконання замовлень	
				у.о.	%	год	%
1	2	3	4	5	6	7	8
1	Алгоритм «бджолоїної колонії»	7,80	6,24	2156,00	4,00	3,00	2,00
2	Алгоритм «хаотичного кажана»	7,70	6,16	2364,00	4,39	3,00	2,00
3	Алгоритм «кажана на основі стратегії пошуку польоту Леві»	7,55	6,04	2345,00	4,35	3,00	2,00
4	Алгоритм «кажана на основі коефіцієнта скорочення»	7,60	6,08	2163,00	4,01	3,00	2,00
5	«Генетичний алгоритм»	7,00	5,60	2420,00	4,49	6,00	4,00
6	Алгоритм пошуку «косяку риб»	6,80	5,44	3523,00	6,54	6,00	4,00

Продовження таблиці 4.6

7	Алгоритм «зграї вовків»	6,80	5,44	3523,00	6,54	6,00	4,00
8	Модифікований алгоритм «зграї вовків»	6,51	3,91	3523,00	6,54	6,00	4,00
9	Алгоритм «мурашиної колонії з використанням елітних мурах»	6,80	5,44	3523,00	6,54	6,00	4,00
10	Модифікований алгоритм «мурашиної колонії»	6,51	3,91	3523,00	6,54	6,00	4,00
11	Комбінований метод на основі комбінації модифікованого алгоритму «зграї вовків» та алгоритму пошуку «косяку риб»	6,18	3,71	3523,00	6,54	6,00	4,00
12	Комбінований метод, заснований на алгоритмах «мурашиної колонії», «зграї вовків» та «генетичного алгоритму»	5,85	3,51	3587,00	6,65	6,00	4,00

З наведених результатів у табл. 4.6 видно, що використання методів та алгоритмів у складі інформаційної технології забезпечує прискорення пошуку оптимального рішення. Це підтвержено діаграмою (рис. 4.10), де на вісі ОХ позначено номер методу з табл. 4.6.

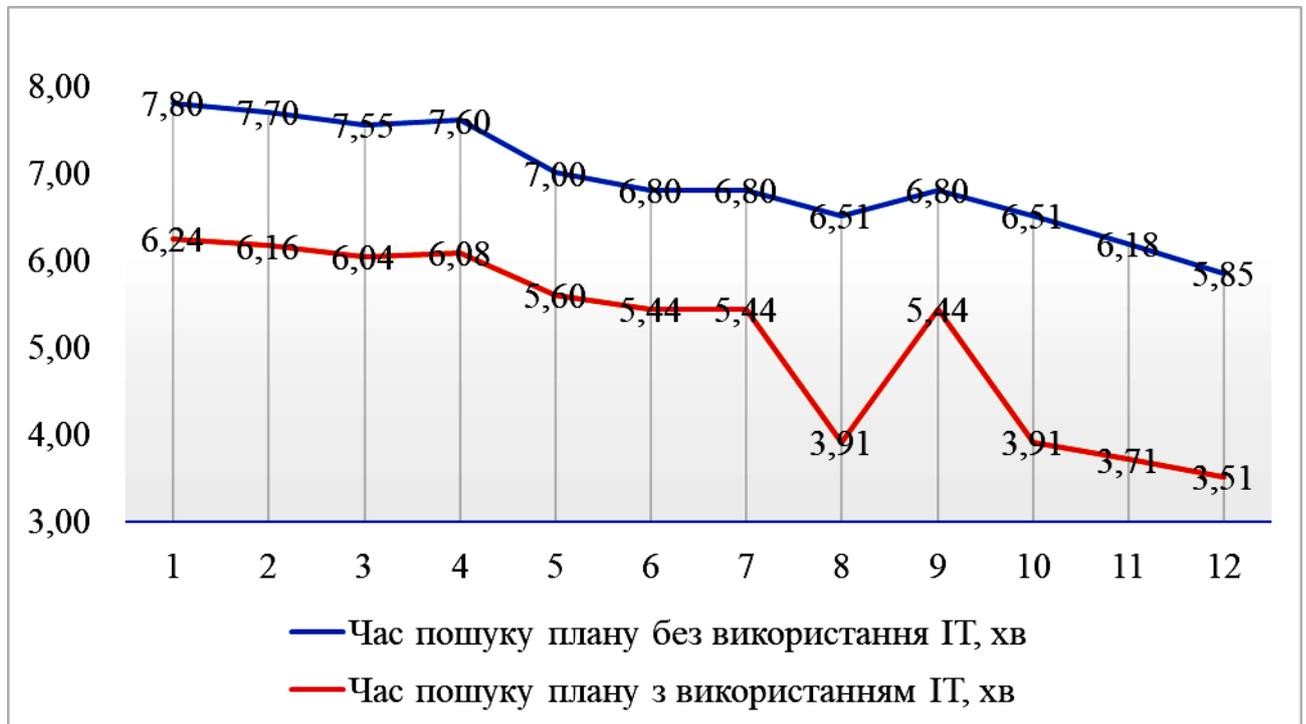


Рисунок 4.10 – Порівняння часу пошуку плану при застосуванні різних методів та алгоритмів із використанням інформаційної технології та без неї для 100 замовлень

При подальших експериментах з алгоритмами були виключені з розгляду такі алгоритми: «генетичний», «бджолоїної колонії», «кажанів на основі Леві», «кажанів», «мурашиної колонії». Це обумовлено тим, що надалі будемо експериментувати з даними великої розмірності з використанням усіх часткових критеріїв, а виключені з експериментів алгоритми мають менший час пошуку у порівнянні з іншими. Для чистоти експерименту було використано інший набір даних із молокозаводу та обрано 300 замовлень за заданий період, а результат порівняння часу виконання представлено у табл. 4.7, де відображено порівняльні

характеристики застосування різних методів та алгоритмів із використанням інформаційної технології та без неї.

Таблиця 4.7 – Порівняння застосування різних методів та алгоритмів із використанням інформаційної технології та без неї для 300 замовлень

№ п/п	Назва алгоритму/методу	Час пошуку плану без використання ІТ, хв	Час пошуку плану з використанням ІТ, хв	Ефективність застосування ІТ, %
1	Алгоритм пошуку «косяку риб»	10,20	7,24	40,85
2	Алгоритм «зграї вовків»	10,20	7,04	44,93
3	Модифікований алгоритм «зграї вовків»	9,77	7,03	38,89
4	Алгоритм «мурашиної колонії з використанням елітних мурах»	10,20	7,24	40,90
5	Модифікований алгоритм «мурашиної колонії»	9,77	5,83	67,38
6	Комбінований метод на основі комбінації модифікованого алгоритму «зграї вовків» та алгоритму пошуку «косяку риб»	9,28	5,26	76,50
7	Комбінований метод, заснований на алгоритмах «мурашиної колонії», «зграї вовків» та «генетичного алгоритму»	7,89	4,73	66,67

З наведених результатів у табл. 4.7 видно, що використання методів та алгоритмів у складі інформаційної технології забезпечує прискорення пошуку оптимального рішення, що підтверджує тенденцію, наведену в табл. 4.6. У колонці «Ефективність застосування» наведено у відсотках, наскільки ефективним є застосування методів та алгоритмів у складі інформаційної технології. З діаграми (рис. 4.11) чітко видно, що використання методів та алгоритмів у складі інформаційної технології дає суттєве зменшення часу пошуку за рахунок того, що аналіз попередніх замовлень дає можливість скоротити час виконання алгоритму, адже на початку роботи отримуємо інформацію, що формується не випадково.

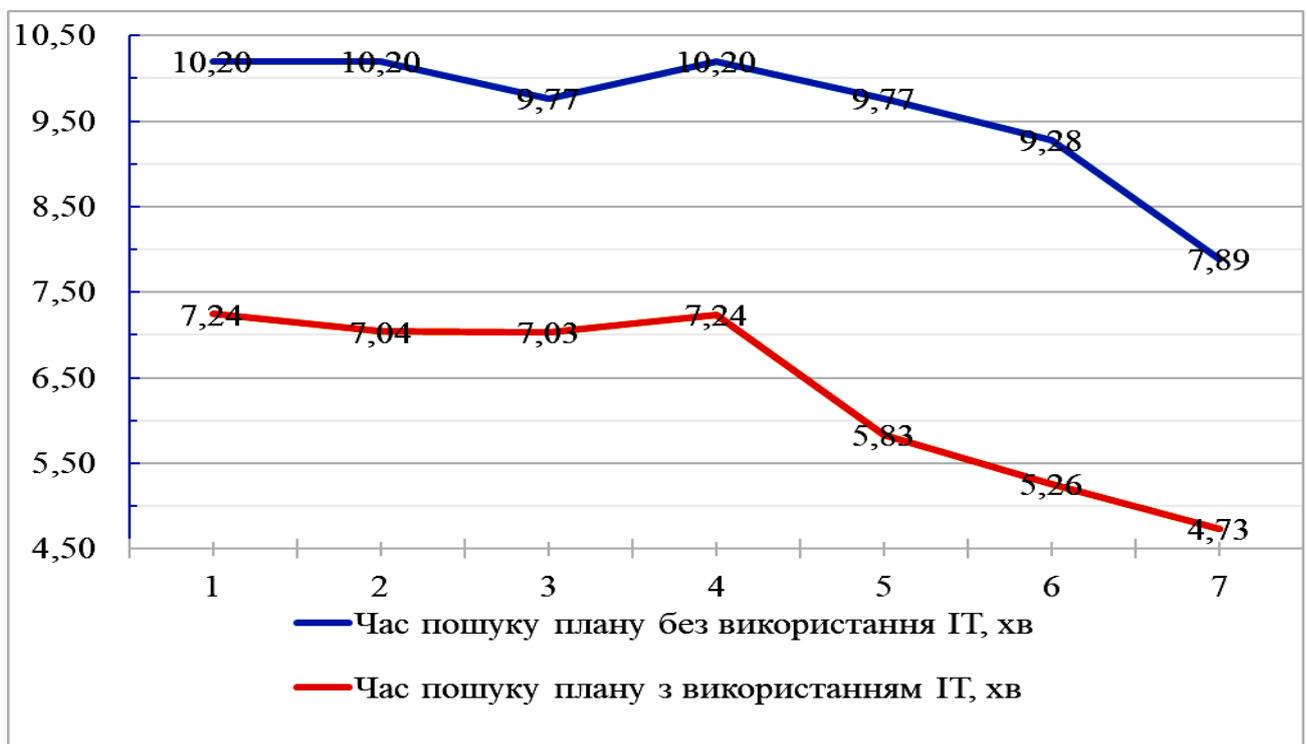


Рисунок 4.11 – Порівняння часу пошуку плану при застосуванні різних методів та алгоритмів із використанням інформаційної технології та без неї для 300 замовлень

Аналізуючи графік, представлений на рис. 4.10, видно, що, застосовуючи модифіковані методи та алгоритми у складі інформаційної технології, можливо скоротити час пошуку оптимального рішення на 3 хв. Це дає можливість швидше

відреагувати на виникнення позаштатних ситуацій, які можуть вимагати реконфігурації плану виконання замовлень.

Максимальне скорочення часу на прийняття рішення про певне коригування приводить до зменшення обсягів браку та некондиційної продукції, а, з урахуванням того, що в харчовій промисловості до готової продукції висуваються вимоги щодо відповідності міжнародним стандартам, задача про максимальне скорочення часу на прийняття рішення стає досить актуальною.

Як приклад можна привести автоматизовані лінії для виготовлення макаронних виробів, адже є лінії, що виготовляють до 100 кг за 1 годину, а є ті, що виготовляють 10 т за годину. Це наведена потужність одного вузла, а саме преса для виготовлення макаронних виробів. Тому затримка у 15 хв може призвести до не виготовлення близько 2 т продукції. В найкращому варіанті буде виготовлена некондиційна продукція. В цьому випадку некондиційна макаронна продукція перемелюється у борошно та додається при виготовлені нової продукції, що дозволено технологією.

Ковбасні вироби певних видів також можуть йти на повторну переробку, адже є види продукції, до складу яких дозволяється включати некондиційні вироби на технологічному рівні. Все це чітко регламентується ДСТУ та ISO, тому при порушеннях підприємство буде мати додаткові збитки.

Але частіше застосовується утилізація некондиційної продукції на підприємстві, що призводить до колосальних збитків. Невчасно виконане замовлення може призвести до незадоволення клієнта, який може відмовитися від подальшої праці з підприємством, що теж позначиться на його доходах.

Необхідно відмітити, що застосування модифікованих методів та алгоритмів у складі інформаційної технології дає можливість знайти більш ефективні рішення, що підтверджуються наведеними результатами порівняння у табл. 4.8.

Таблиця 4.8 – Порівняння ефективності знайденого плану при застосуванні різних методів та алгоритмів із використанням інформаційної технології та без неї для 300 замовлень

№ п/п	Назва алгоритму/методу	Ефективність знайденого плану без використання ІТ		Ефективність знайденого плану з використанням ІТ	
		у.о.	%	у.о.	%
1	Алгоритм пошуку «косяку риб»	11097,45	7,00	11097,45	6,54
2	Алгоритм «зграї вовків»	11097,45	7,00	11097,45	7,00
3	Модифікований алгоритм «зграї вовків»	11527,45	7,27	13832,94	8,73
4	Алгоритм «мурашиної колонії з використанням елітних мурах»	11097,45	7,00	11097,45	7,00
5	Модифікований алгоритм «мурашиної колонії»	11097,45	7,00	14426,69	9,10
6	Комбінований метод на основі комбінації модифікованого алгоритму «зграї вовків» та алгоритму пошуку «косяку риб»	11603,45	7,32	15084,49	9,51
7	Комбінований метод, заснований на алгоритмах «мурашиної колонії», «зграї вовків» та «генетичного алгоритму»	12053,05	7,60	15668,97	9,88

За результатами порівняння, наведених у таблиці ефективностей кожного алгоритму у складі інформаційної технології та без неї, видно, що при застосуванні модифікованих алгоритмів та комбінованих методів є можливість знаходити

альтернативні варіанти, які є більш ефективними. Наприклад, у порівнянні із застосуванням алгоритмів із інформаційною технологією швидкість пошуку може бути зменшена до 70 %.

За результатами порівнянь запропонована інформаційна технологія значно виграє за швидкістю пошуку, адже великий обсяг вхідних даних збільшує час пошуку оптимального плану виконання замовлень, якщо не класифікувати та попередньо не визначати наближену ефективну послідовність виконання.

Необхідно відзначити, що швидкість пошуку оптимального плану виконання замовлень у більшості випадків залежить від підприємства, на якому застосовували дану інформаційну технологію. Якщо у технологічному процесі виготовлення продукції передбачено проходження її через різні технологічні етапи, які потребують різне технологічне обладнання, то це буде збільшувати час пошуку оптимального плану виконання замовлень. Чим простіший технологічний процес виготовлення продукції або для виготовлення використовуються автоматизовані технологічні лінії, які забезпечують потокове виготовлення, тим простіше розробляти план для таких випадків.

Необхідно відмітити, що час пошуку оптимального плану залежить від характеристик комп'ютерної техніки. Більшість випробувань проводилась на Intel Core i5-4690K (3.5ГГц) / RAM 16 ГБ / HDD 1 ТБ.

Від стану автоматизації та використання інформаційних систем залежить можливість із мінімальними витратами реалізувати та впровадити інформаційну технологію на підприємстві. Вона без зайвих витрат може інтегруватися з будь-якою існуючою інформаційною системою.

#### **4.4. Висновки до розділу 4**

Розроблений комбінований метод прийняття рішень на основі алгоритмів «косяку риб» та модифікованого «зграї вовків» стабільно працює на задачах малої та

середньої розмірності, а також при 5–6 часткових критеріях. Комбінований метод на основі алгоритмів «косяку риб» та модифікованого «зграї вовків» дає зменшення часу пошуку майже до 35% при умові врахування не більше 5–6 часткових критеріїв, а в інших випадках не перебільшує швидкість пошуку у 5%. При застосуванні усіх часткових критеріїв модифікований метод забезпечує пошук на 33,87% швидше, ніж модифікований алгоритм «зграї вовків», у порівнянні з алгоритмом «мурашиної колонії» пошук збільшується на 35,15%. Якщо порівнювати роботу запропонованого методу з алгоритмом пошуку «косяку риб», то видно, що швидкість пошуку збільшується на 33,8%. Тому запропонований комбінований алгоритм доцільно застосовувати для підприємств із невеликим асортиментом та мінімальним парком технологічного обладнання, адже використання спрощеного варіанта моделі можливе у випадках використання автоматизованих технологічних комплексів, які забезпечують серійне виготовлення продукції.

Запропоновано комбінований метод на основі «мурашиної колонії», «зграї вовків» та «генетичного алгоритму», який стабільно працює на задачах різної розмірності, а також при використанні усіх критеріїв моделі. За швидкістю пошуку він на 10–45% швидший, ніж інші алгоритми. За інформацією, отриманою після проведення експериментів, є можливість стверджувати, що при застосуванні усіх часткових критеріїв модифікований метод забезпечує пошук швидше, ніж усі інші алгоритми. Крім цього, необхідно відмітити, що, не зважаючи на великі розмірності задачі, застосування методу дає мінімальний виграш у часі, але існує можливість знаходження альтернативного оптимального варіанта, що буде більш ефективнішим у порівнянні з іншими. Якщо на початку виконання комбінованого методу задавати певні альтернативні плани, то швидкість пошуку буде збільшуватися.

Розроблена інформаційна технологія планування виконання замовлень забезпечує створення та коригування оперативно-календарних планів за короткий період часу за рахунок поєднання інтелектуального аналізу даних для попередньої класифікації, розбиття замовлень, а також використання модифікованих алгоритмів та методів, які базуються на комбінуванні алгоритмів. За рахунок можливості

застосування різних алгоритмів та методів у складі запропонованої технології є можливість підібрати набір алгоритмів та методів, які більш оптимальні для певної предметної області. Використання інтелектуального аналізу даних забезпечує збільшення швидкості знаходження рішень за рахунок аналізу даних за попередні періоди, що дає змогу виявити найкращі послідовності використання конкретного технологічного обладнання для виготовлення продукції згідно з актуальними замовленнями.

За результатами апробації є можливість стверджувати, що при застосуванні алгоритмів без технології та з нею швидкість пошуку дуже відрізняється, адже з її використанням швидкість може бути збільшена до 70 %. Використання запропонованої технології дає змогу знайти додаткові альтернативні оптимальні плани виконання замовлень. За рахунок представлення результатів порівняння оптимальних планів у вигляді пелюсткової діаграми людина, що використовує інформаційну технологію, може швидко прийняти рішення в залежності від загальних економіко-соціальних впливів.

Основні результати розділу опубліковані автором у працях [5, 6, 7, 15, 16, 19, 21, 22, 27, 28, 31, 32, 33, 37, 44, 50].

## РОЗДІЛ 5

### ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

#### 5.1 Проєктування архітектури гібридної web-орієнтованої системи підтримки прийняття рішень

При розробленні web-орієнтованої СППР використано практичну реалізацію клієнт-серверної архітектури у вигляді «тонкого» клієнта, де вся логіка зосереджена на серверній частині, а користувальницька частина лише відповідає за відображення даних, тобто результат виконання відображається у браузері користувача. Вибір такої архітектури для побудови системи зумовлено такими перевагами: спрощена інтеграція зі сторонніми програмними продуктами; простота впровадження, що полягає у відсутності потреби у налаштуванні апаратних чи програмних засобів користувачів; підвищена відмовостійкість, оскільки усе необхідне для роботи програмне забезпечення встановлюється лише на сервері. Запропонована структура дозволяє інтегруватися з існуючими на підприємстві інформаційними системами.

Розробка архітектури СППР базується на роботах [5, 15, 16, 18, 23, 26, 35, 36, 39, 40, 52, 53]. До основних компонентів системи належать (рис. 5.1): «озеро даних», засоби завантаження даних, сховище даних, бази знань, серверна частина, web-інтерфейс користувача. Важливо зазначити, що інтерфейс користувача виконано у вигляді web-сторінок для відображення у браузері користувача.

Джерелами даних для СД слугують бази даних інформаційних систем та «озеро даних», що забезпечують адекватність та достовірність інформаційного забезпечення для підтримки прийняття управлінських рішень [283–292]. Накопичення інформації у «озеро даних» та у СД відбувається в автоматичному режимі за рахунок використання засобів завантаження даних. Запропоновано обрати багаторівневу архітектуру СД, що забезпечує інтеграцію та узгодження даних із «озера даних» та інших джерел спочатку у «віртуальне сховище даних», а вже звідти завантажуються у «вітрину даних».

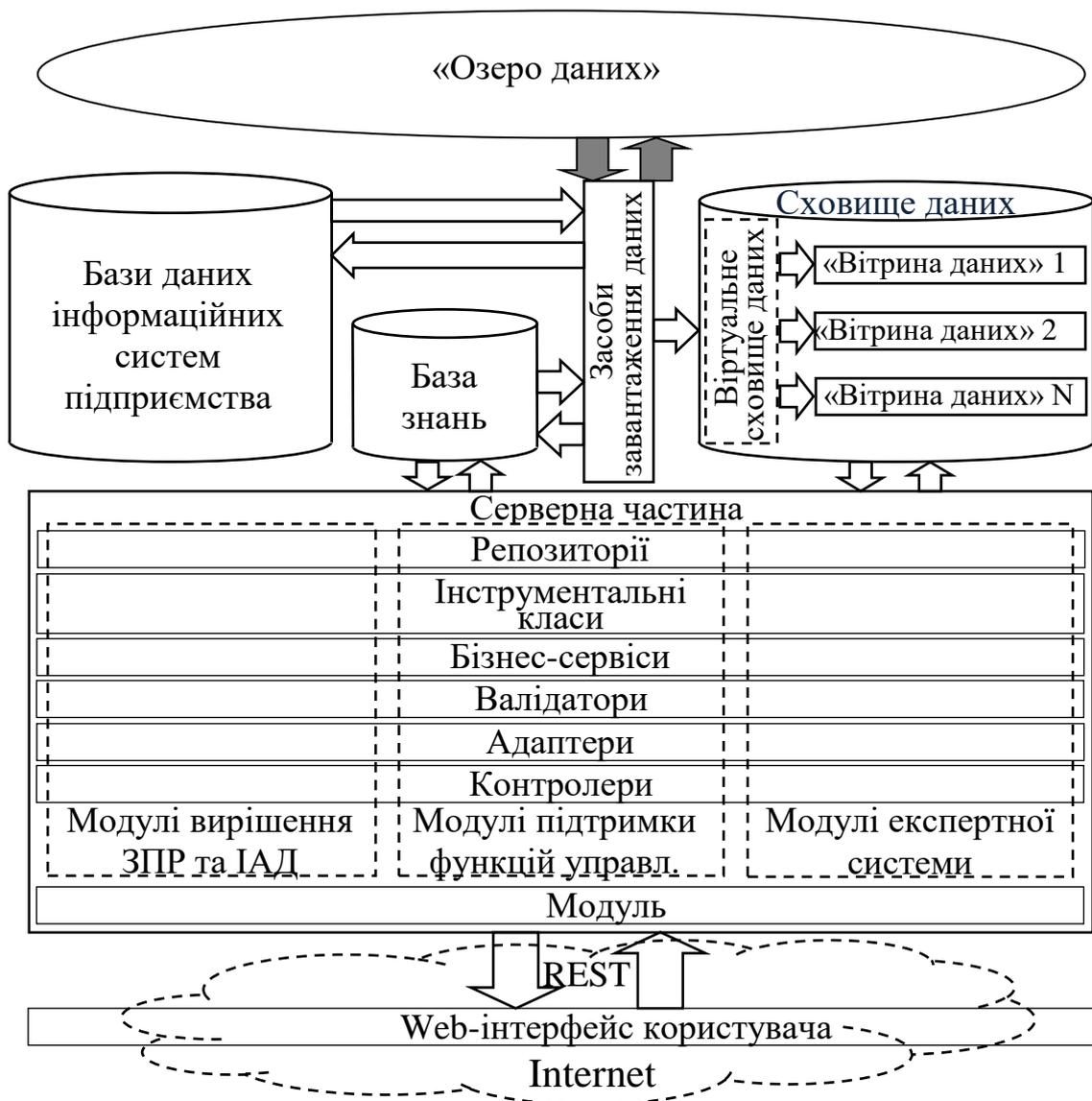


Рисунок 5.1 – Архітектури гібридної web-орієнтованої СППР

«Віртуальне сховище даних» побудовано за правилами реляційних БД. Це забезпечує уникнення надлишковості при роботі з великими об'ємами інформації [283, 296-298]. Також «віртуальне сховище даних» забезпечує проміжне деталізоване зберігання набору даних, отриманого з «озера даних» та інших джерел, для завантаження «вітрин даних».

Кожна «вітрина даних» призначена вміщувати набір даних для підтримки розв'язання певної задачі або надання інформації за певний проміжок часу [296-298].

Дані, що зберігаються у вітринах даних, оновлюються у відповідності до бізнес-логіки, мають певну агрегацію та деталізацію. «Вітрина даних» призначена

для вирішення конкретної локальної задачі, має цільове призначення та містить набір даних, що забезпечує виконання таких завдань: проведення складних аналітичних запитів; вирішення конкретної оптимізаційної задачі; формування звітності; відповіді на певну низку запитів тощо. Наприклад, одна з «вітрин даних» містить інформацію про рецептурні інгредієнти, їх фізико-хімічні та функціонально-технологічні властивості, допоміжні матеріали, показники якості, зберігає призначені для користувача дані про рецептурний склад, фізико-хімічні характеристики інгредієнтів, статус рецептури.

База знань накопичується при формуванні правил та знань у відповідності до ситуації. БЗ вміщує правила щодо технологічних особливостей виготовлення багатокomпонентних продуктів харчування, наприклад, ковбас, морозива [15, 299, 300].

«Озеро даних» (Data lake) забезпечує накопичення неперетвореної інформації з різних джерел, що дозволяє скоротити час на її консолідацію та обробку. «Озеро даних» включає структуровані дані з реляційних баз даних (рядки і колонки), напівструктуровані дані (CSV, лог файли, XML, JSON), неструктуровані дані (поштові повідомлення, документи) і навіть бінарні дані (відео, аудіо, графічні файли) [301–303].

Фактично пропонується використовувати «озеро даних» як «стандартний архів даних», дані потрапляють туди за допомогою пакетів завантаження даних із мінімальною модифікацією. В «озері даних» елементи структур повністю відповідають структурам відповідних джерел. Таким чином, забезпечується централізоване зберігання даних.

До переваг «озер даних» відносять [303]:

- отримання повної інформації без перетворення;
- забезпечення детальної статистики;
- можливість експериментувати з різними типами даних із різних джерел;
- забезпечення доступу до інформації, яка необхідна керівникам в одному місці;
- забезпечення високого рівня централізації та деталізації даних;

- можливість масштабування, що підходить невеликим підприємствам, а також легко інтегрується з іншими системами;
- спрощення роботи користувачів;
- більш дешеві, ніж сховища, адже не потребують попередньої обробки.

При побудові програмної частини використано класичну багат шарову архітектуру з розподіленням на такі складові [10, 18, 52, 53]:

- шар контролерів, за допомогою яких забезпечується взаємодія з клієнтською частиною;
- адаптери, на рівні яких здійснюється перетворення даних із запитів клієнтів у основні бізнес-об'єкти, над якими здійснюється виконання подальшої обробки;
- класи, що здійснюють перевірки вхідних даних на відповідність встановленим правилам;
- бізнес-сервіси, які безпосередньо відповідають за виконання бізнес-логіки;
- допоміжні інструментальні класи, яким бізнес-сервіси делегують виконання окремих операцій;
- репозиторії, призначені для безпосередньої роботи з СД.

Використання «озера даних» забезпечує накопичення усіх даних, представлених різнорідними структурами, а потім дає можливість отримати будь-які набори даних для розв'язання різних задач.

«Засоби завантаження даних» забезпечують у автоматичному режимі отримання наборів даних із баз даних підприємств та бази знань, передачі їх у «озеро даних», а потім отримання необхідної інформації з нього та передачі її до сховища даних. «Засоби завантаження даних» забезпечують отримання, трансформацію, агрегацію та завантаження даних. Для уникнення надлишкової інформації у всіх елементах БЗ, СД, «озера даних» та «вітрин даних» здійснюється перевірка наявної актуальної інформації. За необхідністю «засоби завантаження даних» виконують знищення надлишкової та зайвої інформації. Як інструментарій «засобів завантаження даних» у сучасних клієнт-серверних СУБД виступає інструментарій Data Transformation Services [5, 15, 33, 289–292].

Функціональні можливості серверної частини відповідають потребам користувачів СППР та у загальному вигляді розподілені на модулі, які призначені для вирішення конкретних задач [5, 15]. Кожний модуль при виконанні обробки деякого запиту звертається для отримання даних до СД або БЗ. Після проведення обробки даних та необхідних підрахунків результат відображається клієнтською частиною у заданому вигляді. До основних модулів системи належать: модулі вирішення ЗПР, модулі вирішення задач із використанням ІАД та модулі підтримки функцій управління, що поєднують ведення статистичного обліку та створення звітів, модулі експертної системи.

Запропонована архітектура дає можливість поєднати декілька інформаційних технологій, а саме: використання на підприємствах харчової галузі експертно-моделюючої системи для моделювання складних рецептур продукції, що дає можливість підібрати у виробничих умовах заміну компонентів у рецептурі; використання технології планування виконання замовлень за рахунок застосування комбінованих методів та алгоритмів; використання інформаційної технології для удосконалення процесу аналізу та планування виконання замовлень на харчових підприємствах, яка, на відміну від відомих, ґрунтується на визначенні асортименту та коригуванні рецептур для зменшення собівартості й швидкого виконання замовлень.

У випадку, коли є можливість організувати виготовлення продукції із запасом та є впевненість в її повній реалізації, доцільно на основі ретроспективних даних формувати асортимент виготовлення згідно з аналізом інформації про попит на готову продукцію. Використання інформаційної технології, яка включає в себе адаптовані методи інтелектуального аналізу даних, дозволить підвищити ефективність господарської діяльності підприємств за рахунок формування пропозицій щодо коригування структури асортименту продукції на певний плановий період та за рахунок використання алгоритмів інтелектуального аналізу даних [16, 18, 19, 78, 79]. Застосування такої технології надає ОПР інформаційну підтримку для коригування асортименту продукції. Це проводиться завдяки

введенню нових видів продукції, використанню оригінальних та популярних інгредієнтів у рецептурі, збільшенню обсягів випуску продукції [16, 79].

Коригування структури асортименту продукції пропонується вирішувати із застосуванням інформаційної технології, що складається з 5 основних етапів [16, 79].

Етап 1. Класифікація продукції за прибутковістю методом ABC.

На першому етапі проводиться класифікація продукції за прибутковістю методом ABC. Класифікуємо асортимент продукції за прибутковістю методом ABC на основі принципу Парето. У практиці проведення класичного ABC-аналізу асортимент продукції розбивають на 3 групи [16, 79]:

A – продукція, яка приносить основний дохід.

B – менш затребувані види продукції.

C – найменш затребувані види продукції.

У результаті ранжування асортименту продукції з загального обсягу виділяється продукція групи A, що містить найбільш прибуткові види продукції.

У кінці першого етапу проводимо прогнозування попиту на продукцію групи A. У випадку збільшення попиту на продукцію групи A проводимо коригування плану виготовлення продукції цього типу у бік збільшення, в межах, дозволених резервами виробничих потужностей. В іншому випадку залишаємо план виготовлення продукції групи A без змін.

Етап 2. Виявлення характеристик продукції найбільш привабливої для споживача.

На другому етапі проводимо виявлення характеристик продукції найбільш привабливої для споживача. У роботах [16, 79] запропоновано класифікацію за такими спільними рисами:

- кількісні характеристики (співвідношення ціни та об'єму виробу, калорійність та поживна цінність продукції, терміни придатності);
- якісні характеристики (наявність у рецептурі специфічної та високовартісної сировини, харчових домішок, замінників натуральної сировини, наявність позначки «Без ГМО»);

- рецептурні (склад рецептури, ДСТУ та стандарти виготовлення).

У результаті здійснення другого етапу є можливість отримати такі важливі рішення [16, 79]:

- врахувати залежності між характеристиками продукції, що є суттєвими при розробці нових видів продукції певної групи;
- вивчити можливості застосування виявлених характеристик для підвищення ефективності реалізації існуючих видів продукції.

Етап 3. Пошук сукупностей значень характеристик продукції, що впливають на обсяги реалізації.

На третьому етапі здійснюється пошук сукупностей значень характеристик продукції, що впливають на обсяги реалізації. Для знаходження найбільш популярних сукупностей характеристик будується модель підтримки прийняття рішень методом кластеризації [16, 79]. Виявлення популярних інгредієнтів дає змогу сформулювати рекомендації на заміну чи введення їх до існуючих рецептур.

Етап 4. Пошук потенційно привабливих видів продукції за визначеними вище сукупностями значень характеристик.

На четвертому етапі відбувається пошук потенційно привабливих видів продукції за визначеними вище сукупностями значень характеристик.

Спочатку здійснюється пошук потенційно привабливих видів продукції за визначеними вище сукупностями значень характеристик у групах В та С. Для пошуку потенційно привабливих видів продукції відфільтруємо множину даних у групах В та С за знайденими на третьому етапі сукупностями значень характеристик. Вихідною інформацією є початкова множина даних, відфільтрована згідно з продукцією груп В та С. [16, 79]. Після зазначених дій є можливість чітко виділити групи видів продукції, що потребують:

- проведення додаткових рекламних заходів, для збільшення ефективності продажів;
- пошуку можливостей збільшення обсягів випуску.

Етап 5. Прогнозування собівартості та попиту на продукцію.

На п'ятому етапі відбувається прогнозування собівартості та попиту на кожний із видів продукції, визначених на етапах із першого по четвертий. Це забезпечить визначення рентабельності її виробництва на майбутні періоди.

Джерелом інформації для прогнозування собівартості окремих видів продукції є створений гіперкуб «Собівартість продукції». Для прогнозування витрат по статтях калькуляції використовуються гіперкуби «Собівартість по статтях калькуляції» та «Собівартість по елементах витрат» [16, 79].

Приймаючи рішення про остаточний рівень собівартості продукції, необхідно враховувати виникнення можливих ризиків. Ризики можуть бути пов'язані з ринковими умовами на плановий період. Собівартість готової продукції підприємств харчової галузі дуже залежна від ряду чинників, серед яких поточна ринкова ціна на сировину та матеріали, електроенергію та інші ресурси. Відхилення фактичних витрат від прогнозованих може призвести до неочікуваних підприємством збитків. Таким чином, у конкурентному середовищі успішно будуть функціонувати ті підприємства, які зможуть динамічно пристосуватися до змін навколишнього середовища та управління ризиками. Застосування системи управління ризиками дозволяє вчасно виявити, оцінити, локалізувати та проконтролювати ризик [16, 79, 304].

Спрогнозувавши собівартість продукції та оцінивши ризики прийняття її рівня, виконаємо прогнозування попиту на продукцію методом часових рядів Data Mining, враховуючи, що цей показник є більш прогнозованим і в значній мірі визначається звичками споживачів. Модель, що була використана для прогнозування, оцінюється на основі ретроспективних даних за попередній період, а ризики прогнозування – згідно з системою управління ризиками [16, 79, 304].

Оцінка управлінського рішення проводиться на основі отриманих прогнозів за методом точки беззбитковості. Збільшення обсягу виробництва популярних видів продукції приводить до перерозподілу умовно-постійних витрат на весь обсяг продукції. Це в результаті сприяє зниженню собівартості продукції [16, 79].

Застосування методів інтелектуального аналізу даних вимагає наявності великих обсягів даних [37, 38 78, 79, 305–307]. Але, якщо скористатися

накопиченою інформацією спорідненого підприємства або такого самого підприємства з іншого регіону, є можливість здійснити наблизений адекватний аналіз даних. Накопичену у «озеро даних» інформацію є можливість скоригувати та консолідувати у «віртуальне сховище даних» за необхідними параметрами. З одного боку, це є недоліком, адже якщо як вхідні дані подавати певну консолідовану інформацію, то її може бути недостатньо для роботи алгоритмів інтелектуального аналізу даних. Саме така структура системи дає можливість забезпечити використання даної технології, адже детальна інформація завантажується у «озеро даних», а необхідна інформація зберігається у віртуальному або тимчасовому сховищі даних, а для конкретних задач завантажується у відповідні «вітрини даних» [16, 79].

Крім того, запропонована структура дає можливість реалізувати на її базі експертно-моделюючу систему розрахунку рецептур складних експертно-моделюючих систем багатокomпонентних харчових продуктів. Одним із прикладів такої експертно-моделюючої системи є система для моделювання рецептур морозива, запропонована у роботах [15, 21, 299, 300, 308].

Архітектура СППР дозволяє побудувати на її базі експертно-моделюючу систему, що забезпечує коригування рецептури з урахуванням усіх технологічних властивостей багатокomпонентних харчових систем, використовуючи базу знань. Якщо моделювати оптимальну рецептуру тільки за допомогою одного математичного апарату, без використання експертної системи, то отримана рецептура навряд чи буде придатна для використання, оскільки не будуть враховані численні технологічні властивості. Елементи експертно-моделюючої системи були описані в роботах [15, 21, 299, 300, 308].

Використання експертно-моделюючої системи дозволяє цілеспрямовано управляти якістю готового продукту впродовж технологічного процесу його виробництва. Найбільша значимість розробки полягає в автоматизованому підборі для заміни традиційних рецептурних компонентів на принципово нові на основі натуральної сировини [15, 21, 300].

Застосування експертно-моделюючої системи забезпечує покращення комерційної складової діяльності підприємств за рахунок економії робочого часу та витрат на сировину і допоміжні матеріали. Використання експертно-моделюючої системи, побудованої саме на основі запропонованої структури, забезпечить інтегрування знань учених різних спеціальностей, а також скоротить витрати на складні лабораторні дослідження при підборі інгредієнтів при моделюванні нових рецептур виготовлення продукції [15, 300].

До модулів ЗПР входить модуль формування календарного плану виконання замовлень. Серед модулів запропоновано використання методики BSP (Business Systems Planning – системи організаційного планування) та виконання її запровадження в інформаційній системі підтримки управлінських рішень при виробництві макаронних виробів [5, 18, 39, 40, 43, 46, 47, 49, 51–53].

Модуль формування календарного плану виконання замовлень призначений для формування варіантів календарного плану з використанням запропонованих комбінованих алгоритмів. Також передбачена можливість оцінки кожного з варіантів та їх порівняння, внесення корективів, визначення додаткових параметрів. Побудований та обраний план передається в електронному вигляді до розгляду відповідальним особам, а після затвердження записується до баз даних інформаційних систем, які діють на підприємстві [18, 39, 40, 43, 46, 47, 49, 51–53].

Модуль підтримки функцій управління забезпечує надання повної та актуальної інформації про виконання замовлень, дотримання термінів календарного плану, рух ресурсів та матеріалів тощо. Цей модуль реалізує формування звітної документації, виконує функції створення різноманітних видів звітної документації у обраних користувачем формах та представленнях [18, 39, 40, 49, 51–53].

Розроблена архітектура гібридної web-орієнтованої системи підтримки прийняття рішень, яка реалізує запроповану інформаційну технологію на основі модифікованих методів та алгоритмів, базується на гнучкій інтеграції в «озері даних» інформації із різних інформаційних джерел. Запропонована архітектура поєднує декілька інформаційних технологій, які направлені на підвищення техніко-економічних показників підприємства за рахунок підтримки прийняття

управлінських рішень при формуванні плану виконання замовлень. Запропонована архітектура також забезпечить: оперативне формування та гнучкість при зміні планів виконання замовлень; значно зменшить витрати часу на збір даних та підготовку інформації; можливість швидкого пристосування до поточної ситуації та внесення відповідних змін до планів виконання замовлень.

## **5.2 Обґрунтування підходів розробки СППР та використання рефакторингу**

Сучасна теорія і практика показують, що ефективне управління діяльністю підприємств має бути засновано на постійному вдосконаленні СППР та його компонентів. У свою чергу, розробка та впровадження нової інформаційної системи є складним процесом, що триває від декількох місяців до декількох років. Успіх проєкту по розробці і впровадженню будь-якої інформаційної системи багато в чому залежить від готовності підприємства до ведення проєкту, особистої зацікавленості топ-менеджерів, реальної програми дій, наявності ресурсів, навченого персоналу, здатності до подолання опору, що склався на всіх рівнях організації [309, 310].

У цій ситуації особливо актуальним стає застосування спіральної моделі життєвого циклу з визначеним місцем рефакторингу [20, 311]. На кожній ітерації відбувається визначення та уточнення задач та вимог до програмного продукту, визначення кількісних та якісних характеристик та планування завдання на наступну ітерацію. Кожна ітерація складається з визначеної послідовності операцій. Після завершення кожної ітерації розробники отримують зворотній зв'язок від замовника про нову версію інформаційної системи з новим функціоналом. Після кожної ітерації проєкт стає складнішим за рахунок створення нового функціоналу та удосконалення існуючих функцій на основі деталізації вимог та зауважень від замовника. Використання цієї моделі доцільно при розробці систем із високим рівнем можливих ризиків та у випадках, коли невідомий конкретний перелік вимог

до кінцевого продукту або ці вимоги доволі складні. Спіральна модель поділяється на чотири квадранти, в кожен із яких входять основні та допоміжні дії по розробці системи.

На кожній ітерації виникає так званий «технічний борг», який впливає на доступність сприйняття програмного коду розробниками, підвищення відмовостійкості проєкту в цілому [311].

«Технічний борг» – це метафора програмної інженерії, що позначає накопичені в програмному коді або архітектурі проблеми, пов'язані з недотриманням вимог до якості при розробці програмного забезпечення. При подальшому удосконаленні чи модернізації програмного продукту зумовлюють додаткові витрати часу на усунення «технічного боргу». «Технічний борг» зазвичай непомітний для кінцевих користувачів продукту, пов'язаний з недоліками щодо супроводження, тестування, модифікації. При ігноруванні та невчасному усуненні «технічного боргу» зростають витрати часу на його усунення в подальшому. Збільшення «технічного боргу», як правило, негативно впливає на майбутній розвиток проєкту. Збільшення технічного боргу може бути свідомим компромісним рішенням у відповідності до обставин, що склалися [312–314].

Для усунення технічного боргу застосовується рефакторинг, операції якого поділені на категорії відповідно до мети застосування рефакторингу. Розуміння цих категорій важливе для раціонального проведення рефакторингу [24, 311].

Рефакторинг слід використовувати для декількох цілей [529, 315–317]:

- рефакторинг покращує композицію програмного забезпечення;
- рефакторинг полегшує розуміння програмного забезпечення;
- рефакторинг допомагає знайти помилки;
- рефакторинг пришвидшує створення програмних продуктів.

Застосування та використання рефакторингу обґрунтовано в роботах [17, 20, 24, 25, 29], в яких автор дисертаційного дослідження був співатором, а роботи [311, 315–328] стали базовими для використання рефакторингу.

У роботі [315] автор розділив методи рефакторингу на шість основних категорій: складання методів, переміщення функцій між об'єктами, організація

даних, спрощення умовних виразів, спрощення викликів методів, рішення задач узагальнення.

У роботах виділяється шість категорій рефакторингу баз даних [316, 317]: категорія рефакторинг структури, категорія рефакторинг якості даних, категорія рефакторинг посилальної цілісності, категорія рефакторинг архітектури, категорія рефакторинг методів, зміни, які не входять до операцій рефакторингу.

У роботах [311, 317] пропонується перед впровадженням певної операції рефакторингу відповісти на три запитання:

1. Чи має сенс проведення запропонованої операції рефакторингу?
2. Чи повинні зміни виконуватися невідкладно?
3. Чи виправдовує результат витрачені зусилля?

Для проведення рефакторингу необхідно керуватися підходом, що базується на першочерговому тестуванні. Він дозволяє збільшити шанси знаходження будь-яких порушень у роботі, що є наслідком проведення рефакторингу. Рефакторинг програмного коду направлений на зміну програмного коду для забезпечення гнучкої модернізації в процесі розширення проєкту новими функціями та програмними модулями без зміни наявного семантичного значення програмного коду [17, 20, 311].

Якщо необхідно здійснити зміни в структурі сховища даних або бази знань чи бази даних, необхідно пам'ятати, що зміни в їх структурі впливають на усі компоненти системи. Після внесення змін до структури необхідно оновити всі зовнішні програми, що мають доступ до елементів структури. Вся робота має бути організована з використанням системи керування версіями. Після проведення рефакторингу у виробничому середовищі необхідно повідомити про це усіх, хто працює над проєктом, а також усіх представників компаній, які є адміністраторами СППР [17, 20, 24, 41, 311].

Поточну версію СППР  $DSS_v$  опишемо у вигляді множини характеристик (5.1):

$$DSS_v \{f, pm, KD\{t, tr, r, w, sp, k, per\}, etl, DW\{DM\{ \}, t, tr, r, w, sp, d, per\}\}, \quad (5.1)$$

де  $v$  – версія системи;

$f$  – множина функцій системи;

$pm$  – множина програмних модулів;

$etl$  – множина програмних модулів для отримання, очищення, перетворення, завантаження даних у сховище даних та «озеро даних»;

$KD$  – база знань;

$DW$  – сховище даних;

$DM$  – «вітрини даних», призначені для вирішення певного кола задач.

База знань, сховище даних та «вітрина даних» складаються з таких елементів:

$t$  – множина таблиць;

$tr$  – множина тригерів;

$r$  – множина зв'язків між таблицями;

$w$  – множина представлень;

$sp$  – множина збережуваних процедур та функцій;

$d$  – множина даних, які зберігаються у сховищі або «вітрині даних»;

$per$  – множина користувачів та прав доступу.

Формалізований процес повного рефакторингу представимо виразом (5.2):

$$DSS_v \{ \} \xrightarrow{\text{рефакторинг}} DSS_{vn} \{ \}, \quad (5.2)$$

де  $DSS_{vn} \{ \dots \}$  – СППР після проведення рефакторингу з номером версії  $vn$ .

Процес проведення рефакторингу зображено на рис. 5.2.

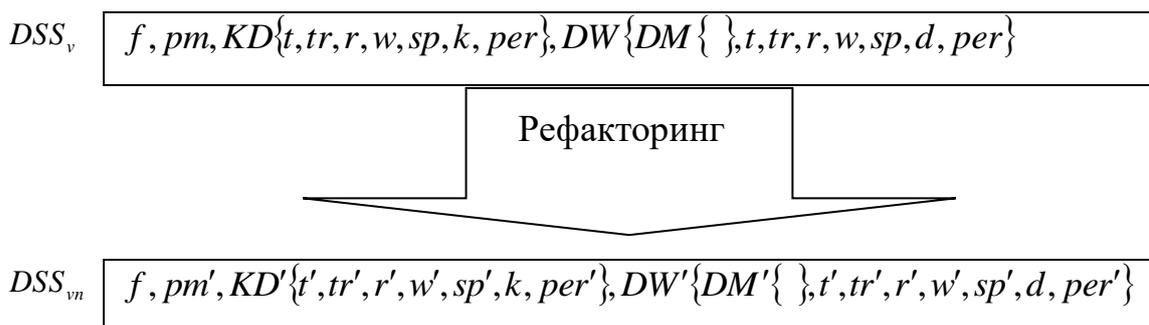


Рисунок 5.2 – Процес повного рефакторингу

Оскільки як рефакторинг коду полягає в зміні програмного коду без зміни його семантичного значення з метою покращення читання та розуміння програмного коду розробником програмного забезпечення, представимо його виразом (5.3):

$$DSS_v \{ \} \xrightarrow{\text{рефакторинг програмного коду}} DSS_{vn} \{ \}, \quad (5.3)$$

де  $DSS_{vn} \{ f, pm', KD \{ t, tr, r, w, sp, k, per \}, etl', DW \{ DM \{ \}, t, tr, r, w, sp, d, per \} \}$  –

СППР після проведення рефакторингу коду;

$v$  – початкова версія системи;

$vn$  – версія системи, що відображає зміни після впровадження доповнення, що передбачає проведення рефакторингу;

$f$  – множина функцій системи, яка в жодному разі не може бути змінена, адже це суперечить визначенню рефакторингу;

$pm'$  – множина програмних модулів із впровадженим доповненням, що передбачає проведення рефакторингу коду;

$etl'$  – множина програмних модулів для отримання, очищення, перетворення, завантаження даних у сховище даних та «озеро даних» із впровадженим доповненням, що передбачає проведення рефакторингу коду;

$KD$  – база знань;

$DW$  – сховище даних;

$DM$  – «вітрини даних», призначені для вирішення певного кола задач.

База знань, сховище даних та «вітрина даних» складаються з таких елементів:

$t$  – множина таблиць;

$tr$  – множина тригерів;

$r$  – множина зв'язків між таблицями;

$w$  – множина представлень;

$sp$  – множина збережуваних процедур та функцій;

$d$  – множина даних, які зберігаються у сховищі або «вітрині даних»;

$per$  – множина користувачів та прав доступу.

У свою чергу, рефакторинг бази знань – це зміна структурної схеми бази знань із метою покращення роботи усієї СППР. Рефакторинг бази знань описується виразом (5.4):

$$DSS_v \{ \} \xrightarrow{\text{рефакторинг бази знань}} DSS_{vn} \{ \}, \quad (5.4)$$

де  $DSS_{vn} \{ f, pm', KD' \{ t', tr', r', w', sp', k, per' \}, etl, DW \{ DM \{ \}, t, tr, r, w, sp, d, per \} \}$  –

СППР після проведення рефакторингу бази знань;

$v$  – початкова версія системи;

$vn$  – версія системи, що відображає зміни після впровадження доповнення, що передбачає проведення рефакторингу;

$f$  – множина функцій системи, яка в жодному разі не може бути змінена, адже це суперечить визначенню рефакторингу;

$pm'$  – множина програмних модулів із впровадженим доповненням, що передбачає проведення рефакторингу коду, адже впровадження змін до множини елементів бази знань може викликати впровадження змін до множини програмних модулів через залежність програмного коду від структури бази знань;

$KD$  – множина елементів бази знань після проведення рефакторингу, що змінили один з наступних елементів:

$t'$  – множина таблиць, у випадку, якщо зміни таблиць не було, маємо  $t'=t$ ;

$tr'$  – множина тригерів, у випадку, якщо зміни тригерів не було, маємо  $tr'=tr$ ;

$r'$  – множина зв'язків між таблицями, у випадку, якщо зміни зв'язків не було, маємо  $r'=r$ ;

$w'$  – множина представлень, у випадку, якщо зміни представлень не було, маємо  $w'=w$ ;

$sp'$  – множина збережуваних процедур та функцій, у випадку, якщо зміни процедур та функцій не було, маємо  $sp'=sp$ ;

$per'$  – множина користувачів та прав доступу, у випадку, якщо зміни користувачів та прав не було, маємо  $per'=per$ ;

$k$  – множина знань, яка зберігається у базі знань і в процесі проведення рефакторингу в жодному разі не змінюється;

$etl$  – множина програмних модулів для отримання, очищення, перетворення, завантаження даних у сховище даних та у «озеро даних»;

$DW$  – сховище даних, що представляє собою сукупність таких елементів:

$DM$  – «вітрини даних», призначені для вирішення певного кола задач та складаються з тих елементарних множин, що і сховище даних;

$t$  – множина таблиць;

$tr$  – множина тригерів;

$r$  – множина зв'язків між таблицями;

$w$  – множина представлень;

$sp$  – множина збережуваних процедур та функцій;

$d$  – множина даних, які зберігаються у сховищі або «вітрині даних»;

$per$  – множина користувачів та прав доступу.

Рефакторинг сховища даних – це зміна структурної схеми сховища даних з метою покращення роботи усієї СППР. Рефакторинг сховища даних описується виразом (5.5):

$$DSS_v \{ \} \xrightarrow{\text{рефакторинг сховища даних}} DSS_{vn} \{ \}, \quad (5.5)$$

де  $DSS_{vn} \{ f, pm', KD \{ t, tr, r, w, sp, k, per \}, etl', DW' \{ DM' \{ \}, t', tr', r', w', sp', d', per' \} \}$  –

СППР після проведення рефакторингу сховища даних;

$v$  – початкова версія системи;

$vn$  – версія системи, що відображає зміни після впровадження доповнення, що передбачає проведення рефакторингу;

$f$  – множина функцій системи, яка в жодному разі не може бути змінена, адже це суперечить визначенню рефакторингу;

$pm'$  – множина програмних модулів із впровадженим доповненням, що передбачає проведення рефакторингу коду, адже впровадження змін до множини елементів сховища даних може викликати впровадження змін до множини програмних модулів через залежність програмного коду від структури сховища даних;

$KD$  – множина елементів бази знань після проведення рефакторингу, що змінили один із таких елементів:

$t$  – множина таблиць;

$tr$  – множина тригерів;

$r$  – множина зв'язків між таблицями;

$w$  – множина представлень;

$sp$  – множина збережуваних процедур та функцій;

$k$  – множина знань, яка зберігається у базі знань і в процесі проведення рефакторингу в жодному разі не змінюється;

$per$  – множина користувачів та прав доступу;

$etl'$  – множина програмних модулів для отримання, очищення, перетворення, завантаження даних у сховище даних та «озеро даних», адже впровадження змін до множини елементів сховища даних може викликати впровадження змін до множини програмних модулів через залежність програмного коду від структури сховища даних;

$DW$  – сховище даних, що представляє собою сукупність таких елементів:

$DM'$  – «вітрини даних», призначені для вирішення певного кола задач та складаються з тих елементарних множин, що і сховище даних;

$t'$  – множина таблиць, у випадку, якщо зміни таблиць не було, маємо  $t'=t$ ;

$tr'$  – множина тригерів, у випадку, якщо зміни тригерів не було, маємо  $tr'=tr$ ;

$r'$  – множина зв'язків між таблицями, у випадку, якщо зміни зв'язків не було, маємо  $r'=r$ ;

$w'$  – множина представлень, у випадку, якщо зміни представлень не було, маємо  $w'=w$ ;

$sp'$  – множина збережуваних процедур та функцій, у випадку, якщо зміни процедур та функцій не було, маємо  $sp' = sp$ ;

$per'$  – множина користувачів та прав доступу, у випадку, якщо зміни користувачів та прав не було, маємо  $per' = per$ ;

$d'$  – множина даних, яка зберігається у сховищі або «вітринах даних», оновлюється повністю, у випадку, якщо були внесені зміни.

На рис. 5.3 представлена спіральна модель розробки програмного забезпечення з визначеним місцем проведення рефакторингу.

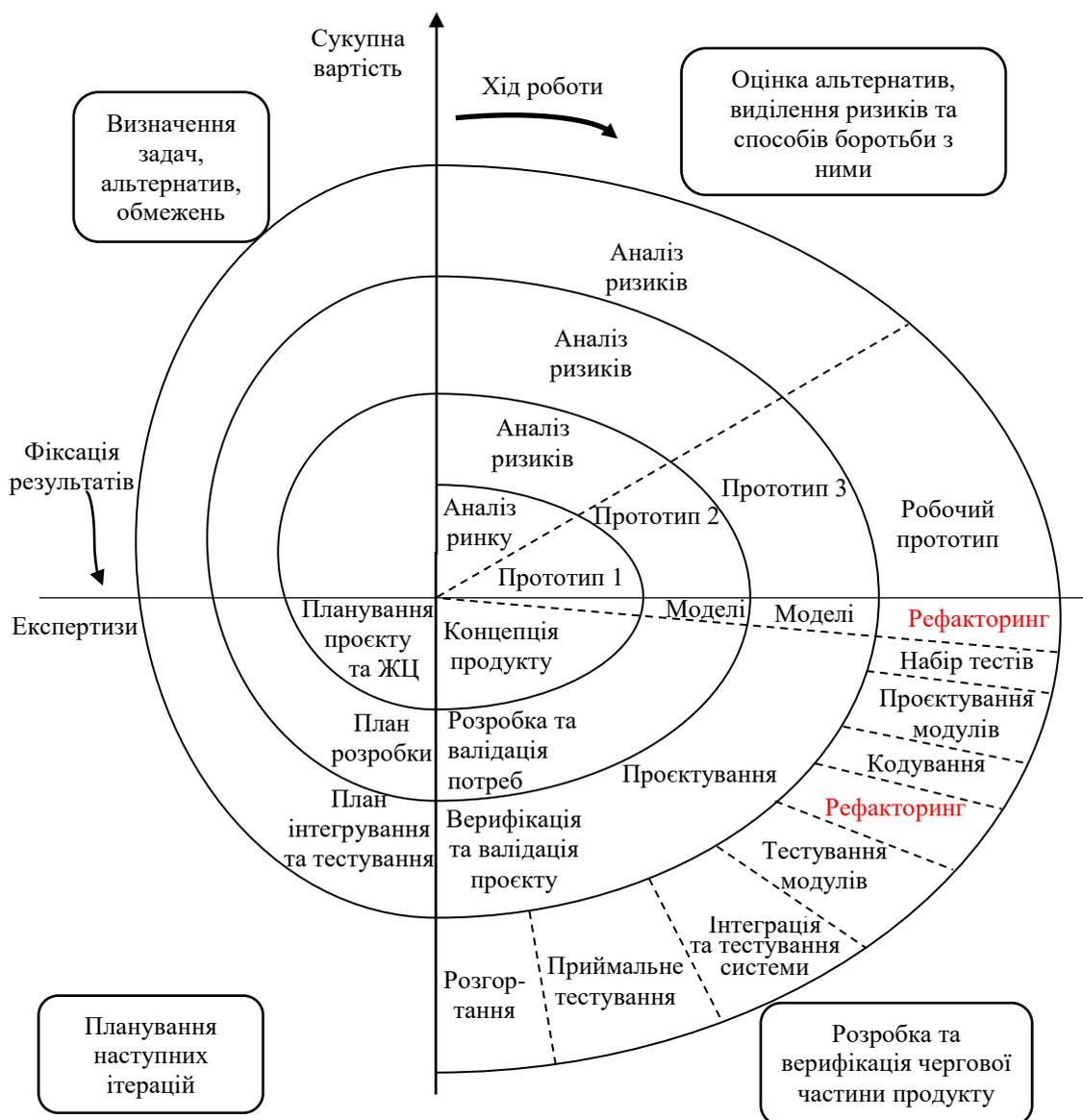


Рисунок 5.3 – Рефакторинг у спіральній моделі розробки програмного забезпечення

Спіральну модель можна описати як гібрид ітераційної та каскадної моделей. Кожна ітерація має послідовність операцій, які відповідають крокам у каскадній моделі. Після завершення кожної ітерації, розробники отримують зворотній зв'язок від замовника про нову версію інформаційної системи з новим функціоналом. На кожній ітерації відбувається визначення, уточнення задач та вимог до програмного продукту, визначення кількісних та якісних характеристик, планується завдання на наступну ітерацію. Рефакторинг проводиться саме у третьому квадранті спіральної методології, що включає в себе розробку моделей та безпосередньо написання програмного коду на основі прототипів програмного забезпечення (рис. 5.3).

Після кожної ітерації проєкт стає складнішим, якщо не використовувати рефакторинг, тоді з кожною ітерацією буде зростати технічний борг [20, 311].

Така модель забезпечує раціональне застосування рефакторингу, а саме [20, 311]: виключає проведення рефакторингу після тестування, щоб не проводити тестування повторно; виключає недоцільності проведення рефакторингу перед етапом кодування, адже його проведення повинно бути при наявності стабільної версії інформаційної системи.

Потреба у рефакторингу виникає при виправленні помилок у прототипах та виникненні конфліктів цілей поточної ітерації з уже готовими напрацюваннями із попередніх ітерацій. Використання рефакторингу можливо також і в другому квадранті, при використанні CASE-засобів у процесі прототипування.

У роботі [311] розроблено імітаційну модель та показано ефективність проведення рефакторингу при розробці інформаційних систем різної складності. Необхідно відмітити, що в імітаційній моделі враховується рівень команди розробників, що також є важливим для процесу розробки. На основі наведених результатів експерименту встановлено, що використання рефакторингу сприяє зменшенню загального часу на проведення розробки та зниженню технічного боргу, відповідно якість кінцевого продукту вища. Проте використання рефакторингу доцільно лише для роботи над довготривалими проєктами. Також підтверджено неефективність застосування рефакторингу на проєктах, які не будуть у подальшому супроводжуватися та розвиватися [311].

Формалізація процесу створення системи підтримки прийняття рішень із застосуванням спіральної моделі розробки програмного забезпечення, в якій, на додаток до традиційної, чітко визначено місце проведення рефакторингу, дала змогу обґрунтувати такий вибір, а також забезпечило скорочення часу на розробку та підвищення ефективності розробки в цілому. Враховуючи, що створювана СППР буде впроваджуватися на різні підприємства з різним ступенем інформатизації, застосування рефакторингу забезпечить у подальшому легкість її інтеграції та розвитку в цілому.

### **5.3 Проєктування та створення елементів баз знань та сховищ даних**

Моделювання та проєктування елементів сховища даних, вітрин даних та баз знань виконано в автоматизованому режимі з використанням CASE-інструментарію AllFusion Erwin Data Modeler [2, 16, 17, 25, 29, 110, 111]. Цей програмний продукт підтримує роботу з різними СУБД [110, 111] та суттєвою особливістю даного програмного забезпечення є те, що воно дозволяє отримувати структуру з існуючих баз та сховищ даних, а також генерувати код для їх створення в різних СУБД.

Створення елементів сховища даних, «вітрин даних» та баз знань починається із побудови їх логічної моделі. Після отримання повної логічної моделі, яка відображає і описує інформацію, якою вона є в реальному світі, відбувається побудова її на фізичному рівні, який відповідає певному СУБД. Результатом моделювання є сукупність пов'язаних між собою таблиць та деталізовані взаємовідношення між ними.

Великою перевагою AllFusion Erwin Data Modeler є можливість порівнювати різні моделі даних, виявляти їх збіжність та розбіжність. Для забезпечення організації зберігання моделей даних був використаний AllFusion Model Manager, який підтримує функціональність та синхронізацію моделей процесів і моделей даних, що забезпечує визначення відповідності цих моделей [2, 78, 110, 111].

Окремим елементом є база знань, яка вміщує набір знань у вигляді правил щодо технологічних особливостей виготовлення продукції БЗ, призначена для зберігання довгострокових фактів, які описують технології та виробництво готового продукту і правил, які описують відносини між цими фактами, та інших типів декларативних знань [15, 21, 229, 300, 308].

Продукційні правила в базі знань мають вигляд (ЯКЩО\_А\_ТО\_С), де А – умова, С – дія. Дія С виконується, якщо А істинно. Дія С, так само, як і умова, являє собою твердження, що може бути виведено системою (тобто стає їй відомою), якщо істинна умова правила А [15, 299, 300]. Структура бази знань має узагальнений характер та підходить для харчових підприємств (рис. 5.4).

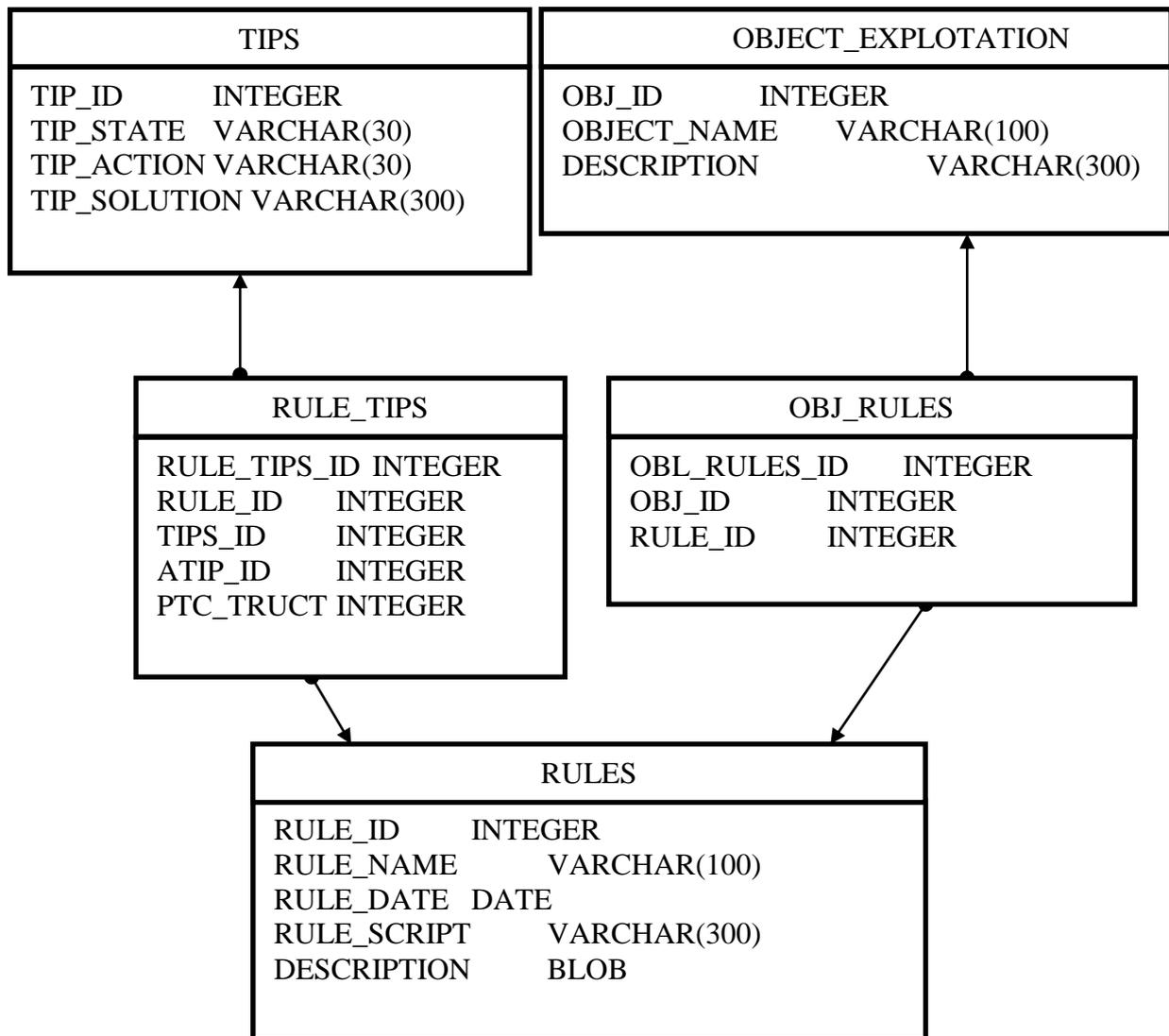


Рисунок 5.4 – Структура бази знань

Запропонована структура бази знань (рис. 5.4) пройшла апробації при створенні експертно-моделюючої експертної системи моделювання рецептур морозива [15, 300].

До основних об'єктів бази знань належать [15, 300]:

1. **OBJECT\_EXPLORATION** – вміщує інформацію про об'єкти, які піддаються аналізу, для виявлення в рецептурі технологічних відхилень та складається з таких атрибутів: **OBJ\_ID** – унікальний номер об'єкта, **OBJ\_NAME** – ім'я об'єкта, **DESCRIPTION** – короткий опис об'єкта.

2. **TIPS** – вміщує опис проблем та рекомендації для їх усунення та складається з таких атрибутів: **TIP\_ID** – унікальний номер рекомендації, **TIP\_STATE** – описує стан, **TIP\_ACTION** – дія (негативна чи позитивна), яка викликає поточний стан досліджуваного об'єкта, **TIP\_SOLUTION** – рекомендації користувачу, виходячи зі стану аналізованого об'єкта.

3. **RULES** – вміщує усі правила, яким може відповідати об'єкт дослідження при тих чи інших подіях. Формат запису є текстовий, між параметрами використовуються логічні операції (**AND**, **OR**, **NOT**). Розбір правила виконується програмно за допомогою спеціального парсера, який забезпечує аналіз записаного правила. Атрибути: **RULE\_ID** – унікальний номер правила, **RULE\_NAME** – назва правила, **RULE\_DATE** – дата створення правила, **RULE\_SCRIPT** – вираз, що описує правило в текстовому форматі з використанням параметрів, в ролі яких можуть виступати об'єкти дослідження. Цей вираз може бути представлений у вигляді рекурсії, тобто коли одне правило вкладено в інше.

4. **OBJ\_RULES** – таблиця зв'язків між елементами таблиць **OBJECT\_EXPLORATION** та **RULES**.

5. **RULE\_TIPS** – таблиця зв'язків між елементами таблиць **TIPS** та **RULES**.

Структура кожної «вітрини даних» розроблялася згідно з забезпеченням певної задачі або наданої інформації за певний період. «Вітрина даних» має цільове призначення та містить набір даних, що забезпечує виконання таких завдань: проведення складних аналітичних запитів; вирішення конкретної оптимізаційної

задачі; формування звітності; відповіді на певну низку запитів тощо. За рахунок того, що є чітке розбиття на частини і кожна «вітрина даних» призначена для вирішення певної задачі, забезпечується повнота доступу до необхідних даних.

Наприклад, «вітрина даних», наведена на рис. 5.5, призначена для підтримки роботи експертно-моделюючого модуля [15, 300].

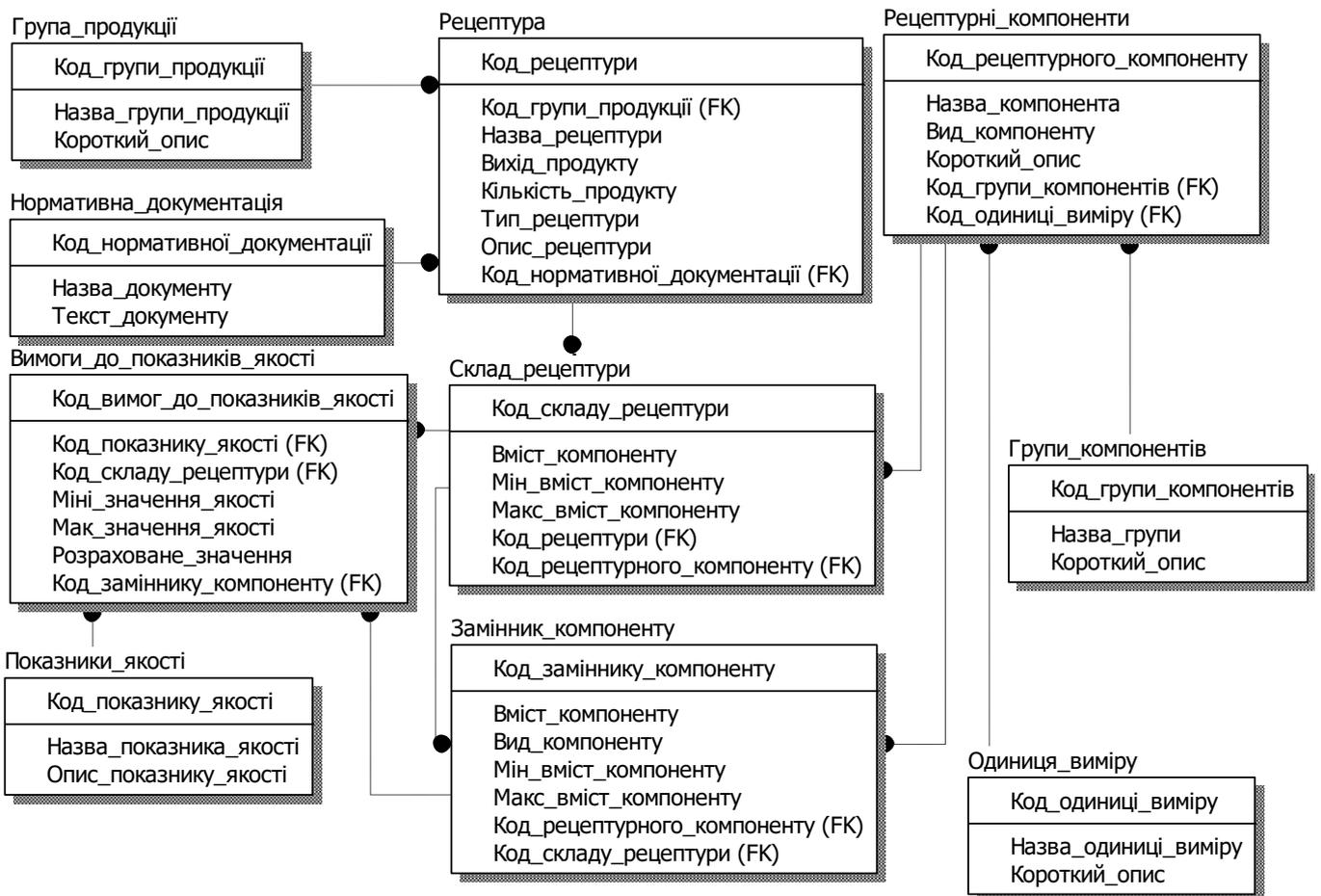


Рисунок 5.5 – Модель вітрини даних для підтримки роботи експертно-моделюючого модуля

Необхідно відзначити, що дані у «вітринах даних» оновлюються у відповідності до бізнес-логіки, мають певну агрегацію та деталізацію. За рахунок того, що «вітрина даних» вміщує тільки необхідну інформацію для розв’язання певної задачі на момент часу, мають доступ до неї лише вузькоспеціалізовані запити, тому це приводить до виключення зайвої обробки інформації чи її пошуку. Таким чином обробка інформації здійснюється у рази швидше. Це досягається за

рахунок того, що, використовуючи засоби завантаження даних, у вітрину завантажуються тільки актуальна інформація, а детальна інформація залишається в «озері даних».

Крім того, були змодельовані окремо «вітрини даних» для кожної задачі, що вирішується СППР. Наприклад, логічна модель «вітрини даних» для моніторингу виконання замовлень за певний період представлена на рис. 5.6.

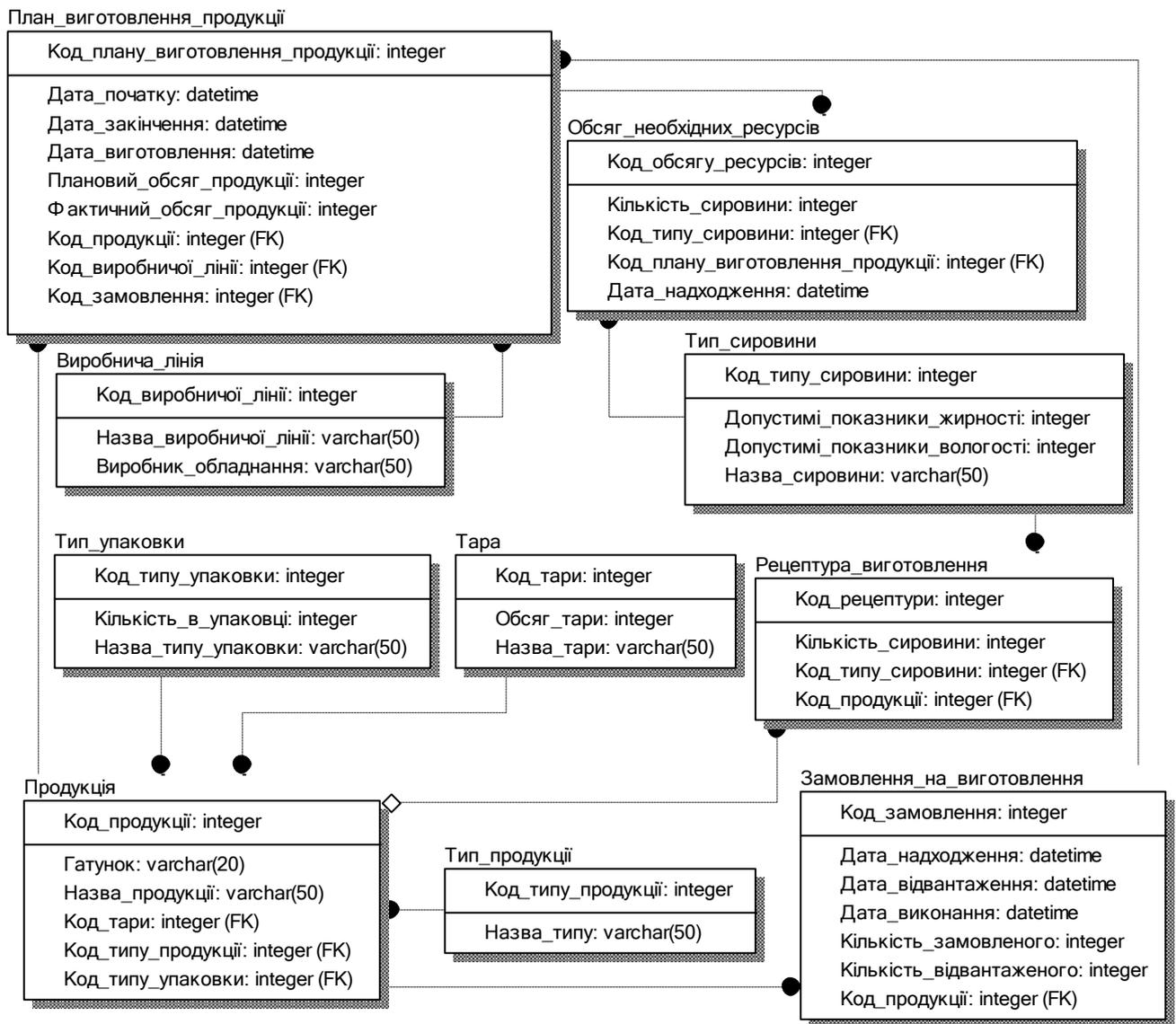


Рисунок 5.6 – Логічна модель «вітрини даних» для моніторингу виконання замовлень за певний період

У роботі [78] був запропонований підхід створення структури сховища даних на основі отриманих моделей структур баз даних, які були джерелами даних для

завантаження. При створенні структур сховища та «вітрин даних» було обрано підхід, який базується на потребах наборів даних, необхідних для розв'язку задач. Спочатку будувалися структури «вітрини даних», а на їх основі – загальна структура «віртуального сховища даних», що дає змогу уникнути надлишковості.

Модель «вітрини даних» для реалізації інформаційної технології представлена у Додатку Ж. Модель складається з таких таблиць: «План виробництва», «Змінне завдання», «Всі\_замовлення», «Продукція», «Сировина», «Альтернативні рішення», «WOLF», «Сформований маршрут», «ANT COLONY» тощо. Фізична модель створюється на основі логічної моделі бази даних, на основі якої вже формується системний код для створення структури в обраній СУБД.

Використання CASE-засобу для проектування і створення моделей даних AllFusion ERwin Data Modeler r7 забезпечило швидку розробку структур «вітрин даних», сховища даних та бази знань. Даний засіб забезпечує порівняння моделей, що дає можливість в автоматизованому режимі виявити усі подібні елементи [2, 110, 111]. За рахунок того, що засіб підтримує перехід між основними популярними СУБД, були отримані SQL-коди створення усіх елементів.

Як СУБД для реалізації сховища та «вітрин даних», а також бази знань був обраний MS SQL Server 2008, тому що він підтримує різні архітектури, а також використовується на підприємствах, де проходили апробацію елементи системи.

MS SQL Server 2008 може бути розгорнутий як на сервері підприємства, так і локально на персональному комп'ютері користувача. Він є універсальною платформою для вирішення широкого кола завдань: зберігання даних, бізнес-аналітики, розробки додатків, консолідації серверів тощо. До складу СУБД як інтегрований компонент входить величезний комплекс засобів бізнес-аналітики. Корпорації розглядають SQL Server як ключовий компонент своєї платформи доступу до даних, до складу якої входять офісний пакет Microsoft Office та засіб розробки Visual Studio [328–333].

Зі збільшенням обсягів даних і кількості користувачів організаціям стає все складніше отримати передбачуваний відгук. SQL Server 2008 містить великий набір компонентів, що забезпечують масштабовану і передбачувану продуктивність для

будь-якого робочого навантаження. Стиснення даних дозволяє зберігати їх більш ефективно і зменшує витрати на накопичувальні пристрої [328–333].

SQL Server 2008 являє собою масштабовану платформу для організації сховищ даних, яка дозволяє швидше і ефективніше інтегрувати інформацію в сховища, а також керувати зростаючими його обсягами, надаючи всім користувачам відомості, необхідні для кращого розуміння цієї інформації. Всебічний аналіз бізнес-даних дозволяє приймати більш обґрунтовані рішення завдяки тому, що користувачі можуть збирати, очищати, зберігати і готувати дані для цієї мети. Орієнтуючись на тенденції ринку бізнес-аналітики, творці SQL Server 2008 включили в нього розширену інфраструктуру, за допомогою якої ІТ-фахівці зможуть збільшувати можливості інтелектуального аналізу даних у масштабі всієї організації. Цьому сприяють управління звітами і аналізом будь-якого обсягу, а також глибока інтеграція з Microsoft Office System. Платформа SQL Server 2008 дозволяє організаціям залучати до бізнес-аналізу всіх співробітників, завдяки чому рішення приймаються швидше і виявляються якіснішими і компетентнішими [328–333].

За рахунок стиснення даних SQL Server 2008 більш ефективно зберігає інформацію (у тому числі, резервні копії), а також підвищує продуктивність завдяки скороченню обсягів введення-виведення. Секціонування полегшує роботу з великими таблицями, тому що поділяє їх на логічні блоки, які є простими в управлінні та досить прозорими для адміністрування. У SQL Server 2008 ці можливості розширені у порівнянні з попередньою версією [328–333].

Обрання СУБД MS SQL Server обумовлено функціональною достатністю вбудованих сервісів, а саме засобів Integration Services (SSIS), що забезпечують підтримку гнучкої та потужної архітектури для ефективної інтеграції даних, розроблення та використання пакетів перетворення даних за заданим алгоритмом розробника, глибоку інтеграцію пакетів із інструментарієм ІАД [328–333].

Генерація структури БД на основі автоматично створеного SQL-коду відбувається при натисканні кнопки Generate. Діалог зв'язку з БД і виконання SQL-коду відбувається в результаті натиснення кнопки Connect. У середовищі MS SQL

Server 2008 отримуємо згенеровану базу даних з усіма визначеними таблицями та зв'язками.

У зв'язку з тим, що проектування та створення всього проєкту відбувалося з використанням спіральної методології, постала задача ведення версіонування елементів системи, а також версій «вітрин даних», сховища даних та бази знань.

Версіонування є невід'ємним інструментом протягом всього процесу розробки програмного забезпечення, яке забезпечує визначення та організацію роботи команди розробників, а також визначає взаємозв'язки між компанією та користувачами. Правильно обрана схема версіонування повинна надати користувачам можливість зрозуміти, які функції та помилки присутні в тому випуску, який використовується. Коли користувач зіштовхнеться з певною помилкою у відповідній версії програмного продукту, він може [25, 311, 333]:

- оновити програмний продукт до більш нової версії, в якій ця помилка виправлена;
- за відсутністю виправлень звернутися до компанії розробника з чітким формулюванням помилки та дочекатися нової версії.

У багатьох різних схемах версіонування є свої особливості, але найчастіше основною метою є надання інформації про відмінності між двома версіями [311, 333, 334].

Після проведеного аналізу було прийнято рішення про використання «семантичного версіонування бази даних» [25, 311, 333, 334] для маркування «версій вітрин» та сховища даних, бази знань.

Специфікація «семантичне версіонування бази даних» заснована на семантичному версіонуванні програмного продукту, який містить в собі три числа, розділених крапкою, де кожне з чисел (X.Y.Z) має відповідну назву – мажорна.мінорна.патч. У специфікації «семантичне версіонування бази даних» зміни мажорної, мінорної та патч-версії відбуваються у таких випадках [25, 311]:

- мажорна версія інкрементується при додаванні нового поля до таблиці, видаленні поля із таблиці, перейменуванні поля, видаленні NULL-значення, додаванні обмежень;

- мінорна версія інкрементується при додаванні нового елементу, нового поля, за умовою, що воно буде містити значення за замовчуванням, при додаванні NULL-значення, при видаленні обмежень, при зміні тригерів;
- патч-версія інкрементується при зміні типів даних, за умови, що початковий і кінцевий типи даних є сумісними, при додаванні NULL-значення, якщо це є виправленням помилки проєктування бази даних, при видаленні обмежень, якщо це є виправленням помилки проєктування.

У процесі проєктування баз знань, вітрин та сховищ даних використано підходи щодо маркування та автоматизованого впровадження оновлення їх компонентів і елементів за рахунок застосування категорії рефакторингу доступу та специфікації семантичного версіонування баз даних, що забезпечило підвищення якості та скорочення часу розробки.

#### **5.4 Обрання та обґрунтування технологій розробки**

Розробка програмної логіки роботи описаних модулів у розділі 5.1 на серверній частині здійснювалась мовою програмування Java, вибір якої зумовлений відомими перевагами [18, 335–337], з використанням Spring Framework та Hibernate. Не зважаючи на те, що мова програмування Java надає значну кількість вбудованих функціональних можливостей для розробки інформаційних систем, засоби для організації складових програмних частин складної системи у єдине ціле не надаються. Вирішення такої проблеми покладається безпосередньо на розробників та архітекторів. Для розв'язання даної задачі обрана програмна платформа Spring Framework, що надає формалізовані засоби об'єднання розрізнених компонентів у єдину програмну систему на основі принципу введення залежностей (Dependency Injection) [338-340].

У відповідності зі вказаним принципом задача побудови залежностей об'єкта вирішується за допомогою зовнішнього механізму. Ключове призначення

програмної платформи Spring Framework полягає у забезпеченні побудови програмної інфраструктури СППР [18, 339, 340].

Spring Framework має модульну структуру, що дозволяє використовувати лише необхідні для реалізації функцій системи елементів [339, 340].

Модулі об'єднані у групи за функціональним призначенням (рис. 5.7), а саме на такі [18]:

- ядро програмної платформи (Core Container);
- модулі доступу до даних (Data Access/Integration);
- модулі для розробки web-орієнтованих систем (Web);
- модулі для роботи з аспектно-орієтованим програмуванням (AOP – Aspect Oriented Programming);
- інструментальні модулі (Instrumentation);
- тестові модулі (Test).

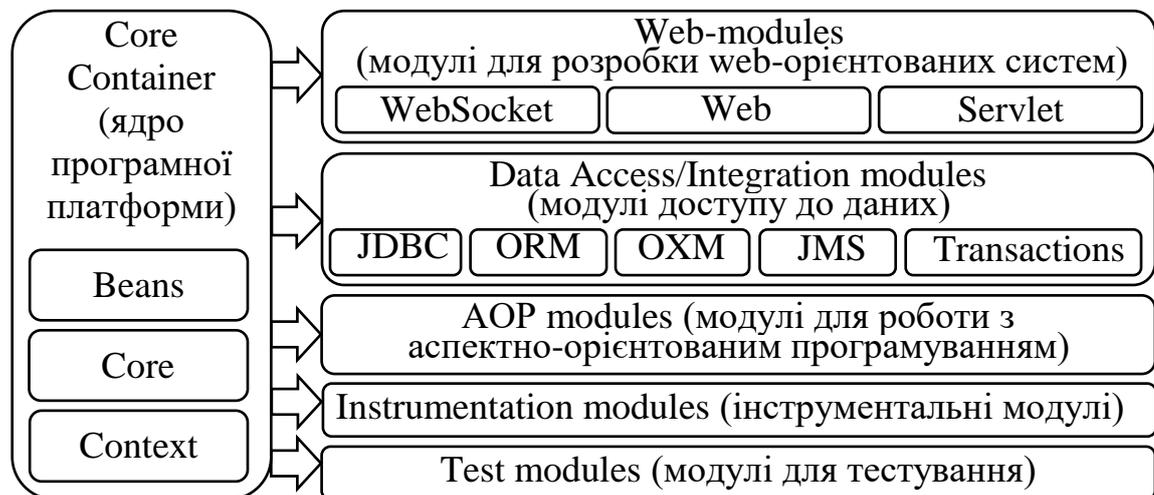


Рисунок 5.7 – Модулі програмної платформи Spring Framework, розподілені на групи за функціональним призначенням

Фундаментальну частину програмної платформи Spring Framework складають модулі Beans та Core, які забезпечують можливості введення залежностей між компонентами СППР. Саме це дозволяє виокремити конфігурацію системи та специфікацію залежностей об'єктів. Модуль Context розширяє Beans, надає вбудовану підтримку наступної функціональності [18]:

- локалізація та інтернаціоналізація з використанням пакетів ресурсів;
- використання та поширення подій між компонентами для керування поведінкою;
- створення програмних середовищ (контекстів) для управління часом існування та взаємодією компонентів (наприклад, контейнер сервлетів).

Модуль JDBC надає власний рівень абстракції для безпосередньої роботи з базою та сховищем даних, що забезпечує зменшення обсягів роботи для розробників, а також виключає необхідність обробки специфічних помилок від конкретних систем управління базами даних.

Модуль ORM представляє собою основу для роботи з технологіями для об'єктно-реляційного відображення, такими, як Hibernate. Модуль для обміну повідомленнями (JMS) надає функції для створення та отримання повідомлень, чим забезпечується взаємодія розроблюваної СППР зі сторонніми інформаційними системами. Модуль Transactions підтримує програмне та декларативне керування транзакціями, що надає можливості для гнучкого керування цілісністю даних, які використовуються системою, та єдність операцій, які виконуються [18, 340].

Spring забезпечує роботу з локальними та глобальними транзакціями, без прив'язки до технологій реалізації. Надається досить широка підтримка реалізації декларативних транзакцій як за допомогою XML-конфігурації, так і анотацій. Крім цього, є можливість управляти транзакціями програмно, що зумовлює гнучкість при налаштуванні в залежності від конкретних вимог до функціональності системи.

Модулі для розробки web-орієнтованих систем надають основні можливості розробки контролерів web-сервісів для web-інтеграції системи, що є ключовим для взаємодії серверної частини з користувальницькою [18, 341-343].

Модуль AOP відповідає за роботу з аспектно-орієтованим програмуванням та забезпечує можливість створення методів-перехоплювачів. Такі методи є досить корисними за необхідності виокремлення наскрізної для різних об'єктів функціональності.

Також можливості аспектно-орієтованого програмування використовуються для додаткового ведення обліку інформації про усі операції, що здійснюються під

час роботи системи з описом усіх викликів та значеннями переданих аргументів. Крім цього, надається можливість інтеграції з бібліотекою AspectJ, яка надає розширені способи перехоплення викликів та впровадження деякої програмної логіки на різних етапах взаємодії з об'єктами [18].

Модулі для тестування, які надають можливість тестування компонентів під управлінням Spring Framework, були використані для проведення інтеграційного тестування компонентів системи у поєднанні з бібліотекою для модульного тестування Junit.

Вибір програмної платформи для побудови архітектури інформаційної системи обумовлений можливістю повної незалежності бізнес-логіки від елементів програмної платформи та ізольованістю залежностей на компоненти від решти програмного коду. Дана перевага надається Spring Framework, що також вплинуло на її вибір як програмну платформу для побудови програмної архітектури [18].

При розробці СППР увагу приділено роботі з джерелами даних, оскільки взаємодія між об'єктами об'єктно-орієнтованої мови програмування та реляційною базою даних зазвичай досить громіздка та трудомістка. Це зумовлено невідповідністю парадигм того, як зберігаються дані, представлені об'єктами, у порівнянні з базами даних.

Для зв'язування баз даних з концепціями об'єктно-орієнтованих мов програмування доцільним є використання технології ORM (Object-Relational Mapping – об'єктно-реляційне відображення). Концепція ORM у Java EE представлена специфікацією JPA (Java Persistence API) [18].

Для роботи з СД використано одну з реалізацій підходу об'єктно-реляційного відображення, а саме Hibernate. Задачею Hibernate є перенесення даних із рядків таблиць бази даних у об'єкти Java і навпаки [18, 344].

Крім цього, Hibernate надає вбудовані можливості для запитів і пошуку, що дозволяє скоротити час розробки, який витрачається на ручну обробку даних у SQL і JDBC.

Проте, на відміну від багатьох подібних рішень, Hibernate не приховує потужності SQL і дозволяє за необхідності використовувати реляційні технології напряду.

Вибір Hibernate також зумовлено відсутністю необхідності у складній взаємодії з базою даних, використанням складних запитів тощо, оскільки оперування даними відбувається на програмному рівні.

У порівнянні з аналогами (наприклад, MyBatis), Hibernate відразу пропонує готове рішення, початкові вимоги до конфігурування якого мінімальні [18].

Взаємодія між кодом інформаційної системи та функціональними можливостями Hibernate може відбуватися через стандартний інтерфейс доступу JPA, а також через розширений інтерфейс Hibernate, який надає додаткові можливості.

Загальну схему взаємодії рівня доступу до даних інформаційної системи з реляційною базою даних з використанням Hibernate зображено на рис. 5.8.

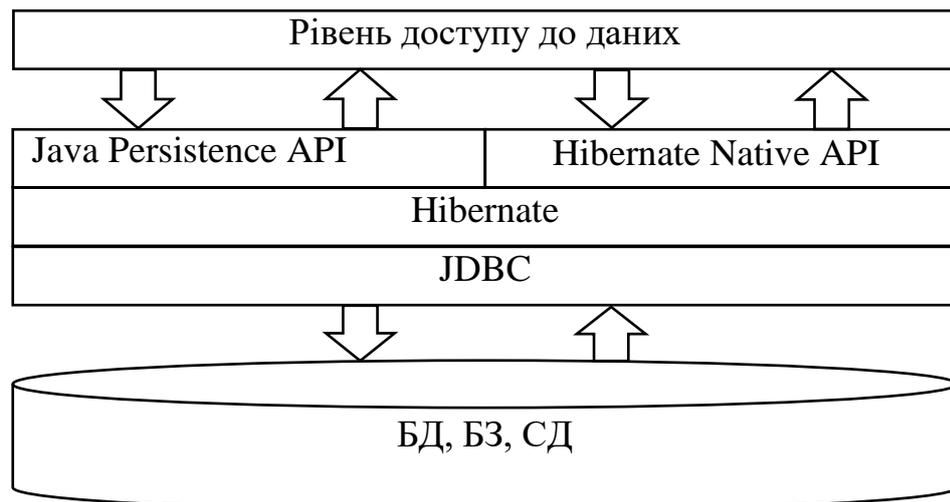


Рисунок 5.8 – Схема взаємодії рівня доступу до даних інформаційної системи з базою та сховищем даних із використанням Hibernate

Взаємодія серверної частини з клієнтською здійснюється мережею Інтернет з використанням ідеології REST (Representational State Transfer), що дозволило

виокремити логіку системи від інтерфейсу користувача, а структуру системи зробити більш простою та масштабованою [345].

REST представляє собою архітектурний стиль для розподілених систем. У відповідності з цим стилем кожний елемент даних однозначно визначається глобальним ідентифікатором таким, як URL (Uniform Resource Locator – уніфікований вказівник на ресурс) [18].

Для формалізованого опису інтерфейсів взаємодії між серверною та користувальницькою частиною була використана специфікація Open API, із застосуванням якої описані усі ресурси, що надаються серверною частиною, та доступні операції над ними.

Фактично дана специфікація визначає стандарт опису REST інтерфейсів, який дозволяє як людині, так і комп'ютеру розуміти можливості серверної частини без доступу до вихідного коду, додаткової технічної документації або прямої взаємодії через мережу [564].

## **5.5 Забезпечення аутентифікації та авторизації користувачів до СППР**

Враховуючи, що СППР є web-орієнтованою, необхідно забезпечити відстеження змін в інформаційних технологіях, які постійно з'являються і швидко змінюються, а з іншого боку, – врахувати реальні характеристики апаратного й програмного забезпечення корпоративних мереж і систем [14].

Оцінкою інформаційної безпеки займаються з початку появи інформаційних технологій [347]. За цією тематикою є багато праць, але найбільш актуальними та фундаментальними з них є нормативні документи, що зробили вагомий теоретичний та практичний внесок у розв'язання задач забезпечення інформаційної безпеки, а саме: «Помаранчева книга» [348], у якій викладені та систематизовані критерії оцінки захисту комп'ютерних систем; Європейські критерії оцінки безпеки інформаційних технологій [349], що врахували усі недоліки та обмеження,

викладені у “Помаранчевій книзі”; Канадські критерії оцінки безпеки надійності комп'ютерних систем [350]; Федеральні критерії США [351], розроблені на замовлення уряду США і спрямовані на усунення обмежень, незручностей практичного застосування і недоліків “Помаранчевої книги”; Міжнародний стандарт ISO/IEC 15408 – “Критерії оцінки безпеки інформаційних технологій” [352-354]; Стандарт SEM-97/017 – “Загальна методологія оцінки безпеки інформаційних технологій” [355]. Окремо необхідно відзначити публікацію [356], у якій розглянуто використання коефіцієнта емерджентності для визначення рівня захисту інформаційних потоків для певного класу архітектури комп'ютерної мережі [1].

При розробці механізму аутентифікації та авторизації користувачів для web-орієнтованої системи важливо взяти до уваги цілу низку концептуальних помилок, які часто допускаються розробниками і дозволять спростити несанкціоноване отримання даних облікового запису. Таким чином, до основних складових частин забезпечення захисту системи, крім механізму аутентифікації та авторизації, також належать: контроль складності паролів; використання підходів захисту від підбору паролів; передача паролів тільки через захищене з'єднання; використання хеш-функції шифрування при зберіганні паролів; надання користувачам можливості зміни паролів та вчасне повідомлення про його зміну; вимога повторної аутентифікації користувача при здійсненні важливих дій, таких як зміна пароля, адреси тощо [14].

Для побудови модуля безпеки розглядаються такі платформи: JAAS, Apache Shiro, Spring Security.

JAAS (Java Authentication and Authorization Service) є стандартною програмною платформою Java SE, що інтегрована безпосередньо у Java Development Kit і призначена для реалізації функцій аутентифікації та авторизації користувачів. Підтримується аутентифікація на основі стандартних протоколів та сертифікатів. Оскільки JAAS є вбудованою платформою з низькорівневим API (Application Programming Interface) інтерфейсом прикладного програмування, а стандартні засоби інтеграції з сервлетами та Java EE (Enterprise Edition) – платформою для розробки корпоративного програмного забезпечення з додатками є відсутні, то це

обмежує можливість її ефективного використання для створення складних промислових систем. Необхідно зазначити, що побудова модуля безпеки виключно на основі JAAS може тривати доволі довго та вимагатиме підключення додаткових програмних платформ і бібліотек. Цю платформу доцільно використовувати у системах, де відсутня складна бізнес-логіка. Крім того, JAAS доцільно використовувати як базовий захист на примітивному рівні, а побудову модуля захисту здійснити на іншій програмній платформі [10].

Apache Shiro є програмною платформою, що дає змогу розробнику забезпечити розподілений доступ до інформації, компонентів системи та інформаційних джерел, і який підтримує різні варіанти зберігання аутентифікаційних даних, а також роботу через різні мережеві протоколи прикладного рівня [10, 14, 357]. Ця платформа забезпечує реалізацію функцій аутентифікації, ідентифікації, керування подіями та контролем доступу, управління сесіями, використання криптографічних функцій тощо. До основних недоліків платформи належить ряд складнощів, які можуть виникати при інтеграції з системою, створеною з використанням програмної платформи Spring; під час використання протоколу авторизації OAuth, що уможливорює відкриття обмеженого доступу до захищених ресурсів без необхідності передачі аутентифікаційних даних користувача, якому вони належать [10, 14].

Spring Security – це програмна платформа, до складу якої входять гнучкі та потужні механізми для організації безпеки будь-якого рівня складності [10, 358-360]. Ця платформа дає можливість реалізувати аутентифікацію та авторизацію, підтримує низку протоколів і способів їх налаштування, дає змогу створити конфігурацію з використанням таких підходів: опис у XML-файлі або створення Java-класів із власною логікою обробки. Spring Security дає можливість реалізувати потужну модель безпеки без будь-якого впливу на бізнес-логіку системи, а також платформа легко адаптується до специфічних вимог [360]. Це досягається за рахунок використання підходів аспектноорієнтованого програмування, які дають змогу здійснювати додаткові операції під час виконання конкретних методів чи модулів, не змінюючи при цьому їх функціонала. Отже, є можливість управляти

усіма етапами виконання транзакцій та операціями, пов'язаними з безпекою. Крім того, надаються механізми контролю усіх, навіть дрібних операцій, що виконуються, а при необхідності існує можливість здійснювати додаткову перевірку легітимності дій з метою захисту даних. Доступ для користувачів надається не тільки на основі ролей, а також із використанням списку контролю доступу (Access Control List) [10].

Spring Security надає можливість використовувати такі технології для забезпечення захисту: x.509 Certificates, LDAP, Kerberos, CA Sitemender, HTTP Basic, HTTP Digest та ін. Ця програмна платформа підтримує можливості інверсії управління (Inversion of Control), що включає основні абстрактні принципи та набір рекомендацій для формування слабо зв'язаного програмного коду. Принципи інверсії управління полягають у тому, що кожен компонент системи повинен бути максимально ізольованим від інших, а під час роботи не залежати від конкретної реалізації інших компонентів. У такий спосіб буде забезпечена гнучкість модуля захисту системи. Розробляючи корпоративну систему, важливо звернути увагу на чотири основні проблеми безпеки: аутентифікацію, безпеку web-запитів, безпеку сервісного рівня системи, а саме методів, що забезпечують безпосередню реалізацію бізнес-логіки, безпеку об'єктів доменного рівня (різні доменні об'єкти з різними рівнями доступу). Специфікація сервлетів надає підхід до перевірки аутентифікаційних даних. Проте для цього необхідно здійснювати специфічне налаштування контейнера сервлетів та редагування його параметрів, що унеможливорює універсальність конфігурації, особливо при необхідності створення власного Java-класу для реалізації інтерфейсу аутентифікації. Spring Security дає змогу досягати повної незалежності конфігурації навіть безпосередньо на рівні WAR (Web Application Archive – формат файлу, що описує, як саме web-додаток запаковується до специфікації Java-сервлетів у файл формату JAR або ZIP) [10, 357-360].

Крім того, Spring Security має кілька дієвих механізмів аутентифікації, що дають змогу обирати підхід до аутентифікації у різних зовнішніх середовищах під час розгортання додатка. У специфікації сервлетів передбачена реалізація функцій

захисту URI (Uniform Resource Identifier – уніфікований ідентифікатор ресурсу) запиту. За специфікацією URI може бути представлений лише в обмеженому форматі, а тому будуть використані можливості Spring Security, які забезпечать задання шляхів із використанням шаблону Ant або регулярних виразів, розгляд окремих частин URI (наприклад, HTTP GET-параметри), реалізацію власного джерела конфігурування даних, що дає змогу динамічно змінювати web-запит під час фактичного використання web-додатка. Відсутність підтримки механізмів захисту безпеки сервісного рівня системи та доменних об'єктів у стандартній специфікації сервлетів призводить до доволі серйозних обмежень під час розробки комплексних систем. Як правило, розробники або ігнорують ці вимоги, або реалізують логіку безпеки у коді MVC-контролера і навіть всередині представлень [10, 358-360].

Такий підхід має кілька значних недоліків [10]:

- авторизація є загальною потребою для безпеки додатка, а об'єднання логіки авторизації з MVC-контролерами або представленнями призводять до ускладнень під час модульного тестування та налагодження, а також до необхідності дублювання коду, що, у свою чергу, порушує SOLID (S – single responsibility, принцип єдиного обов'язку; O – open-closed, принцип відкритості/закритості; L – Liskov substitution, принцип підстановки Барбари Лісков; I – interface segregation, принцип розділення інтерфейсу; D – dependency inversion, принцип інверсії залежностей) принципи об'єктно-орієнтованого програмування та негативно впливає на модульність і загальну архітектуру системи;

- у випадку необхідності створення нового типу клієнтського додатка виникає проблема неможливості повторного використання коду авторизації, вбудованого на web-рівні.

Отже, логіку авторизації доцільно вводити саме на сервісному рівні для підтримки різних типів клієнтських додатків. Під час створення простих додатків специфікації безпеки сервлетів може бути достатньо. Проте, якщо взяти до уваги необхідність в універсальності конфігурації, обмеженість роботи з web-запитами, відсутність можливості захисту на сервісному рівні та захисту доменних об'єктів,

виникає актуальна потреба у використанні стороннього альтернативного рішення. Для конфігурування безпеки доступу з урахуванням особливостей предметної області будуть використані елементи простору імен Spring Security, які доцільно подати такими групами функцій для відображення їх основних можливостей [10, 360]:

- Web/HTTP Security – забезпечує встановлення фільтрів, пов'язаних із сервісними компонентами, захист методів доступу до мережевих ресурсів, роботу аутентифікаційних механізмів програмної платформи, відображення сторінок помилок, аутентифікації тощо;
- BusinessObject (Method) Security – відповідає за захист усіх рівнів сервісів;
- AuthenticationManager – здійснює обробку аутентифікаційних запитів від внутрішніх компонентів програмної платформи;
- AccessDecisionManager – забезпечує засоби доступу до web-сторінок та методів;
- AuthenticationProvider – забезпечує роботу менеджера аутентифікації користувачів та управління ідентифікаторами користувачів;
- UserDetailsService – використовується програмною платформою як репозиторій користувачів, а також як основа стратегії доступу для AuthenticationProvider.

Під час реалізації модуля безпеки системи доцільно розглянути стандартний сценарій аутентифікації користувача і дослідити переваги, які для цього процесу надає Spring Security, з метою їх ефективного застосування. Стандартний сценарій складається з таких кроків [10, 359, 360]:

- користувач намагається увійти у систему з певними даними для авторизації;
- система перевіряє коректність введених даних та відповідність паролю імені користувача;
- у випадку успішної перевірки система отримує контекст із інформацією про користувача, його роль тощо;
- цей контекст встановлюється відповідно до користувача;

- на основі встановленого контексту механізм контролю доступу надає користувачу можливість здійснювати визначений перелік дозволених операцій.

Spring Security визначає такі корективи для стандартного механізму аутентифікації: отримані ім'я користувача та пароль комбінуються в екземпляр класу `UsernamePasswordAuthenticationToken`; цей об'єкт передається до `AuthenticationManager` для проведення валідації; у випадку успішності цієї процедури контекст безпеки стає викликом методу `SecurityContextHolder.getContext().setAuthentication(...)` та передається в середині об'єкта аутентифікації [10, 360].

Також важливо зазначити деякі функції управління сесіями [10, 359]:

- визначення перевищення ліміту відведеного часу підключення для однієї сесії та перехід на потрібний ресурс;
- управління паралельним доступом для одного й того ж користувача, що дає можливість кожному користувачу налаштувати такий доступ;
- автоматичний захист від атаки “виправлення сесії”, коли сесії створюються злоумисником, а потім відбуваються маніпуляції для входу легального користувача під цією сесією з метою перехоплення посилання, що містить ідентифікатор сесії та інші дані.

Spring Security підтримує функції роботи з паролями та їх зберігання у зашифрованому вигляді з використанням алгоритму хешування, що представляє собою односторонню функцію перетворення деяких вхідних даних довільної довжини у вихідні дані фіксованої довжини. Зашифровані паролі зберігаються у базі даних системи, що мінімізує можливість їх викриття. Зламати подібний шифрований пароль можна лише завдяки повному перебору всіх можливих варіантів. Для підвищення надійності шифрування паролів Spring Security надає можливість використовувати додатковий рядок унікальних даних для кожного окремого користувача під час виконання алгоритму шифрування, що значно збільшує час, необхідний для розшифрування такого пароля [10, 359, 360].

Але основними компонентами захисту є розробка системи аутентифікації та авторизації користувачів, а також використання рефакторингу доступу при розробці елементів СППР [14].

Основним методом аутентифікації в сучасних web-додатках є використання файлів cookie, що вміщують ідентифікатор сесії на сервері. При цьому сесія має термін дії, який автоматично збільшується при зверненні користувача на сервер [14, 361].

Аутентифікація на основі серверної сесії та cookie відноситься до групи підходів, які зберігають стан взаємодії між клієнтом і сервером, а також дані, які між ними передаються. При використанні такого підходу дані про сесію зберігаються як на клієнтській, так і на серверній частині. Перелік активних сесій зберігається на сервері (наприклад, у базі даних), а на клієнтській частині створюється файл cookie, що містить ідентифікатор активної сесії [14].

Алгоритм взаємодії зареєстрованого користувача з web-додатком при використанні традиційного підходу у вигляді діаграми послідовності зображений на рис. 5.9.

Алгоритм взаємодії зареєстрованого користувача при використанні традиційного підходу містить наступні кроки [14].

Користувач вводить свої ім'я та пароль на формі для введення клієнтських даних. Введені дані передаються на обробку до модуля аутентифікації та авторизації серверної частини, де здійснюється їх перевірка шляхом звернення до репозиторію даних та пошуку користувача, який повністю відповідає отриманим даним. Якщо користувача з таким іменем та паролем знайдено, створюється та зберігається сесія, ідентифікатор якої передається до клієнтської частини. Дані про створену сесію зберігаються на сервері у вигляді, наприклад, окремої таблиці в базі даних із необхідним набором полів, серед яких ідентифікатор сесії, термін її дії, відповідний ідентифікатор користувача. У браузері клієнта зберігається файл cookie з ідентифікатором створеної на сервері сесії. Зазвичай використовується стандартне ім'я такого файлу – JSESSIONID [14, 361].

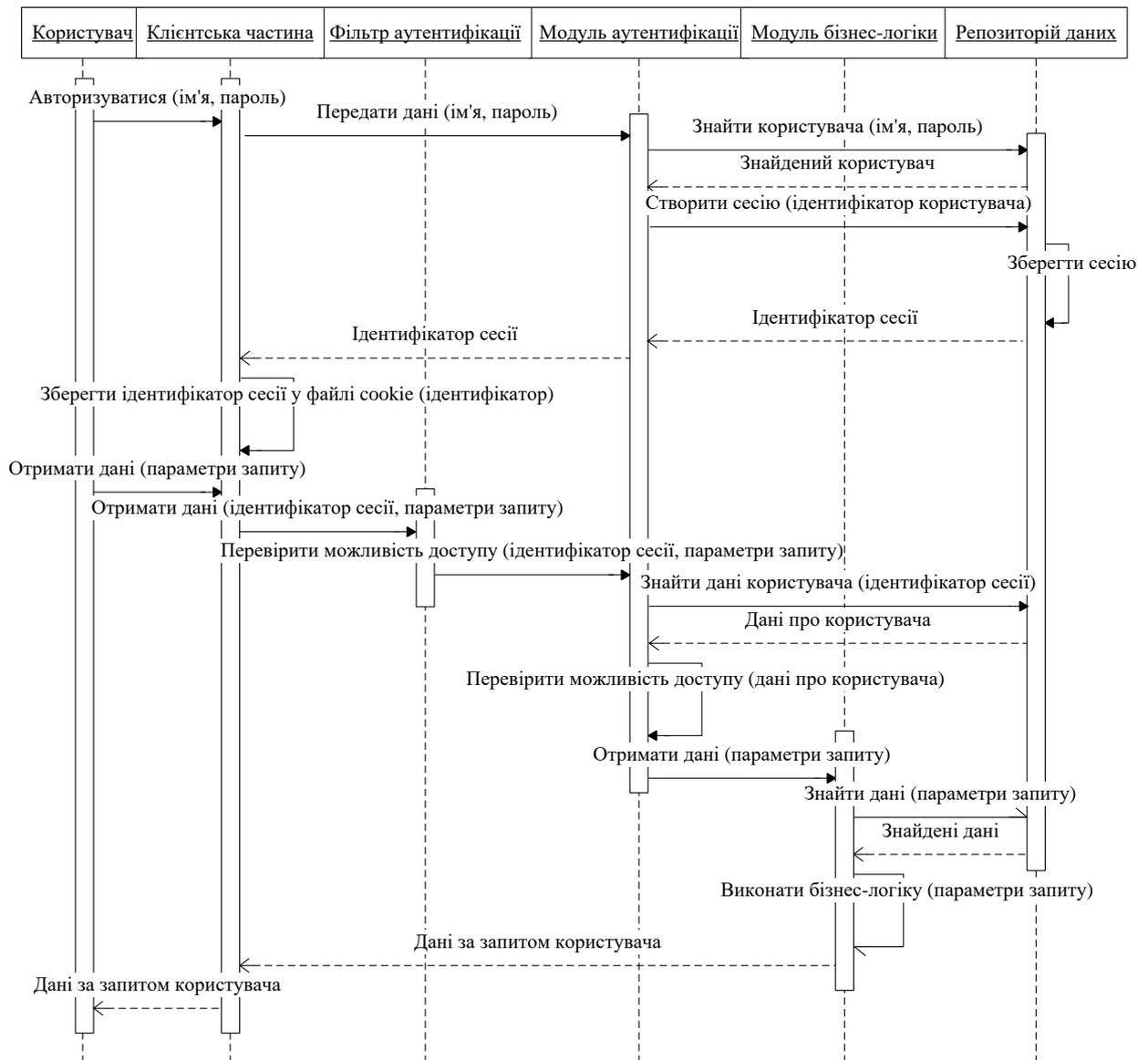


Рисунок 5.9 – Алгоритм взаємодії зареєстрованого користувача з web-додатком при використанні традиційного підходу на основі серверної сесії

При всіх наступних запитах користувача ідентифікатор сесії з файлу cookie відправляється у заголовок запиту до серверної частини, де запит спочатку надходить до фільтра аутентифікації, а далі передається до модуля авторизації та аутентифікації. Саме тут відбувається обробка отриманого ідентифікатора сесії, а саме здійснюється звернення до репозиторія даних із метою пошуку потрібних для визначення можливості доступу до певного ресурсу даних та безпосередньо перевіряється можливість доступу [14].

Якщо перевірка проходить успішно, дані із запиту передаються до модулів бізнес-логіки системи, де здійснюється їх подальша обробка з метою надання користувачу необхідної інформації [14].

Після того, як користувач закінчить роботу з web-додатком, дані про сесію видаляються як на серверній, так і на клієнтській частинах.

JWT-маркер представляє собою текстовий рядок, що містить у зашифрованому вигляді необхідні для аутентифікації та авторизації користувача дані [14, 361].

Основні кроки, які виконувані при роботі користувача з системою за умови використання JWT-маркерів, можна описати таким чином.

Як і при використанні підходу з серверною сесією, користувач здійснює введення своїх імені та паролю на формі для введення клієнтських даних. Аналогічно традиційному підходу, введені дані передаються на обробку до модуля аутентифікації та авторизації серверної частини, де за допомогою звернення до репозиторію даних для пошуку користувача з відповідними даними здійснюється їх перевірка. У випадку успішної перевірки формується JWT-маркер, до інформаційної частини якого вносяться усі необхідні дані (ідентифікатор користувача, його роль у системі, час закінчення дії маркера тощо). Іншими словами, маркер включає дані про користувача та дії, які можуть виконуватися ним у системі, що є необхідним для визначення можливостей доступу при усіх наступних запитах. Сформований маркер надсилається у відповідь та зберігається клієнтською частиною [14].

При кожному наступному запиті користувача отриманий маркер включається до заголовку авторизації (Authorization header) у форматі `Authorization: Bearer` (маркер) та разом із параметрами запиту надсилається до серверної частини. Після отримання такого запиту фільтр аутентифікації передає його до модуля аутентифікації та авторизації, де відбувається перевірка коректності та дійсності маркера на основі його підпису та значень окремих полів, наприклад, поля з терміном дії. Правильність підпису маркера визначається вказаним у заголовку алгоритмом за умови його повного співставлення з алгоритмом, який використовується серверною частиною. Якщо підпис дійсний, тобто, наприклад, він

сформований із використанням певного секретного рядку або за допомогою використаного на сервері відкритого ключа встановлюється автентичність маркера, відбуваються додаткові перевірки, такі як актуальність терміну дії, відповідність ролі користувача можливості звернення до певного ресурсу тощо. Іншими словами, успішне проходження перевірки дозволяє подальшу обробку запиту. Модулі, які реалізують бізнес-логіку системи, отримують дані із запиту та виконують необхідну обробку. Отримані в результаті дані повертаються у відповідь до клієнтської частини, де вони у відповідному вигляді відображаються для користувача [14, 361].

Основною перевагою використання JWT-маркерів є відсутність необхідності зберігання будь-яких даних про маркери чи самі маркери на сервері. Кожен маркер включає всі необхідні дані для перевірки його достовірності, а також дає можливість передавати довільний набір даних. Серверна частина забезпечує формування, підпис та видачу JWT-маркерів, перевірку дійсності маркерів у вхідних запитах [14, 361].

На рис. 5.10 відображено порядок взаємодії зареєстрованого користувача з web-додатком при використанні JWT-маркера на діаграмі послідовності [14].

Недоліком використання маркерів є необхідність обмеження довжини заголовку у запиті для деяких серверів. У таких випадках у інформаційну частину маркера варто включати лише мінімально необхідні дані. Крім цього, маркер містить у собі термін дії, який, відповідно до вимог безпеки, повинен бути не дуже тривалим, а після його закінчення необхідно сформувати новий маркер [14].

На сьогоднішній день існує значна кількість готових реалізацій та бібліотек для роботи з JWT-маркерами з метою розробки механізму аутентифікації та авторизації системи на їх основі. При використанні мови програмування Java для розробки серверної частини системи, на прикладі системи підтримки прийняття рішень, досить зручною у використанні та доступною для розуміння є бібліотека Nimbus JOSE + JWT [14, 361].

Враховуючи, що в структурі присутні база знань, вітрини та сховище даних, при їх створенні та удосконаленні необхідно застосовувати категорію «рефакторинг доступу», що містить у собі операції, які дають можливість контролювано

виконувати зміни у частині обмеження доступу при адмініструванні системи управління базою даних, створювати регламентовані процеси реакції на події, пов'язані з політикою безпеки [17, 20, 24]. Автором категорії «рефакторинг доступу» є Струзік В. А., який виклав її у роботах [24, 311].

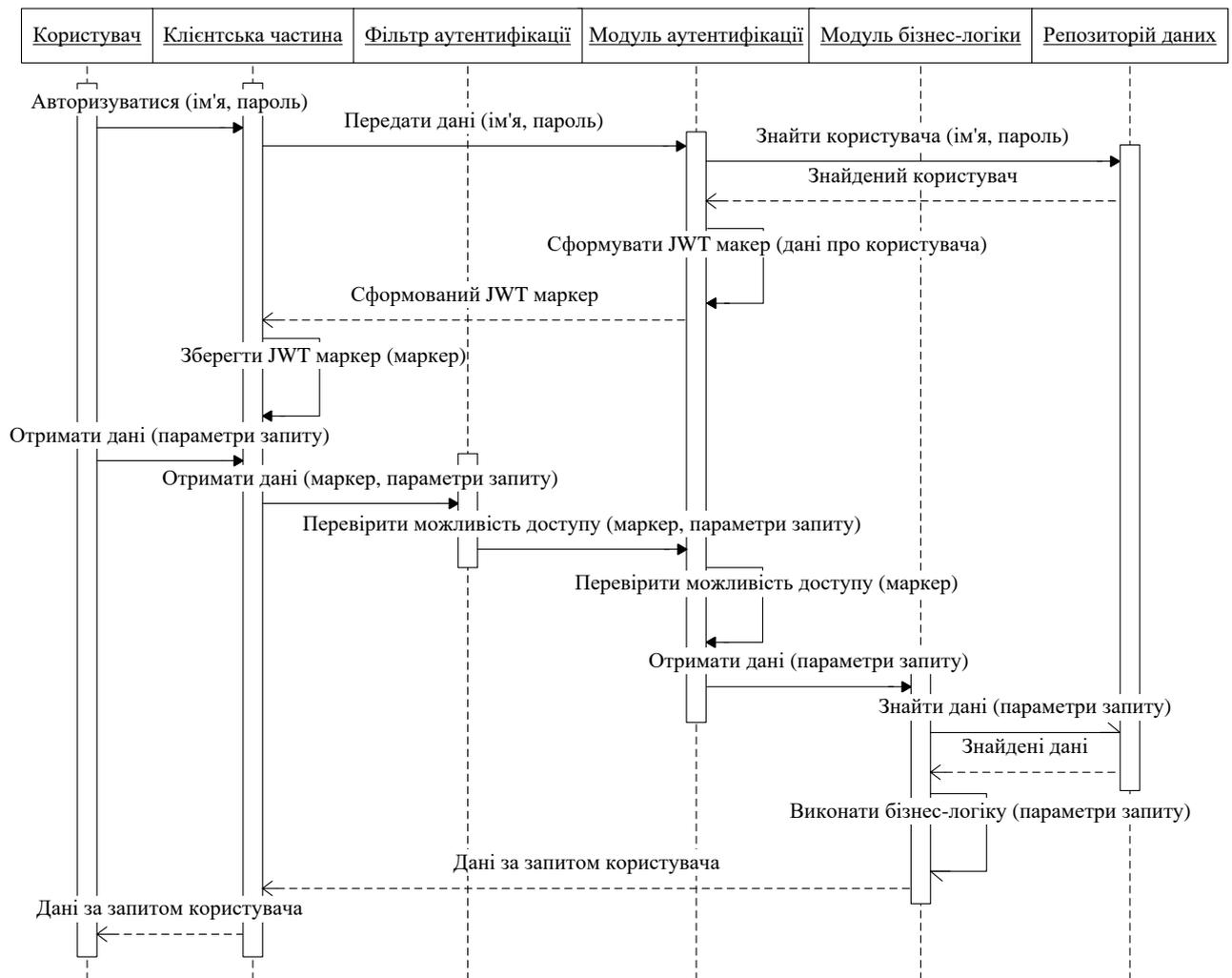


Рисунок 5.10 – Алгоритм взаємодії зареєстрованого користувача з web-додатком при використанні JWT-маркера

До категорії «рефакторинг доступу» належать такі операції [24]:

- зміна атрибутів автентифікації;
- звуження привілеїв доступу;
- розширення привілеїв доступу;

- виділення схеми бази даних;
- злиття схем баз даних.

Метою використання операції «зміна атрибутів автентифікації» є зміна способів або атрибутів автентифікації. Основними причинами застосування цієї операції є компрометація пароля користувача або планова ротація паролів [24].

Процес застосування операції «зміна атрибутів автентифікації» складається з чотирьох етапів проведення операції рефакторингу [24, 311]:

1. Виділяємо елемент «вітрини даних», з якою працює певний додаток, та користувача, який має доступ до цієї «вітрини даних».
2. Створюємо нового користувача.
3. Змінюємо користувача «вітрини даних» у тестовому та виробничому середовищі. Проводимо аудит «вітрини даних» на предмет наявності автентифікації за атрибутами старого користувача.
4. Видаляємо старого користувача.

При використанні операції «зміна атрибутів автентифікації» категорії «рефакторинг доступу» не втрачається працездатність інформаційної системи та всі запити клієнтів будуть успішно оброблені. Метою впровадження операції «звуження привілеїв доступу» є скасування привілеїв доступу до об'єктів бази даних, що на момент впровадження операції рефакторингу були надані [24, 311].

Процес застосування операції «звуження привілеїв доступу» складається з таких кроків проведення операції рефакторингу [24, 311]:

1. Виділяємо вітрину даних з об'єктами, до яких має доступ певний користувач.
2. Створюємо нового користувача.
3. Новий користувач застосовується в місцях звуження привілеїв. Проводиться аудит вітрини даних на предмет наявності авторизації старого користувача за привілеями, які підпали під звуження.
4. Видалення у старого користувача привілеїв, які підпали під звуження.

При звуженні привілеїв доступу до вітрини чи сховища даних розробник може зіштовхнутися з тим, що, вилучивши привілеї у певних користувачів, залишається

частина сховища або «вітрини даних», доступ користувачами до яких взагалі відсутній.

Використання операції «звуження привілеїв доступу» категорії «рефакторинг доступу» дає змогу уникнути такої ситуації. Адже спочатку створюється новий користувач, який застосовується в місцях звуження привілеїв. Таким чином, всі запити можуть бути оброблені [24, 311].

Операція «розширення привілеїв доступу» впроваджується для надання існуючому користувачу прав доступу до об'єктів сховища та вітрин даних, які йому недоступні. Найчастіше використовується після зміни схеми бази даних, що впроваджувалася для доповнення або додавання до відображення в реляційній базі даних сутностей реального світу [24, 311].

Процес застосування операції «розширення привілеїв доступу» складається із двох етапів [24, 311]:

1. Визначається частина сховища або вітрина даних з об'єктами, доступ до яких необхідно отримати певному користувачу.
2. Користувачу надають необхідні привілеї доступу.

При розширенні привілеїв доступу розробнику необхідно в першу чергу розробити оптимальну послідовність виконання дій для досягнення бажаного результату. Таким чином, витрачаються людино-години для розроблення плану дій, що може бути досить різним в залежності від досвіду та кваліфікації розробника [24, 311].

Метою впровадження «виділення схеми бази даних» є виокремлення групи таблиць, представлень та збережуваних процедур, що їх обслуговують (сегмента даних), до окремої структури. Процес застосування операції «виділення схеми бази даних» складається з шести етапів для виділення схеми сховища даних [24, 311]:

1. Визначити сегмент даних у сховищі.
2. Видалити зв'язки між елементами, що переміщуються.
3. Створити нового користувача та перевірити працездатність інформаційної системи у тестовому середовищі. Провести аудит сховища даних на предмет звернень до виокремлених даних від імені старого користувача.

4. Створити нову вітрину даних та виконати міграцію даних.

5. Замінити нового користувача за старою схемою на користувача нової структури.

6. Видалити виокремлений сегмент сховища.

Це можливо зробити з урахуванням того, що сховище даних складається з віртуального сховища та окремих вітрин.

При використанні операції «виділення схеми бази даних» категорії «рефакторинг доступу» працездатність інформаційної системи зберігається так само, як і доступ до всіх елементів сховища даних, в яке вносяться зміни. Тобто система залишається працездатною.

Операція «злиття схем баз даних» націлена на злиття різних вітрин даних в єдину з метою спрощення реалізації бізнес-транзакцій та забезпечення розв'язання задач. Процес застосування операції «злиття схем баз даних» складається з п'яти кроків [311]:

1. Виділяються дві вітрини, які необхідно поєднати.

2. Проводиться уніфікація назв об'єктів вітрин даних, що зливаються.

3. Визначається головна вітрина даних з усіх, які будуть поєднуватися, і здійснюється міграція даних.

4. Замінюється користувач другорядної вітрини даних на відповідного користувача головної вітрини даних. Перевіряється працездатність системи у тестовому середовищі. Здійснюється аудит вітрини даних на предмет звернень до даних від імені користувачів другорядної вітрини даних.

5. Видаляються другорядні вітрини даних.

При виникненні бізнес-потреби злиття декількох вітрин даних, втрачається доступ до кожної з вітрин даних, що зливаються.

При використанні операції «злиття схеми баз даних» категорії «рефакторинг доступу» працездатність інформаційної системи зберігається так само, як і доступ до всіх елементів вітрини даних, в які вносяться зміни. Таким чином, всі запити можуть бути оброблені.

Використання рефакторингу в цілому сприяє підвищенню швидкості розробки системи та якості програмного продукту незалежно від кваліфікації команди розробників, що було доведено у працях [24, 311].

Застосування категорії «рефакторинг доступу» при розробці баз даних підтверджено на основі отриманих результатів від розробників програмного забезпечення. Було встановлено, що використання операцій рефакторингу доступу забезпечує [24, 311]:

- безперервну роботу інформаційної системи під час впровадження змін, пов'язаних із доступом до даних, при адмініструванні системи управління базою даних;
- відсутність сервісного вікна для впровадження операцій, які пов'язані з політикою безпеки;
- відсутність необхідності прийняття рішень при впровадженні змін у частині обмеження доступу при адмініструванні системи управління базою даних.

## **5.6. Результати впровадження досліджень**

Інтерфейс користувача реалізований у вигляді web-додатка. На даний момент реалізовані тільки основні функції СППР.

Система має засоби табличного, текстового, графічного відображення даних. Інтерфейс користувача відповідає принципам і стандартам GUI (Graphical User's Interface). За рахунок використання web-інтерфейсу користувач має змогу отримати доступ до системи з будь-якого джерела.

Розглянемо основні можливості створеної web-орієнтованої СППР для планування виконання замовлень на виготовлення продукції.

Вкладка «Планування виконання замовлень» (рис. 5.11) забезпечує можливість для перегляду або редагування існуючих календарних планів, а також планування виконання робіт на майбутні періоди.

СППР планування виконання замовлень

Грибков С.В. [Вихід](#)

Планування виконання замовлень | Контроль виконання | Технологічне обладнання

Період планування: від  до:

Поточний план | Замовлення що надійшли | Виконанні замовлення

№	Номер замовлення	Продукції	Замовник	Від	До	Штраф	Обсяг (кг)	Пріоритет
1	1301	Продукція 1	ТОВ "Підприємство 1"	02/10/2020	05/10/2020	Так	110	40
2	1302	Продукція 2	ТОВ "Підприємство 2"	02/10/2020	05/10/2020	Так	200	30
3	1303	Продукція 3	ТОВ "Підприємство 3"	04/10/2020	08/10/2020	Ні	160	30
4	1304	Продукція 4	ТОВ "Підприємство 2"	03/10/2020	07/10/2020	Так	320	0
5	1305	Продукція 5	ТОВ "Підприємство 1"	05/10/2020	08/10/2020	Ні	90	0

Previous | 1 | 2 | 3 | 4 | 5 | Next

К-ть на стор.

Параметри пошуку плану | Оберіть метод пошуку: "згряя вовків" | [Сформувати](#)

Реконфігурувати поточний  діалоговий режим

Рисунок 5.11 – Вкладка «Планування виконання замовлень»

Після введення початкової та кінцевої дати планового періоду формується перелік замовлень, що необхідно виконати за встановлений період.

Такий перелік представлено у табличному вигляді з короткими відомостями про кожне замовлення (порядковий номер, номер замовлення, продукція, замовник, термін виконання від і до, чи є штрафні санкції за невчасне виконання, пріоритет).

Для кожного замовлення є можливість переглянути детальну інформацію, а також переглянути, яке саме обладнання необхідне для його виконання, який набір сировини необхідний та його обсяги для виконання, терміни відвантаження.

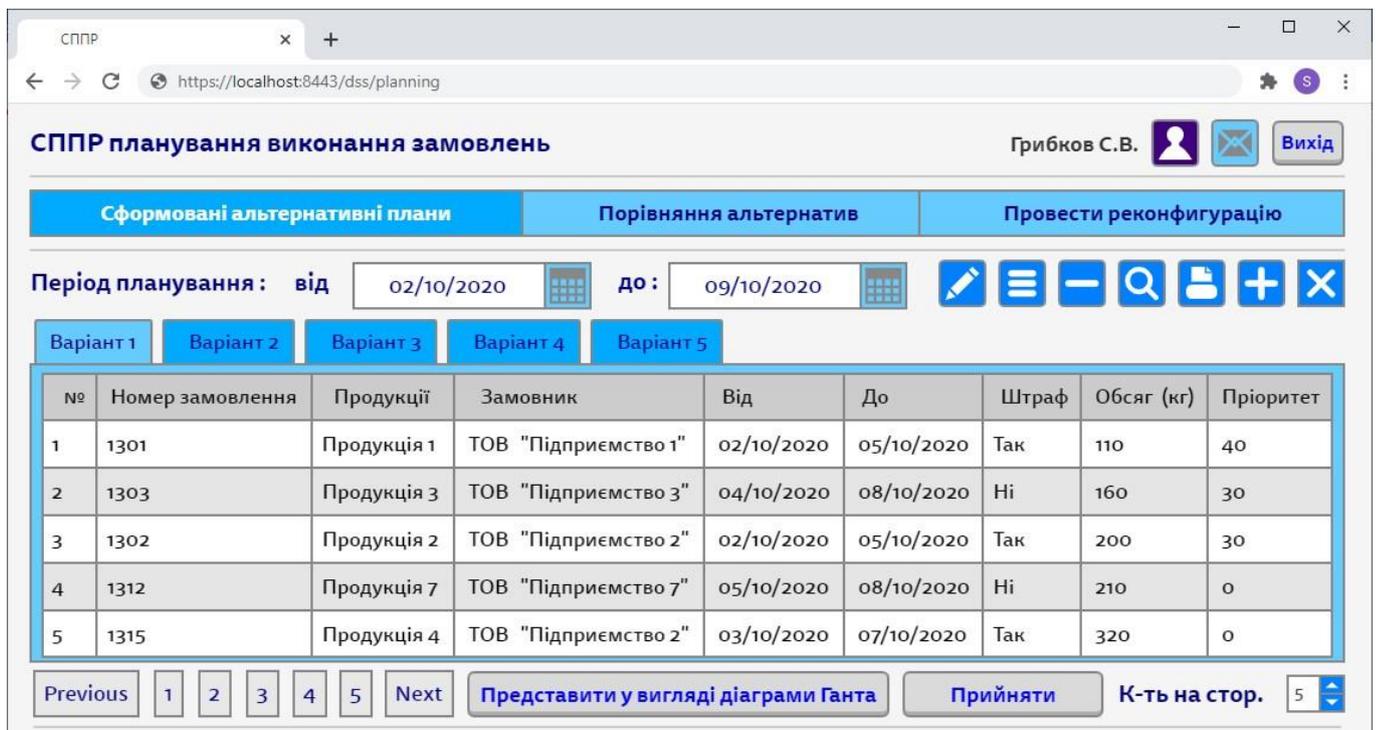
До сформованого переліку замовлень можливо включити інші замовлення або видалити наявні. Для кожного замовлення можна встановити пріоритет виконання у вигляді числа від 1 до 100. За замовчанням встановлене значення пріоритету становить 1 (найменше). При розрахунку критерію ефективності введені пріоритети перераховуються за формулою (5.6) [18]:

$$u_i = \frac{v_i}{\sum_i v_i}, \quad (5.6)$$

де  $v_i$  – введене користувачем значення пріоритету для кожного договору.

При натисканні кнопки «Параметри»  з'являється вікно, в якому є можливість задати параметри пошуку, а саме обрати, які критерії враховувати при пошуку оптимального плану, який метод буде використовуватися для пошуку, які дані використовувати для аналізу можливих варіантів, враховувати постачання сировини чи враховувати необхідність відвантаження тощо.

Після натискання кнопки «Сформувати» на екрані буде відображено перелік альтернативних планів виконання замовлень (рис. 5.12).



СППР планування виконання замовлень Грибков С.В.   [Вихід](#)

Сформовані альтернативні плани Порівняння альтернатив Провести реконфігурацію

Період планування: від  до:          

Варіант 1 Варіант 2 Варіант 3 Варіант 4 Варіант 5

№	Номер замовлення	Продукції	Замовник	Від	До	Штраф	Обсяг (кг)	Пріоритет
1	1301	Продукція 1	ТОВ "Підприємство 1"	02/10/2020	05/10/2020	Так	110	40
2	1303	Продукція 3	ТОВ "Підприємство 3"	04/10/2020	08/10/2020	Ні	160	30
3	1302	Продукція 2	ТОВ "Підприємство 2"	02/10/2020	05/10/2020	Так	200	30
4	1312	Продукція 7	ТОВ "Підприємство 7"	05/10/2020	08/10/2020	Ні	210	0
5	1315	Продукція 4	ТОВ "Підприємство 2"	03/10/2020	07/10/2020	Так	320	0

Previous      Next [Представити у вигляді діаграми Ганта](#) [Прийняти](#) К-ть на стор.

Рисунок 5.12 – Сформовані альтернативні плани

Візуальне представлення варіанта виконання замовлення є таблиця, яка відображає послідовність виконання замовлень за заданий проміжок часу.

На кожному етапі відбувається перевірка та удосконалення певних частин плану. Особа, що складає план, у діалогову режимі намагається досягти оптимального плану виконання договорів.

Є можливість обраний план передати й представити у вигляді діаграми Ганта у Microsoft Excel. Приклад сформованого плану випуску продукції менеджером та системою представлені на рис. 5.13 та 5.14.



Рисунок 5.13 – План, сформований менеджером



Рисунок 5.14 – План, запропонований СППР

Із графіка видно, що сформований план економить загальний час, загальне виконання замовлень і здійснюється на 1 годину раніше, а це забезпечує економію коштів при експлуатації технологічного обладнання та заробітну плату.

Якщо варіантів буде декілька, є можливість порівняти обрані плани з використанням пелюсткової діаграми (рис. 5.15).

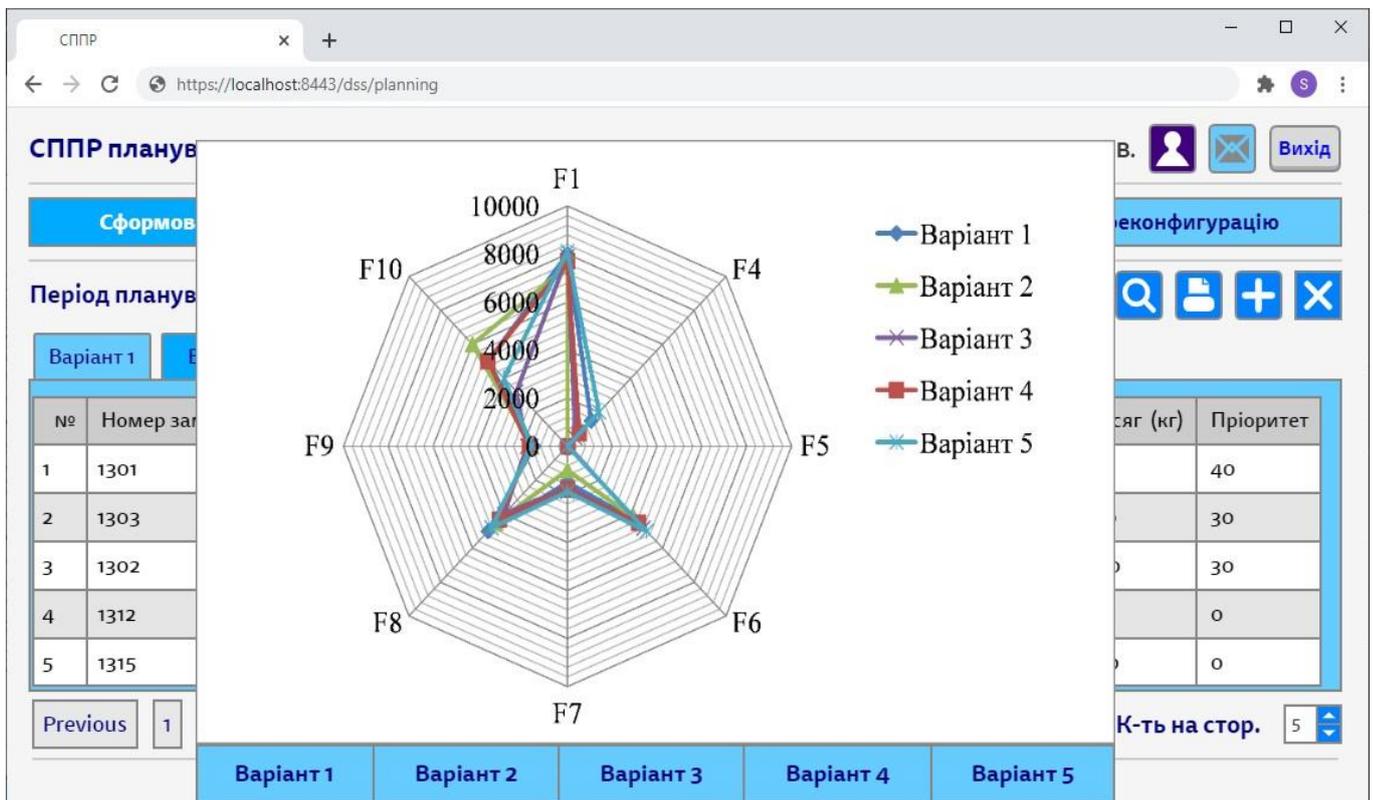


Рисунок 5.15 – Приклад порівняння п'яти варіантів планів на пелюстковій діаграмі

Вкладка «Контроль виконання» (рис. 5.16) дозволяє переглянути, які замовлення були виконані за певний період.

За допомогою кнопки «Завантажити» автоматично сформується звіт відповідно до обраних фільтрів та буде експортований у Microsoft Word.

СППР планування виконання замовлень

Грибков С.В. [Вихід](#)

Планування виконання замовлень | **Контроль виконання** | Технологічне обладнання

Період планування: від  до:

Цех 1 | Цех 2 | Цех 3 | Зміна:  | Зміна:

№	Номер замовлення	Продукції	Замовник	Від	До	Штраф	Обсяг (кг)	Пріоритет
1	1275	Продукція 6	ТОВ "Підприємство 7"	25/09/2020	01/10/2020	Ні	170	o
2	1204	Продукція 3	ТОВ "Підприємство 4"	25/09/2020	02/10/2020	Ні	210	o
3	1300	Продукція 4	ТОВ "Підприємство 6"	27/10/2020	01/10/2020	Так	210	o
4	1299	Продукція 6	ТОВ "Підприємство 7"	28/10/2020	01/10/2020	Так	210	o
5	1290	Продукція 3	ТОВ "Підприємство 8"	29/10/2020	01/10/2020	Так	120	o

Previous | 1 | 2 | 3 | 4 | 5 | Next | [Представити у вигляді діаграми Ганта](#) | [Завантажити](#) | К-ть на стор.

Рисунок 5.16 – Вкладка «Контроль виконання»

При використанні СППР відмічено скорочення часу реконфігурації оперативно-календарних планів за короткі проміжки часу при виникненні термінових замовлень та позаштатних ситуацій. Виникнення позаштатних ситуацій обумовлено в перебоях постачання сировини та матеріалів, а також при термінових ремонтно-профілактичних роботах на технологічному обладнанні.

Використання гібридних web-орієнтованих систем підтримки прийняття рішень, в основі яких лежить запропонована інформаційна технологія з сукупністю модифікованих методів та алгоритмів, для вирішення задач управління харчовими підприємствами дає можливість вивести весь процес управління на якісно новий рівень.

Використання такої СППР забезпечить виконання замовлень із такими перевагами:

- оперативно формує оперативно-календарний план виконання замовлень із мінімізацією витрат та направлений на максимізацію прибутку;

- дозволяє оперативно коригувати існуючий календарний план замовлень, що дає можливість реагувати на замовлення в реальному часі та забезпечувати оптимальне використання технологічного обладнання;
- значно збільшує ефективність використання сировини та матеріалів, а також забезпечує мінімізацію витрат на їх зберігання;
- забезпечує швидке реагування при виникненні негативних та позаштатних ситуацій шляхом внесення відповідних змін до поточного плану виконання замовлень;
- чітко розподіляє усі задачі щодо виконання кожного замовлення між виробничими підрозділами, що дає можливість врахувати послідовність виконання та необхідні ресурси з прив'язкою у часі;
- дозволяє оптимізувати використання виробничих потужностей.

У 2019 році здійснено апробацію результатів дослідження на ТОВ «Слобожанський бекон». Діяльність цього підприємства повністю охоплює технологічний процес виготовленням м'ясних та ковбасних виробів, а також напівфабрикатів, починається безпосередньо від постачання сировини й закінчується відвантаженням готової продукції замовнику. ТОВ «Слобожанський бекон» здійснює найбільш повний набір основних функцій, що притаманні для більшості підприємств даної галузі, а саме:

1. Утримання живої худоби з дотриманням усіх міжнародних та українських правил щодо утримання худоби з метою забезпечення підвищення якості цієї худоби для подальшої її переробки.
2. Забій худоби відбувається з використанням сучасного технічного обладнання з метою дотримання технології та вимог до забою худоби.
3. Первинна обробка м'ясної сировини (туш) відбувається згідно з технологічними нормами щодо дотримання визначених операцій стосовно первинної обробки та сортування сировини за видами й цілями подальшого використання сировини у виробництві.
4. Забезпечення якості готової продукції згідно з ДСТУ та ISO з виконанням: контролю якості продукції на всіх етапах виробництва; проведення

лабораторних випробувань на всіх етапах виробництва; формування заключних експертиз щодо якості готової продукції; контроль некондиційної продукції та проведення лабораторних випробувань для прийняття рішення відносно подальшої переробки.

5. Виробництво м'ясних та ковбасних виробів на основі прийнятих замовлень відповідного асортименту м'ясної та ковбасної продукції згідно з вимогами замовників та споживачів із дотриманням технологічних вимог та державних стандартів якості продукції згідно з рецептурами.

6. Виробництво напівфабрикатів, у тому числі сировини для інших виробництв із дотриманням рецептур та технологічних вимог, ДСТУ, а також сортування сировини для подальшого продажу або використання її у виробництві.

7. Переробка м'ясо-ковбасних виробів у випадку їх повернення від замовника, що включає такі операції: визначення можливості переробки продукції, яку повернули; організація переробки згідно з технологічними умовами; утилізація продукції, що не підлягає переробці.

8. Забезпечення зберігання сировини та готової продукції з використанням необхідного обладнання та пристроїв для правильного зберігання продукції для забезпечення її якості.

9. Оптова та роздрібна торгівля готовою продукцією у повному асортиментному обсязі як через власну мережу торгових точок, так і за договорами.

10. Забезпечення відвантаження, доставки та прийому готової продукції, що була повернена замовниками.

За рахунок того, що діяльність підприємства охоплює повний технологічний процес виготовлення м'ясних та ковбасних виробів, а також напівфабрикатів, є можливість стверджувати про високий рівень перевірки апробації на даному підприємстві. На ТОВ «Слобожанський бекон» здійснено апробацію результатів дисертаційного дослідження, а саме інформаційної технології, що включає комбіновані алгоритми і забезпечує:

- формування альтернативних планів виготовлення продукції за короткий період часу;

- швидке реагування на виникнення позаштатних ситуацій;
- зменшення збитків;
- збільшення прибутку, скорочення витрат на виготовлення продукції, оптимальне використання виробничих та складських приміщень при збереженні сировини та готової продукції.

За результатами апробації керівництво ТОВ «Слобожанський бекон» рекомендувало впровадження інформаційної технології, що включає комбіновані алгоритми на інші підприємства харчової галузі. Результати впровадження відображені в акті від 16.12.2019 (Додаток А).

У 2019 році здійснено апробацію результатів дисертаційного дослідження на ТОВ «Продеко», яке займається такими видами діяльності:

- виробництвом хліба та хлібобулочних виробів;
- виробництвом продуктів борошномельно-круп'яної промисловості;
- виробництвом макаронних виробів і подібних борошняних виробів;
- оптовою торгівлею будівельними матеріалами, деревиною та санітарно-технічним обладнанням;
- виробництвом тортів і тістечок, борошняних кондитерських виробів нетривалого зберігання;
- роздрібною торгівлею різноманітними продуктами харчування в спеціалізованих магазинах;
- оптовою торгівлею насінням та зерном, кормами для тварин та необробленим тютюном;
- оптовою торгівлею продуктами харчування, у тому числі ракоподібними, рибою та молюсками;
- наданням послуг перевезення, транспортування та використання вантажного автомобільного транспорту.

Отже, підприємство має широкий спектр діяльності, що вимагає постійного удосконалення методів управління. На підприємстві пройшли апробацію:

- розроблена математична модель планування виконання замовлень, яка враховує основні особливості діяльності підприємств харчової галузі і дозволяє

коригувати й оцінювати ефективність виконання замовлень у залежності від об'єктивних та суб'єктивних переваг, наданих особою, яка приймає рішення, а також забезпечує як урахування, так і виключення певних часткових критеріїв у залежності від певної ситуації;

- розроблена інформаційна технологія планування виконання замовлень забезпечує створення та коригування оперативно-календарних планів за рахунок використання комбінованих методів та алгоритмів.

За результатами апробації керівництво ТОВ «Продеко» рекомендувало впровадження результатів дослідження на інші харчові підприємства. Результати впровадження відображені у акті від 10.12.2019 (Додаток А).

У 2020 році здійснено апробацію науково-практичних результатів дисертаційного дослідження на ТОВ «Козятинський м'ясокомбінат», основною діяльністю якого є виробництво напівфабрикатів та ковбасних виробів, забій великої рогатої худоби та свиней. На даний час підприємство виробляє за місяць понад 150 т (більше, як 50 найменувань) напівфабрикатів, 20 т ковбасних виробів (більше, як 60 найменувань), біля 300 т м'яса. Завдяки своїй стабільній якості та безпечності продукція ТОВ «Козятинський м'ясокомбінат» давно завоювала довіру у споживачів. Протягом багатьох років (починаючи з 2003 року) комбінат є основним постачальником яловичини для виробництва гамбургерів для мережі ресторанів Mc'Donalds, з 2006 року безперервно працює з такими торгівельними мережами, як «МЕТРО», «БІЛЛА» та іншими. Продукція комбінату поставляється у всі області України. Зараз до складу підприємства входять: цех забою та первинної переробки худоби, холодильник, ділянка по обвалюванню м'яса ВРХ та свиней, цех по виробництву м'ясних напівфабрикатів, ковбасний цех, виробнича лабораторія, допоміжні обслуговуючі підрозділи. Основні його види діяльності співпадають із ТОВ «Слобожанський бекон» та притаманні підприємствам даної галузі.

Метою промислової апробації на ТОВ «Козятинський м'ясокомбінат» була практична перевірка ефективності застосування модифікованих методів та алгоритмів, а також інформаційної технології прийняття рішень для розв'язання задачі планування виконання замовлень у виробничих умовах.

Результати перевірки застосування модифікованих методів та алгоритмів у складі інформаційної технології прийняття рішень показали свою ефективність у виробничих умовах, тому керівництво ТОВ «Козятинський м'ясокомбінат» рекомендувало їх до широкого впровадження на підприємствах харчової галузі, що підтвержено актом від 20.10.2020 (Додаток А).

У 2021 році інформаційні технології прийняття рішень для розв'язання задачі планування виконання договорів передано на ТОВ «Лайм Системс». Діяльність даного підприємства не відноситься до харчової промисловості. Основною діяльністю даного підприємства є:

- діяльність у сфері комп'ютерного програмування;
- здійснення ремонту комп'ютерів і периферійного обладнання; надання консультаційних послуг в області комп'ютерних технологій;
- здійснення налаштування управління комп'ютерного обладнання.

Метою промислової апробації була практична реалізація моделей та алгоритмів прийняття рішень для розв'язання задачі планування виконання замовлень і перевірка ефективності застосування створених інформаційних технологій у виробничих умовах. Результати перевірки застосування інформаційних технологій прийняття рішень у виробничих умовах показали ефективне планування виконання замовлень, що обумовило підвищення техніко-економічних показників підприємства.

За результатами апробації керівництво ТОВ «Лайм Системс» вважає, що запропоновані інформаційні технології повністю задовольняють виробничим потребам і можуть бути рекомендовані для широкого впровадження на різних підприємствах. Результати впровадження відображені у відповідному акті від 18.01.2021 (Додаток А).

У 2021 році здійснено апробацію науково-практичних результатів дисертаційного дослідження на ТОВ «Феракс», основною діяльністю якого є виробництво продуктів із м'яса і м'яса свійської птиці. Підприємство є типовим представником своєї галузі, його основні види діяльності співпадають із діяльностями ТОВ «Слобожанський бекон» та ТОВ «Козятинський м'ясокомбінат».

Виробнича потужність підприємства складає до 35 т на добу. Високий рівень автоматизації на ТОВ «Феракс» гарантує найсуворіше дотримання гігієнічних та технологічних норм, а ще привабливий та апетитний вигляд пакованої продукції. Його партнерами є такі торговельні мережі, як «МЕТРО», «Сільпо», Велика Кишеня», «МегаМаркет», «Караван», «Фуршет», «Фора», «ЕКО МАРКЕТ», «Ашан» та ін.

На підприємстві пройшли практичну перевірку програмні модулі, що реалізують запропоновану в дисертації інформаційну технологію прийняття рішень на основі модифікованих методів та алгоритмів для розв'язання задач планування виконання замовлень. Застосування наданої інформаційної технології у виробничих умовах забезпечило ефективне планування виконання замовлень, що обумовило підвищення техніко-економічних показників підприємства. За результатами апробації керівництво ТОВ «Феракс» вважає, що запропонована інформаційна технологія повністю відповідає виробничим потребам та може бути рекомендована до широкого впровадження на підприємствах харчової галузі, що підтверджено актом від 27.01.2021 (Додаток А).

У 2021 році науково-практичні результати дисертаційного дослідження, що полягають у інформаційній технології прийняття рішень для розв'язання задач планування виконання замовлень на основі модифікованих евристичних і еволюційних методів та алгоритмів, передані на ПрАТ «Оболонь». Необхідно відмітити, що корпорація «Оболонь» об'єднує 10 виробничих підприємств по всій Україні, її основними видами діяльності є: виробництво пива, виробництво фруктових і овочевих соків; дистиляція, ректифікація та змішування спиртних напоїв; виробництво солоду; виробництво безалкогольних напоїв; виробництво мінеральних та інших вод, розлитих у пляшки. Продукція корпорації користується попитом як на внутрішньому, так і зовнішньому ринках. Передані результати дисертаційного дослідження пройшли перевірку у виробничих умовах на ПрАТ «Оболонь», підтвердили свою ефективність при плануванні виконання замовлень, використовуються при вирішенні задач управління підприємством. Передана

інформаційна технологія може бути рекомендована для впровадження на підприємствах харчової галузі (акт впровадження 18.02.2021) (Додаток А).

Результати дисертаційного дослідження використовуються для проведення лекційних та лабораторних занять, а також при виконанні курсових та дипломних проєктів на кафедрі інформаційних систем Національного університету харчових технологій. Певні результати використовуються у дисциплінах: «Управління ІТ-проєктами» для здобувачів освіти ступеня бакалавр за спеціальністю 122 – комп’ютерні науки освітньо-професійної програми «Комп’ютерні науки»; «Клієнт-серверні технології» для здобувачів освіти ступеня магістр за спеціальністю 122 – комп’ютерні науки освітньо-професійних програм «Інформаційні управляючі системи та технології» та «Комп’ютерний еколого-економічний моніторинг».

Результати дисертаційного дослідження впроваджено у Національному університеті харчових технологій при виконанні НДР:

- «Наукове обґрунтування та розроблення активних пакувальних систем харчових продуктів» (№ ДР 0118U003558, 2019 р.) Проблемною науково-дослідною лабораторією НУХТ, а саме інформаційна технологія для розв’язання задачі планування виконання замовлень (акт впровадження 29.06.2019);

- НДР «Дослідження та впровадження інформаційних технологій у галузях харчової промисловості та освіти» (№ ДР 0117U003475) 2017-2022 рр. кафедрою інформаційних систем, а саме інформаційна технологія для розв’язання задачі планування виконання замовлень (акт впровадження 29.12.2020).

Акти впровадження у Національному університеті харчових технологій наведено у Додатку А.

Інформаційна технологія прийняття рішень та відповідний програмний продукт передані у Київський міський Радіоклуб Ліги Радіоаматорів України і використовуються для вирішення організаційно-управлінських задач при підготовці та проведенні змагань, що підтверджено довідкою від 26.09.2020 (Додаток А). Необхідно відмітити, що Київський міський Радіоклуб Ліги Радіоаматорів України не здійснює виробничу діяльність, але дана громадська організація займається

змаганнями з радіоспорту. Це вимагає ефективного планування підготовки їх проведення, адже все здійснюється за кошти членів клубу.

## 5.7 Висновки до розділу 5

Запропонована структура web-орієнтованої СППР дає можливість поєднати декілька інформаційних технологій, а саме: підходи експертно-моделюючої системи рецептур виготовлення, що дає змогу підібрати у виробничих умовах заміну компонентів у рецептурі; технологію планування виконання замовлень, що забезпечує створення та коригування оперативно-календарних планів за рахунок використання комбінованих методів та алгоритмів. Ця інформаційна технологія забезпечує формування альтернативних планів виготовлення продукції; використання модифікованих методів; інформаційну технологію для удосконалення процесу аналізу та планування виконання замовлень на харчових підприємствах, яка, на відміну від відомих, ґрунтується на визначенні асортименту та коригуванні рецептур для зменшення собівартості і швидкого виконання замовлень.

У процесі проектування баз знань, вітрин та сховищ даних запропоновано й обґрунтовано використання специфікації «семантичне версіонування баз даних» для маркування і впровадження автоматизованого оновлення її компонентів та елементів.

Обґрунтовано сукупність програмно-технологічних рішень, що входять до моделі web-орієнтованої програмної підсистеми підтримки прийняття рішень, а також підходи реалізації механізмів аутентифікації та авторизації у web-орієнтованих системах.

При створенні елементів СППР обґрунтовано використання моделей життєвих циклів програмного забезпечення, які, на відміну від традиційних, мають чітко визначений етап рефакторингу, що забезпечило ефективність розробки та скорочення часу на її розроблення.

Обґрунтовано застосування до процесу створення і супроводу сховищ та «вітрин даних» категорії рефакторингу доступу, яка дає змогу впроваджувати зміни, пов'язані з доступом до даних, без втрати працездатності інформаційної системи.

Результати дисертаційного дослідження пройшли апробацію та передані на ТОВ «Слобожанський бекон», ТОВ «Продеко», ТОВ «Козятинський м'ясокомбінат», ТОВ «Лайм Системс», ТОВ «Феракс», ПрАТ «Оболонь» для розв'язання задачі планування виконання замовлень, що підтвержено відповідними актами та довідками. За результатами промислової апробації керівництво підприємств надало рекомендації щодо впровадження інформаційної технології на підприємствах харчової галузі.

Результати дисертаційного дослідження впроваджено у навчальний процес Національного університету харчових технологій при викладанні дисципліни «Управління ІТ-проектами» для здобувачів освіти ступеня бакалавр за спеціальністю 122 – комп'ютерні науки освітньо-професійної програми «Комп'ютерні науки», дисципліни «Клієнт-серверні технології» для здобувачів освіти ступеня магістр за спеціальністю 122 – комп'ютерні науки освітньо-професійних програм «Інформаційні управляючі системи та технології» та «Комп'ютерний еколого-економічний моніторинг».

Практичне значення роботи полягає в тому, що розроблені моделі, методи, алгоритми та інформаційна технологія формують основу для створення web-орієнтованих систем підтримки прийняття ефективних рішень при управлінні підприємствами харчової галузі, які дадуть змогу вивести весь процес управління на якісно новий рівень.

Використання СППР, що буде включати запропоновану інформаційну технологію, забезпечить: оперативне формування оперативних планів виконання замовлень із мінімізацією витрат та направлених на максимізацію прибутку; оперативні реконфігурації існуючого календарного плану виконання замовлень, що забезпечить оптимальне використання технологічного обладнання та мінімізацію витрат при виникненні позаштатних ситуацій під час появи термінових до виконання замовлень або позаштатних ситуацій; мінімізацію витрат на зберігання сировини та

матеріалів; чітке розподілення усіх задач щодо виконання кожного замовлення між виробничими підрозділами, це дає можливість урахувати послідовність виконання та необхідні ресурси з прив'язкою у часі.

Наукові висновки та практичні результати, одержані під час проведення досліджень відповідно до теми дисертації, пройшли апробацію на міжнародних науково-практичних конференціях і наукових семінарах і викладені у матеріалах наукових конференцій та наукових статтях у виданнях міжнародного значення.

Основні результати розділу опубліковані автором у працях [1, 2, 10, 14–18, 20–26, 29, 33, 35, 36, 38–44, 46–49, 51–53].

## ВИСНОВКИ

Загальним результатом дисертаційного дослідження є вирішення науково-прикладної проблеми підвищення ефективності процесів управління харчовим підприємством шляхом розробки інформаційних технологій підтримки прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів.

Конкретні одержані результати та висновки полягають у такому.

1. Розроблено математичну модель процесу планування виконання замовлень, яка, на відміну від традиційних, що передбачають розбиття за рівнями управління та підрозділами виконання, забезпечує оцінку ефективності сформованого варіанта плану з урахуванням зовнішніх та внутрішніх впливів на основі запропонованих часткових критеріїв та зазначених обмежень. Математична модель враховує усі умови діяльності харчового підприємства та процесу планування виконання замовлень: терміни зберігання сировини та готової продукції; можливість виникнення та необхідність переробки некондиційної продукції; особливості виконання кожного окремого замовлення; особливості використання технологічного обладнання тощо. Для формування узагальненої цільової функції використовується адитивна згортка критеріїв. Розроблена математична модель дозволяє сформулювати план виконання замовлень із врахуванням усіх операцій технологічного процесу при виготовленні продукції. Створена математична модель дозволяє коригувати та оцінювати ефективність виконання замовлень у залежності від об'єктивних та суб'єктивних переваг, наданих ОПР, а також забезпечує як урахування, так і виключення певних часткових критеріїв у залежності від певної ситуації.

2. Досліджено застосування таких алгоритмів для розв'язання задачі планування виконання замовлень: алгоритм «бджолої колонії», алгоритм «хаотичного кажана», алгоритм «кажана на основі стратегії пошуку польоту Леві», алгоритм «кажана на основі коефіцієнта скорочення», «генетичний алгоритм, алгоритм «косяку риб», алгоритм «зграї вовків», модифікований алгоритм «зграї вовків», алгоритм «мурашиної колонії з використанням елітних мурах»,

модифікований «алгоритм мурашиної колонії». При кількості замовлень менше 25 усі алгоритми знаходять однакові значення, але кращі показники за часом пошуку мають стандартні версії алгоритмів; при кількості замовлень 50 і більше кращі показники мають модифіковані алгоритми. За результатами дослідження запропоновано модифікації алгоритмів «мурашиної колонії» та «зграї вовків», які дозволяють при їх застосуванні проводити реконфігурацію планів при зменшенні часу пошуку оптимуму майже на 20% у порівнянні з іншими версіями алгоритмів. При використанні немодифікованих версій час пошуку оптимуму збільшується на 4,5%. Модифікація алгоритму «мурашиної колонії» дозволяє скоротити час за рахунок використання підходів локальної оптимізації, а модифікація алгоритму «зграї вовків» – за рахунок виділення шаблонів із найкращих альтернативних рішень задачі.

3. Розроблено комбінований метод прийняття рішень на основі алгоритмів «косяку риб» та модифікованого алгоритму «зграї вовків», який стабільно працює на задачах малої та середньої розмірності й дає зменшення часу пошуку до 35% при умові врахування до 5–6 часткових критеріїв. У багатокритеріальних випадках комбінований метод забезпечує прискорення пошуку на 33,9% у порівнянні з модифікованим алгоритмом «зграї вовків», на 35,1% у порівнянні з алгоритмом «мурашиної колонії», на 33,8% у порівнянні з алгоритмом «косяку риб». Тому запропонований комбінований алгоритм доцільно застосовувати для підприємств із невеликим асортиментом та з мінімальним парком технологічного обладнання, адже він має більшу ефективність при пошуку обмежено критеріальних альтернативних оптимальних планів (до 5–6 часткових критеріїв оптимальності). Тобто, для підприємств, що використовують автоматизовані технологічні комплекси, які забезпечують серійне виготовлення продукції.

4. Розроблено комбінований метод на основі алгоритмів «мурашиної колонії», «зграї вовків» та «генетичного алгоритму», який стабільно працює на багатокритеріальних задачах різної розмірності й має швидкість пошуку на 10–45% більшу, ніж інші алгоритми. За інформацією, отриманою після проведення експериментів, є підстави стверджувати, що для багатокритеріальних задач

комбінований метод забезпечує більш швидший пошук, ніж усі інші алгоритми. Його доцільно застосовувати для підприємств, які виготовляють великий асортимент багатокомпонентних видів продукції і потребують складних технологічних операцій, розрізнених за певними ділянками виробництва.

5. Розроблена інформаційна технологія планування виконання замовлень, яка забезпечує формування та оперативну реконфігурацію планів за короткий період часу за рахунок поєднання інтелектуального аналізу даних для попередньої класифікації, а також використання модифікованих алгоритмів та методів, які базуються на комбінуванні алгоритмів. Запропонована технологія дає можливість підібрати сукупність алгоритмів та методів, адекватну для певної предметної області. Використання інтелектуального аналізу даних забезпечує збільшення швидкості знаходження рішень за рахунок аналізу даних за попередні періоди, що дає змогу виявити найкращі послідовності використання конкретного технологічного обладнання для виготовлення продукції згідно з актуальними замовленнями. Застосування запропонованої технології дає змогу знайти додаткові альтернативні оптимальні плани виконання замовлень, а представлення результатів їх порівняння у вигляді наочної пелюсткової діаграми допомагає відповідальній особі швидко прийняти ефективне рішення.

6. Розроблено архітектуру гібридної web-орієнтованої системи підтримки прийняття рішень, яка реалізує запропоновану інформаційну технологію і базується на гнучкій інтеграції в «озері даних» інформації із статистичних та експертних джерел. Запропонована архітектура дає можливість поєднати декілька інформаційних технологій, а саме: технологію планування виконання замовлень, що забезпечує створення та коригування оперативно-календарних планів за рахунок використання модифікованих та комбінованих алгоритмів; підходи експертно-моделюючої системи рецептур виготовлення, що дає можливість підібрати у виробничих умовах заміну компонентів у рецептурі; технологію для удосконалення процесу аналізу та планування виконання замовлень на харчових підприємствах, яка, на відміну від відомих, ґрунтується на визначенні асортименту та коригуванні рецептур для зменшення собівартості та швидкого виконання замовлень.

7. Формалізовано процес створення системи підтримки прийняття рішень із застосуванням спіральної моделі розробки програмного забезпечення, в якій, на додаток до традиційної, чітко визначено місце проведення рефакторингу, що дає можливість скорочення часу на розробку та підвищення ефективності розробки в цілому. Обґрунтовано застосування категорії «рефакторингу доступу», яка дає змогу впроваджувати зміни, пов'язані з доступом до даних, при адмініструванні сховища та «вітрин даних» без втрати працездатності системи. У процесі проєктування баз знань, вітрин та сховищ даних використано підходи щодо маркування та автоматизованого впровадження оновлення їх компонентів і елементів за рахунок застосування категорії рефакторингу доступу та специфікації семантичного версіонування баз даних, що забезпечило підвищення якості та скорочення часу розробки.

8. Результати дисертаційного дослідження пройшли апробацію та передані на ТОВ «Слобожанський бекон», ТОВ «Продеко», ТОВ «Козятинський м'ясокомбінат», ТОВ «Лайм Системс», ТОВ «Феракс», ПрАТ «Оболонь» для розв'язання задачі планування виконання замовлень, що підтверджено відповідними актами та довідками. Результати дисертаційного дослідження впроваджено у навчальний процес Національного університету харчових технологій при викладанні дисципліни «Управління ІТ-проєктами» для здобувачів освіти ступеня бакалавр за спеціальністю 122 – комп'ютерні науки освітньо-професійної програми «Комп'ютерні науки», дисципліни «Клієнт-серверні технології» для здобувачів освіти ступеня магістр за спеціальністю 122 – комп'ютерні науки освітньо-професійних програм «Інформаційні управляючі системи та технології» та «Комп'ютерний еколого-економічний моніторинг».

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Грибков С. В., Кононова В. О., Харкянен О. В. Оцінка засобів захисту інформаційних ресурсів. *Вісник Національного університету «Львівська політехніка». Комп'ютерні системи та мережі*. 2014. № 806. С. 99–105.
2. Грибков С. В., Загоровська Л. Г. Використання СА Erwin Model Manager для моделювання структури сховища даних. *Наукові праці НУХТ*. 2014. № 6, Т. 20. С. 125–130.
3. Грибков С. В., Харкянен О. В., Логвин Т. В. Дослідження пакетів програмних модулів для ідентифікації динамічних об'єктів. *Технологічний аудит та резерви виробництва*. 2015. № 25 (5/2). С. 42–49.
4. Грибков С. В., Литвинов В. В., Олійник Г. В. Задача планування виконання договорів та підходи до її ефективного вирішення. *Математичні машини і системи*. 2015. № 1. С. 61–70.
5. Hrybkov S. V., Oliinyk H. V. Modeling of the decision support system structure in the planning and controlling of contracts implementation. *Ukrainian Journal of Food Science*. 2015. N 3 (1). P. 123–130.
6. Грибков С. В., Харкянен О. В., Пархоменко І. І. Прогнозування стану технологічного обладнання харчового підприємства методом сингулярного спектрального аналізу. *Вісник інженерної академії України*. 2015. № 3. С. 87–91.
7. Грибков С. В., Харкянен О. В., М'якшило О. М. Моделювання процесу моніторингу та планування собівартості продукції багатомоделного харчового підприємства. *Наукові праці НУХТ*. 2015. № 6, Т. 21. С. 100–108.
8. Грибков С. В., Загоровская Л. Г., Бондарь Н. П., Губеня В. А. Анализ и реинжиниринг процесса планирования закупок для предприятий ресторанного хозяйства. *Вестник Алмаатинского технологического университета*. 2015. №4 (109). С. 31-36.
9. Грибков С. В., Маковецька С. В., М'якшило О. М. Дослідження та математичне моделювання процесу постачання сировини на цукровий завод з

врахуванням генетико-детермінованих властивостей цукрових буряків. *Наукові праці НУХТ*. 2016. № 6, Т. 22. С. 7–15.

10. Грибков С. В., Литвинов В. А., Олійник Г. В. Обрання програмної платформи для побудови модуля безпеки web-орієнтованої системи підтримки прийняття рішень. *Вісник Національного університету «Львівська політехніка». Автоматика, вимірювання та керування*. 2016. № 852. С. 137–142.

11. Грибков С. В., Павелко В. І., Заславський А. І., Дмитренко Д. С. Математичне моделювання процесу термічної обробки м'ясних виробів». *Харчова промисловість*. 2016. № 20. С. 115–120.

12. Грибков С. В., Олійник Г. В. Модифікований АСО алгоритм побудови календарного плану виконання договорів. *Математичне та комп'ютерне моделювання. Технічні науки*. 2017. № 15. С. 156–162.

13. Грибков С. В., Маковецька С. В. Функціональне моделювання організації та управління забезпечення сировиною цукрового заводу. *Науковий вісник Національного університету біоресурсів і природокористування України. Техніка та енергетика АПК*. 2017. № 286. С. 100–111.

14. Грибков С. В., Олійник Г. В. Використання JWT-маркерів для аутентифікації та авторизації користувачів у web-додатках. *Вісник Національного університету «Львівська політехніка». Автоматика, вимірювання та керування*. 2017. № 880. С. 60–67.

15. Hrybkov S. V., Breus N. M., Polischuk G. Ye. Hybrid expert system to model the ice cream recipes. *Ukrainian Journal of Food Science*. 2017. № 5 (2). P. 294–304.

16. Hrybkov S. V., Kharkianen O. V., Myakshylo O. M., Kostikov M. P. Development of information technology for supporting the process of adjustment of the food enterprise assortment. *Eastern-european journal of enterprise technologies*. 2018. № 3 (91). P. 77–87.

17. Грибков С. В., Струзік В. А., Литвин А. О. Дослідження методів і підходів проведення рефакторингу баз даних. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2018. № 30. С. 151–155.

18. Hrybkov S. V., Litvinov V. V., Oliinyk H. V. Web-oriented decision support system for planning agreements execution. *Eastern-european journal of enterprise technologies*. 2018. № 2 (93). P. 13–24.
19. Грибков С. В., Бойко Р. О. Мережеві структури при керуванні складними організаційно-технічними (технологічними) системами. *Харчова промисловість*. 2019. № 25. С. 116–123.
20. Струзік В. А., Грибков С. В., Чобану В. В. Визначення місця рефакторингу в сучасних методологіях розробки інформаційних систем. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Технічні науки*. 2019. № 5, Т. 30 (69). С. 173–177.
21. Hrybkov S. V., Breus N. M., Seidykh O. L., Polischuk G. Ye. Development of mathematical apparatus of the expert system for modelling ice cream recipes with specified quality parameters. *Science and innovation*. 2019. Vol. 15 (5). P. 57–66.
22. Hrybkov S. V., Kharkianen O. V., Ovcharuk V. O., Ovcharuk I. Development of information technology for planning order fulfillment at a food enterprise. *Eastern-european journal of enterprise technologies*. 2020. N 3 (103). P. 62–73.
23. Грибков С. В., Литвинов В. В., Олійник Г. В. Інструментальна модель веб-орієнтованої програмної реалізації підсистеми підтримки прийняття рішень у складі програмного комплексу ситуаційного центру. *Математичні машини і системи*. 2020. № 1. С. 73–81.
24. Грибков С. В., Струзік В. А., Чобану В. В. Категорія рефакторинг доступу. *Наукові праці НУХТ*. 2020. № 2, Т. 26. С. 31–49.
25. Грибков С. В., Струзік В. А., Чобану В. В. Специфікація семантичного версіонування бази даних. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Технічні науки*. 2020. № 2, Т. 31 (70). С. 196–201..
26. Грибков С. В., Сєдих О. Л. Розробка СППР управління виробництвом. *Almanahul SWorld*. 2020. № 4 (1). С. 52–56.
27. Hrybkov S. V., Seidykh O. L. Manufacturing management methods at the expense of using bat algorithm. *Modern engineering and innovative technologies*. 2020. Issue 12 (3). P. 68–73.

28. Hrybkov S. V., Seidykh O. L. Hybrid algorithm based on fish school search and grey wolf optimizer algorithms for food enterprise management. *European science review*. Vienna: Premier Publishing s.r.o, 2020. N 7–8. P. 19–22.
29. Hrybkov S. V., Struzik V. A., Chobanu V. V. Evolution of refactoring. *The Austrian Journal of Technical and Natural Sciences*. Vienna: Premier Publishing s.r.o, 2020. N 7–8 P. 11–16.
30. Hrybkov S. V., Kharkianen O. V., Gladka Y. U. Finding the best versions of schedule for order fulfillment at food companies. *MIND journal Semiannual*. Poland: Bielsko-Biała. 2020. N 9. P. 1–12.
31. Hrybkov S. V., Seidykh O. L. Solution of control tasks using bat algorithm modifications. *Modern engineering and innovative technologies*. 2020. Issue 13 (2). P. 60–64.
32. Грибков С. В., Сєдих О. Л. Розробка модифікованих ітераційних алгоритмів для розв'язання задачі формування оптимальних варіантів розкладу виконання замовлень. *Харчова промисловість*. 2020. № 27. С. 126–137.
33. Грибков С. В., Олійник Г. В. Високопродуктивні пакети завантаження даних до сховища даних. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті: матеріали 81 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 23–24 квітня 2015 р.)*. Київ: НУХТ, 2015. Ч. 2. С. 356.
34. Грибков С. В., Нерадович М. С., Олійник Г. В. Використання методу бджолиного рою при формуванні оптимального розкладу виконання замовлень. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами: матеріали II міжнар. наук.-техн. конф. Київ : НУХТ, 2015. С. 194.*
35. Грибков С. В., Воловик О. О. Web-орієнтована інформаційна система підтримки контролю якості сировини та готової продукції. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами: матеріали III міжнар. наук.-техн. конф. Київ: НУХТ, 2016. С. 219.*

36. Hrybkov S. V., Ollinyk H. V., Litvinov V. V. Decision support system for the contracts execution planning based on genetic algorithms and collective mind algorithms. *Food Science for Well-being (CEFood 2016): Book of Abstracts of 8th Central European Congress on Food (Kiyv, 23–26 May 2016)*. Kiyv: NUFT, 2016. P. 58.

37. Грибков С. В., Буряченко М. М., Харкянен О. В. Використання еволюційних методів для формування виробничої програми харчового підприємства. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті: матеріали 83 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 5–6 квітня 2017 р.)*. Київ, 2017. С. 285.

38. Грибков С. В., Буряченко М. М. Використання середовища R для аналізу та виявлення причин виникнення некондиційної продукції у макаронному виробництві. *Обчислювальний інтелект (результати, проблеми, перспективи): матеріали 4 міжнар. наук.-практ. конф. (м. Київ, 16–18 травня 2017 р.)*. Київ-Черкаси: ВПЦ «Київський університет», 2017. С. 204–205.

39. Грибков С. В., Олійник Г. В. Розробка web-орієнтованої системи підтримки прийняття рішень при плануванні виконання договорів. *Автоматика–2017: матеріали XXIV міжнар. конф. з автоматичного управління (м. Київ, 13–15 вересня 2017 р.)*. Київ: НУБіП, 2017. С. 192–193.

40. Грибков С. В., Олійник Г. В. Web-орієнтована система підтримки прийняття рішень планування виконання договорів. *Проблеми інформатизації: міжнар. наук-техн. конф.: зб. тез*. Київ: ДУТ, 2017. С. 239.

41. Грибков С. В., Струзік В. А., Литвин А. О. Використання еволюційного підходу при рефакторингу баз даних. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами: матеріали IV міжнар. наук.-техн. Internet-конф. (м. Київ, 22 листопада 2017 р.)*. Київ: НУХТ, 2017. С. 281.

42. Hrybkov S. V., Struzik V. A., Lytvyn A. O., Chobanu V. V. Study Of The Methods And Approaches Of Databases Refactoring. *Програмовані логічні інтегральні схеми та мікропроцесорна техніка в освіті і виробництві: зб. тез міжнар. наук.-практ. семінару молодих вчених та студентів*. Луцьк: Вежа-Друк, 2018. С. 12–13.

43. Грибков С. В., Ольшевська М. А. Підсистема підтримки діяльності начальника виробництва макаронного підприємства ТОВ «Продоко». *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 84 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 23–24 квітня 2018 р.). Київ: НУХТ, 2018. Ч. 2. С. 337.

44. Грибков С. В., Фурта О. О. Використання інтелектуального аналізу даних для планування виготовлення продукції на ТОВ «Слобожанський бекон». *Проблеми інформатизації*: матеріали XII міжнар. наук.-техн. конф. Київ: ДУТ, НТУ; Полтава: ПНТУ; Катовице: КЕУ; Париж: Університет Париж VII Венсент-Сен-Дені; Вільнюс: ВДТУ; Харків: ХНДІТМ; Білорусь: БДАЗ, 2018. С. 28–29.

45. Грибков С. В., Олійник Г. В., Чернишов І. Використання модифікованих методів мурашиного алгоритму для розв'язку задачі формування розкладу. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 85 Ювілейної міжнар. наук. конф. молодих учених, аспірантів і студентів, присвяченої 135-річчю НУХТ (м. Київ, 11–12 квітня 2019 р.). Київ: НУХТ, 2019. Ч. 2. С. 435.

46. Грибков С. В., Ольшевська М. А. Створення інформаційної системи підтримки управління виробництва макаронних виробів. *LXXV наук. конф. професорсько-викладацького складу, аспірантів, студентів та співробітників відокремлених структурних підрозділів університету*: зб. тез. Київ: НТУ, НУХТ, 2019. Т. 2. С. 462.

47. Грибков С. В., Фурта О. О. Підсистема підтримки прийняття рішення головного технолога ТОВ «Слобожанський бекон». *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами*: матеріали VI міжнар. наук.-техн. Internet-конф. (м. Київ, 20 листопада 2019 р.). Київ: НУХТ, 2019. С. 176.

48. Грибков С. В., Чорнобай К. Ю. Дослідження та проектування підсистеми управління молочним балансом молокозаводу. *Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій*: наук. праці

Другої міжнар. наук.-практ. конф. (м. Київ, 19 грудня 2019 р.). Київ: НУХТ, 2019. С. 298–302.

49. Грибков С. В., Сєдих О. Л., Ольшевська М. А. Дослідження та розроблення СППР управління виробництвом ТОВ «Продеко». *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 86 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 2–3 квітня 2020 р.). Київ: НУХТ, 2020. Ч. 2. С. 403.

50. Hrybkov S. V., Seidykh O. L. Combined algorithm based on algorithms for konyake fish and gray wolves for solving complex problems. *Sworld-Us conference proceedings: conf. proceed. of Global scien. and educ. in the modern real*. Washington, USA: «ISE&E» & SWorld in conjunction with KindleDP Seattle, 2020. P. 54–57.

51. Грибков С. В., Балашева А. М. Дослідження і створення веб-орієнтованої СППР при управлінні розвитку компанії ТОВ «Доктор Філін». *Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій*: матеріали I міжнар. наук.-практ. конф. (м. Краків, 7–8 лютого 2019 р.). Краків, Польща, 2019. С. 120–123.

52. Грибков С. В., Литвинов В. В., Олійник Г. В. До проблеми типізації проектних рішень у сфері програмної реалізації підсистем ситуаційного центру. *Цифрова економіка та інформаційні технології*: матеріали Міжнар. наук.-практ. конф. (м. Київ, 15–16 квітня 2020 р.). Київ: ДУІТ, 2020. С. 37–38.

53. Грибков С. В., Литвинов В. В., Олійник Г. В. Инструментальная модель подсистемы поддержки принятия решений. *Математичне та імітаційне моделювання систем*: матеріали XV міжнар. наук.-практ. конф. (м. Чернігів, 29 червня – 01 липня 2020 р.). Чернігів, 2020. С. 212–214.

54. Shaw C. Manufacturing Planning and Execution Software Interfaces. *Journal of Manufacturing Systems*. 2000. Vol. 19, N 1. P. 1–17.

55. Жукова И. С. Проблемы организации производства в условиях смены технологических укладов. *Теоретические основы и практика организации производства*: Юбилейный сб. науч. трудов. Воронеж: ВГТУ. 2010. С. 13–20.

56. Lean manufacturing. URL: [https://en.wikipedia.org/wiki/Lean\\_manufacturing](https://en.wikipedia.org/wiki/Lean_manufacturing) (дата звернення: 12.09.2020).
57. Омеляненко Т.В., Щербіна О. В., Барабась Д. О., Вакуленко А. В. Ощадливе виробництво: концепції, інструменти, досвід : наук.-практ. видання. К : КНЕУ, 2009. 157 с.
58. Єрохин Є. А. Актуальність методології ощадливого виробництва в умовах світової економічної кризи. URL: <http://www.ekportal.ru/page-id-1360.html> (дата звернення: 12.10.2020).
59. Маскелл Б., Баггали Б. Практика бережливого учета: управленческий, финансовый учет и система отчетности на бережливых предприятиях. М.: ИСКИ, 2010. 384 с.
60. Вумек Дж. П. Бережливое производство: как избавиться от потерь и добиться процветания вашей компании. М.: Альпина Бизнес Букс, 2006. 470 с.
61. Марчвински Ч., Щук Д. Иллюстрированный глоссарий по бережливому производству. М. : Альпина Бизнес Букс, 2005. 270 с.
62. Харченко І. В., Романюк Л. М. Впровадження системи «бережливого виробництва» на українських підприємствах машинобудівної галузі та стратегія підвищення її конкурентоспроможності. *Наукові праці Кіровоградського національного університету. Економічні науки*. 2016. Вип. 29. С. 186–192.
63. Оно Т. Производственная система Тойоты: уходя от массового производства. М.: ИКСИ, 2005. 187 с.
64. Синго С. Изучение производственной системы Тойоты с точки зрения организации производства / пер. с англ. М.: ИКСИ, 2006. 312 с.
65. Имаи М. Гемба кайдзен: путь к снижению затрат и повышению качества. М. : Альпина Паблишерз, 2010. 340 с.
66. Имаи М. Кайдзен. Ключ к успеху японских компаний. М. : Альпина Паблишер. 2009. 280 с.
67. Goldratt E. M. Cost accounting: the number one enemy of productivity. *American Production and Inventory Control Society: annual International conference proc.* Chicago: 1983. 446 p.

68. Goldratt E. M. From cost world to throughput world. *Advances in Management Accounting*. 1992. N 1. P. 35–53.
69. Noreen E. W., Smith D. A., MacKey T. J. The theory of constraints and its implications for management accounting. Great Barrington: North River Press, 1995. 216 p.
70. Детмер У. Теория ограничений Голдратта: системный подход к непрерывному совершенствованию / пер. с англ. 2-е изд. М.: Альпина Бизнес Букс, 2008. 444 с.
71. Маргіта Н. О., Маргіта М. В. Практичні аспекти застосування теорії обмежень. *Науковий вісник Мукачівського державного університету. Економіка*. 2017. Вип. 2 (8). С. 101–105.
72. Суська М. М., Малигіна В. Д., Бідаш В. І. Забезпечення якості та безпечності м'ясної продукції на підприємствах птахопереробної галузі. *Науковий вісник Луганського національного аграрного університету. Технічні науки. Технології харчових виробництв*. 2010. № 22. С. 190–197.
73. НАССР. URL: <https://uk.wikipedia.org/wiki/НАССР> (дата звернення: 01.10.2020).
74. Леньшин В. Н., Куминов В. В., Фролов Е. Б., Будник Р. А. Производственные исполнительные системы (MES) – путь к эффективному предприятию. URL: <https://sapr.ru/article/7478> (дата звернення: 05.10.2020).
75. Орлов О. О. Планування діяльності промислового підприємства. Київ : Скарби, 2008. 336 с.
76. Михалевич М. В., Сергиенко И. В. Моделирование переходной экономики: модели, методы, информационные технологии: монография. К.: Наукова думка, 2005. 670 с.
77. Ладанюк А. П., Шумигай Д. А., Бойко Р.О. Системна задача координації в технологічних комплексах неперервного типу. *Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація: зб. наук. праць*. Кіровоград: КНТУ, 2012. Вип. 25, Ч. I. С. 288–294.

78. Грибков С. В. Розроблення системи підтримки прийняття рішень для управління макаронним виробництвом: дис. ... канд. техн. наук: 05.13.06. Київ: НУХТ, 2013. 160 с.
79. Харкянен О. В. Інформаційна підтримка прийняття рішень в задачах планування багатоміноменклатурного харчового підприємства: дис. ... канд. техн. наук: 05.13.06 Київ: НУХТ, 2013. 115 с.
80. Бойко Р. О. Інформаційні технології управління технологічним комплексом цукрового заводу в умовах невизначеності: дис. ... канд. техн. наук: 05.13.06. Київ: НУХТ, 2014. 218 с.
81. Дейнека В. С., Сергиенко И. В. Анализ многокомпонентных распределенных систем и оптимальное управление: монография. К.: Наукова думка, 2007. 704 с.
82. Распределенные информационные системы. Архитектура информационных систем. URL: [https://sites.google.com/site/arhitektura\\_informacionnyhsist/lekcia-5-raspredelennye-informacionnye-sistemy/](https://sites.google.com/site/arhitektura_informacionnyhsist/lekcia-5-raspredelennye-informacionnye-sistemy/) (дата звернення: 12.11.2020).
83. Цветков В. Я., Алпатов А. Н. Проблемы распределенных систем. *Перспективы науки и образования*. 2014. № 6 (12). С.31–36.
84. Горбачев В. Г. Методология. Некоторые проблемы создания больших информационных систем в учреждениях. URL: [http://www.inmeta.ru/metod/big\\_syst.htm](http://www.inmeta.ru/metod/big_syst.htm) (дата звернення: 12.11.2020).
85. Сергиенко И. В., Михалевич М. В., Стецюк П. И., Кошлай Л. Б. Модели и информационные технологии для поддержки принятия решений при проведении структурно-технологических преобразований. *Кибернетика и системный анализ*. 2009. № 2. С. 26–49.
86. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. Москва: Мир, 1973. 344 с.
87. Згуровский М. З., Панкратова Н. Д. Системный анализ: проблемы, методология, приложения: монография. К. : Наукова думка, 2011. 727 с.

88. Сергієнко І. В. Інформатика в Україні: становлення, розвиток, проблеми: монографія. Київ: Наукова думка, 1999. 354 с.
89. Павлов А. А. и др. Математические модели иерархического планирования и принятия решений. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2008. № 48. С. 63–66.
90. Павлов А. А., Иванова А. А., Зигура Р. А. Метод группового учёта аргументов и анализа иерархий (МГУАИАИ) в задачах принятия решений. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2007. № 47. С. 205–214.
91. Павлов А. А., Лищук Е. И. Оперативные алгоритмы принятия решений в иерархической системе Саати, основанные на замещении критериев. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2008. № 48. С. 78–81.
92. Згуровский М. З., Попов А. А., Мисюра Е. Б., Мельников О. В. Методология построения эффективного решения многоэтапных задач календарного планирования на основе принципа иерархии и комплекса взаимосвязанных моделей и методов. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2009. № 50. С. 8–18.
93. Згуровский М. З., Павлов А. А., Мисюра Е. Б. ПДС-алгоритмы и труднорешаемые задачи комбинаторной оптимизации. *Системні дослідження та інформаційні технології*. 2009. № 4. С. 14–31.
94. Згуровский М. З., Павлов А. А., Мисюра Е. Б., Мельников О. В. Методы оперативного планирования и принятия решений в сложных организационно-технологических системах. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2009. № 50. С. 3–7.
95. Згуровский М. З., Павлов А. А. Иерархическое планирование в системах, имеющих сетевое представление технологических процессов и ограниченные ресурсы, как задача принятия решений. *Системні дослідження та інформаційні технології*. 2009. № 3. С. 70–75.

96. Павлов А. А. Признаки оптимальности допустимых решений труднорешаемых задач комбинаторной оптимизации. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*. 2013. № 59. С. 4–12.
97. Кастеллани К. Автоматизация решения задач управления / пер. с франц. К. Кастеллани. Москва: Мир. 1982. 472 с.
98. Ладанюк А. П. Оптимальные регуляторы в подсистемах поддержки принятия решений. *Проблемы управления и информатики*. 2004. № 2. С. 138–143.
99. Сергиенко И. В., Лебедева Т. Т., Семенова Н. В. О существовании решений в задачах векторной оптимизации. *Кибернетика и системный анализ*. 2000. № 6. С. 39–46.
100. Galba E. F., Deineka V. S., Sergienko I. V. Weighted pseudoinverses and weighted normal pseudosolutions with singular weights. *Computational Mathematics and Mathematical Physics*. 2009. N 49 (8). P. 1281–1297.
101. Зайченко Ю. П., Сидорук І. А. Аналіз багатокритеріальної задачі оптимізації інвестиційного портфеля на основі прогнозування прибутковості акцій. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка* : зб. наук. пр. 2015. Вип. 62. С. 79-88.
102. Павлов А. А., Мисюра Е. Б., Мельников О. В., Ганзина Е. Решение задачи агрегации в трехуровневой модели планирования мелкосерийного производства. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2009. № 51. С. 26–27.
103. Павлов А. А., Иванова А. А., Штанькевич А. С. Инверсный модифицированный алгоритм метода группового учета аргументов и анализа иерархий в задаче принятия решений. *Адаптивні системи автоматичного управління*: міжвідомчий науково-технічний збірник. 2009. № 14 (34). С. 99–104.
104. Павлов А. А., Иванова А. А., Штанькевич А. С., Федотов А. П. Модифицированный метод анализа иерархий (версии 2, 3). *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2009. № 51. С. 3–12.

105. Павлов А. А., Мисюра Е. Б., Костик Д. Ю. Минимизация суммарного запаздывания при наличии заданий с отрицательными значениями директивных сроков. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2011. № 53. С. 3–5.

106. Павлов А. А., Лисецкий Т. Н. Нестационарный метод анализа иерархий в задачах иерархического планирования и принятия решений. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2011. № 53. С. 82–86.

107. Павлов А. А., Мисюра Е. Б., Лисецкий Т. Н. Объединение работ в группы с учетом их приоритетов, готовности к выполнению и директивных сроков. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2011. № 53. С. 209–211.

108. Павлов А. А. и др. Формальное описание трехуровневой модели оперативного планирования систем с сетевым представлением технологических процессов. Постановка новых задач исследования. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*: зб. наук. праць. 2012. № 55. С. 5–10.

109. Sergienko I. V., Deineka V. S. Optimal control of distributed systems with conjugation conditions. Berlin: Springer Science & Business Media, 2006. 399 с.

110. Маклаков С. В. Моделирование бизнес-процессов с AllFusion Process Modeler. Москва: Диалог-МИФИ, 2008. 224 с.

111. Маклаков С. В. Создание информационных систем с AllFusion Modeling Suite. Москва: Диалог-МИФИ, 2007. 396 с.

112. Загорювська Л. Г. Функціонально-вартісна модель автоматизованої системи управління хлібопекарським виробництвом у ринкових умовах: дис. ... канд. техн. наук: 05.13.06. Київ: НУХТ, 2000. 123 с.

113. Александров Д. В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы. Москва : Финансы и статистика, 2011. 224 с.

114. Грибков С. В., Загоровська Л. Г. Дослідження та моделювання діяльності ВАТ «Макаронна фабрика». *Наукові праці Національного університету харчових технологій*. К. : НУХТ, 2004. № 15. С. 81-82.
115. Сергеева В. И. Корпоративная логистика. 300 ответов на вопросы профессионалов. М : ИНФРА-М, 2005. 976 с.
116. Петухова О. М., Ткачук О. С. Удосконалення організації логістичних процесів на складі підприємства. *Ефективна економіка*. 2018. № 1. URL: <http://www.economy.nayka.com.ua/?op=1&z=6045> (дата звернення: 13.09.2020).
117. Мороз О. В., Музика О. В. Системні фактори ефективності логістичної концепції постачання на підприємствах: монографія. Вінниця : УНІВЕРСУМ, 2007. 165 с.
118. Кальченко А. Г. Логістика. Київ : КНЕУ, 2003. 284 с.
119. Захаров К. В., Бочарников П. В., Захаров А. К., Циганок А. В. Логистика, эффективность и риски внешнеэкономических операций. К.: ИНЭКС, 2000. 217 с.
120. Гришова И. Ю., Щербата М. Ю. Аналитическое обеспечение управления структурой оборотных активов молокоперерабатывающих предприятий. *Экономический вестник Донбасса*. 2013. № 32. С. 101–106.
121. Кузьо Н. Є. Мінімізація витрат у логістичному ланцюгу поставок з використанням економіко-математичного моделювання. *Вісник Національного університету «Львівська політехніка»*. 2012. № 735. С. 112–117.
122. Белікова В. І. Оптимізація товарних запасів лакофарбових матеріалів. *Вісник КНУТД*. 2014. № 2. С. 186–191.
123. Гаврилов Д. Управление производством на базе стандартов MRP II. СПб.: Питер, 2008. 416 с.
124. О'Лири Д. ERP-системы. Современное планирование и управление ресурсами предприятия. Москва : Вершина, 2004. 320 с.
125. Manufacturing execution system. URL: [https://uk.wikipedia.org/wiki/Manufacturing\\_execution\\_system](https://uk.wikipedia.org/wiki/Manufacturing_execution_system) (дата звернення: 12.11.2020).

126. Загидуллин Р. Р. Управление машиностроительным производством с помощью систем MES, APS, ERP. Старый Оскол: ТНТ, 2011. 372 с.
127. Кратко о том, что такое MES-системы. MEScenter. URL: <http://www.mescenter.ru/index.php/home/30-whatismes> (дата звернения: 12.11.2020).
128. Фролов Е. Б., Загидуллин Р. Р. MES-системы как они есть или эволюция систем планирования производства. *Генеральный директор*. 2008. № 4. С. 84–91.
129. C-MES – Collaborative Manufacturing Execution System – Одна из моделей системы оперативного управления производством. TADVISER. URL: <http://www.tadviser.ru/a/53198> (дата звернения: 12.11.2020).
130. MES системы и их задачи. Линчакин: компьютерный портал. URL: <https://linchakin.com/posts/mes-systems/> (дата звернения: 12.11.2020 2018).
131. Внедрение Систем APS. URL: <http://www.opengl.org.ru/sistema-upravleniya-predpriyatiem-tipa-erp/vnedrenie-sistem-aps.html> (дата звернения: 12.11.2020).
132. Advanced Planning & Scheduling. URL: [https://ru.wikipedia.org/wiki/Advanced\\_Planning\\_%26\\_Scheduling](https://ru.wikipedia.org/wiki/Advanced_Planning_%26_Scheduling) (дата звернения: 12.11.2020).
133. Advanced Planning and Scheduling software for balancing demand and capacity. Siemens Digital Industry Software. URL: <https://www.plm.automation.siemens.com/global/ru/products/manufacturing-operations/advanced-planning-scheduling.html> (дата звернения: 02.11.2020).
134. Asprova Advanced Planning and Scheduling (APS). Asprova. URL: <https://www.asprova.com/en/asprova.html> (дата звернения: 05.11.2020).
135. APS 4FACTORY: Advanced production planning and scheduling. DSR. URL: <https://www.dsr.com.pl/en/planning-production-aps/> (дата звернения: 12.11.2020).
136. Seiki's real time manufacturing planning software. SeikiSOFTWARE. URL: <https://www.seikisystems.co.uk/manufacturing-planning-software/> (дата звернения: 12.11.2020).
137. Advanced Planning and Scheduling Software CyberPlan, to manage the Supply Chain processes in a single solution integrated with your ERP. CYBERTC

ZUCCHETTI. URL: <https://www.cybertec.it/en/software-advanced-planning-scheduling/> (дата звернення: 12.11.2020).

138. Manufacturing optimization. Demand Solutions. URL: <https://www.demandsolutions.com/advanced-planning-production-planning-scheduling-software.html> (дата звернення: 12.11.2020).

139. Advanced planning and scheduling with delfoi planner. Delfoi. URL: <https://www.delfoi.com/products/delfoi-planner/delfoi-planner-aps/> (дата звернення: 12.11.2020).

140. Fastreact. Capterra. URL: <https://www.capterra.com/p/14890/Fastreact/> (дата звернення: 12.11.2020 ).

141. FastReactPlan – Fashion Production Planning & Control Dynamic, highly visual planning of capacity, pre-production and materials. Coats DIGITAL. URL: <https://www.fastreact.com/solutions/fastreactplan-manufacturing/> (дата звернення: 12.11.2020).

142. Planning Solutions for Automotive Industry. URL: <https://www.flexis.com/en/solutions/automotive> (дата звернення: 12.11.2020).

143. Терелянский П. В. Системы поддержки принятия решений. Опыт проектирования: монография. Волгоград: ВолгГТУ, 2009. 127 с.

144. Інтелектуальні системи прийняття рішень, загальна теорія, класифікація та їх місце у наш час. URL: <https://m.habr.com/ru/company/ods/blog/359188/> (дата звернення: 15.11.2020).

145. Power D. J. A Brief History of Decision Support Systems. DSSResources. URL: <http://DSSResources.COM/history/dsshhistory.html> (дата звернення: 17.11.2020).

146. Высоцкий В. А. Влияние информационных технологий на развитие систем поддержки принятия решений. Москва: Национальный исследовательский университет «Высшая школа экономики», 2013. С. 27–29.

147. Компьютерные технологии. Классификация информационных систем (ИС). StudIzba. URL: <https://studizba.com/lectures/10-informatika-i-programmirovanie/307-kompyuternye-tehnologii/4079-32-klassifikaciya-informacionnyh-sistem-is.html> (дата звернення: 21.11.2020).

148. Мелихов А. Н., Бернштейн Л. С., Коровин С. Л. Ситуационные советующие системы с нечеткой логикой: монография. Москва: Наука, 1990. 272 с.
149. Борисов В. В., Зернов М. М. Реализация ситуационного подхода на основе нечеткой иерархической ситуационно-событийной сети. *Искусственный интеллект и принятие решений*. 2009. № 1. С. 17–30.
150. М'якшило О. М., Харкянен О. В. Планування собівартості продукції харчового підприємства на основі аналітичних моделей OLAP-кубів *Харчова промисловість*. 2011. № 10–11. С. 332–337.
151. М'якшило О. М., Харкянен О. В. Проблеми оперативного планування для підприємств спиртової галузі. *Наукові праці НУХТ*. 2004. № 15. С. 81–82.
152. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям. СПб. : Питер, 2009. 624 с.
153. Бідюк П. І., Кузнєцова Н. В., Терентьев О. М. Система підтримки прийняття рішень для аналізу фінансових даних *Наукові вісті НТУУ “КПІ”*. Київ : НТУУ “КПІ” ВПІ ВПК “Політехніка”, 2011. № 1. С. 48–61.
154. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных: OLAP и Data Mining. СПб.: БХВ-Петербург, 2004. 336 с.
155. Пендс Н. Что относится к OLAP? URL: [http://www.cfin.ru/itm/olap/olap\\_fasmi.shtml](http://www.cfin.ru/itm/olap/olap_fasmi.shtml) (дата звернення: 25.02.2012).
156. Târnavăeanu D. Pentaho Business Analytics: a Business Intelligence Open Source Alternative. *Database Systems Journal*. 2012. Vol. 3, N 3. P. 23–34.
157. Codd E. F., Codd S. B., Salley C. T. Providing OLAP. On-line Analytical Processing to User-Analists: An IT Mandate. URL: [http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem\\_dwh/lit/Cod93.pdf](http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf) (дата звернення: 12.11.2020, 01.02.2012).
158. Комарцова Л. Г. Исследование нейросетевых и гибридных методов и технологий в интеллектуальных системах поддержки принятия решений: дис. ... д-ра техн.наук: 05.13.11. Москва: ПЦ МЭИ (ТУ), 2003. 438 с.
159. Data mining and knowledge discovery handbook. New York: Springer, 2005. 1383 p.

160. Онищенко А. Г., Листунов Л. Система управления производством. Управление качеством в MES HYDRA (ООО «ИндаСофт»). *Автоматизация Производства-2010*: збір. допов. міжгалуз. конф. Москва, 2010. С. 17-24.
161. Программное обеспечение MES HYDRA. Индасофт. URL: <https://www.indusoft.ru/products/mpdv/HYDRA/> (дата звертання: 12.11.2020 ).
162. MES HYDRA. MEScenter. URL: <http://www.mescenter.ru/index.php/mescat-list/262-mescatalog-hydra> (дата звертання: 12.11.2020 ).
163. Грибков С. В., Загоровська Л. Г. Моделювання структури системи підтримки прийняття рішень для підприємств харчової галузі. *Харчова промисловість*. 2007. № 5. С. 100–102.
164. Сперкач М. О. Інформаційна технологія оперативно-календарного планування дрібносерійного виробництва за концепцією «точно в строк»: дис. ... канд. техн. наук: 05.13.06. Київ: КПІ, 2016. 222 с.
165. Танаев В. С., Шкурба В. В. Введение в теорию расписаний. М.: Наука, 1975. 256 с.
166. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. М.: Физматгиз Наука, 1975. 359 с.
167. Шкурба В. В. и др. Задачи календарного планирования и методы их решения. К.: Наукова думка, 1966. 155 с.
168. Domschke W., Scholl A., Vob S. Produktionsplanung. Ablauforganisatorische Aspekte. Berlin: Heidelberg, Springer Verlag, 2005. 456 p.
169. Подчасова Т. П., Португал В. М. и др. Эвристические методы календарного планирования. К. : Техника, 1980. 140 с.
170. Pinedo M. L. Scheduling. Theory, Algorithms and Systems. Berlin: Springer, 2008. 671 p.
171. Зак Ю. А. Прикладные задачи теории расписаний и маршрутизации перевозок. М.: Книжный дом «ЛИБРОКОМ», 2012. 394 с.
172. Сергиенко И. В., Шило В. П. Задачи дискретной оптимизации: проблемы, методы исследования, решения: монография. К.: Наукова думка, 2003. 262 с.

173. Sergienko I. V., Hulianytskyi L. F., Sirenko S. I. Classification of applied methods of combinatorial optimization. *Cybernetics and Systems Analysis*. 2009. N 45 (5). С. 732–741.
174. Гуляницький Л. Ф., Сергиєнко І. В. Метаевристический метод деформованного многогранника в комбинаторній оптимізації. *Кибернетика и системный анализ*. 2007. № 6. С. 70–79.
175. Сергієнко І. В., Задірака В. К., Литвин О. М. Елементи загальної теорії оптимальних алгоритмів та суміжні питання: монографія. Київ: Наукова думка, 2012. 404 с.
176. Sergienko I. V., Deineka V. S. Solution of inverse boundary-value problems for multicomponent parabolic distributed systems. *Cybernetics and Systems Analysis*. 2007. N 43 (4). P. 507–526.
177. Galba E. F., Deineka V. S., Sergienko I. V. Expansions and polynomial limit representations of weighted pseudoinverses. *Computational Mathematics and Mathematical Physics*. 2007. N 47. P. 713–191.
178. Gantt H. L. Agraphical daily balance in manufacture. *ASME Transactions*. 1903. Vol. 24. P. 1322–1336.
179. Johnson S. M. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*. 1954. Vol. 1. P. 61–68.
180. Bellman R. Mathematical aspects of scheduling theory. *Journal of the Society of Industrial and Applied Mathematics*. 1956. Vol. 4. P. 168–205.
181. Jackson J. R. Scheduling a production line to minimize maximum tardiness. *Manag. Sci. Res. Project. Research Report*. 1955. N 43. P. 75–92.
182. Smith W. E. Various optimizers for single stage production. *Naval Research Logistics Quarterly*. 1956. Vol. 3. P. 59–66.
183. Conway R. W., Maxwell W. L., Miller L. W. Theory of Scheduling. Boston: Addison- Wesley, Reading, MA. 1967. 304 p.
184. Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Optimization*. 1979. Vol. 2. P. 287–325.

185. Brooks G N., White C R. An algorithm for finding optimal or near – optimal solutions to the production scheduling problem. *J. Ind.Eng.* 1965. Vol. 16. N 1. P. 34–40.
186. Carlier J. The one-machine sequencing problem. *European J. of Oper. Res.* 1982. Vol. 11. N 1. P. 42–47.
187. Lageweg B. J., Lenstra J. K., Rinnooy Kan A. H. G. Minimizing maximum lateness on one machine: computational experience and applications. *Statistica Neerlandica.* 1976. Vol. 30. P. 25–41.
188. Алексеев О.Г. Комплексное применение методов дискретной оптимизации. М. : Наука, 1986. 248 с.
189. Jabarzadeh A., Rostami M., Shahin M., Shahanaghi K. Two-stage fuzzy-stochastic programming for parallel machine scheduling problem with machine deterioration and operator learning effect. *Journal of Industrial and Systems Engineering.* 2017. Vol. 10. N 3, P. 16–32.
190. Brucker P., Knust S. Complex scheduling. Springer-Verlag Berlin, Heidelberg, 2006. 267 p.
191. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М. : Мир, 1985. 512 с.
192. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи / пер. с англ. М. : Мир, 1982. 416 с.
193. Беллман Р. Динамическое программирование. М. : ИЛ, 1960. 400 с.
194. Корбут А. А., Сигал И. Х., Финкельштейн Ю. Ю. Гибридные методы в дискретном программировании. *Известия АН СССР. Техническая кибернетика.* 1988. № 1. С. 65–77.
195. Корбут А. А., Финкельштейн Ю. Ю. Приближенные методы дискретного программирования. *Известия АН СССР. Техническая кибернетика.* 1983. № 1. С. 165–176.
196. Каширских К. Н., Поттс К. Н., Севастьянов С. В. Улучшенный алгоритм решения двухмашинной задачи flow shop с неодновременным поступлением работ. *Дискретный анализ и исследование операций.* 1997. Т. 4, № 1. С. 13–32.

197. Sevastianov S. V., Woeginger G. J. Makespan minimization in open shops: A polynomial time approximation scheme. *Math. Prog.* 1998. Vol. 82. P. 191–198.

198. Shih-Wei Lin, Kuo-Ching Ying A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems. *International Journal of Production: ResearchGate* . URL: [https://www.researchgate.net/publication/269280769\\_A\\_multi-point\\_simulated\\_annealing\\_heuristic\\_for\\_solving\\_multiple\\_objective\\_unrelated\\_parallel\\_machine\\_scheduling\\_problems](https://www.researchgate.net/publication/269280769_A_multi-point_simulated_annealing_heuristic_for_solving_multiple_objective_unrelated_parallel_machine_scheduling_problems) (дата звертання: 1.12.2019).

199. Ruiz-Torres A., Paletta G., Perez-Roman E. Parallel machine scheduling to minimize the makespan with sequence dependent deteriorating effects. *Computers & Operations Research*. 2013. Vol. 40, P. 2051–2061.

200. Holland J. H. *Adaptation in Natural and Artificial Systems*. Cambridge, Massachusetts: re-issued by MIT Press, 1992. 217 p.

201. Goldberg D. E. *Genetic Algorithms in Search, Optimization, and Machine learning*. Boston : Addison-Wesley Longman Publishing Co, 1989. 372 p.

202. Mitchell M. *An Introduction to Genetic Algorithms*. London : MIT Press, 1996. 162 p.

203. Garg R., Mittal S. Optimization by Genetic Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2014. Vol. 4 (4). P. 587–589.

204. Sivaraj R., Ravichandran T., Devi Priya R. Boosting Performance of genetic algorithm through Selective initialization. *European Journal of Scientific Research*. 2012 Vol. 68. N 1, P. 93–100.

205. Santosh K. S., Vinod, K. G. Genetic Algorithms: Basic Concepts and Real World Applications. *International Journal of Electrical. Electronics and Computer Systems (IJECS)*. 2015. N 3 (12). P. 116–123.

206. Сетлак Г. Решение задач многокритериальной оптимизации с использованием генетических алгоритмов. *System Research & Information Technologies*. 2002. N 3. С. 32–42.

207. Cleveland G. A. Using genetic algorithms to schedule flow shop releases. *Proceeding of the Third International Conference on Genetic Algorithms* (San Mateo, California, 1989). San Mateo, 1989. P. 160–169.

208. Белецкая С. Ю., Асанов Ю. А., Поваляев А. Д., Гаганов А. В. Исследование эффективности генетических алгоритмов многокритериальной оптимизации. *Вестник ВГТУ*. 2015. № 1. С. 10–14.

209. Грибков С. В., Загоровська Л. Г. Використання генетичних алгоритмів для побудови розкладів виконання заявок на виготовлення макаронних виробів. *Новітні технології, обладнання, безпека та якість харчових продуктів: сьогодення та перспективи: тези доповідей Міжнар. наук.-практ. конф. (м. Київ, 27–28 жовтня 2010 р.)*. К.: НУХТ, 2010. Ч. 2. С. 62.

210. Грибков С. В., Загоровська Л. Г., Бровченко Н. Н. Використання генетичних алгоритмів для розв'язання задачі складання розкладу замовлень. *Восточно-европейский журнал передовых технологий*. 2011. № 6/4 (54). С. 45–48.

211. Батищев Д. И., Власов С. Е., Булгаков И. В. Решение задачи «слепого» упорядочения при помощи генетических алгоритмов. *Обозрение прикладной и промышленной математики*. М.: ГВП, 1996. Т. 3, Вып. 5. С. 725–734.

212. Ротштейн О. П., Штовба С. Д. Оптимізація багатовимірних технологічних процесів генетичними алгоритмами. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 1999. № 2. С. 7–13.

213. Курейчик В. М., Кажаров А. А. Роевой интеллект в решении графовых задач. *Международная конференция по мягким вычислениям и измерениям*. 2013. Т. 1. С. 31–34.

214. Курейчик В. М., Кажаров А. А. Алгоритмы эволюционного роевого интеллекта в решении задачи разбиения графа. *Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС)*. 2012. № 1. С. 237–242.

215. Курейчик В. М., Кажаров А. А. Использование роевого интеллекта в решении NP-трудных задач. *Известия ЮФУ. Технические науки*. 2011. № 7 (120). С. 30–36.

216. Курейчик В. В., Курейчик В. М., Родзин С. И. Основы теории эволюционных вычислений. Ростов-на-Дону: ЮФУ, 2010. 320 с.
217. Pham D. T., Castellani M. Benchmarking and Comparison of Nature-Inspired Population-Based Continuous Optimisation Algorithms. *Soft Computing*. 2013. N 18. P. 1–33.
218. Карпенко А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов. *Информационные технологии* (Приложение). 2012. № 7. С. 1–32.
219. Желдак Т. А. Застосування методу моделювання колонії мурах до розв’язання комбінаторних задач планування виконання замовлень металургійними підприємствами. *Математичні машини і системи*. 2013. № 4. С. 95–106.
220. Senthilkumar K. M., Raja K. A. Hybrid Algorithm Based on PSO and ACO Approach for Solving Combinatorial Fuzzy Unrelated Parallel Machine Scheduling Problem. *European Journal of Scientific Research*. 2011. N 3 (93). P. 87–104.
221. Сагун А., Хайдуров В., Кунченко-Харченко В. Метод стаи волков и его модификация для решения задачи поиска оптимального пути. *Фізико-математична освіта*. 2017. № 2. С. 135–139.
222. Частикова В. А., Новикова Е. Ф. Алгоритм летучих мышей для решения задачи глобальной оптимизации. *Научные труды КубГТУ* (электронный сетевой политематический журнал). URL: <http://ntk.kubstu.ru/file/348> (дата звернення: 30.06.2020).
223. Yang X. S. Bat algorithm for multi-objective optimization. *International Journal of Bio-Inspired Computation*. 2011. Vol. 3. N 5. P. 267–274.
224. Neto L., Lacerda M. G. Multimodal fish school search algorithms based on local information for school splitting. *BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*. Ipojuca, 2013. P. 158–165.
225. Mirjalili S., Lewis A. Grey Wolf Optimizer. *Advanced in Engineering Software*. 2014. Vol. 69. P. 46–61.
226. Лагунова А. Д. Алгоритм стаи серых волков (GWO) для задач оптимизации. *Оригинальные исследования (ОРИС)*. 2019. № 4. С. 52–62.

227. Nikabadi M, Naderi R. A hybrid algorithm for unrelated parallel machines scheduling. *International Journal of Industrial Engineering Computations*. 2016. Vol. 7 (4). P. 681–702.
228. Rodriguez F., Lozano M., García-Martínez C., González-Barrera J. An artificial bee colony algorithm for the maximally diverse grouping problem. *Information Sciences*. 2013. N 230. P. 183–196.
229. Goldberg D. E., Kumara S. A Practical Schema Theorem for Genetic Algorithm Design and Tuning. Illinois, 2001. 11 p.
230. Kennedy J., Eberhart R. Particle Swarm Optimization. *Proc. of ICNN'95 – International Conference on Neural Networks*. (Perth, 27 Nov.-1 Dec., 1995). Purdue School of Engineering and Technology, 1995. P. 1942–1948.
231. Jeyasingh S. Modified Bat Algorithm for Feature Selection with the Wisconsin Diagnosis Breast Cancer (WDBC) Dataset. *PMC US National Library of Medicine National Institutes of Health*. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5555532/> (дата звертання: 02.10.2020).
232. Filho B., Neto L. A novel search algorithm based on fish school. *Systems, Man and Cybernetics*. 2008. N 1. P. 2646–2651.
233. Filho B., Neto L. Популяційний алгоритм, оснований на поведінці косяка рыб. URL: <https://habr.com/ru/post/178309/> (дата звернення: 12.11.2020).
234. Rodriguez F. J., Lozano M., García-Martínez C., González-Barrera J. D. An artificial bee colony algorithm for the maximally diverse grouping problem *Information Sciences*. 2013. N 230. P. 183–196.
235. Ajorlou S., Shams I. Artificial bee colony algorithm for CONWIP production control system in a multi-product multi-machine manufacturing environment. *Journal of Intelligent Manufacturing*. 2013. Vol. 24, Issue 6. P. 1145–1156.
236. Khan L., Ullah I., Saeed T., Lo K. L. Virtual Bees Algorithm Based Design of Damping Control System for TCSC. *COMSATS Institute of Information Technology Abbottabad*. 2010. N 4. C. 1–18.

237. Toksari M. D. Minimizing the earliness/tardiness costs on parallel machine with learning effects and deteriorating jobs: a mixed nonlinear integer programming approach. London : Springer-Verlag, 2008. P. 801–807.
238. Karaboga D. An Idea Based On Honey Bee Swarm for Numerical Optimization. *Erciyes University: Kayser* . URL: [https://abc.erciyes.edu.tr/pub/tr06\\_2005.pdf](https://abc.erciyes.edu.tr/pub/tr06_2005.pdf) (дата звертання: 17.01.2020).
239. Singh A., Nagaraju A. Network coding: ABC based COPE in wireless sensor and Mesh network. *Proc. of the IEEE Contemporary Computing and Informatics* (Mysore, India, 27–29 November 2014). Mysore, 2014. P. 320–325.
240. Singh A., Nagaraju A. An Artificial Bee Colony-Based COPE Framework for Wireless Sensor Network. *Computers*. 2016. N 5. 2. P. 1–15.
241. Pham D. T., Castellani M. The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimisation Problems. *Proc. ImechE. Part C*. 2009. N 223 (12). P. 2919–2938.
242. Бджолиний алгоритм. URL: [https://uk.wikipedia.org/wiki/Бджолиний\\_алгоритм](https://uk.wikipedia.org/wiki/Бджолиний_алгоритм) (дата звернення: 05.03.2020).
243. Yuce B., Packianather M S., Mastrocinque E., Pham D. T., Lambiase A. Honey Bees Inspired Optimization Method: The Bees Algorithm. *Insects*. 2013. N 4. P. 646–662.
244. Artificial bee colony algorithm. URL: [https://en.wikipedia.org/wiki/Artificial\\_bee\\_colony\\_algorithm](https://en.wikipedia.org/wiki/Artificial_bee_colony_algorithm) (дата звернення: 23.06.2020).
245. Естественные алгоритмы. Алгоритм поведения роя пчёл. *Хабр*. URL: <https://habr.com/ru/post/104055/> (дата звернення: 23.06.2020).
246. Yang X. S. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. 2010. Vol. 284. P. 65–74.
247. Nakamura R., Pereira L., Costa K. BBA : A binary bat algorithm for feature selection. *Conference on graphics, patterns and images (25th SIBGRAPI)*. IEEE Publication (Ouro Preto, August 22–25 2012). Ouro Preto, 2012. P. 291–297.
248. Fister I. J., Fister D., Yang X. S. A hybrid bat algorithm. *Electrotechnical Review*. 2013. Vol. 80. N 1–2. P. 1–7.

249. Guo S.-S., Wang J.-S., Ma X.-X. Improved bat algorithm based on multipopulation strategy of island model for solving global function optimization problem. URL: <https://www.hindawi.com/journals/cin/2019/6068743/> (дата звертання: 02.10.2020).
250. Wang G., Guo L. A Novel hybrid bat algorithm with harmony search for global numerical optimization. *Hindawi Publishing Corporation*. 2013. Vol 2013. P. 1–21.
251. Yang X. S., He X. Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*. 2013. Vol. 5, N 3. P. 141–149.
252. Yang X. S., Gandomi H. Chaotic bat algorithm. *Journal of Computational Science*. 2014. Vol. 5, Issue 2. P. 224–232.
253. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия-Телеком, 2008. 452 с.
254. Grefenstette J. J. Optimization of control parameters for genetic algorithms. *IEEE Transaction on Systems, Man and Cybernetics*. SMC-16. 1986. N 1. P. 122–128.
255. Madadi A., Motlagh M. Optimal Control of DC motor using Grey Wolf Optimizer Algorithm. *Technical Journal of Engineering and Applied Science*. 2014. Vol. 4 (4). P. 373–379.
256. Dorigo M., Di Caro G. Ant colony optimization: a new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*. Washington, USA, 1999. Vol. 2. P. 1470–1477.
257. Corne D., Dorigo M., Glover F. New Idea in Optimization. *Control and Cybernetics*. 2000. Vol. 29, N 3. P. 653–684.
258. Dorigo M., Stützle T. Ant colony optimization. Cambridge, UK: Mit Press, 2004. 319 p.
259. Dorigo M., Birattari M., Stutzle T. Ant colony optimization. *Stutzle IEEE computational intelligence magazine*. 2006. N 1 (4). P. 28–39.
260. Dorigo M., Blum C. Ant colony optimization theory: A survey. *Theoretical computer science*. 2005. N 344 (2–3). P. 243–278.

261. Kureichik V., Kazharov A., Lyapunova I. Parameter Analysis of Ant Algorithm. *Life Science Journal*. 2014. Т. 11, N 10. P. 402–406.

262. Kureichik V. M., Kazharov A. A., Lyapunova I. A. Analysis of Ant Algorithm. *Communications in Computer and Information Science (CIT&DS): Book Subtitle of First Conference (September 15-17, 2015)*. Volgograd, Russia: 2015. Part III. P. 179–187.

263. Кажаров А. А., Курейчик В. М. Обзор задач коммивояжера и маршрутизации автотранспорта. *Труды конгресса по интеллектуальным системам и информационным технологиям «IS&IT 2013»*. Научное издание: в 4 т.: сб. трудов. Россия, Ростов-на-Дону: Министерство образования и науки Российской Федерации, Российская ассоциация искусственного интеллекта, Южный федеральный университет, 2013. С. 117–123.

264. Кажаров А. А., Курейчик В. М. Муравьиные алгоритмы для решения транспортных задач. *Известия Российской академии наук. Теория и системы управления*. 2011. № 1. С. 32.

265. Kureichik V. M., Kazharov A. Using Fuzzy Logic Controller in Ant Colony Optimization. *Advances in Intelligent Systems and Computing*. 2015. Т. 347. С. 151–158.

266. Кажаров А. А., Курейчик В. М., Ляпунова И. А. Определение зависимости параметров муравьиного алгоритма от исходных данных. *Вестник ростовского государственного университета путей сообщения*. Ростов-на-Дону: Ростовский государственный университет путей сообщения. 2014. № 4 (56). С. 63–70.

267. Курейчик В. М., Кажаров А. А. О некоторых модификациях муравьиного алгоритма. *Известия ЮФУ. Технические науки*. 2008. № 4 (81). С. 7–12.

268. Кажаров А. А., Курейчик В. М. Использование шаблонных решений в муравьиных алгоритмах. *Известия ЮФУ. Технические науки*. 2013. № 7 (144). С. 17–22.

269. Kureichik V., Kazharov A. The Development of the Ant Algorithm for Solving the Vehicle Routing Problems. *World Applied Sciences Journal*. 2013. Т. 26. N 1. P. 114–121.

270. Кажаров А. А., Курейчик В. М. Муравьиные алгоритмы для решения транспортных задач. *Известия Российской академии наук. Теория и системы управления*. 2010. № 1. С. 32–45.
271. Shtovba S. Ant algorithms: theory and applications. *Programming and Computer Software*. 2005. Vol 31, N 4. P. 167–178.
272. Штовба С. Д. Муравьиные алгоритмы: теория и применение. *Программирование*. 2005. № 4. С. 1–16.
273. Штовба С. Д. Муравьиные алгоритмы. *Exponenta Pro. Математика в приложениях*. 2003. № 4. С. 70–75.
274. Штовба С. Д., Рудий О. Мурашині алгоритми оптимізації. *Вісник Вінницького політехнічного інституту*. 2004. № 4. С. 62–69.
275. Stutzle Th., Hoos H. H. Max-min ant system. *Future Generation Computer Systems*. 2000. Vol. 16 (8). P. 889–914.
276. Rocki K., Suda R. Accelerating 2-opt and 3-opt Local Search Using GPU in the Travelling Salesman Problem. *CCGRID*. 2012. Vol. 1. P. 705–706.
277. Тарасюк В. В. Інформаційні технології. *Енциклопедія Сучасної України. Інститут енциклопедичних досліджень НАН України*. URL: <http://esu.com.ua/search/articles.php?id=12474> (дата звернення: 05.10.2020 ).
278. Інформаційні технології. URL: [https://uk.wikipedia.org/wiki/Інформаційні\\_технології](https://uk.wikipedia.org/wiki/Інформаційні_технології) (дата звернення: 05.10.2020 ).
279. Казаринов Л. С., Шнайдер Д. А., Хасанов А. Р. Метод оценки текущего состояния контролируемого оборудования в задаче оперативного планирования ремонтно-профилактических работ. *Вестник ЮУрГУ. Компьютерные технологии, управление, радиоэлектроника*. 2006. № 14 (69). С. 84–87.
280. Лепеш Г. В., Куртов В. Н., Мотылев Н. Г., Красильников А. Ю., Телятников С. В., Чилипенко А. Л. Оперативный контроль и диагностика оборудования. *Технико-технологические проблемы сервиса*. 2009. № 3 (9). С. 8–16.
281. Barker K., Wilson K. Decision trees with single and multiple interval-valued objectives. *Decis Anal*. 2012. N 9 (4). P. 348–358.

282. Kamiński B., Jakubczyk, M., Szufel, P. A framework for sensitivity analysis of decision trees. *Central European Journal of Operations Research*. 2017. N 26 (1). P. 135–159.

283. Грибков С. В., Загоровська Л. Г. Моделювання структури сховища даних системи підтримки прийняття рішень для підприємств макаронної галузі. *Матеріали міжнародної науково-практичної конференції «Інтелектуальні системи прийняття рішень та інформаційні технології: матеріали міжнар. наук.-практ. конф. (м. Чернівці, 17–19 травня 2006 р.)*. Чернівці, 2006. С. 44–45.

284. Томашевський В. М., Яцишин А. Ю. Математична модель задачі проектування гібридних сховищ даних з врахуванням структур джерел даних. *IT-вісник КНУ*. URL: [http://it-visnyk.kpi.ua/wp-content/uploads/2011/07/53\\_10.pdf](http://it-visnyk.kpi.ua/wp-content/uploads/2011/07/53_10.pdf) (дата звернення: 12.12.2020).

285. Туманов В. Е., Маклаков С. В. Проектирование реляционных хранилищ данных. Москва : Диалог-МИФИ, 2007. 333 с.

286. Inmon W. H. Building the Data Warehouse. 4 ed. New York : John Wiley & Sons, 2005. 576 p.

287. Rainardi V. Building a Data Warehouse: With Examples in SQL Server. Second Edition. New York: Apress, 2008. 541 p.

288. Архипенков С. Я., Голубев Д. В., Максименко О. Б. Хранилища данных. От концепции до внедрения. Москва : Диалог-МИФИ, 2002. 528 с.

289. Спирли Э. Корпоративные хранилища данных. Планирование, разработка, реализация. М. : Вильямс, 2001. Т. 1. 400 с.

290. Tobin Donal Warehouse vs. Database: 7 Key Differences. URL: <https://www.xplenty.com/blog/data-warehouse-vs-database-what-are-the-key-differences/> (дата звернення: 10.10.2020).

291. Асєєв Г. Г. Методологія створення сховищ даних: стандарти та моделювання. *Вісник Книжкової палати*. 2009. № 5. С. 30–32.

292. Kimball R. The Data Warehouse Toolkit (3rd ed.). New York : John Wiley & Sons, 2013. 388 p.

293. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Москва : Вильямс, 2017. 1440 с.
294. Дейт К. Введение в системы баз данных. Boston: Addison-Wesley, 2018. 1328 р.
295. Тарасов С. В. СУБД для программиста. Базы данных изнутри. Москва : Солон-Пресс, 2015. 320 с.
296. Demarest M. Building The Data Mart. *DBMS Magazine*. 1994. N 7. P. 44–50.
297. Асеев Г. Г. Вітрини даних – необхідна ланка в концепціях побудови сховищ даних. *Вісник Книжкової палати*. 2012. № 7. С. 26–29.
298. Almeida S. M., Ishikawa M., Reinschmidt J., Roeber T. Getting Started with DataWarehouse and Business Intelligence. *Inter. Technic. Support Organization*. Город, 1999. 244 р.
299. Hrybkov S. V., Breus N. M. Expert System for Ice Cream Recipes Modeling. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті, присвяченої 135-річчю Національного університету харчових технологій: матеріали 85 Ювілейної міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 11–12 квітня 2019 р.)*. К. : НУХТ, 2019. Ч. 2. С. 412.
300. Бреус Н.М. Інформаційна технологія моделювання рецептур морозива: дис. ... канд. техн. наук: 05.13.06. Київ: НУХТ, 2019. 138 с.
301. Harris Jim The growing importance of big data quality. URL: <https://blogs.sas.com/content/datamanagement/2016/11/21/growing-import-big-data-quality/> (дата звернення: 21.12.2020).
302. What is a data lake? URL: <https://aws.amazon.com/ru/big-data/datalakes-and-analytics/what-is-a-data-lake/> (дата звернення: 21.12.2020).
303. Data lake. URL: [https://en.wikipedia.org/wiki/Data\\_lake](https://en.wikipedia.org/wiki/Data_lake) (дата звернення: 21.12.2020).
304. М'якшило О. М., Харкянен О. В. Дослідження і розробка методів управління ризиками в діяльності харчового підприємства. *Наукові праці Національного університету харчових технологій*. 2014. Т. 20, № 10–11. С. 105–114.

305. Алгоритмы интеллектуального анализа данных (службы Analysis Services – интеллектуальный анализ данных). URL: <http://msdn.microsoft.com/ru-ru/library/ms175595.aspx> (дата обращения: 19.02.2012).
306. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. 2-е изд. СПб. : БХВ-Петербург, 2007. 384 с.
307. Бергер А. В. Microsoft SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных. СПб. : БХВ-Петербург, 2007. 928 с.
308. Бреус Н. М., Грибков С. В. Гібридна експертна система для моделювання рецептур морозива. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами: IV міжнар. наук.-техн. Internet-конф. К. : НУХТ. 2017. С. 243.*
309. Фаулер М. Архитектура корпоративных программных приложений. М.: Издательский дом «Вильямс», 2006. 544 с.
310. Проблемы при проектировании и разработке информационных систем. URL: [http://www.softholm.com/news/development/article\\_2611.html](http://www.softholm.com/news/development/article_2611.html) (дата звернення: 29.12.2020).
311. Струзік В. А. Вдосконалення технологій проведення рефакторингу баз даних для інформаційних систем: дис. ... канд. техн. наук: 05.13.06 НУХТ. Київ, 2020. 169 с.
312. Технический долг. URL: [https://ru.wikipedia.org/wiki/Технический\\_долг](https://ru.wikipedia.org/wiki/Технический_долг) (дата звернення: 21.12.2020).
313. Технический долг. URL: <https://refactoring.guru/ru/refactoring/technical-debt> (дата звернення: 21.12.2020).
314. Технический долг. URL: <https://habr.com/ru/post/453280/> (дата звернення: 21.12.2020).
315. Fowler M., Beck K., Brant J., Opdyke W., Roberts D. Refactoring: Improving the Design of Existing. Addison-Wesley, 1999. 464 p.

316. Ambler S. Refactoring for Fitness. URL: <https://www.drdoobs.com/refactoring-for-fitness/184414821?cid=Ambysoft> (дата звернення: 02.01.2021).
317. Ambler S. The Process of Database Refactoring: Strategies for Improving Database Quality. URL: <http://www.agiledata.org/essays/databaseRefactoring.html> (дата звернення: 02.01.2021).
318. Fowler M. Evolutionary Database Design URL: <https://martinfowler.com/articles/evodb.html/> (дата звернення: 12.11.2020 ).
319. What Is Refactoring? URL: <https://dzone.com/articles/what-is-refactoring/> (дата звернення: 12.11.2020 ).
320. Turning an example by Martin Fowler into a Refactoring Kata. URL: <https://dzone.com/articles/what-is-refactoring/> (дата звернення: 12.11.2020 ).
321. Фаро С. Лерми П. Рефакторинг SQL-приложений. СПб.: Символ-плюс, 2009. 336 с.
322. Martin R. The Clean Coder: A Code of Conduct for Professional Programmers, 1st Edition. Pearson, 2011. 242 p.
323. Drake M. Understanding Database Sharding. URL: <https://www.digitalocean.com/community/tutorials/understanding-database-sharding> (дата звернення: 12.11.2020 ).
324. Пшеничный Д. А. Автоматизация технологических процессов рефакторинга баз данных промышленных предприятий: дис. ... канд. техн. наук: 05.13.06. Москва, 2013. 167 с.
325. Suryanarayana G., Samarthyam G., Sharma T. Refactoring for Software Design Smells: Managing Technical Debt 1st Edition. 2014. 258 p.
326. Тестирование базы данных – Краткое руководство. URL: [https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/izuchite-testirovanie-bazy-dannykh/testirovani\\_e-bazy-dannykh-kratkoe-rukovodstvo](https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/izuchite-testirovanie-bazy-dannykh/testirovani_e-bazy-dannykh-kratkoe-rukovodstvo) (дата звернення: 12.11.2020 15.07.2018).
327. Эмблер С., Садаладж П. Рефакторинг баз данных. Эволюционное проектирование. Москва : Вильямс, 2016. 368 с.

328. Макконнелл С. Совершенный код. Мастер-класс. СПб. : БХВ-Петербург, 2017. 896 с.
329. Лобел Л., Браст Э. Дж., Форте С. Разработка приложений на основе Microsoft SQL Server 2008. СПб. : БХВ-Петербург, 2010. 1024 с.
330. Ларсон Б. Разработка бизнес-аналитики в Microsoft SQL Server 2005. СПб. : Питер, 2008. 688 с.
331. Макленнен Дж., Танг Чж., Криват Б. Microsoft SQL Server 2008: Data Mining – интеллектуальный анализ данных. СПб. : БХВ-Петербург, 2009. 720 с.
332. Харинатх С., Куинн С. SQL Server 2005 Analysis Services и MDX для профессионалов. Москва : Диалектика, 2008. 848 с.
333. Назаров К. Экстремально предвзятый взгляд на версионирование программных продуктов. URL: <https://www.slideshare.net/racktear/semver> (дата звертання: 12.01.2021 ).
334. Престон-Вернер Т. Семантичне Версіонування 2.0.0. URL: <https://semver.org/> (дата звертання: 12.01.2021 ).
335. Построение распределенных систем на Java. URL: <https://www.intuit.ru/studies/courses/633/489/lecture/24846> (дата звернення: 12.11.2020 ).
336. Bass L., Clements P., Kazman R. Software Architecture in Practice. Addison-Wesley Professional, 2013. 640 p.
337. Niemeyer P., Leuck P. Learning Java, 4th Edition. O'Reilly Media, 2013. 1010 p.
338. Ritter S. 4 Reasons Why Java is Still #1. Azul Systems. URL: <https://www.azul.com/4-reasons-java-still-1/> (дата звернення: 21.01.2021).
339. Johnson R., Hoeller J. Spring Framework. Reference Documentation. URL: <https://docs.spring.io/spring/docs/4.3.9.RELEASE/spring-framework-reference/html/>. (дата звернення: 25.02.2018).
340. Олійник Г. В., Грибков С. В. Підтримка механізму транзакцій програмною платформою Spring. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та*

*технологічними комплексами*: Міжнар. наук.-техн. конф. К. : НУХТ. 2016. С. 250–251.

341. Москаленко Н. В. Web-ориентированная информационная система рекламной компании. URL: <https://novainfo.ru/article/?nid=11602> (дата звернення: 28.12.2020).

342. Ахаев А. В. WEB-ориентированная экспертная система выбора программных продуктов. *Наука. Технологии. Инновации*: материалы Всерос. науч. конф. студентов, аспирантов и молодых ученых. Новосибирск. 2012. № 3 С. 263–266.

343. Лобода Ю Г., Орлова О. Ю. Технологии разработки Web-приложений. *Наукові праці. Одеська національна академія харчових технологій*. 2014. № 46. С 239–244.

344. Mihalcea M., Ebersole S. Hibernate ORM 5.2.13. Final User Guide. URL: [https://docs.jboss.org/hibernate/orm/5.2/userguide/html\\_single/Hibernate\\_User\\_Guide.html](https://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html) (дата звернення: 03.01.2021).

345. Kalin M. Java Web Services: Up and Running. 2nd Edition. O'Reilly Media, 2013. 360 p.

346. OpenAPI Specification. URL: <https://swagger.io/specification/> (дата звернення: 03.01.2021).

347. Форристал Д. Защита от хакеров WEB-приложений. Москва: ДМК, 2004. 496 с.

348. Trusted Computer Systems Evaluation criteria. US DoD 5200.28-STD. US, 1985. 116 p.

349. Information Tecnhology Security Evaluation Criteria, v. 1.2. Office for Official ublications of the European Communities, 1991. 163 p.

350. Canadian Trusted Computer Product Evaluation Criteria, v. 3.0. Canadian System Security Centre, Communications Security Establishment, Government of Canada, 1993. 27 p.

351. Federal Criteria for Information Technology security. NIST, NSA, US Government, 1993. 71 p.

352. ISO/IEC 15408-1:1999 – Information technology – Security techniques – Evaluation criteria for IT security. Part 1: Introduction and general model. 53 p.
353. ISO/IEC 15408-2:1999 – Information technology – Security techniques – Evaluation criteria for IT security. Part 2: Security functional requirements. 343 p.
354. ISO/IEC 15408-3:1999. Security techniques – Evaluation criteria for IT security – Part 3: Security assurance requirements. 213 p.
355. CEM-97/017. Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model. 37 p.
356. Якименко І. З. Критерії оцінки рівня захисту комп'ютерних мереж з врахуванням їх архітектури. *Інформатика та математичні методи в моделюванні*. 2013. Т. 3, № 1. С. 82–90.
357. Apache Shiro Documentation. URL: <http://shiro.apache.org/documentation.html> (дата звернення: 03.12.2020).
358. Schaefer C., Ho C., Harrop R. Pro Spring 4th ed. Edition. Apress, 2014. 728 p.
359. Mularien P. Spring Security. PACKT Publications, 2010. 396 p.
360. Уоллс К. Spring в действии. М. : ДМК Пресс, 2013. 752 с
361. JSON Web Token и sliding expiration в web-приложении. URL: <https://habrahabr.ru/post/267349> (дата звернення: 03.12.2020).

**ДОДАТКИ**

**Додаток А**  
**Акти впровадження**



ЗАТВЕРДЖУЮ

Директор ТОВ «Слобожанський бекон»

М.І. Мартиненко

17 грудня 2019 р.

**АКТ**

**промислової апробації результатів дисертаційних досліджень  
Грибкова С.В на ТОВ «Слобожанський бекон»**

Ми, що нижче підписалися, директор Мартиненко М.І., інженер-технолог Фурта О.Ю. та представник Національного університету харчових технологій (НУХТ) к.т.н., доцент Грибков С.В. склали цей акт про те, що у виробничих умовах ТОВ «Слобожанський бекон» здійснено апробацію інформаційної технології для удосконалення процесу управління, що є науково-практичним результатом досліджень Грибкова С. В.

Метою промислової апробації є перевірка ефективності застосування інформаційної технології для удосконалення процесу управління у виробничих умовах та практична реалізація програмних модулів, математичних моделей, алгоритмів, баз даних для оптимізації формування календарно-оперативних планів виконання замовлень.

За результатами перевірки ефективності застосування інформаційної технології, що включає комбіновані алгоритми, забезпечує: формування альтернативних планів виготовлення продукції за короткий період часу; швидке реагування на виникнення позаштатних ситуацій; зменшення збитків; збільшення прибутку, скорочення витрат на виготовлення продукції; оптимальне використання виробничих та складських приміщень при збереженні сировини та готової продукції.

За результатами апробації керівництво ТОВ «Слобожанський бекон» вважає, що програмний продукт, розроблений на основі інформаційної технології, забезпечує підтримку функцій управління та може бути рекомендований до впровадження на підприємствах галузі.

Від ТОВ «Слобожанський бекон»

Від НУХТ

  
М.І. Мартиненко  
О.Ю. Фурта

  
С.В.Грибков

ЗАТВЕРДЖУЮ

Директор ТОВ "Продеко"

О. З. Гандзюк

" 12 " 20 19 р.



## АКТ

**впровадження результатів дисертаційної роботи Грибкова С.В  
на ТОВ "Продеко"**

Ми, що нижче підписалися, директор Гандзюк О.З., інженер-технолог Березовський І.В. та представник Національного університету харчових технологій (НУХТ) к.т.н., доцент Грибков С.В. склали цей акт про те, що у виробничих умовах ТОВ "Продеко" здійснено апробацію інформаційних технологій прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів, що є науково-практичним результатом досліджень Грибкова С. В.

Грибков Сергій Віталійович отримав ряд прикладних наукових результатів з метою розробки інформаційних технологій прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів, що направлено на удосконалено методів управління в умовах ризику та невизначеності. Ним зроблено, зокрема, наступне:

1. Розроблено математичну модель планування виконання замовлень з урахуванням основних особливостей діяльності підприємств харчової галузі. Створена математична модель дозволяє коригувати та оцінювати ефективність виконання замовлень в залежності від об'єктивних та суб'єктивних переваг наданих ОПР, а також забезпечує як врахування так і виключення певних часткових критеріїв в залежності від певної ситуації.

2. Розроблена інформаційна технологія планування виконання замовлень забезпечує створення та коригування оперативно-календарних планів за рахунок комбінування алгоритмів мурашиної колонії, сірих вовків та генетичного

За результатами апробації керівництво ТОВ "Продеко" вважає, що отримані в ході виконання дослідження результати дисертаційної роботи можуть бути рекомендовані до впровадження на підприємствах галузі в якості додаткових модулів до єдиної інформаційної системи підприємства.

від ТОВ "Продеко"

О.З. Гандзюк

І.В. Березовський

від НУХТ

С.В.Грибков



ТОВ «КОЗЯТИНСЬКИЙ М'ЯСОКОМБІНАТ»

Україна, 22100, Вінницька обл.  
м.Козятин вул., Довженка, 33  
Тел.: +380 (4342) 2-2242 Факс: +380 (4342) 2-1129

ЗАТВЕРДЖУЮ  
Директор ТОВ  
«Козятинський м'ясокомбінат»  
Зікрань М.М.  
2020 р.

### АКТ

#### промислової апробації результатів дисертаційних досліджень Грибкова С.В. на ТОВ «Козятинський м'ясокомбінат»

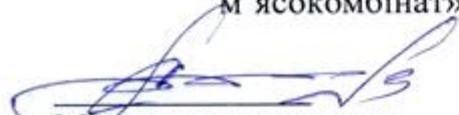
Ми, що нижче підписалися, менеджер з розвитку Рудь М.С.. та представник Національного університету харчових технологій (НУХТ) к.т.н., доцент Грибков С.В. склали цей акт про те, що у виробничих умовах ТОВ «Козятинський м'ясокомбінат» здійснено апробацію науково-практичних результатів дисертаційного дослідження Грибкова С.В., а саме модифіковані методи та алгоритми, інформаційна технологія прийняття рішень для розв'язання задачі планування виконання замовлень.

Метою промислової апробації є практична перевірка ефективності застосування модифікованих методів та алгоритмів, а також інформаційної технології прийняття рішень для розв'язання задачі планування виконання замовлень у виробничих умовах.

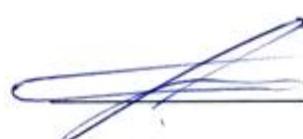
Результати перевірки застосування модифікованих методів та алгоритмів, а також інформаційної технології прийняття рішень показали свою ефективність у виробничих умовах.

Керівництво ТОВ «Козятинський м'ясокомбінат» вважає, що запропоновані результати дисертаційного дослідження Грибкова С.В., а саме модифіковані методи та алгоритми, інформаційна технологія прийняття рішень для розв'язання задачі планування, повністю задовольняють виробничим потребам та можуть бути рекомендовані до широкого впровадження на підприємствах харчової галузі.

Від ТОВ «Козятинський  
м'ясокомбінат»

  
Менеджер з розвитку,  
к.т.н. Рудь М.С.

Від НУХТ

  
С.В. Грибков

ЗАТВЕРДЖУЮ

Директор

ТОВ «Лайм Системс»

С. А. Гладков

«18» січня 2021 р.



## АКТ

**апробації результатів дисертаційних досліджень Грибкова С.В.**

Ми, що нижче підписалися, директор Гладков С.А. та представники Національного університету харчових технологій (НУХТ) к.т.н., доцент Грибков С.В. склали цей акт про те, що у виробничих умовах ТОВ «Лайм Системс» здійснено апробацію інформаційних технологій прийняття рішень для розв'язання задачі планування виконання замовлень, що є науково-практичним результатом дисертаційних досліджень Грибкова С.В.

Метою промислової апробації є практична реалізація моделей та алгоритмів прийняття рішень для розв'язання задачі планування виконання замовлень і перевірка ефективності застосування створених інформаційних технологій у виробничих умовах.

Результати перевірки застосування інформаційних технологій прийняття рішень у виробничих умовах показали ефективне планування виконання замовлень, що обумовило підвищення техніко-економічних показників підприємства.

За результатами апробації керівництво ТОВ «Лайм Системс» вважає, що запропоновані інформаційні технології повністю задовольняють виробничим потребам та можуть бути рекомендовані до широкого впровадження на різних підприємствах.

Від ТОВ «Лайм Системс»

Від НУХТ

С.А. Гладков

С.В. Грибков





**АКТ**

**промислової апробації результатів дисертаційних досліджень  
Грибкова С. В. на ТОВ «ФЕРАКС»**

Ми, що нижче підписалися, головний технолог Шафранович В.С. та представник Національного університету харчових технологій (НУХТ) к.т.н., доцент Грибков С. В. склали цей акт про те, що у виробничих умовах ТОВ «ФЕРАКС» проведено апробацію науково-практичних результатів дисертаційного дослідження Грибкова С.В.

Метою промислової апробації є практична перевірка ефективності застосування інформаційної технології прийняття рішень на основі модифікованих методів та алгоритмів для розв'язання задач планування виконання замовлень у виробничих умовах.

Результати перевірки застосування інформаційної технології прийняття рішень на основі модифікованих методів та алгоритмів у виробничих умовах показали ефективне планування виконання замовлень, що обумовило підвищення техніко-економічних показників підприємства.

За результатами апробації керівництво ТОВ «ФЕРАКС» вважає, що запропонована інформаційна технологія повністю задовольняє виробничим потребам та може бути рекомендована до широкого впровадження на підприємствах харчової галузі.

Від ТОВ «ФЕРАКС»



С. Шафранович

Від НУХТ

С.В. Грибков

ЗАТВЕРДЖУЮ

Директор з інформаційних  
технологій

ПрАТ «ОБОЛОНЬ»  
Пшеничний О.М.

«18» лютого 2021 р.



### АКТ

#### впровадження результатів дисертаційних досліджень Грибкова С. В. на ПрАТ «ОБОЛОНЬ»

Ми, що нижче підписалися, директор з інформаційних технологій ПрАТ «ОБОЛОНЬ» Пшеничний О.М та представник Національного університету харчових технологій (НУХТ) доцент кафедри Інформаційних систем к.т.н. Грибков С. В. склали цей акт про наступне.

Грибковим С. В. була надана інформаційна технологія прийняття рішень для розв'язання задач планування виконання замовлень на основі модифікованих евристичних і еволюційних методів та алгоритмів.

Надані результати дисертаційного дослідження пройшли перевірку їх використання у виробничих умовах, підтвердили свою ефективність при плануванні виконання замовлень, і використовуються при вирішенні задач управління підприємством.

Передана інформаційна технологія може бути рекомендована для впровадження на підприємствах харчової галузі.

Директор з інформаційних  
технологій ПрАТ «ОБОЛОНЬ»



Пшеничний О.М.

Доцент кафедри Інформаційних  
систем НУХТ, к.т.н.

С.В. Грибков

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**



**ЗАТВЕРДЖУЮ**

**Ректор НУХТ**

**А. І. Українець**

*30.06.2019*

**АКТ**

**впровадження результатів науково-дослідної роботи у  
навчальний процес**

**Замовник** Національний університет харчових технологій

(найменування організації, установи)

Ректор Українець А. І.

(П.І.Б. керівника підприємства)

**Даним актом підтверджується, що результати прикладних наукових досліджень «Наукове обґрунтування та розроблення активних пакувальних систем харчових продуктів», 0118U003558**

(найменування виду роботи, найменування теми, № держ. реєстрації)

виконаної на кафедрі інформаційних систем

(найменування кафедри)

виконуваної з 01.04.2019 р. по 30.06.2019 р.

(термін виконання)

**впроваджені у навчальний процес кафедри інформаційних систем НУХТ**

(найменування структурного підрозділу, де здійснювалось впровадження)

**1. Вид впроваджених результатів: інформаційна технологія для розв'язання задачі планування виконання замовлень**

(технологія, обладнання, методики тощо)

**2. Форма впровадження: Розділи дисциплін. Матеріали для проведення лабораторних та лекційних занять.**

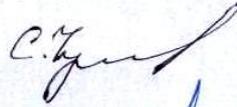
**3. Новизна результатів науково-дослідних робіт: якісно нові**

(піонерське, принципово нові, якісно нові, модифікація старих розробок)

**4. Перелік курсів і дисциплін, у рамках яких викладені результати НДР: Дисципліна «Клієнт-серверні технології» для здобувачів освіти ступеня магістр спеціальності 122 «Комп'ютерні науки» освітньо-професійних програм «Інформаційні управляючі системи та технології»**

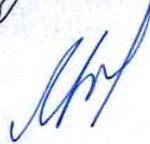
**5. Ефект від впровадження (соціальний, науково-технічний, тощо) Полягає у підвищенні ефективності планування виконання замовлень, що обумовлює підвищення техніко-економічних показників підприємства.**

**Завідувач кафедри  
інформаційних систем**  
д.т.н., старш. наук. співроб.,



С.М. Чумаченко

**Завідувач ПНДЛ**  
старш. наук. співроб., канд. техн. наук.



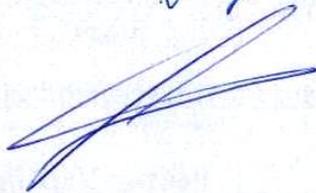
А.І. Маринін

**Керівник НДР**  
старш. наук. співроб., канд. техн. наук.



А.І. Маринін

**Виконавець**



С.В. Грибков

Міністерство освіти і науки України  
 Національний університет харчових технологій

ЗАТВЕРДЖУЮ:

Ректор НУХТ

Шевченко О.Ю.

20 10 р.



АКТ

впровадження результатів науково-дослідних, дослідно-конструкторських і дисертаційних робіт у навчальний процес вищих навчальних закладів

Замовник Національний університет харчових технологій  
(найменування організації, установи)

Ректор Шевченко О.Ю.  
(П.І.Б. керівника)

Даним актом підтверджується, що результати прикладних наукових досліджень за темою «Дослідження та впровадження інформаційних технологій у галузях харчової промисловості та освіти, № ДР 0117U003475  
(найменування виду роботи)

(найменування теми, № держ. реєстрації)

виконаної на кафедрі інформаційних систем  
(найменування кафедри)

виконуваної з 06.2017 по теперішній час  
(термін виконання)

впроваджені у навчальний процес кафедри інформаційних систем  
(найменування структурного підрозділу, де здійснювалось впровадження)

1. Вид впроваджених результатів інформаційна технологія для розв'язання задачі планування виконання замовлень

2. Форма впровадження: матеріали використовуються для проведення лекційних та лабораторних занять, а також при виконанні курсових та дипломних проектів.

3. Новизна результатів науково-дослідних робіт: Якісно нові та модифікація попередніх розробок.  
(піонерське, принципово нові, якісно нові, модифікація старих розробок)

4. Перелік курсів і дисциплін, у рамках яких викладені результати НДР: \_\_\_\_\_

Дисципліна «Управління ІТ-проектами» для здобувачів освіти ступеня бакалавр спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки», дисципліна «Клієнт-серверні технології» для здобувачів освіти ступеня магістр спеціальності 122 «Комп'ютерні науки» освітньо-професійних програм «Інформаційні управляючі системи та технології» та «Комп'ютерний еколого-економічний моніторинг».

**5. Соціальний і науково-технічний ефект** Полягає у підвищенні ефективності прийняття рішень проєктантами інформаційно-управляючих систем та фахівцями харчової промисловості для розв'язання задачі планування виконання замовлень по виготовленню продукції в умовах невизначеності та ризику. Застосування запропонованої технології забезпечить підвищення техніко-економічних показників виробництва.

**Завідувач кафедри  
інформаційних систем**  
д.т.н., старш. наук. співроб.



С.М. Чумаченко

**Керівник НДР**  
к.т.н., професор



В.В. Самсонов

**Виконавець**



С.В. Грибков



# ЛІГА РАДІОАМАТОРІВ УКРАЇНИ

## UKRAINIAN AMATEUR RADIO LEAGUE

Зареєстровано Міністерством юстиції України 3 вересня 1992 р. Свідоцтво № 286  
ПАТ КБ "Інтербанк", Київ, п/р N 2600 7 030113001, МФО 300216, ЄРДПОУ 14361078

UARL HQ, Office 29, 52/2 Peremohy Ave., Kyiv, 03057, Ukraine  
Phone: +38 (067) 463-21-71  
Ukrainian QSL-bureau UARL, PO Box 56, Kyiv-1, 01001, Ukraine

ЦК ТСО України, Штаб-квартира ЛРУ,  
пр-т Перемоги, 52/2, офіс 29, м. Київ, 03057, Укра  
Телефон: +38 (067) 463-21-71

№ 1

Від 26 09 2020

### ДОВІДКА

Видана доценту кафедри інформаційних систем Національного університету харчових технологій, к.т.н, Грибкову Сергію Віталійовичу, в тому, що результати його дисертаційних досліджень (інформаційна технологія прийняття рішень та відповідний програмний продукт) передані Київському міському Радіоклубу і використовуються для вирішення організаційно-управлінських задач при підготовці та проведенні змагань.

За результатами застосування запропонованих інформаційних технологій підвищилася ефективність планування підготовки та проведення змагань, що забезпечило мінімізацію відповідних витрат. Спільнота радіозв'язківців м. Києва рекомендує апробовані інформаційні технології к впровадженню на інших підприємствах для вирішення задач планування в умовах невизначеності та ризику

Довідка видана за проханням запитом Грибкова Сергія Віталійович.

Голова КРК:



Мошенський А.О.

## Додаток Б

## Список публікацій здобувача за темою дисертації

## Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Грибков С. В., Кононова В. О., Харкянен О. В. Оцінка засобів захисту інформаційних ресурсів. *Вісник Національного університету «Львівська політехніка». Комп'ютерні системи та мережі.* 2014. № 806. С. 99–105. Бази: *Google Scholar, Index Copernicus International.*
2. Грибков С. В., Загоровська Л. Г. Використання СА Erwin Model Manager для моделювання структури сховища даних. *Наукові праці НУХТ.* 2014. № 6, Т. 20. С. 125–130. Бази: *Index Copernicus, EBSCOhost, Google Scholar.*
3. Грибков С. В., Харкянен О. В., Логвин Т. В. Дослідження пакетів програмних модулів для ідентифікації динамічних об'єктів. *Технологічний аудит та резерви виробництва.* 2015. № 25 (5/2). С. 42–49. Бази: *EBSCO, Ifindr, Directory of Open Access Journals (DOAJ), EconBiz, IDEAS, ERIH PLUS (The European Reference Index for the Humanities and the Social Sciences), FSTA, OpenAIRE, Bielefeld Academic Search Engine (BASE), Index Copernicus Journals Master List, Наукова періодика України – проєкт Національної бібліотеки України імені В. І. Вернадського (НБУВ), Ulrich's Periodicals Directory, WorldCat, CrossRef.*
4. Грибков С. В., Литвинов В. В., Олійник Г. В. Задача планування виконання договорів та підходи до її ефективного вирішення. *Математичні машини і системи.* 2015. № 1. С. 61–70. Бази: *РИНЦ(eLIBRARY), CrossRef, Google Scholar, ВІНІТІ РАН.*
5. Hrybkov S. V., Oliinyk H. V. Modeling of the decision support system structure in the planning and controlling of contracts implementation. *Ukrainian Journal of Food Science.* 2015. N 3 (1). P. 123–130. Бази: *EBSCO, Chemical Abstracts Service Source Index (CASSI), Food Science and Technology Abstracts (FSTA), Index Copernicus International, Ulrichsweb, Directory of Open Access scholarly Resources (ROAD), Google Scholar.*

6. Грибков С. В., Харкянен О. В., Пархоменко І. І. Прогнозування стану технологічного обладнання харчового підприємства методом сингулярного спектрального аналізу. *Вісник інженерної академії України*. 2015. № 3. С. 87–91. *Бази: Innovative Journal Impact Factor (IJIF), Google Scholar*.
7. Грибков С. В., Харкянен О. В., М'якшило О. М. Моделювання процесу моніторингу та планування собівартості продукції багатоміноменклатурного харчового підприємства. *Наукові праці НУХТ*. 2015. № 6, Т. 21. С. 100–108. *Бази: Index Copernicus, EBSCOhost, Google Scholar*.
8. Грибков С. В., Загоровская Л. Г., Бондарь Н. П., Губеня В. А. Анализ и реинжиниринг процесса планирования закупок для предприятий ресторанного хозяйства. *Вестник Алмаатинского технологического университета*. 2015. №4 (109). С. 31-36. *Бази: CrossRef, РИИЦ(eLIBRARY)*.
9. Грибков С. В., Маковецька С. В., М'якшило О. М. Дослідження та математичне моделювання процесу постачання сировини на цукровий завод з врахуванням генетико-детермінованих властивостей цукрових буряків. *Наукові праці НУХТ*. 2016. № 6, Т. 22. С. 7–15. *Бази: Index Copernicus, EBSCOhost, Google Scholar*.
10. Грибков С. В., Литвинов В. А., Олійник Г. В. Обрання програмної платформи для побудови модуля безпеки web-орієнтованої системи підтримки прийняття рішень. *Вісник Національного університету «Львівська політехніка». Автоматика, вимірювання та керування*. 2016. № 852. С. 137–142. *Бази: РИИЦ(eLIBRARY), Google Scholar, Index Copernicus International*.
11. Грибков С. В., Павелко В. І., Заславський А. І., Дмитренко Д. С. Математичне моделювання процесу термічної обробки м'ясних виробів». *Харчова промисловість*. 2016. № 20. С. 115–120. *Бази: Google Scholar, Index Copernicus International*.
12. Грибков С. В., Олійник Г. В. Модифікований АСО алгоритм побудови календарного плану виконання договорів. *Математичне та комп'ютерне моделювання. Технічні науки*. 2017. № 15. С. 156–162. *Бази: Bielefeld Academic Search Engine (BASE), Citefactor, Cosmos Impact Factor, The Journals Impact Factor (JIF)*,

*General Impact Factor (GIF), Google Scholar, InfoBase Index, International Citation Index OF JOURNAL IMPACT FACTOR & INDEXING, Open Access Infrastructure for Research in Europe (OpenAIRE), Public Knowledge Project (PKP), ResearchBib, Scientific Indexing Services (SIS), WorldCat.*

13. Грибков С. В., Маковецька С. В. Функціональне моделювання організації та управління забезпечення сировиною цукрового заводу. *Науковий вісник Національного університету біоресурсів і природокористування України. Техніка та енергетика АПК.* 2017. № 286. С. 100–111. Бази: *Directory of Open Access Journals (DOAJ), CrossRef.*

14. Грибков С. В., Олійник Г. В. Використання JWT-маркерів для аутентифікації та авторизації користувачів у web-додатках. *Вісник Національного університету «Львівська політехніка». Автоматика, вимірювання та керування.* 2017. № 880. С 60–67. Бази: *РИНЦ(eLIBRARY), Google Scholar, Index Copernicus International.*

15. Hrybkov S. V., Breus N. M., Polischuk G. Ye. Hybrid expert system to model the ice cream recipes. *Ukrainian Journal of Food Science.* 2017. № 5 (2). P. 294–304. Бази: *EBSCO, Chemical Abstracts Service Source Index (CASSI), Food Science and Technology Abstracts (FSTA), Index Copernicus International, Ulrichsweb, Directory of Open Access scholarly Resources (ROAD), Google Scholar.*

16. Hrybkov S. V., Kharkianen O. V., Myakshylo O. M., Kostikov M. P. Development of information technology for supporting the process of adjustment of the food enterprise assortment. *Eastern-european journal of enterprise technologies.* 2018. № 3 (91). P. 77–87. Бази: *Scopus, CrossRef, Index Copernicus International, Applied Science & Technology Source, Chemical Abstracts Plus (CAplus), Computers & Applied Sciences Complete, Directory of Open Access Journals (DOAJ), Google Scholar, Directory of Open Access scholarly Resources (ROAD), MIAR, Ifindr, OpenAIRE (Open Access Infrastructure for Research in Europe), Polska Bibliografia Naukowa (PBN), WorldCat, Scientific Periodicals of Ukraine (SPU VNLU), Bielefeld Academic Search Engine (BASE), China National Knowledge Infrastructure (CNKI), Scilit, WorldWideScience.org, Zeitschriftendatenbank (ZDB), ResearchBib, EuroPub,*

*JournalTOCs, Dimensions, Scientific Indexing Services (SIS), The General Impact Factor (GIF), Genamics JournalSeek, SHERPA RoMEO, Ulrich's Periodicals Directory, CORE (COncnecting REpositories), Neliti, Socionet, Researchgate, ERIH PLUS, Lens, Korea Open Access Platform for Researchers (KOAR), Microsoft Academic, Paperity, Root Society, Wizdom.ai, KindCongress.*

17. Грибков С. В., Струзік В. А., Литвин А. О. Дослідження методів і підходів проведення рефакторингу баз даних. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2018. № 30. С. 151–155. Бази: РИИЦ(eLIBRARY), *Universal Impact Factor, Open Academic Journals Index.*

18. Hrybkov S. V., Litvinov V. V., Oliinyk H. V. Web-oriented decision support system for planning agreements execution. *Eastern-european journal of enterprise technologies*. 2018. № 2 (93). P. 13–24. Бази: *Scopus, CrossRef, Index Copernicus International, Applied Science & Technology Source, Chemical Abstracts Plus (CAplus), Computers & Applied Sciences Complete, Directory of Open Access Journals (DOAJ), Google Scholar, Directory of Open Access scholarly Resources (ROAD), MIAR, Ifindr, OpenAIRE (Open Access Infrastructure for Research in Europe), Polska Bibliografia Naukowa (PBN), WorldCat, Scientific Periodicals of Ukraine (SPU VNLU), Bielefeld Academic Search Engine (BASE), China National Knowledge Infrastructure (CNKI), Scilit, WorldWideScience.org, Zeitschriftendatenbank (ZDB), ResearchBib, EuroPub, JournalTOCs, Dimensions, Scientific Indexing Services (SIS), The General Impact Factor (GIF), Genamics JournalSeek, SHERPA RoMEO, Ulrich's Periodicals Directory, CORE (COncnecting REpositories), Neliti, Socionet, Researchgate, ERIH PLUS, Lens, Korea Open Access Platform for Researchers (KOAR), Microsoft Academic, Paperity, Root Society, Wizdom.ai, KindCongress.*

19. Грибков С. В., Бойко Р. О. Мережеві структури при керуванні складними організаційно-технічними (технологічними) системами. *Харчова промисловість*. 2019. № 25. С. 116–123. Бази: *Google Scholar, Index Copernicus International.*

20. Струзік В. А., Грибков С. В., Чобану В. В. Визначення місця рефакторингу в сучасних методологіях розробки інформаційних систем. *Вчені*

записки Таврійського національного університету імені В. І. Вернадського. Технічні науки. 2019. № 5, Т. 30 (69). С. 173–177. Бази: РИИЦ(eLIBRARY), Index Copernicus International.

21. Hrybkov S. V., Breus N. M., Seidykh O. L., Polischuk G. Ye. Development of mathematical apparatus of the expert system for modelling ice cream recipes with specified quality parameters. *Science and innovation*. 2019. Vol. 15 (5). P. 57–66. Бази: Web of Science, Scopus, Directory of Open Access Journals (DOAJ), CrossRef, EuroPub, Ulrich's Periodicals Directory, Index Copernicus International.

22. Hrybkov S. V., Kharkianen O. V., Ovcharuk V. O., Ovcharuk I. Development of information technology for planning order fulfillment at a food enterprise. *Eastern-european journal of enterprise technologies*. 2020. N 3 (103). P. 62–73. Бази: Scopus, CrossRef, Index Copernicus International, Applied Science & Technology Source, Chemical Abstracts Plus (CAplus), Computers & Applied Sciences Complete, Directory of Open Access Journals (DOAJ), Google Scholar, Directory of Open Access scholarly Resources (ROAD), MIAR, Ifindr, OpenAIRE (Open Access Infrastructure for Research in Europe), Polska Bibliografia Naukowa (PBN), WorldCat, Scientific Periodicals of Ukraine (SPU VNLU), Bielefeld Academic Search Engine (BASE), China National Knowledge Infrastructure (CNKI), Scilit, WorldWideScience.org, Zeitschriftendatenbank (ZDB), ResearchBib, EuroPub, JournalTOCs, Dimensions, Scientific Indexing Services (SIS), The General Impact Factor (GIF), Genamics JournalSeek, SHERPA RoMEO, Ulrich's Periodicals Directory, CORE (COncnecting REpositories), Neliti, Socionet, Researchgate, ERIH PLUS, Lens, Korea Open Access Platform for Researchers (KOAR), Microsoft Academic, Paperity, Root Society, Wizdom.ai, KindCongress.

23. Грибков С. В., Литвинов В. В., Олійник Г. В. Інструментальна модель веб-орієнтованої програмної реалізації підсистеми підтримки прийняття рішень у складі програмного комплексу ситуаційного центру. *Математичні машини і системи*. 2020. № 1. С. 73–81. Бази: РИИЦ(eLIBRARY), CrossRef, Google Scholar, ВІНІТИ РАН.

24. Грибков С. В., Струзік В. А., Чобану В. В. Категорія рефакторинг доступу. *Наукові праці НУХТ*. 2020. № 2, Т. 26. С. 31–49. Бази: *Index Copernicus, EBSCOhost, Google Scholar*.
25. Грибков С. В., Струзік В. А., Чобану В. В. Специфікація семантичного версіонування бази даних. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Технічні науки*. 2020. № 2, Т. 31 (70). С. 196–201. Бази: *РИНЦ(eLIBRARY), Index Copernicus International*.
26. Грибков С. В., Сєдих О. Л. Розробка СППР управління виробництвом. *Almanahul SWorld*. 2020. № 4 (1). С. 52–56. Бази: *Google Scholar, Index Copernicus International*.
27. Hrybkov S. V., Seidykh O. L. Manufacturing management methods at the expense of using bat algorithm. *Modern engineering and innovative technologies*. 2020. Issue 12 (3). P. 68–73. Бази: *Google Scholar, Index Copernicus International*.
28. Hrybkov S. V., Seidykh O. L. Hybrid algorithm based on fish school search and grey wolf optimizer algorithms for food enterprise management. *European science review*. Vienna: Premier Publishing s.r.o., 2020. N 7–8. P. 19–22. Бази: *РИНЦ(eLIBRARY), CrossRef, CyberLeninka, EBSCOhost, Google Scholar, Index Copernicus International*.
29. Hrybkov S. V., Struzik V. A., Chobanu V. V. Evolution of refactoring. *The Austrian Journal of Technical and Natural Sciences*. Vienna: Premier Publishing s.r.o., 2020. N 7–8 P. 11–16. Бази: *РИНЦ(eLIBRARY), CrossRef, CyberLeninka, EBSCOhost, Google Scholar, OpenAIRE, Ulrich's Periodicals Directory, WorldCat*.
30. Hrybkov S. V., Kharkianen O. V., Gladka Y. U. Finding the best versions of schedule for order fulfillment at food companies. *MIND journal Semiannual*. Poland : Bielsko-Biała. 2020. N 9. P. 1–12. Бази: *Index Copernicus International, BuzHul*.
31. Hrybkov S. V., Seidykh O. L. Solution of control tasks using bat algorithm modifications. *Modern engineering and innovative technologies*. 2020. Issue 13 (2). P. 60–64. Бази: *Google Scholar, Index Copernicus International*.

32. Грибков С. В., Сєдих О. Л. Розробка модифікованих ітераційних алгоритмів для розв'язання задачі формування оптимальних варіантів розкладу виконання замовлень. *Харчова промисловість*. 2020. № 27. С. 126–137. Бази: *Google Scholar, Index Copernicus International*.

### **Наукові праці, які засвідчують апробацію матеріалів дисертації:**

33. Грибков С. В., Олійник Г. В. Високопродуктивні пакети завантаження даних до сховища даних. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 81 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 23–24 квітня 2015 р.). Київ: НУХТ, 2015. Ч. 2. С. 356.

34. Грибков С. В., Нерадович М. С., Олійник Г. В. Використання методу бджолиного рою при формуванні оптимального розкладу виконання замовлень. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами*: матеріали II міжнар. наук.-техн. конф. Київ : НУХТ, 2015. С. 194.

35. Грибков С. В., Воловик О. О. Web-орієнтована інформаційна система підтримки контролю якості сировини та готової продукції. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами*: матеріали III міжнар. наук.-техн. конф. Київ: НУХТ, 2016. С. 219.

36. Hrybkov S. V., Ollinyk H. V., Litvinov V. V. Decision support system for the contracts execution planning based on genetic algorithms and collective mind algorithms. *Food Science for Well-being (CEFood 2016)*: Book of Abstracts of 8th Central European Congress on Food (Kyiv, 23–26 May 2016). Kyiv: NUFT, 2016. P. 58.

37. Грибков С. В., Буряченко М. М., Харкянен О. В. Використання еволюційних методів для формування виробничої програми харчового підприємства. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 83 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 5–6 квітня 2017 р.). Київ, 2017. С. 285.

38. Грибков С. В., Буряченко М. М. Використання середовища R для аналізу та виявлення причин виникнення некондиційної продукції у макаронному виробництві. *Обчислювальний інтелект (результати, проблеми, перспективи)*: матеріали 4 міжнар. наук.-практ. конф. (м. Київ, 16–18 травня 2017 р.). Київ-Черкаси: ВПЦ «Київський університет», 2017. С. 204–205.

39. Грибков С. В., Олійник Г. В. Розробка web-орієнтованої системи підтримки прийняття рішень при плануванні виконання договорів. *Автоматика–2017*: матеріали XXIV міжнар. конф. з автоматичного управління (м. Київ, 13–15 вересня 2017 р.). Київ: НУБіП, 2017. С. 192–193.

40. Грибков С. В., Олійник Г. В. Web-орієнтована система підтримки прийняття рішень планування виконання договорів. *Проблеми інформатизації*: міжнар. наук.-техн. конф.: зб. тез. Київ: ДУТ, 2017. С. 239.

41. Грибков С. В., Струзік В. А., Литвин А. О. Використання еволюційного підходу при рефакторингу баз даних. *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами*: матеріали IV міжнар. наук.-техн. Internet-конф. (м. Київ, 22 листопада 2017 р.). Київ: НУХТ, 2017. С. 281.

42. Hrybkov S. V., Struzik V. A., Lytvyn A. O., Chobanu V. V. Study Of The Methods And Approaches Of Databases Refactoring. *Програмовані логічні інтегральні схеми та мікропроцесорна техніка в освіті і виробництві*: зб. тез міжнар. наук.-практ. семінару молодих вчених та студентів. Луцьк: Вежа-Друк, 2018. С. 12–13.

43. Грибков С. В., Ольшевська М. А. Підсистема підтримки діяльності начальника виробництва макаронного підприємства ТОВ «Продеко». *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті*: матеріали 84 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 23–24 квітня 2018 р.). Київ: НУХТ, 2018. Ч. 2. С. 337.

44. Грибков С. В., Фурта О. О. Використання інтелектуального аналізу даних для планування виготовлення продукції на ТОВ «Слобожанський бекон». *Проблеми інформатизації*: матеріали XII міжнар. наук.-техн. конф. Київ: ДУТ,

НТУ; Полтава: ПНТУ; Катовице: КЕУ; Париж: Університет Париж VII Венсент-Сен-Дені; Вільнюс: ВДТУ; Харків: ХНДІТМ; Білорусь: БДАЗ, 2018. С. 28–29.

45. Грибков С. В., Олійник Г. В., Чернишов І. Використання модифікованих методів мурашиного алгоритму для розв'язку задачі формування розкладу. *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті: матеріали 85 Ювілейної міжнар. наук. конф. молодих учених, аспірантів і студентів, присвяченої 135-річчю НУХТ (м. Київ, 11–12 квітня 2019 р.)*. Київ : НУХТ, 2019. Ч. 2. С. 435.

46. Грибков С. В., Ольшевська М. А. Створення інформаційної системи підтримки управління виробництва макаронних виробів. *LXXV наук. конф. професорсько-викладацького складу, аспірантів, студентів та співробітників відокремлених структурних підрозділів університету: зб. тез*. Київ: НТУ, НУХТ, 2019. Т. 2. С. 462.

47. Грибков С. В., Фурта О. О. Підсистема підтримки прийняття рішення головного технологу ТОВ «Слобожанський бекон». *Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами: матеріали VI міжнар. наук.-техн. Internet-конф. (м. Київ, 20 листопада 2019 р.)*. Київ: НУХТ, 2019. С. 176.

48. Грибков С. В., Чорнобай К. Ю. Дослідження та проектування підсистеми управління молочним балансом молокозаводу. *Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій: наук. праці Другої міжнар. наук.-практ. конф. (м. Київ, 19 грудня 2019 р.)*. Київ: НУХТ, 2019. С. 298–302.

49. Грибков С. В., Сєдих О. Л., Ольшевська М. А. Дослідження та розроблення СППР управління виробництвом ТОВ «Продеко». *Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті: матеріали 86 міжнар. наук. конф. молодих учених, аспірантів і студентів (м. Київ, 2–3 квітня 2020 р.)*. Київ: НУХТ, 2020. Ч. 2. С. 403.

50. Hrybkov S. V., Seidykh O. L. Combined algorithm based on algorithms for konyak fish and gray wolves for solving complex problems. *Sworld-Us conference*

*proceedings: conf. proceed. of Global scien. and educ. in the modern real.* Washington, USA: «ISE&E» & SWorld in conjunction with KindleDP Seattle, 2020. P. 54–57.

**Наукові праці, які додатково відображають наукові результати дисертації:**

51. Грибков С. В., Балашева А. М. Дослідження і створення веб-орієнтованої СППР при управлінні розвитку компанії ТОВ «Доктор Філін». *Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій*: матеріали I міжнар. наук.-практ. конф. (м. Краків, 7–8 лютого 2019 р.). Краків, Польща, 2019. С. 120–123.

52. Грибков С. В., Литвинов В. В., Олійник Г. В. До проблеми типізації проектних рішень у сфері програмної реалізації підсистем ситуативного центру. *Цифрова економіка та інформаційні технології*: матеріали Міжнар. наук.-практ. конф. (м. Київ, 15–16 квітня 2020 р.). Київ: ДУІТ, 2020. С. 37–38.

53. Грибков С. В., Литвинов В. В., Олійник Г. В. Инструментальная модель подсистемы поддержки принятия решений. *Математичне та імітаційне моделювання систем*: матеріали XV міжнар. наук.-практ. конф. (м. Чернігів, 29 червня – 01 липня 2020 р.). Чернігів, 2020. С. 212–214.

## Додаток В

### Функціональні моделі

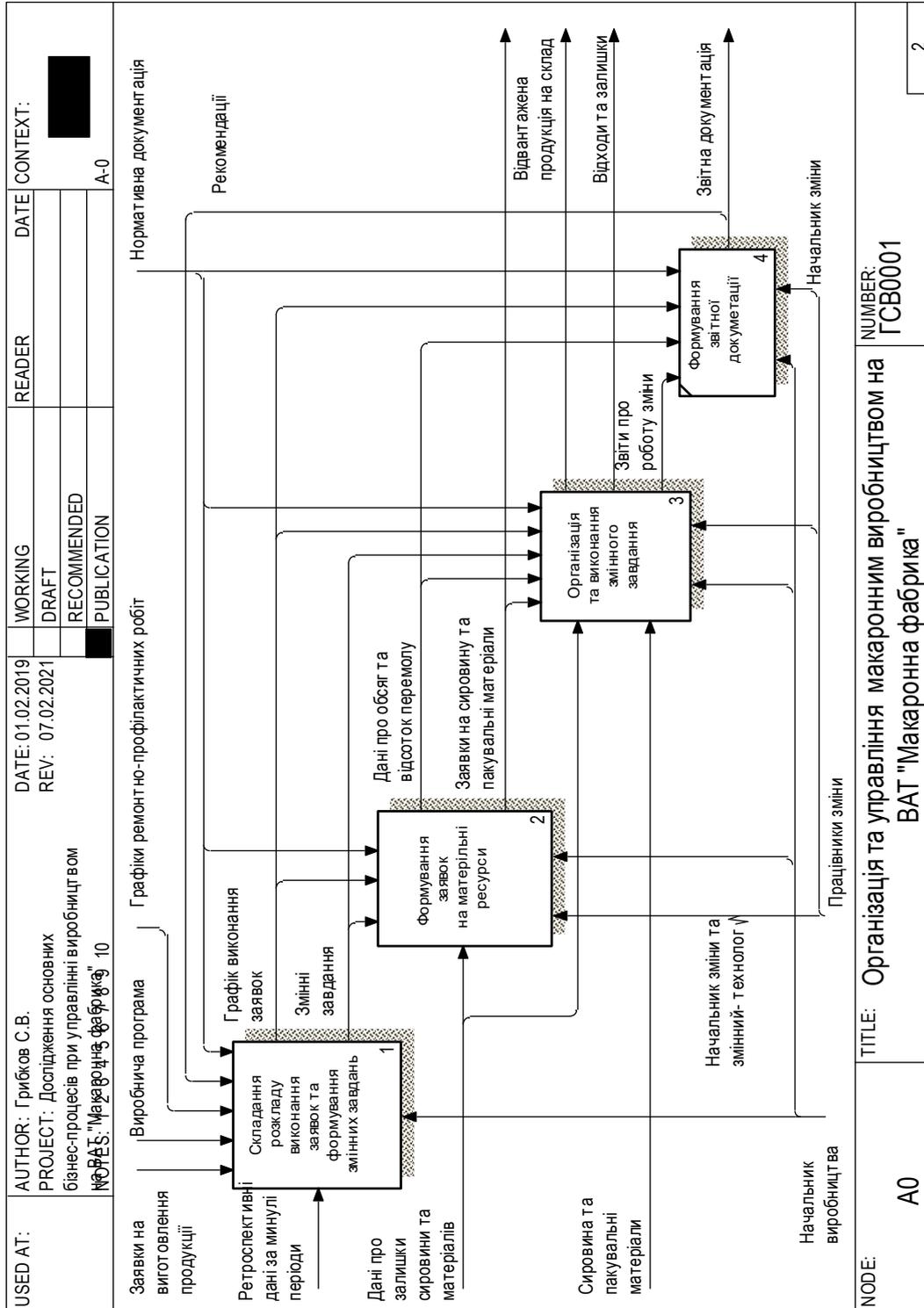


Рисунок В.1 – Декомпозиція процесу «Організація та управління макаронним виробництвом на ВАТ «Макаронна фабрика»

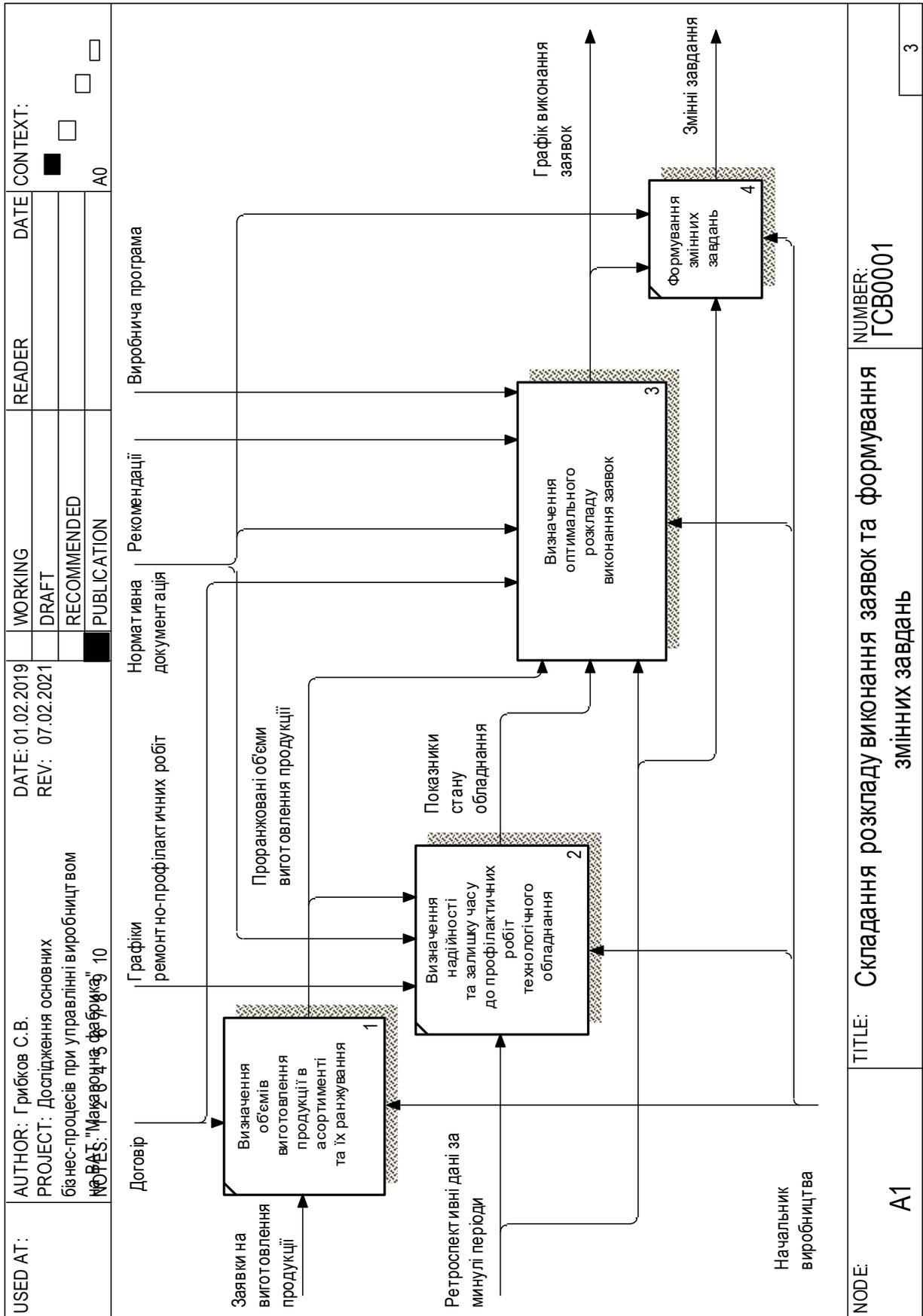


Рисунок В.2 – Декомпозиція процесу «Складання розкладу виконання заявок та формування змінних завдань»

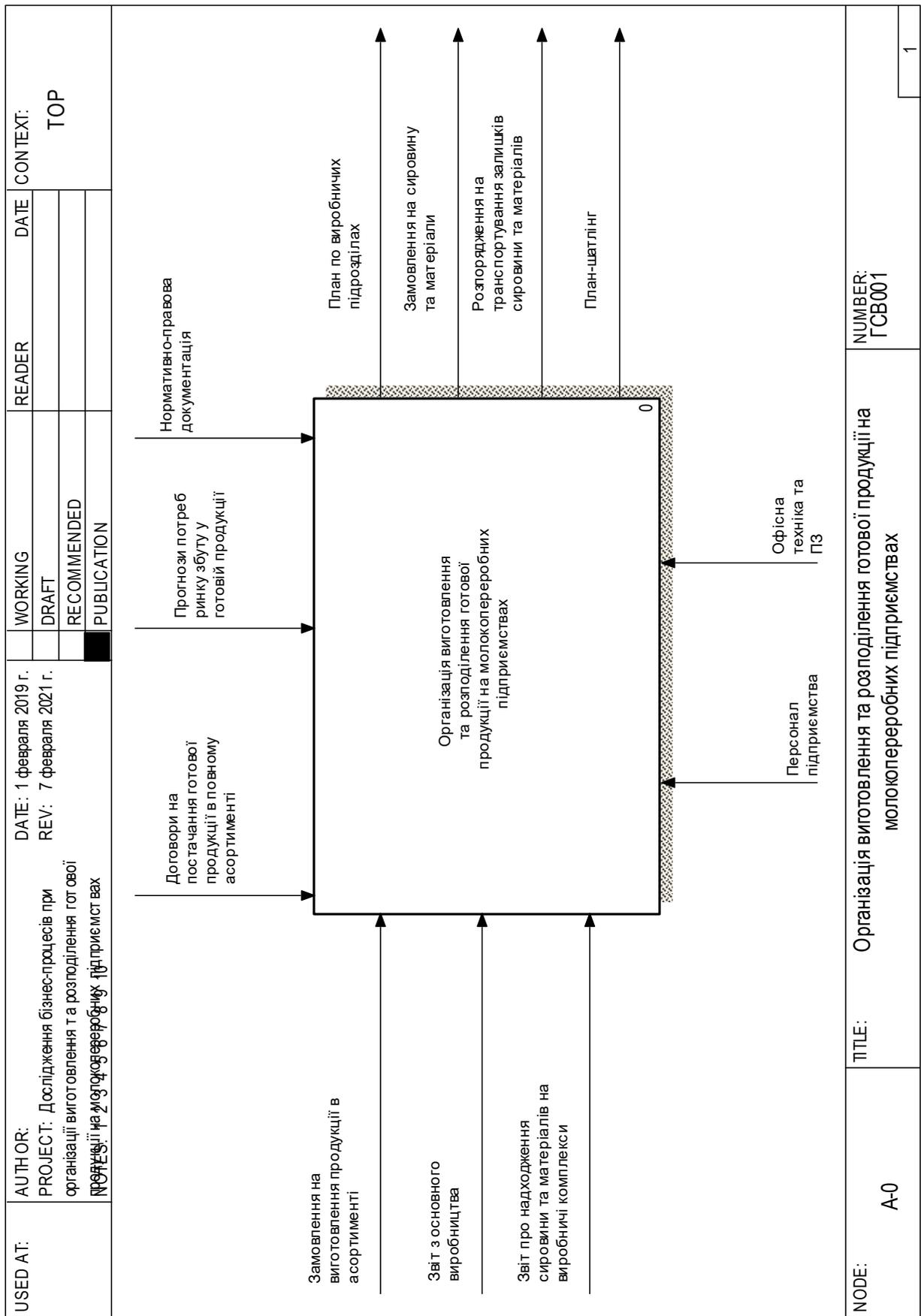
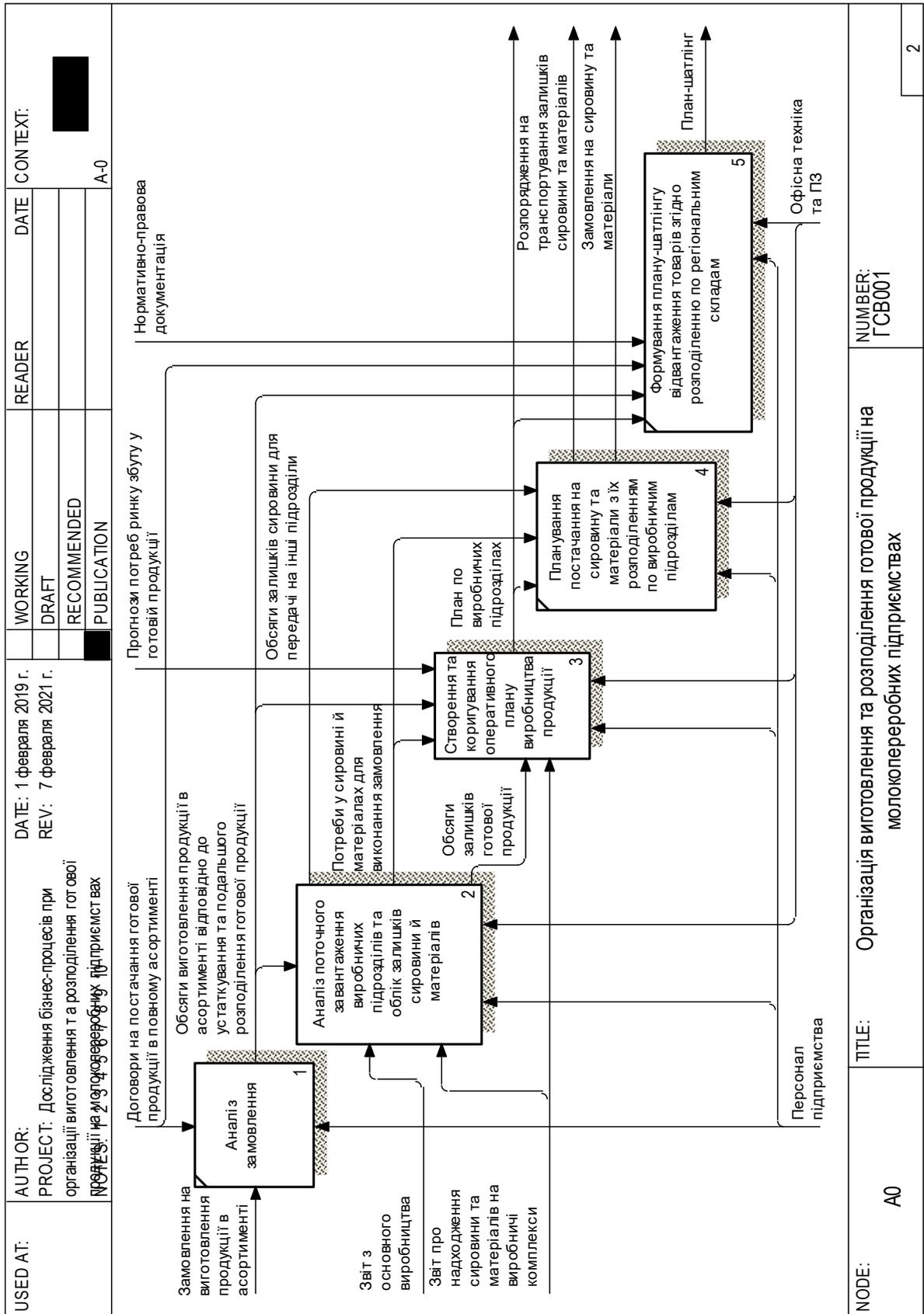


Рисунок В.3 – Контекстна діаграма «Організація виготовлення та розподілення готової продукції на молокопереробних підприємствах»



NUMBER:  
ГСВ001

TITLE: Організація виготовлення та розподілення готової продукції на молокопереробних підприємствах

NOTE: A0

2

Рисунок В.4 – Декомпозиція функції «Організація виготовлення та розподілення готової продукції на молокопереробних підприємствах»

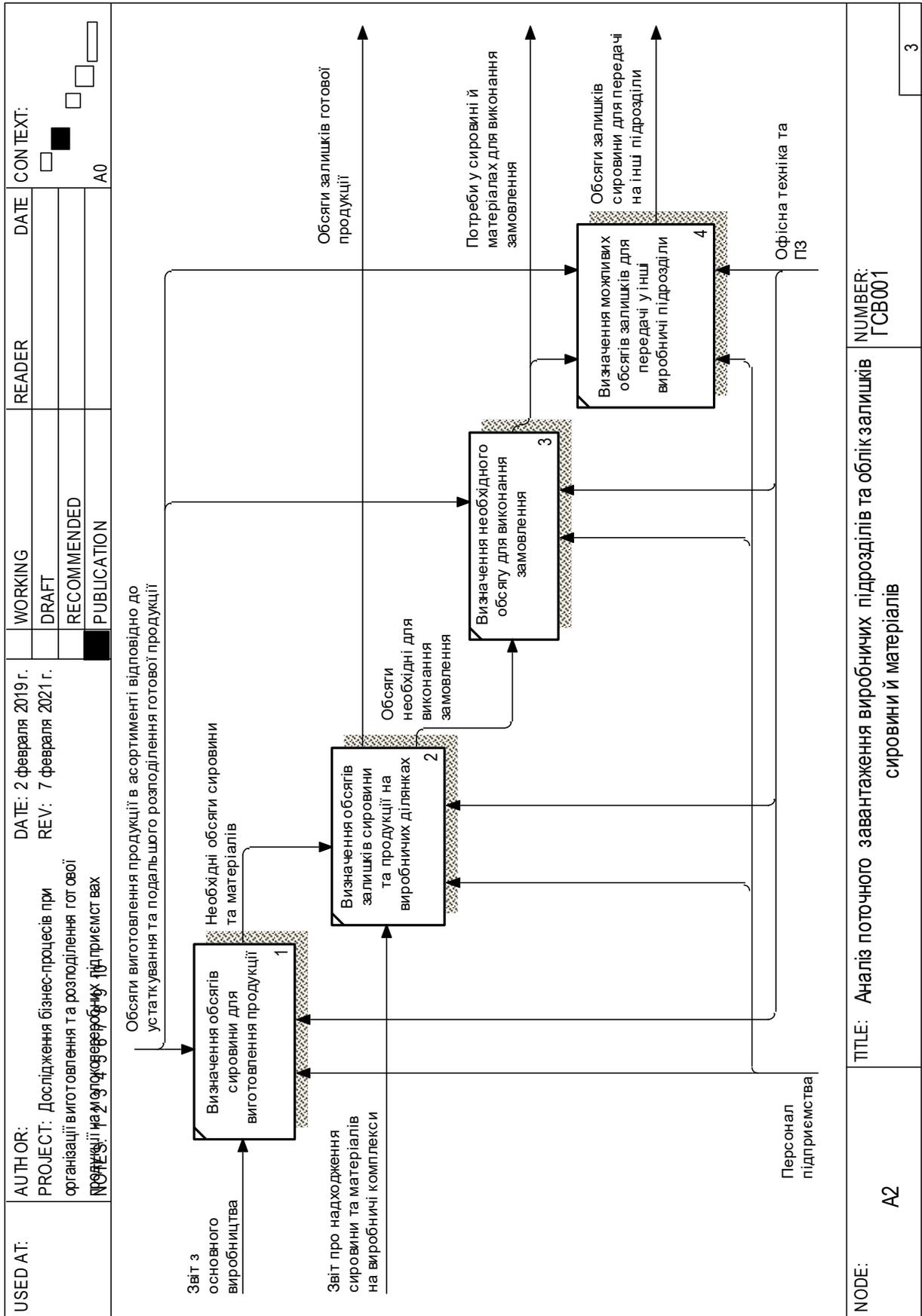


Рисунок В.5 – Декомпозиція процесу «Аналіз поточного завантаження виробничих підрозділів та облік залишків сировини й матеріалів»

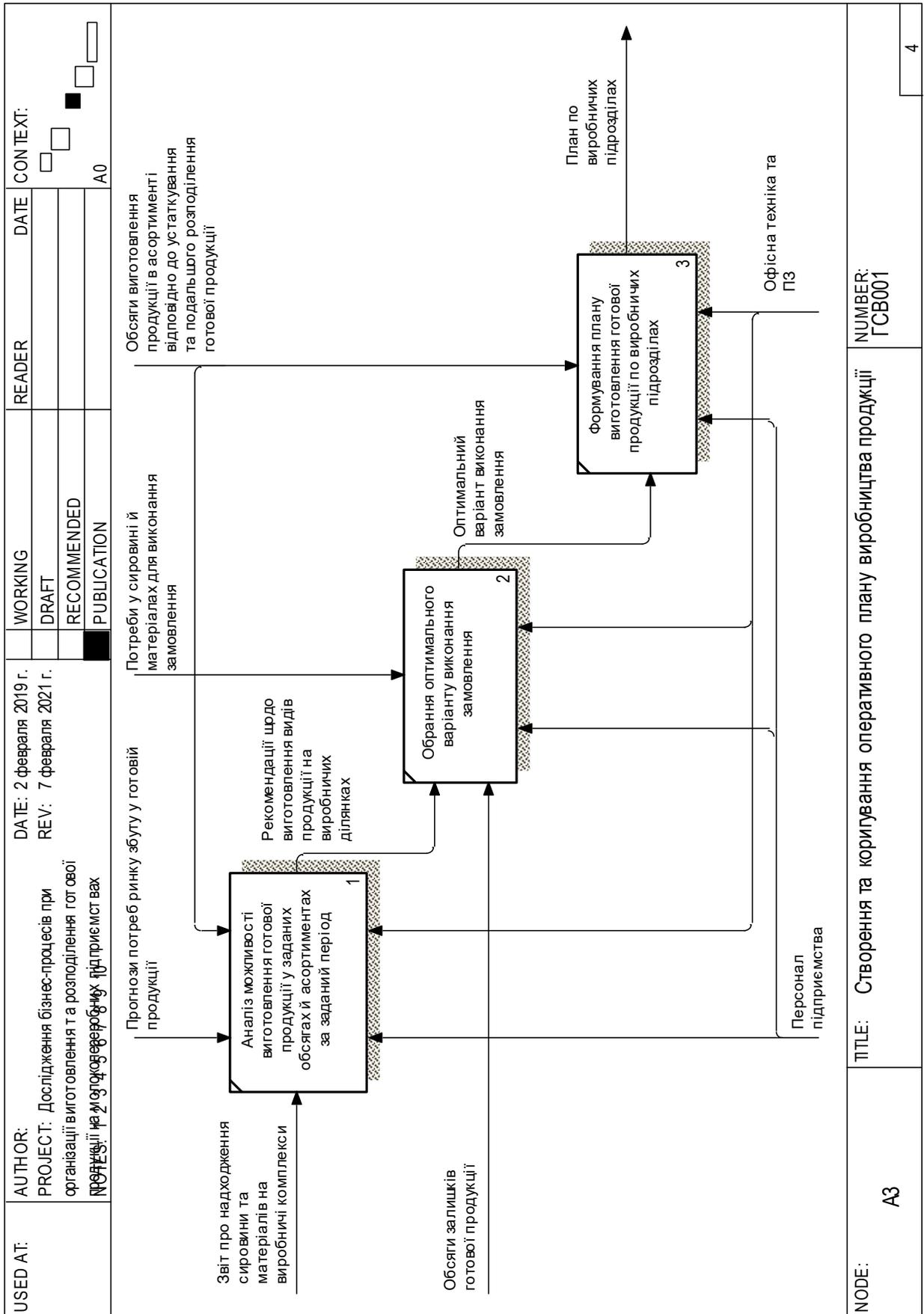


Рисунок В.6 – Декомпозиція процесу «Створення та коригування оперативного плану виробництва продукції»

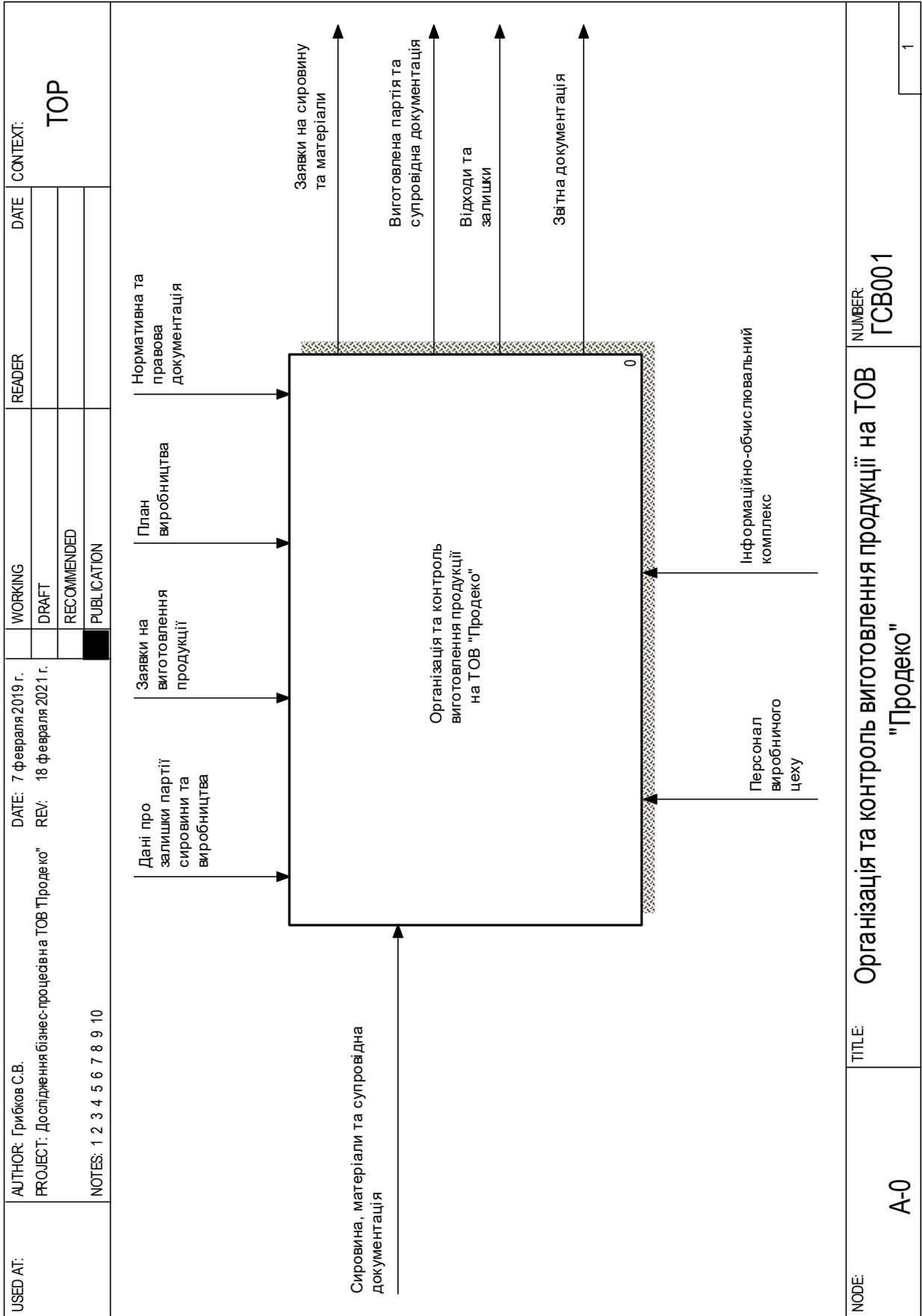


Рисунок В.7 – Контекстна діаграма «Організація та контроль виготовлення продукції на ТОВ «Продеко»»

NUMBER:  
ГСВ001

TITLE: Організація та контроль виготовлення продукції на ТОВ "Продеко"

NODE: А-0

1

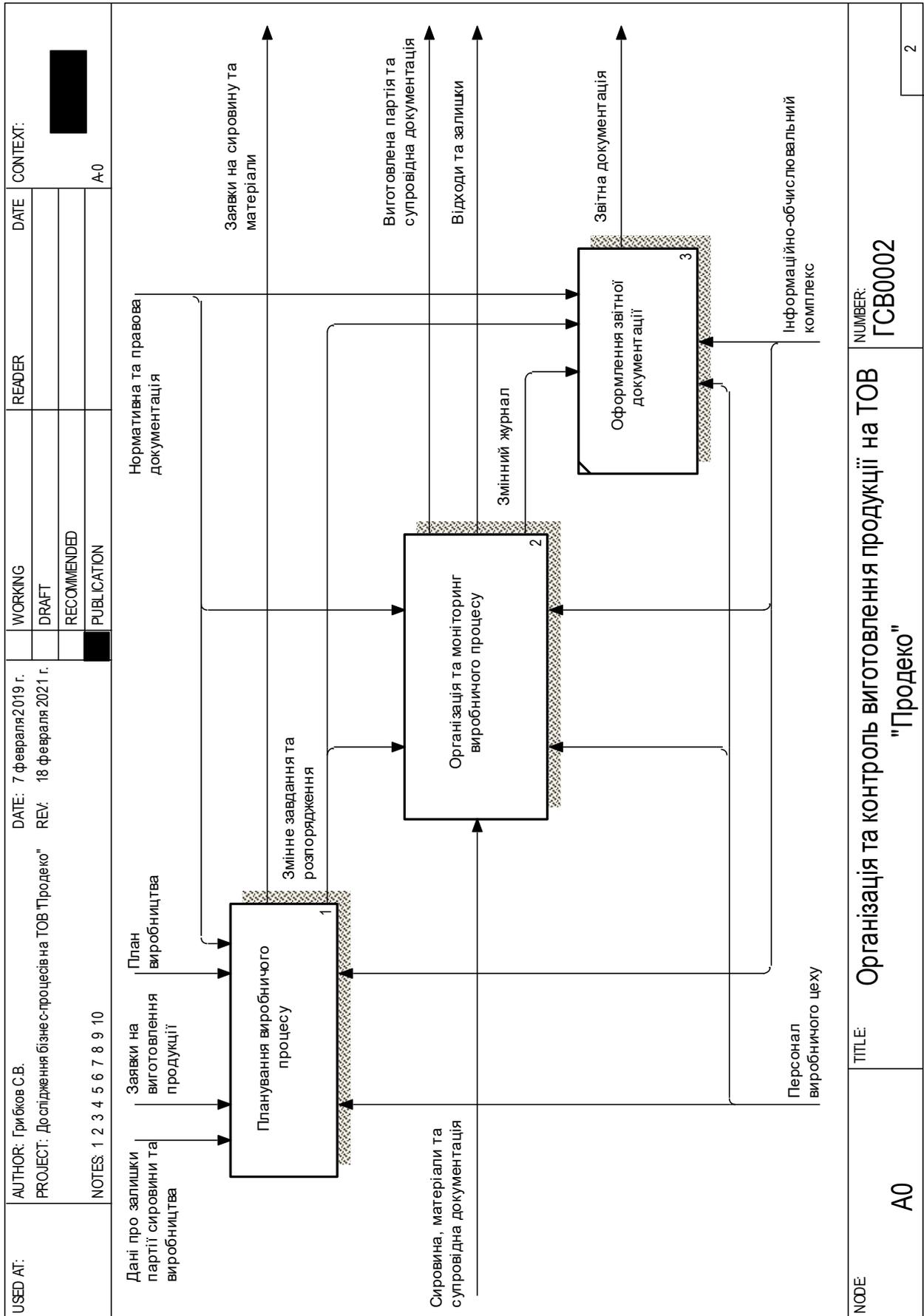


Рисунок В.8 – Діаграма декомпозиції першого рівня функціональної моделі «Організація та контроль виготовлення продукції на ТОВ «Продеко»»

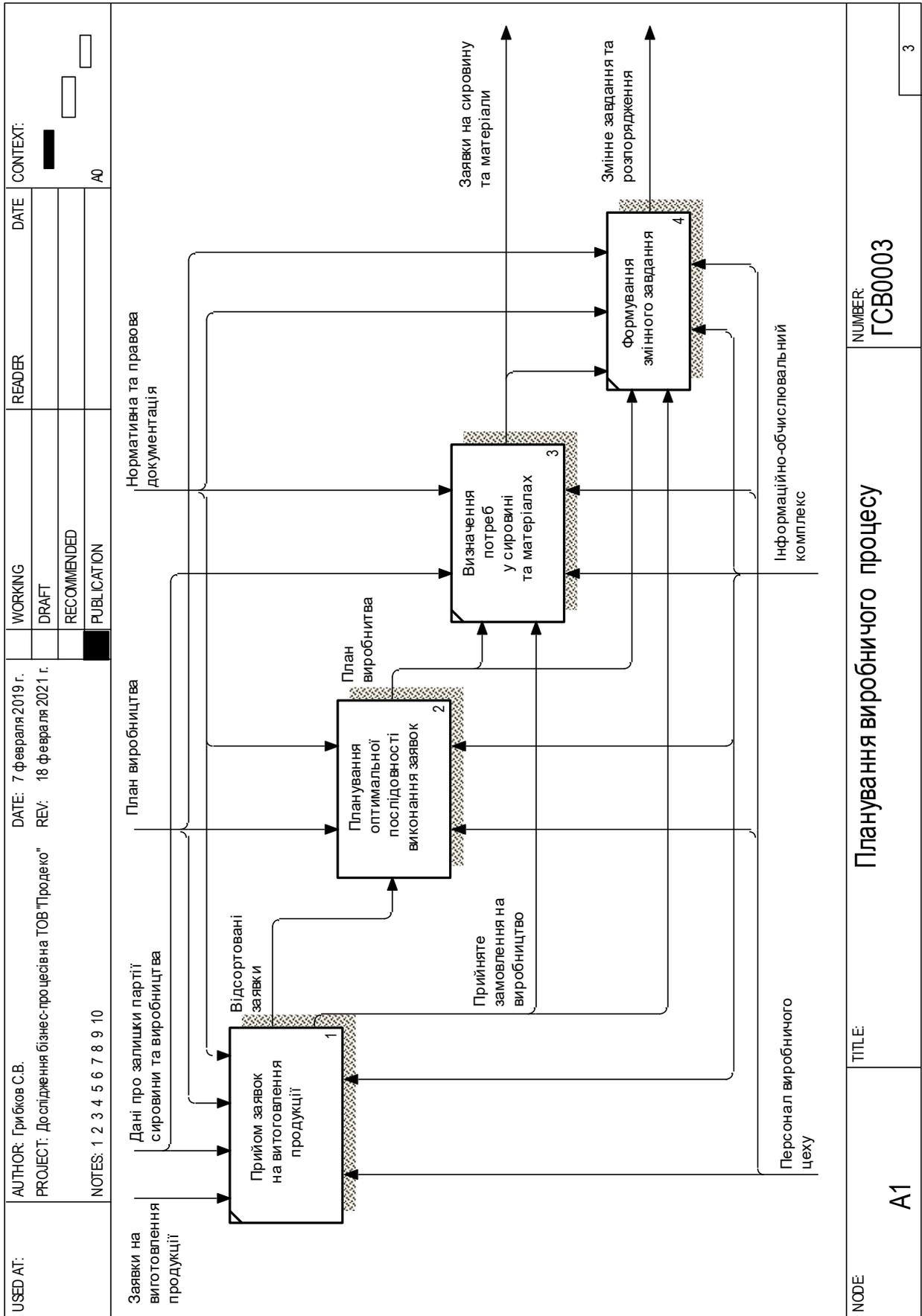


Рисунок В.9 – Діаграма декомпозиції першого рівня функціональної моделі «Планування виробничого процесу»

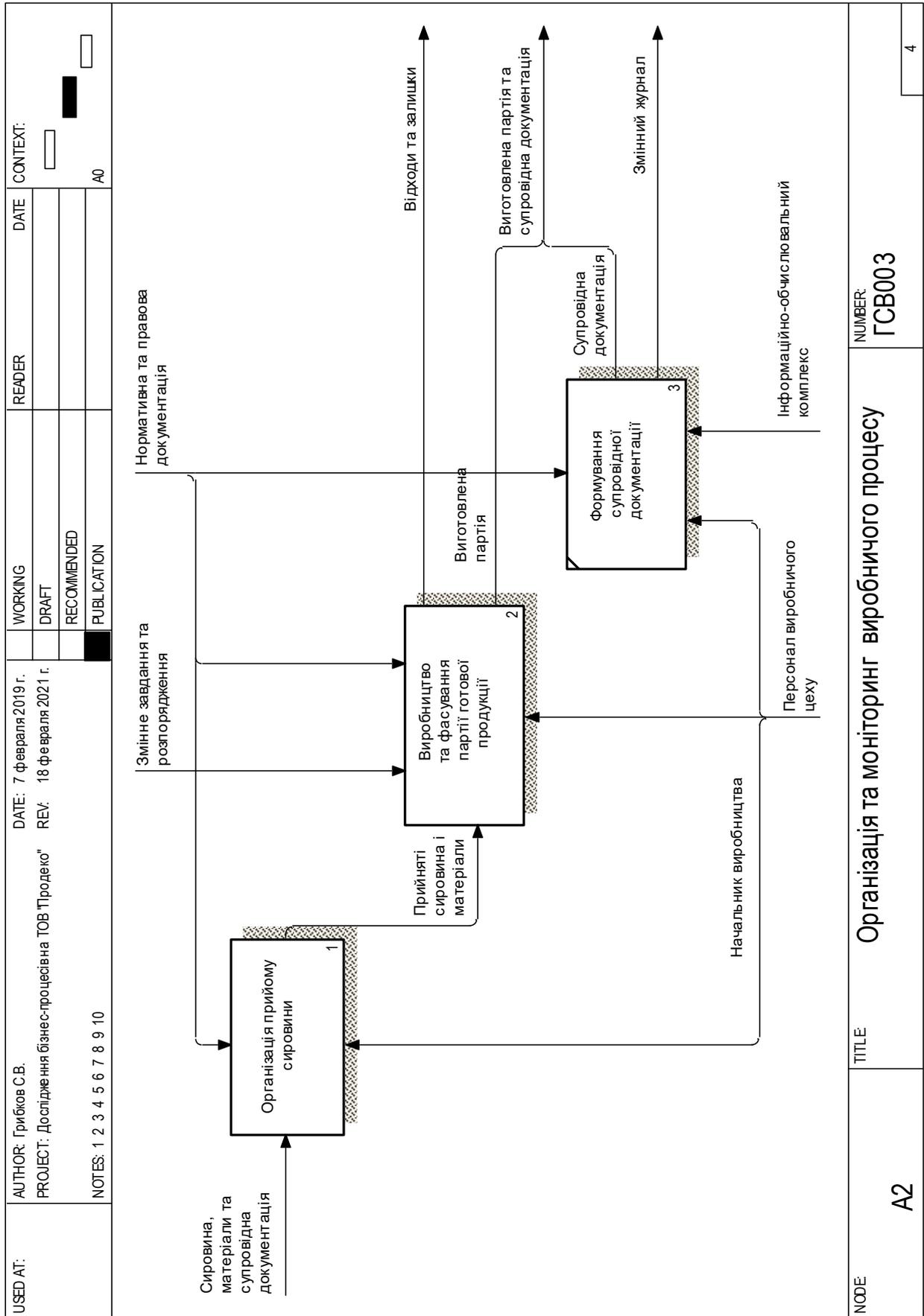


Рисунок В.10 – Діаграма декомпозиції функції «Організація та моніторинг виробничого процесу»

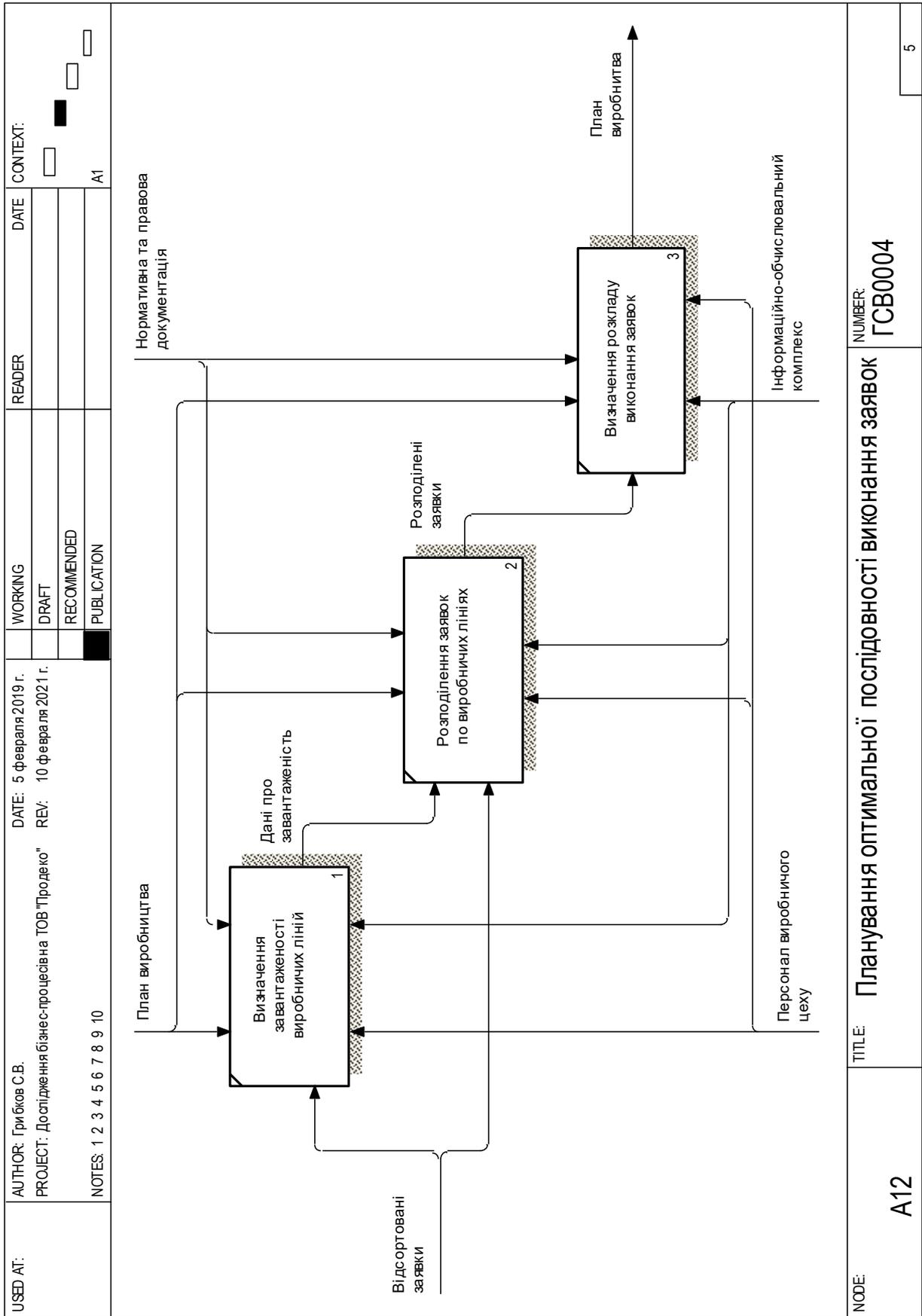


Рисунок В.11 – Діаграма декомпозиції функції «Планування оптимальної послідовності виконання заявок»

NUMBER: ГСВ0004

TITLE: Планування оптимальної послідовності виконання заявок

NODE: A12

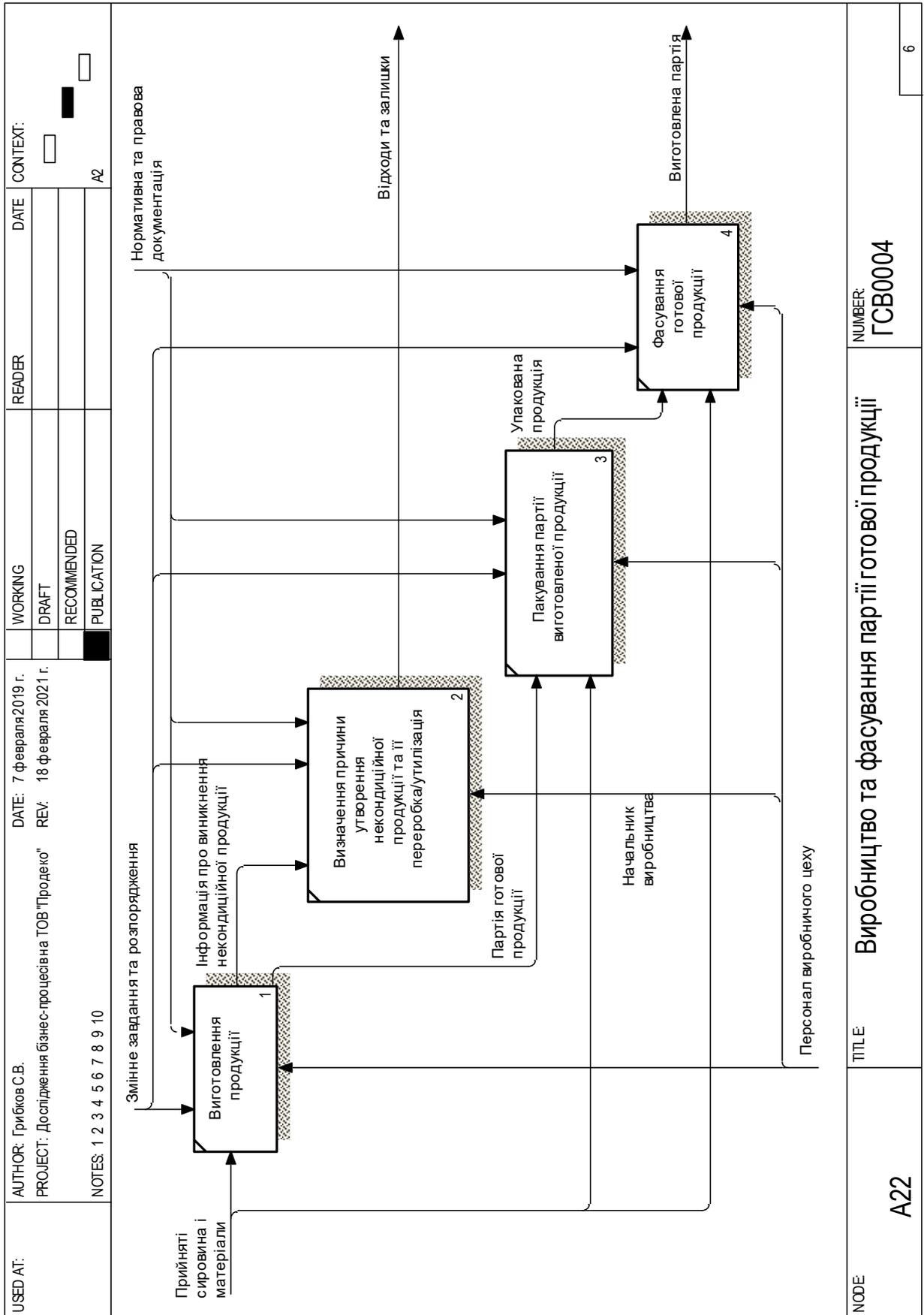
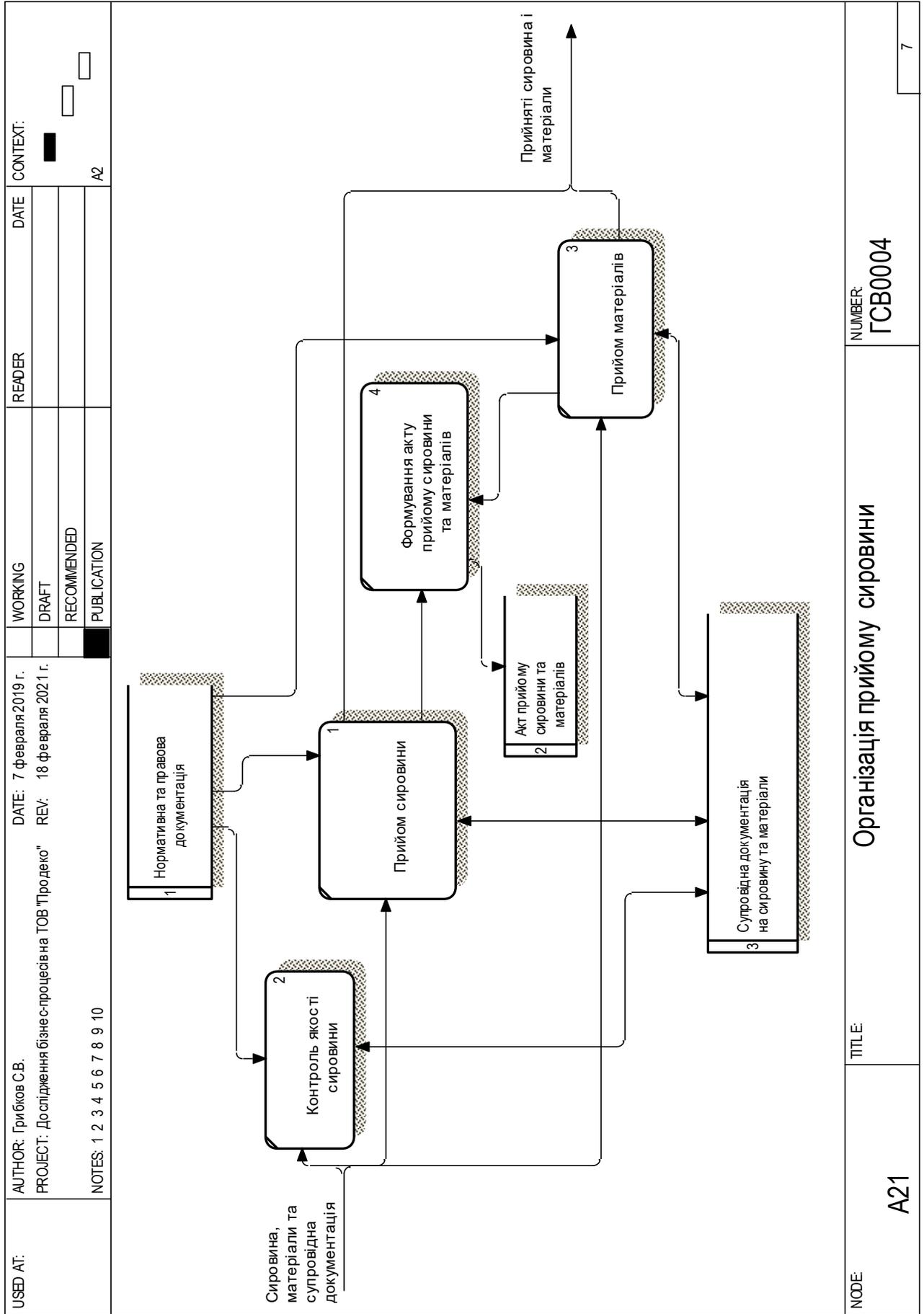


Рисунок В.12 – Діаграма декомпозиції функції «Виробництво та фасування партії готової продукції»



NODE: A21	TITLE: Організація прийому сировини	NUMBER: ГСВ0004
		7

Рисунок В.13 – Діаграма декомпозиції функції «Організація прийому сировини»

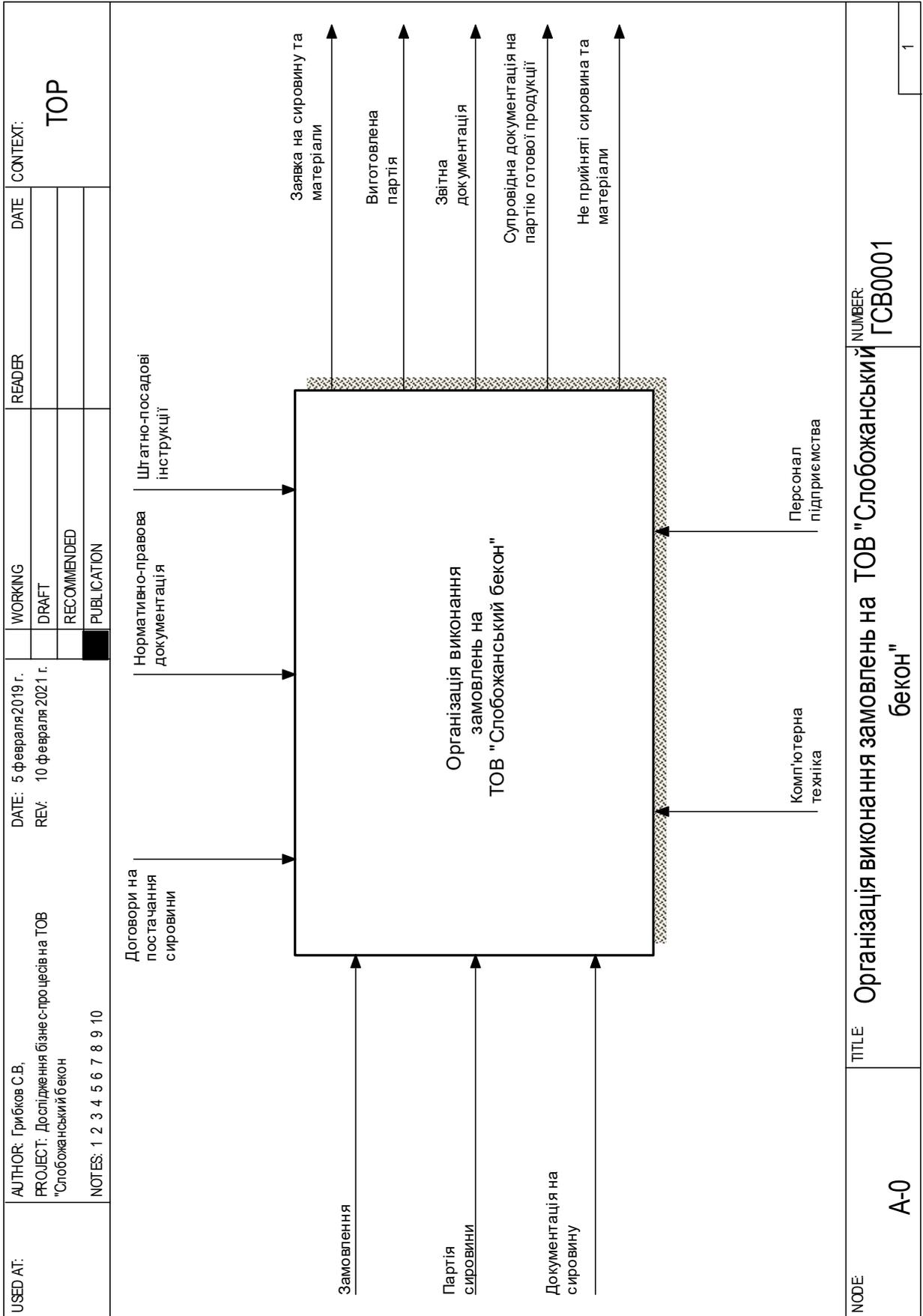


Рисунок В.14 – Контекстна діаграма «Організація виконання замовлень на ТОВ «Слобожанський бекон»»

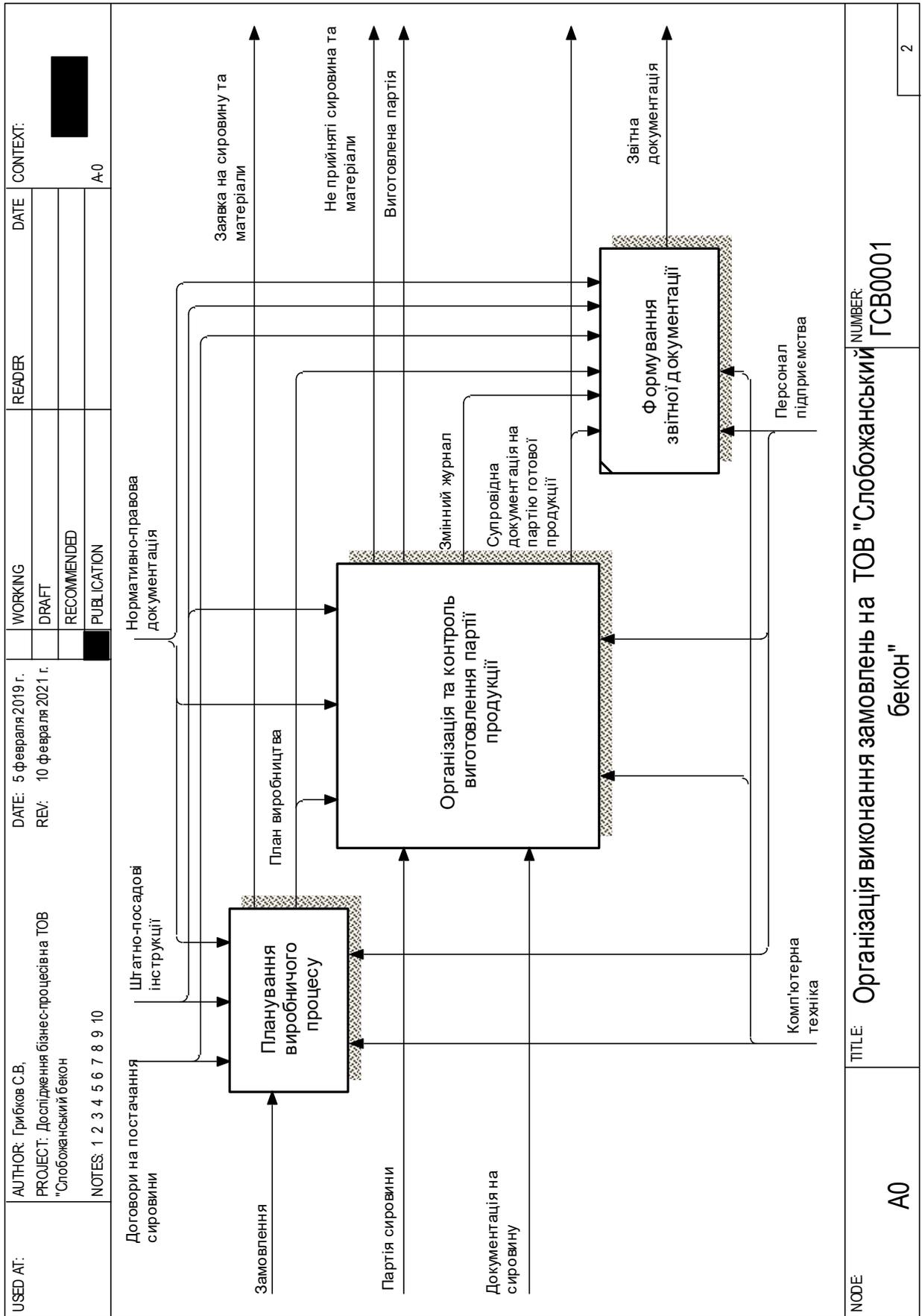
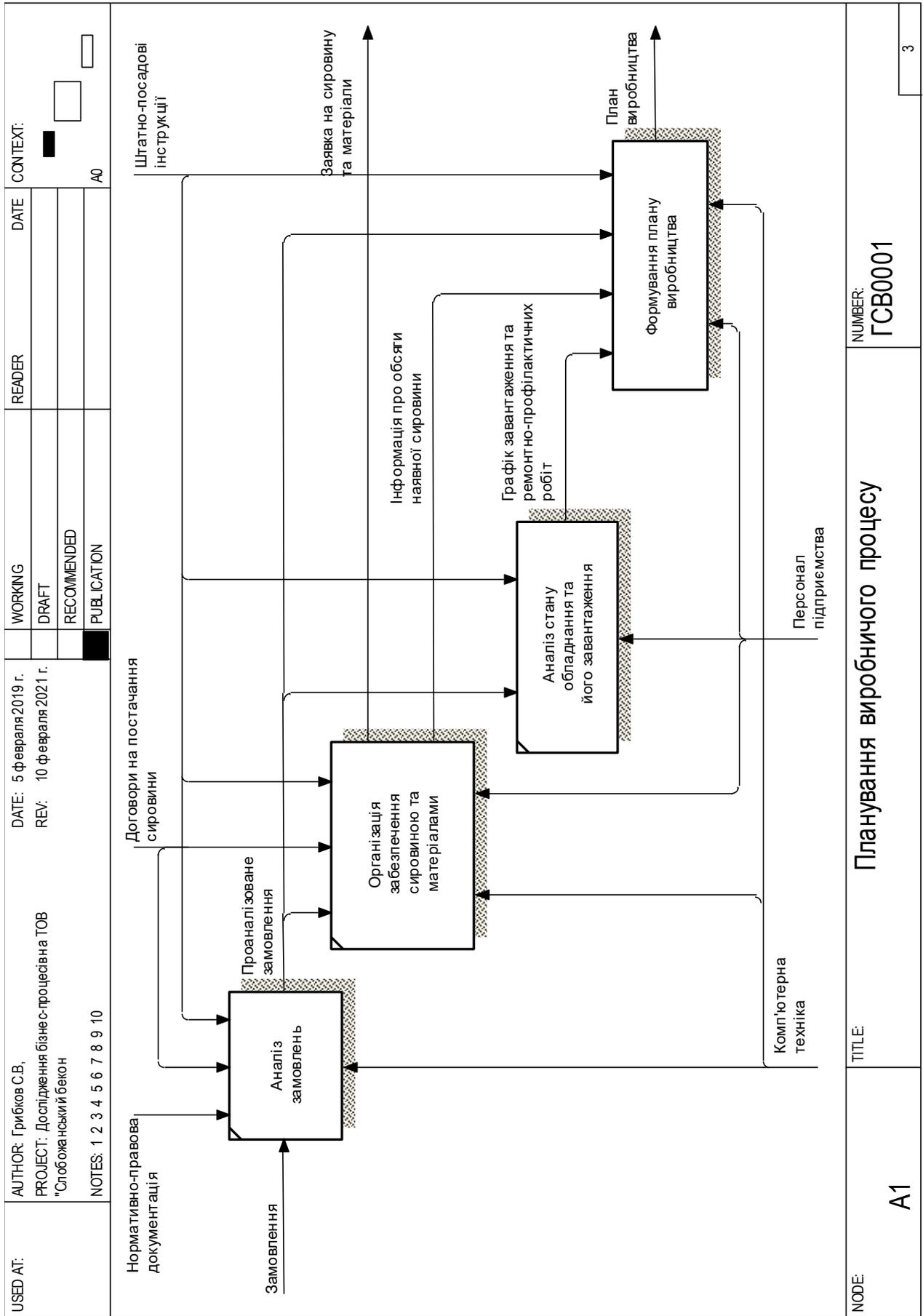


Рисунок В.15 – Декомпозиція функції «Організація виконання замовлень на ТОВ «Слобожанський бекон»»



NUMBER: ГСВ0001

TITLE: Планування виробничого процесу

CODE: А1

3

Рисунок В.16 – Декомпозиція функції «Планування виробничого процесу»



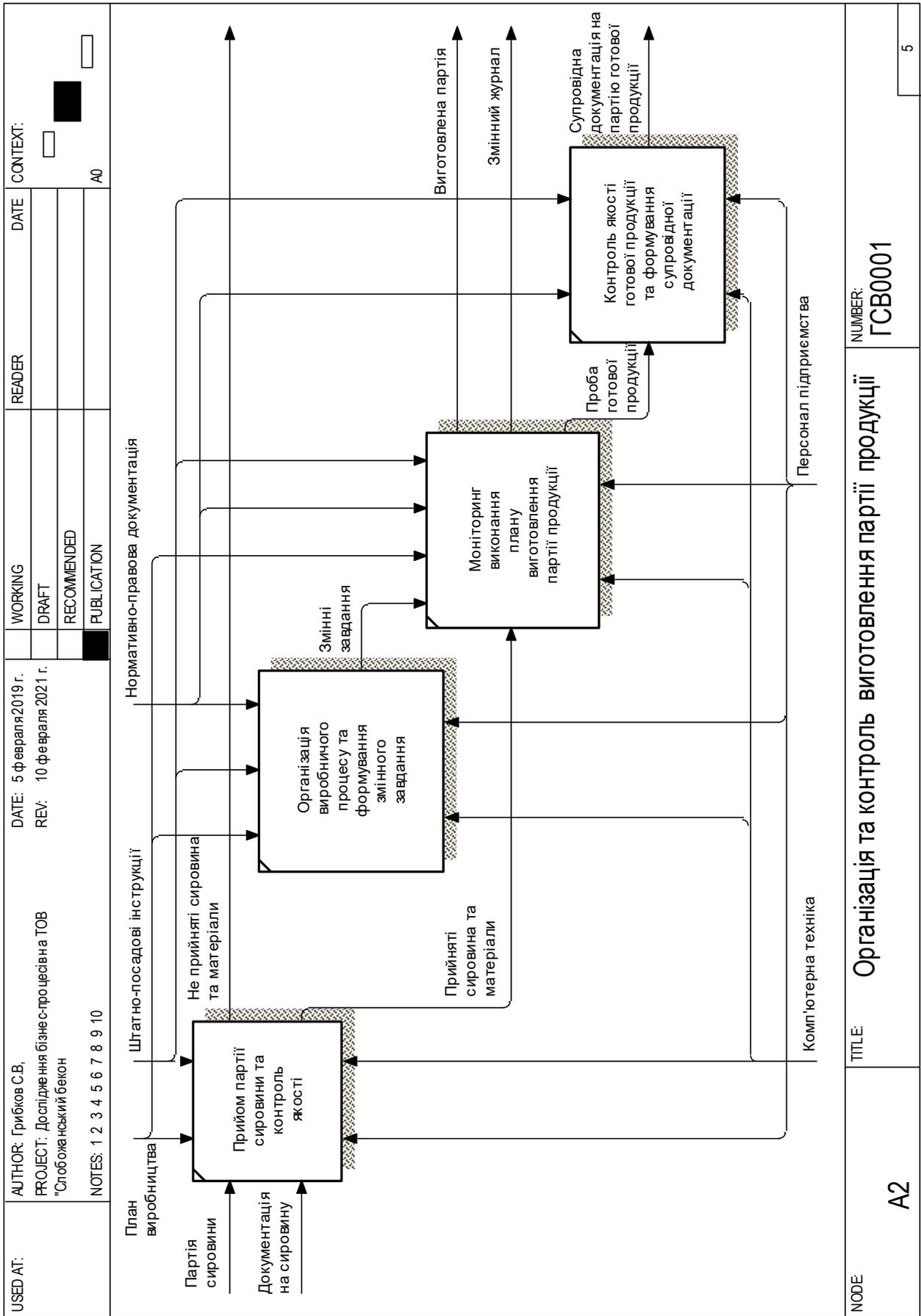


Рисунок В.18 – Декомпозиція функції «Організація та контроль виготовлення партії продукції»

NUMBER:  
ГСВ0001

TITLE:  
Організація та контроль виготовлення партії продукції

A2

5

Додаток Д  
Структура СППР

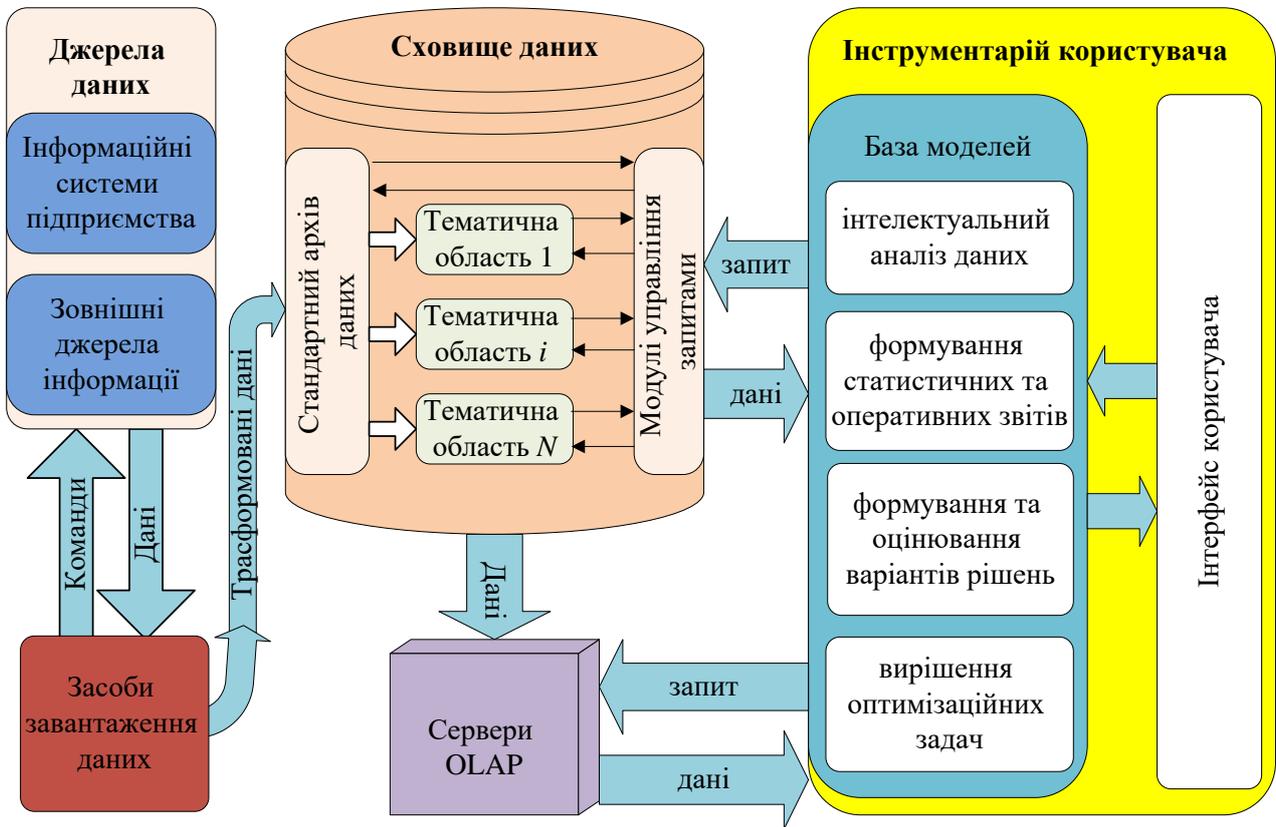


Рисунок Д.1 – Структура СППР

## Додаток Е

## Математична модель процесу термічної обробки м'ясних виробів

Будь-який ковбасний виріб розглядаємо як предмет овально-циліндричної форми, тобто представляємо його у вигляді фігури (рис. Е.1), де  $R_h$  – радіус.

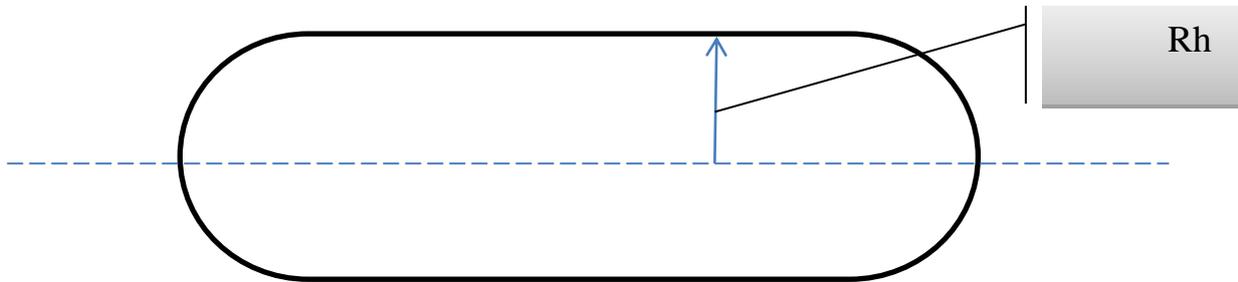


Рисунок Е.1 – Форма ковбасного виробу

Приймаємо  $D$  за діаметр ковбасного виробу, тобто  $D=2 \cdot R_h$ , а довжину  $L$  – за 1 метр.

Як оболонки використовується поліамід з такими показниками: теплопровідність  $\lambda=0,28$  (або в межах  $\lambda=0,214 \div 0,315$ ), товщина оболонок складає  $\delta=50 \div 80$  мк (або  $\delta=45 \div 80$  мк).

Якщо виріб має овальну, квадратну або прямокутну форму в поперечному перерізі, то приймаємо за найменший радіус або діаметр вписаного кола (рис. Е.2).

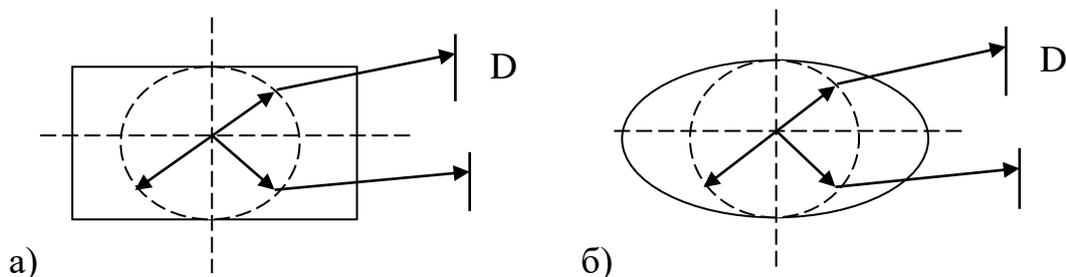


Рисунок Е.2 – Розріз ковбасного виробу впоперек

При цьому  $R_h = R_0$ , а  $R_0$  – діаметр продукту на початковій стадії.

Загальна тривалість процесу термічної обробки варених ковбас визначається за формулою (E.1):

$$\tau_{заг} = \omega_1 * \tau_{підсуш} + \omega_2 * \tau_{підсмаж} + \omega_3 * \tau_{вар}, \quad (E.1)$$

де  $\tau_{підсуш}$  – час нагрівання,  $\tau_{підсмаж}$  – час на підсмажування,  $\tau_{вар}$  – час на варку,  $\omega_i$  – коефіцієнт наявності  $i$ -ої стадії термічної обробки ( $i=1$  – нагрівання,  $i=2$  – підсмажування,  $i=3$  – варка) в технологічному процесі виготовлення продукту  $\omega_i = [0,1]$ , якщо 0 – стадія не використовується, 1 – якщо використовується.

На кожній із стадій діаметр  $Rh$  змінює своє значення у співвідношенні з  $i$ -ою стадією обробки і розраховується за формулою (E.2).

$$Rs_i = K_i R_0, \quad (E.2)$$

де  $K_i$  – емпіричний коефіцієнт, який характеризує збільшення радіуса ковбасного виробу при термічній обробці ( $K_1 = 1$ ,  $K_2 = 1.023$ ,  $K_3 = 1.045$ ).

#### 1. Стадія підсушування.

Час підсушування визначається як сума часу сушіння виробу ( $\tau_{підсуш}$ ) і час випаровування конденсату ( $\tau_{конден}$ ) (E.3):

$$\tau_{підсуш} = \tau_{сушки} + 60 * \tau_{конден}. \quad (E.3)$$

Розрахунок часу сушіння виробу визначається за такою формулою (E.4):

$$\tau_{сушки} = \frac{F_{0,підсушки} * Rh^2}{\alpha} = \frac{F_{0,підсушки} * R_0^2}{\alpha} = \frac{F_{0,підсушки} * Rs_2^2}{\alpha}, \quad (E.4)$$

де  $F_{0, підсушки}$  – тривалість підсушування у безрозмірному вигляді,  $a$  – коефіцієнт температуропровідності (для варених ковбас  $a=5 \cdot 10^{-4}$  м/хв) [11].

Значення критерію Фур'є, який відповідає часу, впродовж якого для процесу підсушування температура в центрі ковбасного виробу досягає потрібного значення, визначається за формулою (Е.5):

$$F_{0, підсушки} = \left( \frac{Bi_{підсушки} + 4}{8 * Bi_{підсушки}} \right) * \left( \ln \left( \frac{2}{(Bi_{підсушки} + 2)(1 - T_{n, підсушки})} \right) + F'_0 \right), \quad (E.5)$$

де  $Bi_{підсушки}$  – критерій Біо для стадії підсушки,  $T_{n, підсушки}$  – безрозмірне значення температури поверхні продукту в процесі підсушки,  $F'_0$  – час проходження «температурного фронту», яке визначається за номограмою.

Значення критерію Біо визначається за формулою (Е.6):

$$Bi_{підсушки} = \frac{a_{підсушки}}{\lambda} * R_0 = 25.47 * R_0, \quad (E.6)$$

де  $a_{підсушки}$  – коефіцієнт тепловіддачі,  $\lambda$  – коефіцієнт теплопровідності (для варених ковбас  $\lambda = 0,465$  Вт/м\*К).

Значення коефіцієнта тепловіддачі при термічній обробці ковбасних виробів можна визначити за виразом (Е.7):

$$a_{підсушки} = a_{середов} * (1 + 1,9 * d) = (6.16 + 4.49 * W) * (1 + 1,9 * d) = 11,84, \text{ Вт}/(\text{м}^2 \cdot \text{К}), \quad (E.7)$$

де коефіцієнт тепловіддачі від пароповітряного середовища, який визначається за емпіричною формулою Юргеса:

$a_{середов} = 6.16 + 4.49 * W$  Вт/(м<sup>2</sup>·К), де  $W$  – швидкість руху середовища, м/с ( $W=1$  м/с) [3];

$d$  – вологовміст продукту (при температурі середовища  $t_{cp}=100^{\circ}\text{C}$  і відносній вологості повітря  $\varphi = 10\%$ , вологовміст  $d = 0,076$ );

$F'_0$  – час проходження «температурного фронту», який визначається за номограмою [11], можна вирахувати, знаючи число Біо (Е.6), використовуючи формулу (Е.8):

$$\begin{aligned} F'_0 &\approx 0.7 * \left( \frac{1}{12} + \frac{1}{3 * Bi_{підсушки}} - \frac{2}{3 * (Bi_{підсушки})^2} * \ln(1 + 0.5 * Bi_{підсушки}) \right) = \\ &= 0.7 * \left( \frac{1}{12} + \frac{1}{3 * 25.47 * R_0} - \frac{2}{3 * (25.47 * R_0)^2} * \ln(1 + 0.5 * 25.47 * R_0) \right) = \quad . \quad (E.8) \\ &= 0.7 * \left( \frac{1}{12} + \frac{1}{76.41 * R_0} - \frac{2}{1946.16 * R_0^2} * \ln(1 + 12.74 * R_0) \right) \end{aligned}$$

Безрозмірне значення температури поверхні продукту в процесі підсушки визначається за формулою (Е.9):

$$T_{n, підсушки} = \frac{t_{кін, сушки} - t_{0, сушки}}{t_{cp, сушки} - t_{0, сушки}}, \quad (E.9)$$

де  $t_{кін, підсушки}$  – температура поверхні ковбасного виробу в кінці процесу підсушки,  $^{\circ}\text{C}$ ;  $t_{0, підсушки}$  – початкова температура ковбасного виробу (прийнято вважати, що початкова температура дорівнює  $15^{\circ}\text{C}$ ),  $^{\circ}\text{C}$ ,  $t_{cp, підсушки}$  – температура середовища,  $^{\circ}\text{C}$ .

Підставивши вираз (Е.6) і (Е.9) в формулу (Е.5), отримаємо значення критерію Фур'є, який відповідає часу, протягом якого для підсушки температура в центрі ковбасного виробу досягає необхідного значення (Е.10):

$$F_{0, \text{підсушки}} = \left( \frac{25.47 * R_0 + 4}{8 * 25.47 * R_0} \right) * \ln \left( \frac{2}{(25.47 * R_0 + 2) \left( 1 - \frac{t_{\text{кін, сушки}} - t_{0, \text{сушки}}}{t_{\text{ср, сушки}} - t_{0, \text{сушки}}} \right)} \right) + F'_0. \quad (\text{E.10})$$

Інакше необхідно використовувати нанограму з використанням  $Bi_{\text{підсушки}}$  і

параметра  $\rho = \frac{Bi_{\text{підсушки}} - T_{n, \text{підсушки}} (Bi_{\text{підсушки}} + 2)}{Bi_{\text{підсушки}} (1 - T_{n, \text{підсушки}})}$ , що дає можливість визначити  $F'_0$ .

Час випаровування конденсату (E.11):

$$\tau_{\text{конден}} = (2 - 0.46 * Bi_{\text{підсушки}}) * (21 * T_{n, \text{підсушки}} - 8) = (2 - 0.46 * 25.47 * R_0) * (21 * T_{n, \text{підсушки}} - 8). \quad (\text{E.11})$$

Результуючий час підсушки визначається за формулою

$$\begin{aligned} \tau_{\text{підсуш}} &= \frac{F_{0, \text{підсушки}} * R_0^2}{\alpha} + 60 * \tau_{\text{конден}} = \\ &= \frac{\left( \frac{Bi_{\text{підсушки}} + 4}{8 * Bi_{\text{підсушки}}} \right) * \ln \left( \frac{2}{(Bi_{\text{підсушки}} + 2) (1 - T_{n, \text{підсушки}})} \right) + F'_0}{\alpha} * R_0^2 + 60 * \tau_{\text{конден}} \end{aligned}$$

2. Стадія обсмажування.

Необхідно визначити час обсмажування (E.12)

$$\tau_{\text{обс}} = \frac{F_{0, \text{обс}} * Rs_2^2}{\alpha}, \quad (\text{E.12})$$

де  $Rs_2 = R_0 * K_2$  – середній радіус виробу з урахуванням процесу обжарювання і дорівнює радіусу в кінці процесу (E.13).

$$Bi_{обсмаж} = \frac{a_{обсмаж}}{\lambda} * Rs_2 = \frac{a_{обсмаж}}{\lambda} * R_0 * K_2, \quad (E.13)$$

де  $a_{обжарки}$  – коефіцієнт тепловіддачі, який  $a_{обсмаж} = a_{нідсушки} = a_{середов} * (1 + 1,9 * d) = 11,84$  Вт/(м<sup>2</sup>·К),  $\lambda$  – коефіцієнт теплопровідності (для варених ковбас  $\lambda = 0,465$  Вт/м·К) [11]. Тривалість обсмажування (E.14–E.15):

$$F_{0,обсмаж} = \left( \frac{Bi_{обсмаж} + 4}{8 * Bi_{обсмаж}} \right) * \left[ \ln \left( \frac{2}{(Bi_{обсмаж} + 2)(1 - T_{п,обсмаж})} \right) + F'_0 - F_{0,нідсушки} \right], \quad (E.14)$$

$$T_{п,обсмаж} = \frac{t_{кін,обсмаж} - t_{0,обсмаж}}{t_{ср,обсмаж} - t_{0,обсмаж}}, \quad (E.15)$$

$$t_{кін,сушка} = t_{0,обсмаж}$$

### 3. Стадія варки.

Розмірний час процесу варки ( $\tau_{вар}$ ) ковбасних виробів за формулою (E.4), а критерій Біо – за формулою (E.6).

Розмір ковбасного виробу на стадії варки визначається таким емпіричним співвідношенням (E.16).

$$R_{вар} = R_0 * K_3 = 1,045 * R_0. \quad (E.16)$$

Значення коефіцієнта тепловіддачі при термічній обробці ковбасних виробів можна визначити по виразу (E.17):

$$\alpha = \alpha_c (1 + 1,9 * d) = 35,68 \text{ Вт/(м}^2\text{·К)}, \quad (E.17)$$

де  $d$  – вологовміст продукту (при температурі середовища  $t_{ср}=85$  °С і відносній вологості повітря  $\varphi = 90$  %, вологовміст  $d=0,7653$  кг/кг) [3].

$$Bi_{вар} = 79,42 * R_0. \quad (E.18)$$

Критерій Фур'є знаходимо за формулою (E.19), підставивши вираз (E.15):

$$F_0 = 0.7 \left( \frac{1}{12} + \frac{1}{238,26 * R_0} - \frac{2 * \ln(1 + 39.71 * R_0)}{18922,61 * R_0^2} \right). \quad (E.19)$$

Тривалість варки в безрозмірному виразі визначаємо за формулою (E.20):

$$F_{0,вар} = \left( \frac{Bi_{вар} + 4}{8 * Bi_{вар}} \right) * \left[ \ln \left( \frac{t_{вар,ср} - t_{вар,0}}{t_{вар,ср} - t_{вар,к}} \right) + F'_0 \right] = \left( \frac{79,42 * R_0 + 4}{8 * 79,42 * R_0} \right) * \left[ \ln \left( \frac{t_{вар,ср} - t_{вар,0}}{t_{вар,ср} - t_{вар,к}} \right) + F'_0 \right], \quad (E.20)$$

де  $t_{вар,к}$  – температура в центрі виробу в кінці процесу варки.

При розрахунку тривалості варки вважаємо, що початковий стан продукту ( $t_{вар,0}$ ) можна охарактеризувати середньозваженою температурою (E.21):

$$t_{вар,0} = \frac{t_{обсмаж} + t_{н,обсмаж}}{2}. \quad (E.21)$$

Розмірний час процесу варки ( $\tau_{вар}$ ) ковбасних виробів визначаємо за формулою (E.22):

$$\tau_{вар} = 2142,45 * F_{0,вар} * R_0^2. \quad (E.22)$$

Математична модель теплової обробки варених ковбас (E.23).

$$\tau = 2000 * F_{0,нідсуні} * R_0^2 + (2 - 11.72 * R_0^2) (21 * T_n - 8) + 2048.29 * F_{0,вар} * R_0^2 + 2142.45 * F_{0,вар} * R_0^2. \quad (E.23)$$

## Додаток Ж

### Моделі «вітрин даних»

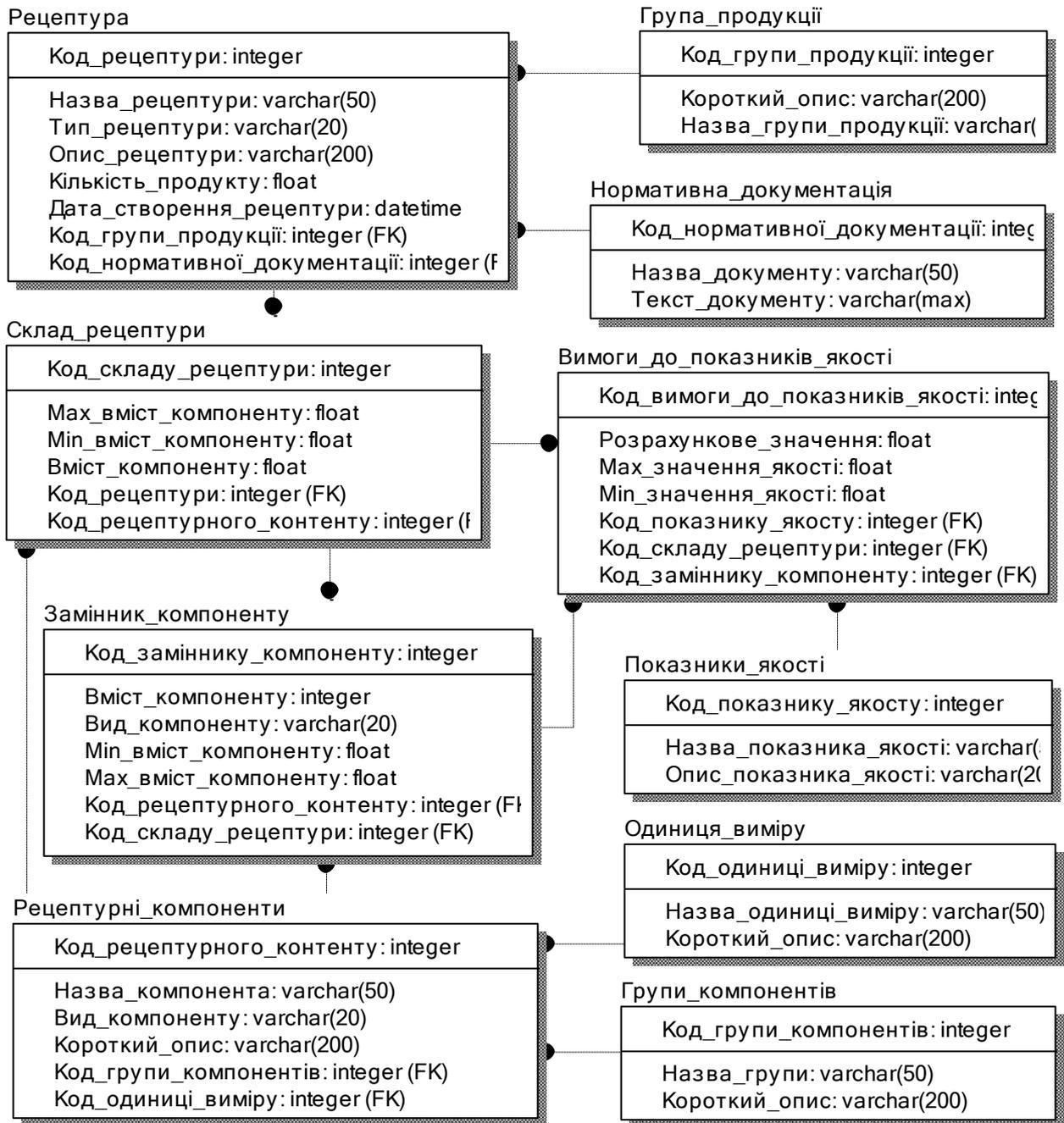


Рисунок Ж.1 – Фізична модель «вітрини даних» для підтримки роботи експертно-моделюючого модуля для формування рецептур виготовлення продукції

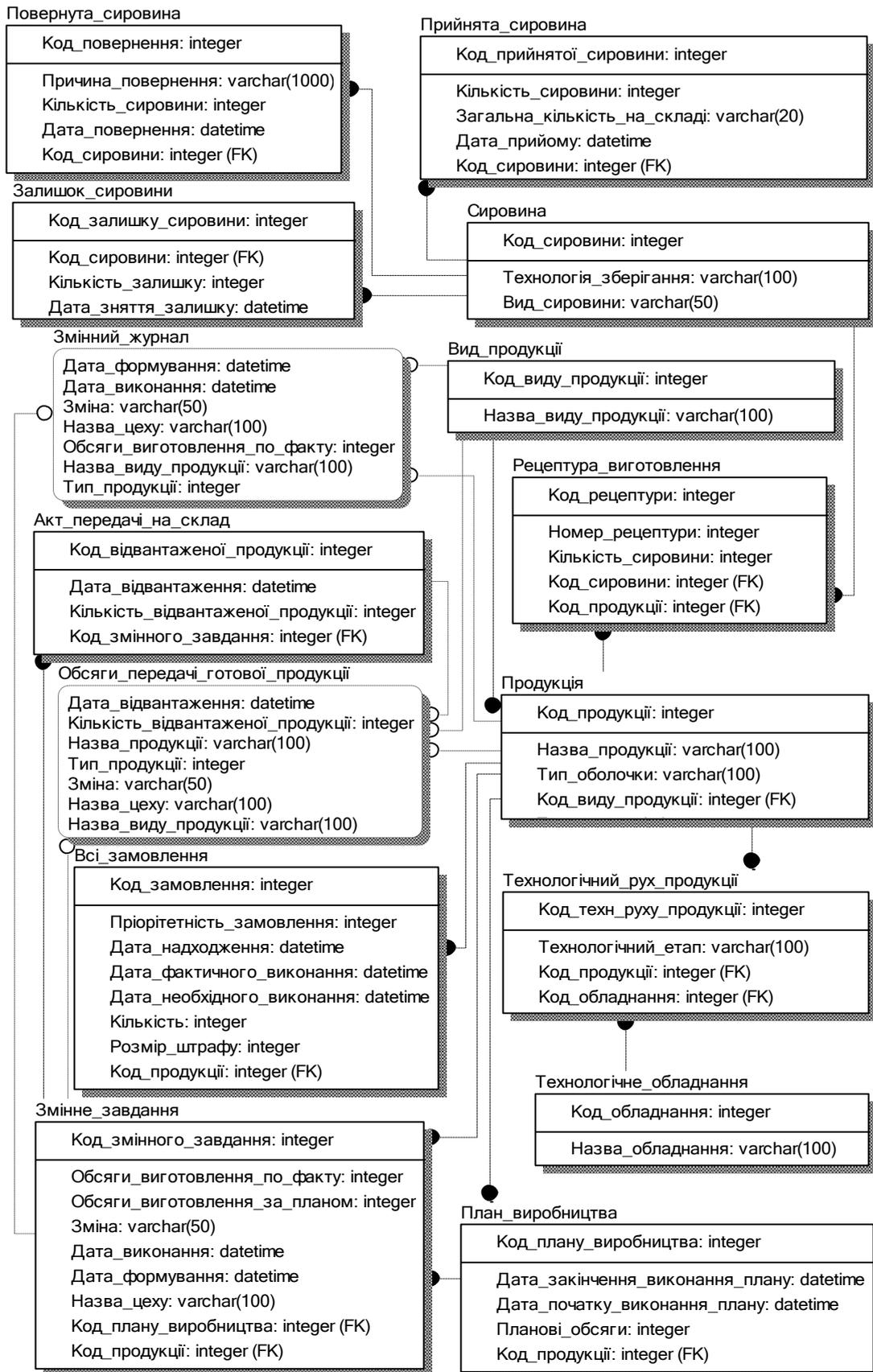


Рисунок Ж.2 – Фізична модель «вітрини даних» для моніторингу виконання замовлень за певний період

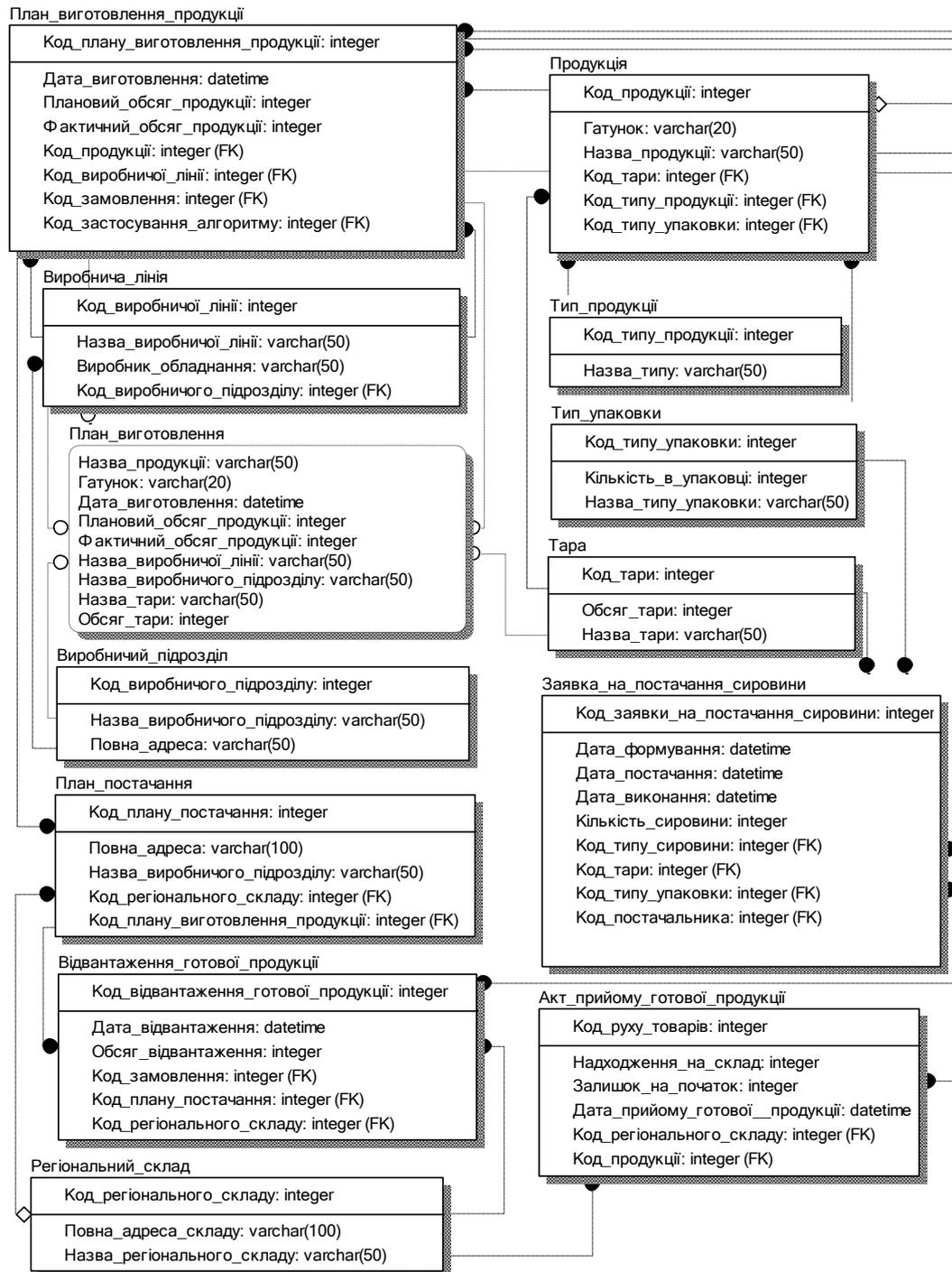


Рисунок Ж.3 – Фізична модель «вітрини даних» для формування плану виконання замовлень (Частина 1)

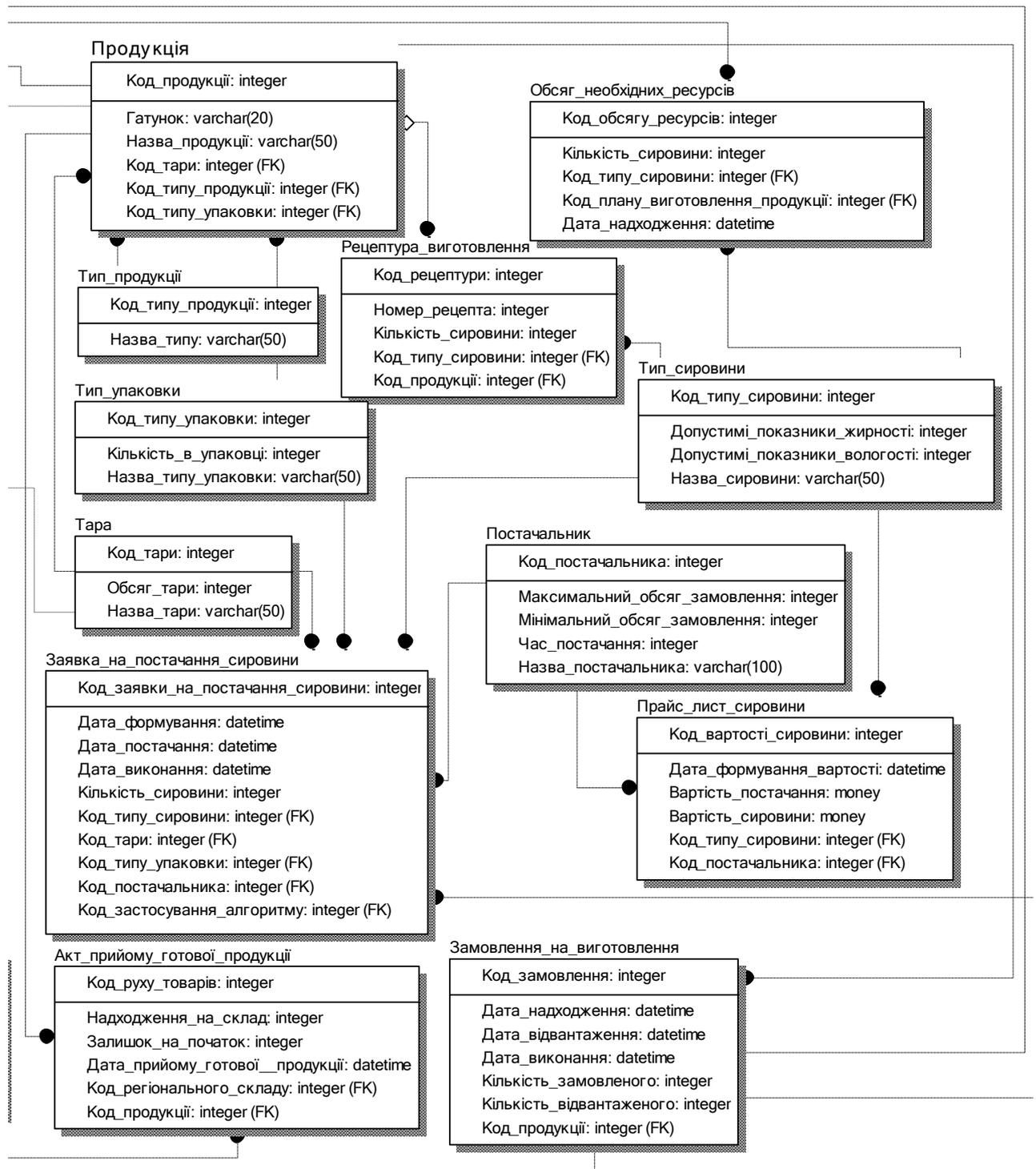


Рисунок Ж.4 – Фізична модель «вітрини даних» для формування плану виконання замовлень (Частина 2)

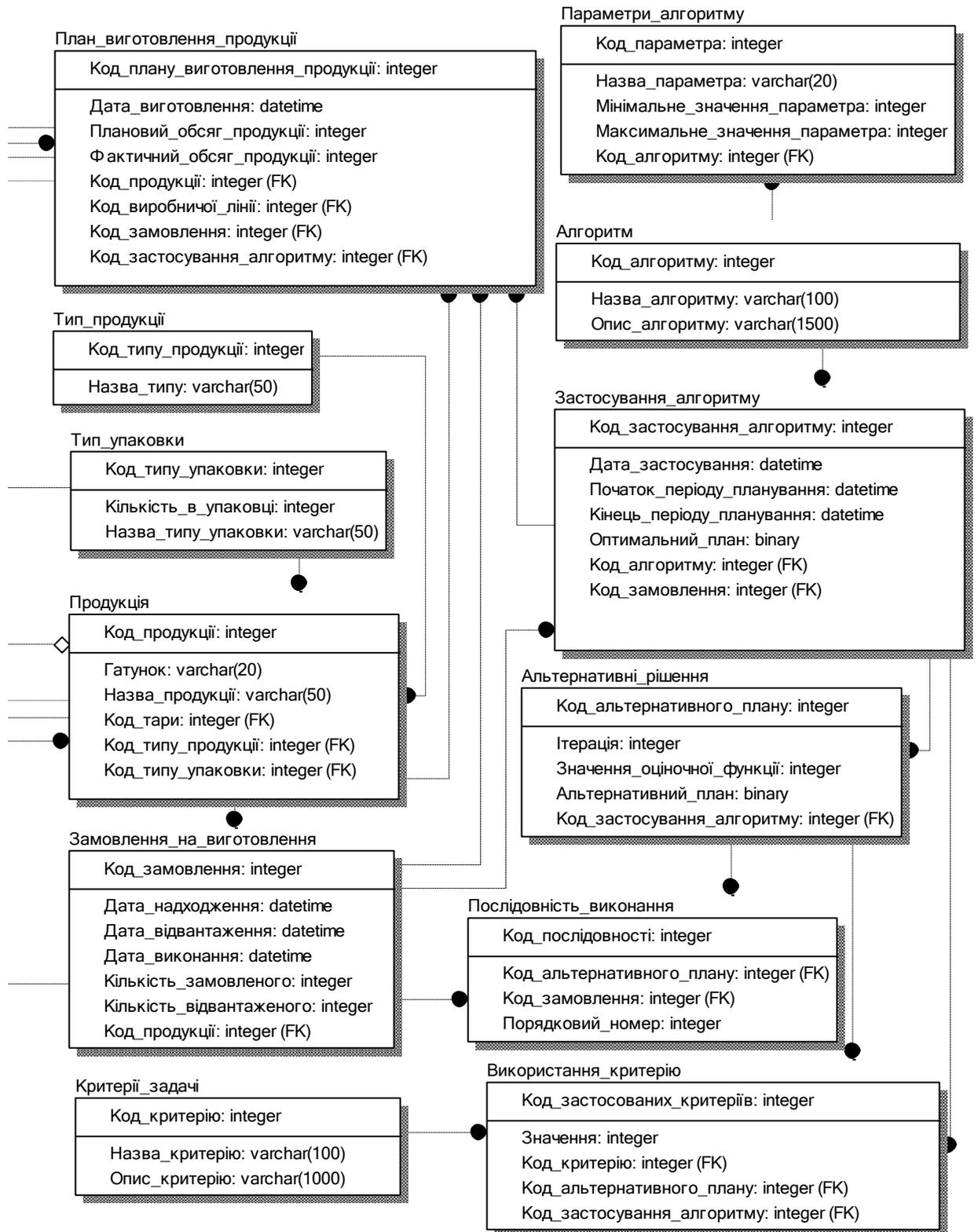


Рисунок Ж.5 – Фізична модель «вітрини даних» для формування плану виконання замовлень (Частина 3)

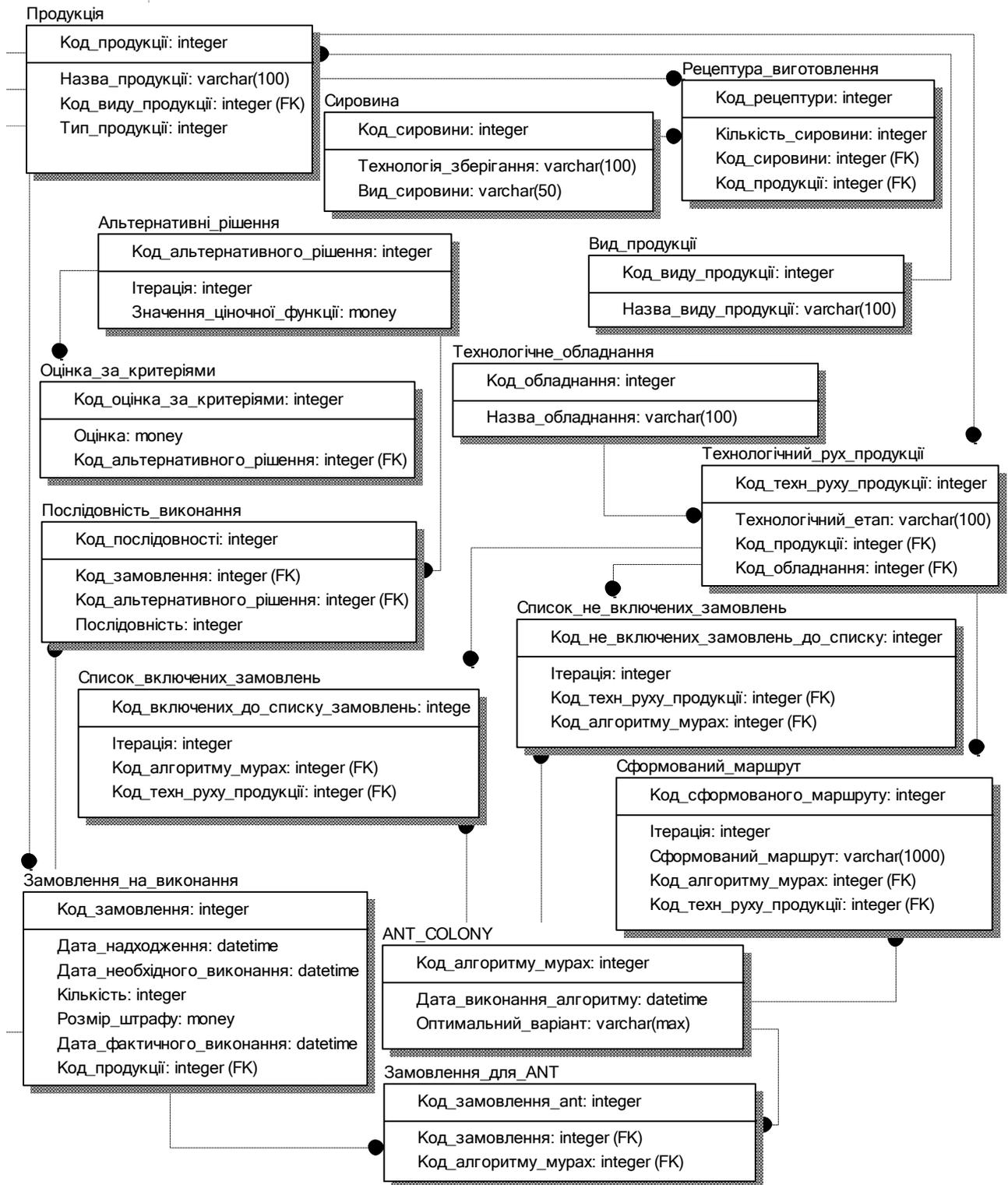


Рисунок Ж.6 – Фізична модель «вітрини даних» для формування плану виконання замовлень (Частина 4)

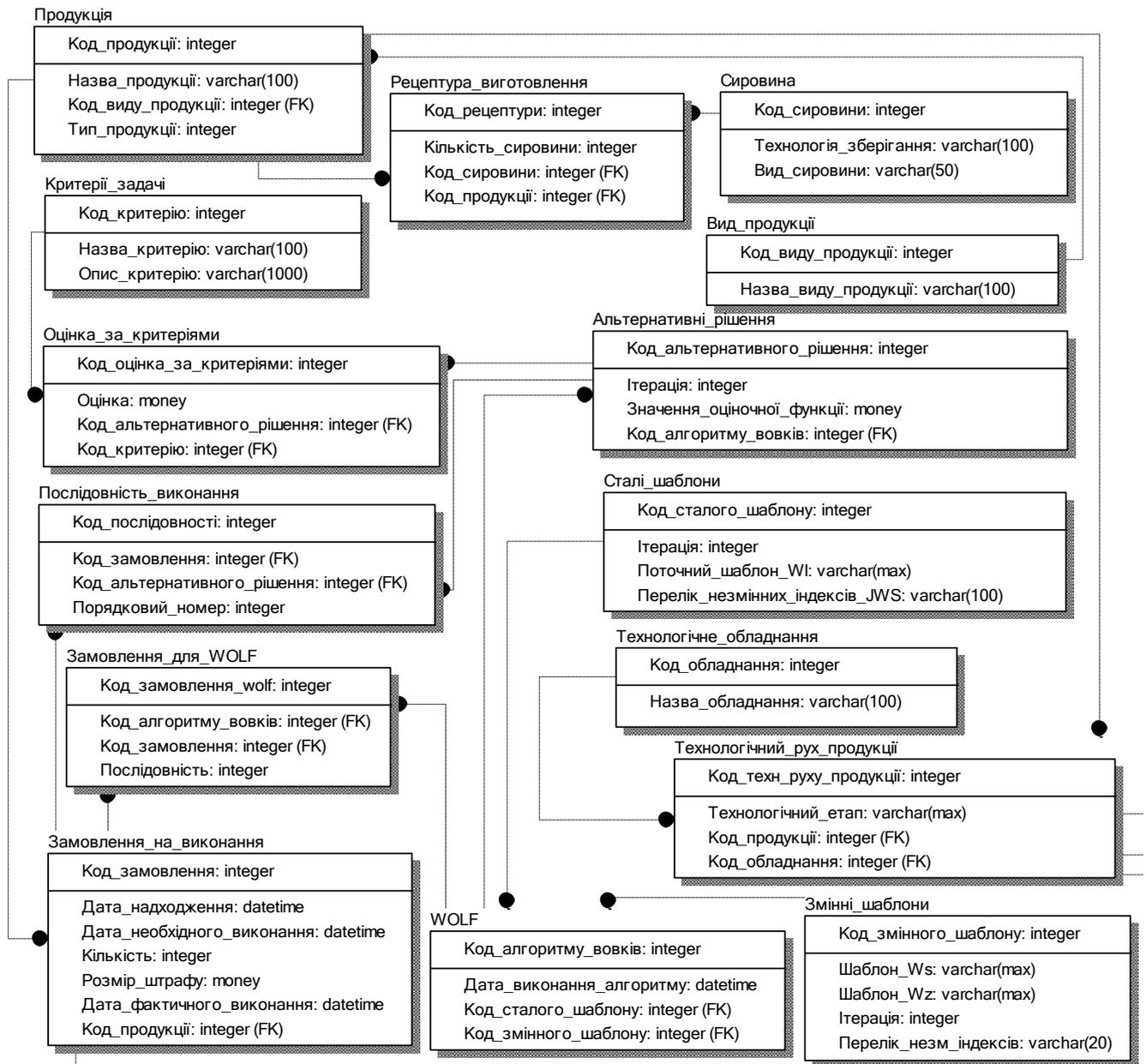


Рисунок Ж.7 – Фізична модель «вітрини даних» для формування плану виконання замовлень (Частина 5)

## Додаток 3

## Фрагменти SQL-кодів для створення «вітрин даних»

**3.1 SQL-код створення «вітрини даних» для формування плану виконання замовлень**

```
CREATE TABLE ANT_COLONY
(
    Код_алгоритму_мурах integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
    Дата_виконання_алгоритму datetime NULL ,
    Оптимальний_варіант varchar(max) NULL
)
go
```

```
ALTER TABLE ANT_COLONY
    ADD PRIMARY KEY CLUSTERED (Код_алгоритму_мурах ASC)
go
```

```
CREATE TABLE WOLF
(
    Код_алгоритму_вовків integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
    Дата_виконання_алгоритму datetime NULL ,
    Код_сталого_шаблону integer NOT NULL ,
    Код_змінного_шаблону integer NOT NULL
)
go
```

```
ALTER TABLE WOLF
    ADD PRIMARY KEY CLUSTERED (Код_алгоритму_вовків ASC)
go
```

```
CREATE TABLE Альтернативні_рішення
```

```
(  
    Код_альтернативного_рішення integer IDENTITY ( 1,1 ) NOT FOR  
    REPLICATION ,  
    Ітерація integer NULL ,  
    Значення_оціночної_функції money NULL ,  
    Код_алгоритму_вовків integer NOT NULL  
)  
go
```

```
ALTER TABLE Альтернативні_рішення
```

```
    ADD PRIMARY KEY CLUSTERED (Код_альтернативного_рішення ASC)  
go
```

```
CREATE TABLE Вид_продукції
```

```
(  
    Код_виду_продукції integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
    Назва_виду_продукції varchar(100) NULL  
)  
go
```

```
ALTER TABLE Вид_продукції
```

```
    ADD PRIMARY KEY CLUSTERED (Код_виду_продукції ASC)  
go
```

```
CREATE TABLE Заовлення_для_ANT
```

```
(  
    Код_заовлення_ant integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
    Код_заовлення integer NOT NULL ,
```

```
Код_алгоритму_мурах integer NOT NULL
)
go
```

```
ALTER TABLE Заовлення_для_ANT
ADD PRIMARY KEY CLUSTERED (Код_заовлення_ant ASC)
go
```

```
CREATE TABLE Заовлення_для_WOLF
(
Код_заовлення_wolf integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Код_алгоритму_вовків integer NOT NULL ,
Код_заовлення integer NOT NULL ,
Послідовність integer NULL
)
go
```

```
ALTER TABLE Заовлення_для_WOLF
ADD PRIMARY KEY CLUSTERED (Код_заовлення_wolf ASC)
go
```

```
CREATE TABLE Заовлення_на_виконання
(
Код_заовлення integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Дата_надходження datetime NULL ,
Дата_необхідного_виконання datetime NULL ,
Кількість integer NULL ,
Розмір_штрафу money NULL ,
Дата_фактичного_виконання datetime NULL ,
Код_продукції integer NOT NULL
```

```
)  
go
```

```
ALTER TABLE Заовлення_на_виконання  
  ADD PRIMARY KEY CLUSTERED (Код_заовлення ASC)  
go
```

```
CREATE TABLE Змінні_шаблони  
(  
  Код_змінного_шаблону integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
  Шаблон_Ws          varchar(max) NULL ,  
  Шаблон_Wz          varchar(max) NULL ,  
  Ітерація           integer NULL ,  
  Перелік_незм_індексів varchar(20) NULL  
)  
go
```

```
ALTER TABLE Змінні_шаблони  
  ADD PRIMARY KEY CLUSTERED (Код_змінного_шаблону ASC)  
go
```

```
CREATE TABLE Критерії_задачі  
(  
  Код_критерію       integer NOT NULL ,  
  Назва_критерію     varchar(100) NULL ,  
  Опис_критерію      varchar(1000) NULL  
)  
go
```

```
ALTER TABLE Критерії_задачі
```

```
ADD PRIMARY KEY CLUSTERED (Код_критерію ASC)
go

CREATE TABLE Оцінка_за_критеріями
(
    Код_оцінка_за_критеріями integer NOT NULL ,
    Оцінка money NULL ,
    Код_альтернативного_рішення integer NOT NULL ,
    Код_критерію integer NOT NULL
)
go

ALTER TABLE Оцінка_за_критеріями
ADD PRIMARY KEY CLUSTERED (Код_оцінка_за_критеріями ASC)
go

CREATE TABLE Послідовність_виконання
(
    Код_послідовності integer NOT NULL ,
    Код_замовлення integer NOT NULL ,
    Код_альтернативного_рішення integer NOT NULL ,
    Порядковий_номер integer NULL
)
go

ALTER TABLE Послідовність_виконання
ADD PRIMARY KEY CLUSTERED (Код_послідовності ASC)
go

CREATE TABLE Продукція
```

```
(  
    Код_продукції    integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
    Назва_продукції  varchar(100) NULL ,  
    Код_виду_продукції integer NOT NULL ,  
    Тип_продукції    integer NULL  
)  
go
```

```
ALTER TABLE Продукція  
    ADD PRIMARY KEY CLUSTERED (Код_продукції ASC)  
go
```

```
CREATE TABLE Рецепттура_виготовлення  
(  
    Код_рецептури    integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
    Кількість_сировини integer NULL ,  
    Код_сировини     integer NOT NULL ,  
    Код_продукції    integer NOT NULL  
)  
go
```

```
ALTER TABLE Рецепттура_виготовлення  
    ADD PRIMARY KEY CLUSTERED (Код_рецептури ASC)  
go
```

```
CREATE TABLE Сировина  
(  
    Код_сировини     integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
    Вид_сировини     varchar(50) NULL ,  
    Технологія_зберігання varchar(100) NULL  
)
```

```
)  
go
```

```
ALTER TABLE Сировина  
  ADD PRIMARY KEY CLUSTERED (Код_сировини ASC)  
go
```

```
CREATE TABLE Список_включених_замовлень  
(  
  Код_включених_до_списку_замовлень integer IDENTITY ( 1,1 ) NOT FOR  
  REPLICATION ,  
  Ітерація integer NULL ,  
  Код_алгоритму_мурах integer NOT NULL ,  
  Код_техн_руху_продукції integer NOT NULL  
)  
go
```

```
ALTER TABLE Список_включених_замовлень  
  ADD PRIMARY KEY CLUSTERED (Код_включених_до_списку_замовлень  
  ASC)  
go
```

```
CREATE TABLE Список_не_включених_замовлень  
(  
  Код_не_включених_замовлень_до_списку integer IDENTITY ( 1,1 ) NOT FOR  
  REPLICATION ,  
  Ітерація integer NULL ,  
  Код_техн_руху_продукції integer NOT NULL ,  
  Код_алгоритму_мурах integer NOT NULL  
)
```

go

```
ALTER TABLE Список_не_включених_замовлень  
ADD PRIMARY KEY CLUSTERED  
(Код_не_включених_замовлень_до_списку ASC)
```

go

```
CREATE TABLE Сталі_шаблони
```

```
(  
Код_сталого_шаблону integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Ітерація integer NULL ,  
Поточний_шаблон_W1 varchar(max) NULL ,  
Перелік_незмінних_індексів_JWS varchar(100) NULL
```

```
)
```

go

```
ALTER TABLE Сталі_шаблони
```

```
ADD PRIMARY KEY CLUSTERED (Код_сталого_шаблону ASC)
```

go

```
CREATE TABLE Сформований_маршрут
```

```
(  
Код_сформованого_маршруту integer IDENTITY ( 1,1 ) NOT FOR  
REPLICATION ,  
Ітерація integer NULL ,  
Сформований_маршрут varchar(1000) NULL ,  
Код_алгоритму_мурах integer NOT NULL ,  
Код_техн_руху_продукції integer NOT NULL
```

```
)
```

go

```
ALTER TABLE Сформований_маршрут
```

```
ADD PRIMARY KEY CLUSTERED (Код_сформованого_маршруту ASC)
```

go

```
CREATE TABLE Технологічне_обладнання
```

```
(
```

```
Код_обладнання integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
```

```
Назва_обладнання varchar(100) NULL
```

```
)
```

go

```
ALTER TABLE Технологічне_обладнання
```

```
ADD PRIMARY KEY CLUSTERED (Код_обладнання ASC)
```

go

```
CREATE TABLE Технологічний_рух_продукції
```

```
(
```

```
Технологічний_етап varchar(max) NULL ,
```

```
Код_техн_руху_продукції integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
```

```
Код_продукції integer NOT NULL ,
```

```
Код_обладнання integer NOT NULL
```

```
)
```

go

```
ALTER TABLE Технологічний_рух_продукції
```

```
ADD PRIMARY KEY CLUSTERED (Код_техн_руху_продукції ASC)
```

go

ALTER TABLE WOLF

ADD FOREIGN KEY (Код\_сталого\_шаблону) REFERENCES

Сталі\_шаблони(Код\_сталого\_шаблону)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE WOLF

ADD FOREIGN KEY (Код\_змінного\_шаблону) REFERENCES

Змінні\_шаблони(Код\_змінного\_шаблону)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Альтернативні\_рішення

ADD FOREIGN KEY (Код\_алгоритму\_вовків) REFERENCES

WOLF(Код\_алгоритму\_вовків)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Замовлення\_для\_ANT

ADD FOREIGN KEY (Код\_замовлення) REFERENCES

Замовлення\_на\_виконання(Код\_замовлення)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Замовлення\_для\_ANT

```
ADD FOREIGN KEY (Код_алгоритму_мурах) REFERENCES  
ANT_COLONY(Код_алгоритму_мурах)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Замовлення_для_WOLF  
ADD FOREIGN KEY (Код_алгоритму_вовків) REFERENCES  
WOLF(Код_алгоритму_вовків)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Замовлення_для_WOLF  
ADD FOREIGN KEY (Код_замовлення) REFERENCES  
Замовлення_на_виконання(Код_замовлення)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Замовлення_на_виконання  
ADD FOREIGN KEY (Код_продукції) REFERENCES  
Продукція(Код_продукції)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Оцінка_за_критеріями  
ADD FOREIGN KEY (Код_альтернативного_рішення) REFERENCES  
Альтернативні_рішення(Код_альтернативного_рішення)
```

```
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
```

```
ALTER TABLE Оцінка_за_критеріями
ADD FOREIGN KEY (Код_критерію) REFERENCES
Критерії_задачі(Код_критерію)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
```

```
ALTER TABLE Послідовність_виконання
ADD FOREIGN KEY (Код_замовлення) REFERENCES
Замовлення_на_виконання(Код_замовлення)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
```

```
ALTER TABLE Послідовність_виконання
ADD FOREIGN KEY (Код_альтернативного_рішення) REFERENCES
Альтернативні_рішення(Код_альтернативного_рішення)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
```

```
ALTER TABLE Продукція
ADD FOREIGN KEY (Код_виду_продукції) REFERENCES
Вид_продукції(Код_виду_продукції)
go
```

```
ALTER TABLE Рецепттура_виготовлення  
  ADD FOREIGN KEY (Код_сировини) REFERENCES Сировина(Код_сировини)  
go
```

```
ALTER TABLE Рецепттура_виготовлення  
  ADD FOREIGN KEY (Код_продукції) REFERENCES  
  Продукція(Код_продукції)  
go
```

```
ALTER TABLE Список_включених_замовлень  
  ADD FOREIGN KEY (Код_алгоритму_мурах) REFERENCES  
  ANT_COLONY(Код_алгоритму_мурах)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Список_включених_замовлень  
  ADD FOREIGN KEY (Код_техн_руху_продукції) REFERENCES  
  Технологічний_рух_продукції(Код_техн_руху_продукції)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Список_не_включених_замовлень  
  ADD FOREIGN KEY (Код_техн_руху_продукції) REFERENCES  
  Технологічний_рух_продукції(Код_техн_руху_продукції)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Список_не_включених_замовлень
  ADD FOREIGN KEY (Код_алгоритму_мурах) REFERENCES
  ANT_COLONY(Код_алгоритму_мурах)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Сформований_маршрут
  ADD FOREIGN KEY (Код_алгоритму_мурах) REFERENCES
  ANT_COLONY(Код_алгоритму_мурах)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Сформований_маршрут
  ADD FOREIGN KEY (Код_техн_руху_продукції) REFERENCES
  Технологічний_рух_продукції(Код_техн_руху_продукції)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Технологічний_рух_продукції
  ADD FOREIGN KEY (Код_продукції) REFERENCES
  Продукція(Код_продукції)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Технологічний_рух_продукції
```

```
ADD FOREIGN KEY (Код_обладнання) REFERENCES  
Технологічне_обладнання(Код_обладнання)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION
```

```
go
```

```
CREATE TABLE Акт_прийому_готової_продукції  
(  
Код_руху_товарів integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Дата_прийому_готової__продукції datetime NULL ,  
Залишок_на_початок integer NULL ,  
Надходження_на_склад integer NULL ,  
Код_регіонального_складу integer NOT NULL ,  
Код_продукції integer NOT NULL  
)
```

```
go
```

```
ALTER TABLE Акт_прийому_готової_продукції  
ADD PRIMARY KEY CLUSTERED (Код_руху_товарів ASC)
```

```
go
```

```
CREATE TABLE Алгоритм  
(  
Код_алгоритму integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Назва_алгоритму varchar(100) NULL ,  
Опис_алгоритму varchar(1500) NULL  
)
```

```
go
```

```
ALTER TABLE Алгоритм
```

```
ADD PRIMARY KEY CLUSTERED (Код_алгоритму ASC)
go

CREATE TABLE Відвантаження_готової_продукції
(
    Код_відвантаження_готової_продукції integer IDENTITY ( 1,1 ) NOT FOR
    REPLICATION ,
    Дата_відвантаження datetime NULL ,
    Обсяг_відвантаження integer NULL ,
    Код_плану_постачання integer NOT NULL ,
    Код_регіонального_складу integer NOT NULL
)
go

ALTER TABLE Відвантаження_готової_продукції
ADD PRIMARY KEY CLUSTERED (Код_відвантаження_готової_продукції
ASC)
go

CREATE TABLE Використання_критерію
(
    Код_застосованих_критеріїв integer NOT NULL ,
    Значення integer NULL ,
    Код_альтернативного_плану integer NOT NULL ,
    Код_застосування_алгоритму integer NOT NULL ,
    Код_критерію integer IDENTITY
)
go

ALTER TABLE Використання_критерію
```

```
ADD PRIMARY KEY CLUSTERED (Код_застосованих_критеріїв ASC)
go
```

```
CREATE TABLE Виробнича_лінія
(
    Код_виробничої_лінії integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
    Назва_виробничої_лінії varchar(50) NULL ,
    Виробник_обладнання varchar(50) NULL ,
    Код_виробничого_підрозділу integer NOT NULL
)
go
```

```
ALTER TABLE Виробнича_лінія
    ADD PRIMARY KEY CLUSTERED (Код_виробничої_лінії ASC)
go
```

```
CREATE TABLE Виробничий_підрозділ
(
    Код_виробничого_підрозділу integer IDENTITY ( 1,1 ) NOT FOR
    REPLICATION ,
    Назва_виробничого_підрозділу varchar(50) NULL ,
    Повна_адреса      varchar(50) NULL
)
go
```

```
ALTER TABLE Виробничий_підрозділ
    ADD PRIMARY KEY CLUSTERED (Код_виробничого_підрозділу ASC)
go
```

```
CREATE TABLE Застосування_алгоритму
```

```
(  
Код_застосування_алгоритму integer NOT NULL ,  
Дата_застосування datetime NULL ,  
Початок_періоду_планування datetime NULL ,  
Кінець_періоду_планування datetime NULL ,  
Оптимальний_план binary NULL ,  
Код_алгоритму integer NOT NULL ,  
Код_замовлення integer IDENTITY  
)  
go
```

```
ALTER TABLE Застосування_алгоритму  
ADD PRIMARY KEY CLUSTERED (Код_застосування_алгоритму ASC)  
go
```

```
CREATE TABLE Заявка_на_постачання_сировини  
(  
Код_заявки_на_постачання_сировини integer NOT NULL ,  
Дата_формування datetime NULL ,  
Дата_постачання datetime NULL ,  
Дата_виконання datetime NULL ,  
Кількість_сировини integer NULL ,  
Код_типу_сировини integer NOT NULL ,  
Код_тари integer NOT NULL ,  
Код_типу_упаковки integer NOT NULL ,  
Код_постачальника integer NOT NULL ,  
Код_замовлення integer IDENTITY  
)  
go
```

```
ALTER TABLE Заявка_на_постачання_сировини
  ADD PRIMARY KEY CLUSTERED (Код_заявки_на_постачання_сировини
  ASC)
```

```
go
```

```
CREATE TABLE Обсяг_необхідних_ресурсів
(
  Код_обсягу_ресурсів integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
  Кількість_сировини integer NULL ,
  Код_типу_сировини integer NOT NULL ,
  Код_плану_виготовлення_продукції integer NOT NULL ,
  Дата_надходження datetime NULL
)
```

```
go
```

```
ALTER TABLE Обсяг_необхідних_ресурсів
  ADD PRIMARY KEY CLUSTERED (Код_обсягу_ресурсів ASC)
```

```
go
```

```
CREATE TABLE Параметри_алгоритму
(
  Код_параметра integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
  Назва_параметра varchar(20) NULL ,
  Мінімальне_значення_параметра integer NULL ,
  Максимальне_значення_параметра integer NULL ,
  Код_алгоритму integer NOT NULL
)
```

```
go
```

```
ALTER TABLE Параметри_алгоритму
```

```
ADD PRIMARY KEY CLUSTERED (Код_параметра ASC)
go
```

```
CREATE TABLE План_виготовлення_продукції
(
    Код_плану_виготовлення_продукції integer NOT NULL ,
    Дата_виготовлення datetime NULL ,
    Плановий_обсяг_продукції integer NULL ,
    Фактичний_обсяг_продукції integer NULL ,
    Код_продукції integer NOT NULL ,
    Код_виробничої_лінії integer NOT NULL ,
    Код_застосування_алгоритму integer NOT NULL ,
    Код_замовлення integer IDENTITY
)
go
```

```
ALTER TABLE План_виготовлення_продукції
ADD PRIMARY KEY CLUSTERED (Код_плану_виготовлення_продукції
ASC)
go
```

```
CREATE TABLE План_постачання
(
    Код_плану_постачання integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
    Назва_виробничого_підрозділу varchar(50) NULL ,
    Повна_адреса varchar(100) NULL ,
    Код_регіонального_складу integer NULL ,
    Код_плану_виготовлення_продукції integer NOT NULL
)
go
```

```
ALTER TABLE План_постачання  
  ADD PRIMARY KEY CLUSTERED (Код_плану_постачання ASC)  
go
```

```
CREATE TABLE Постачальник  
(  
  Код_постачальника integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
  Назва_постачальника varchar(100) NULL ,  
  Час_постачання integer NULL ,  
  Мінімальний_обсяг_замовлення integer NULL ,  
  Максимальний_обсяг_замовлення integer NULL  
)  
go
```

```
ALTER TABLE Постачальник  
  ADD PRIMARY KEY CLUSTERED (Код_постачальника ASC)  
go
```

```
CREATE TABLE Прайс_лист_сировини  
(  
  Код_вартості_сировини integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
  Дата_формування_вартості datetime NULL ,  
  Вартість_постачання money NULL ,  
  Вартість_сировини money NULL ,  
  Код_типу_сировини integer NOT NULL ,  
  Код_постачальника integer NOT NULL  
)  
go
```

```
ALTER TABLE Прайс_лист_сировини
  ADD PRIMARY KEY CLUSTERED (Код_вартості_сировини ASC)
go
```

```
CREATE TABLE Регіональний_склад
(
  Код_регіонального_складу integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
  Повна_адреса_складу varchar(100) NULL ,
  Назва_регіонального_плану varchar(50) NULL
)
go
```

```
ALTER TABLE Регіональний_склад
  ADD PRIMARY KEY CLUSTERED (Код_регіонального_складу ASC)
go
```

```
CREATE TABLE Тара
(
  Код_тари          integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
  Назва_тари        varchar(50) NULL ,
  Обсяг_тари        integer NULL
)
go
```

```
ALTER TABLE Тара
  ADD PRIMARY KEY CLUSTERED (Код_тари ASC)
go
```

```
CREATE TABLE Тип_продукції
(
```

```
Код_типу_продукції integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Назва_типу          varchar(50) NULL  
)  
go
```

```
ALTER TABLE Тип_продукції  
ADD PRIMARY KEY CLUSTERED (Код_типу_продукції ASC)  
go
```

```
CREATE TABLE Тип_сировини  
(  
Код_типу_сировини integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Назва_сировини    varchar(50) NULL ,  
Допустимі_показники_вологості integer NULL ,  
Допустимі_показники_жирності integer NULL  
)  
go
```

```
ALTER TABLE Тип_сировини  
ADD PRIMARY KEY CLUSTERED (Код_типу_сировини ASC)  
go
```

```
CREATE TABLE Тип_упаковки  
(  
Код_типу_упаковки integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Назва_типу_упаковки varchar(50) NULL ,  
Кількість_в_упаковці integer NULL  
)  
go
```

```
ALTER TABLE Тип_упаковки
```

```
ADD PRIMARY KEY CLUSTERED (Код_типу_упаковки ASC)
```

```
go
```

```
ALTER TABLE Акт_прийому_готової_продукції
```

```
ADD FOREIGN KEY (Код_регіонального_складу) REFERENCES
```

```
Регіональний_склад(Код_регіонального_складу)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE Акт_прийому_готової_продукції
```

```
ADD FOREIGN KEY (Код_продукції) REFERENCES
```

```
Продукція(Код_продукції)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE Відвантаження_готової_продукції
```

```
ADD FOREIGN KEY (Код_плану_постачання) REFERENCES
```

```
План_постачання(Код_плану_постачання)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE Відвантаження_готової_продукції
```

```
ADD FOREIGN KEY (Код_регіонального_складу) REFERENCES
```

```
Регіональний_склад(Код_регіонального_складу)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

go

ALTER TABLE Використання\_критерію

ADD FOREIGN KEY (Код\_альтернативного\_плану) REFERENCES

Альтернативні\_рішення(Код\_альтернативного\_плану)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Використання\_критерію

ADD FOREIGN KEY (Код\_застосування\_алгоритму) REFERENCES

Застосування\_алгоритму(Код\_застосування\_алгоритму)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Виробнича\_лінія

ADD FOREIGN KEY (Код\_виробничого\_підрозділу) REFERENCES

Виробничий\_підрозділ(Код\_виробничого\_підрозділу)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Застосування\_алгоритму

ADD FOREIGN KEY (Код\_алгоритму) REFERENCES

Алгоритм(Код\_алгоритму)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

```
ALTER TABLE Заявка_на_постачання_сировини
  ADD FOREIGN KEY (Код_типу_сировини) REFERENCES
  Тип_сировини(Код_типу_сировини)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Заявка_на_постачання_сировини
  ADD FOREIGN KEY (Код_тари) REFERENCES Тара(Код_тари)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Заявка_на_постачання_сировини
  ADD FOREIGN KEY (Код_типу_упаковки) REFERENCES
  Тип_упаковки(Код_типу_упаковки)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Заявка_на_постачання_сировини
  ADD FOREIGN KEY (Код_постачальника) REFERENCES
  Постачальник(Код_постачальника)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
go
```

```
ALTER TABLE Обсяг_необхідних_ресурсів
  ADD FOREIGN KEY (Код_типу_сировини) REFERENCES
  Тип_сировини(Код_типу_сировини)
```

```
ON DELETE NO ACTION
ON UPDATE NO ACTION
go

ALTER TABLE Обсяг_необхідних_ресурсів
ADD FOREIGN KEY (Код_плану_виготовлення_продукції) REFERENCES
План_виготовлення_продукції(Код_плану_виготовлення_продукції)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go

ALTER TABLE Параметри_алгоритму
ADD FOREIGN KEY (Код_алгоритму) REFERENCES
Алгоритм(Код_алгоритму)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go

ALTER TABLE План_виготовлення_продукції
ADD FOREIGN KEY (Код_продукції) REFERENCES
Продукція(Код_продукції)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go

ALTER TABLE План_виготовлення_продукції
ADD FOREIGN KEY (Код_виробничої_лінії) REFERENCES
Виробнича_лінія(Код_виробничої_лінії)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE План_виготовлення_продукції  
  ADD FOREIGN KEY (Код_застосування_алгоритму) REFERENCES  
  Застосування_алгоритму(Код_застосування_алгоритму)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION
```

go

```
ALTER TABLE План_постачання  
  ADD FOREIGN KEY (Код_регіонального_складу) REFERENCES  
  Регіональний_склад(Код_регіонального_складу)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION
```

go

```
ALTER TABLE План_постачання  
  ADD FOREIGN KEY (Код_плану_виготовлення_продукції) REFERENCES  
  План_виготовлення_продукції(Код_плану_виготовлення_продукції)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION
```

go

```
ALTER TABLE Прайс_лист_сировини  
  ADD FOREIGN KEY (Код_типу_сировини) REFERENCES  
  Тип_сировини(Код_типу_сировини)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION
```

go

```
ALTER TABLE Прайс_лист_сировини
ADD FOREIGN KEY (Код_постачальника) REFERENCES
Постачальник(Код_постачальника)
ON DELETE NO ACTION
ON UPDATE NO ACTION
Go
```

### **3.2 SQL-код створення «вітрини даних» для моніторингу виконання замовлень за певний період формування плану виконання замовлень**

```
CREATE TABLE Акт_передачі_на_склад
(
Код_відвантаженої_продукції integer IDENTITY ( 1,1 ) NOT FOR
REPLICATION ,
Дата_відвантаження datetime NULL ,
Кількість_відвантаженої_продукції integer NULL ,
Код_змінного_завдання integer NOT NULL
)
go

ALTER TABLE Акт_передачі_на_склад
ADD PRIMARY KEY CLUSTERED (Код_відвантаженої_продукції ASC)
go

CREATE TABLE Вид_продукції
(
Код_виду_продукції integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Назва_виду_продукції varchar(100) NULL
```

)

go

```
ALTER TABLE Вид_продукції
```

```
ADD PRIMARY KEY CLUSTERED (Код_виду_продукції ASC)
```

go

```
CREATE TABLE Всі_замовлення
```

(

```
Код_замовлення integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
```

```
Дата_надходження datetime NULL ,
```

```
Дата_необхідного_виконання datetime NULL ,
```

```
Кількість integer NULL ,
```

```
Розмір_штрафу integer NULL ,
```

```
Дата_фактичного_виконання datetime NULL ,
```

```
Код_продукції integer NOT NULL ,
```

```
Пріоритетність_замовлення integer NULL
```

)

go

```
ALTER TABLE Всі_замовлення
```

```
ADD PRIMARY KEY CLUSTERED (Код_замовлення ASC)
```

go

```
CREATE TABLE Залишок_сировини
```

(

```
Код_залишку_сировини integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
```

```
Кількість_залишку integer NULL ,
```

```
Дата_зняття_залишку datetime NULL ,
```

```
Код_сировини integer NOT NULL
```

)

go

```
ALTER TABLE Залишок_сировини
ADD PRIMARY KEY CLUSTERED (Код_залишку_сировини ASC)
go
```

```
CREATE TABLE Змінне_завдання
(
Код_змінного_завдання integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Дата_формування    datetime NULL ,
Дата_виконання    datetime NULL ,
Зміна              varchar(50) NULL ,
Обсяги_виготовлення_за_планом integer NULL ,
Обсяги_виготовлення_по_факту integer NULL ,
Назва_цеху        varchar(100) NULL ,
Код_плану_виробництва integer NOT NULL ,
Код_продукції    integer NOT NULL
)
go
```

```
ALTER TABLE Змінне_завдання
ADD PRIMARY KEY CLUSTERED (Код_змінного_завдання ASC)
go
```

```
CREATE TABLE План_виробництва
(
Код_плану_виробництва integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Планові_обсяги    integer NULL ,
Дата_початку_виконання_плану datetime NULL ,
```

```
Дата_закінчення_виконання_плану integer NULL ,  
Код_продукції integer NOT NULL  
)  
go
```

```
ALTER TABLE План_виробництва  
ADD PRIMARY KEY CLUSTERED (Код_плану_виробництва ASC)  
go
```

```
CREATE TABLE Повернута_сировина  
(  
Код_повернення integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Дата_повернення datetime NULL ,  
Кількість_сировини integer NULL ,  
Причина_повернення varchar(1000) NULL ,  
Код_сировини integer NOT NULL  
)  
go
```

```
ALTER TABLE Повернута_сировина  
ADD PRIMARY KEY CLUSTERED (Код_повернення ASC)  
go
```

```
CREATE TABLE Прийнята_сировина  
(  
Кількість_сировини integer NULL ,  
Дата_прийому datetime NULL ,  
Загальна_кількість_на_складі varchar(20) NULL ,  
Код_прийнятої_сировини integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Код_сировини integer NOT NULL
```

```
)  
go
```

```
ALTER TABLE Прийнята_сировина  
ADD PRIMARY KEY CLUSTERED (Код_прийнятої_сировини ASC)  
go
```

```
CREATE TABLE Продукція  
(  
Код_продукції integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Назва_продукції varchar(100) NULL ,  
Тип_оболочки varchar(100) NULL ,  
Код_виду_продукції integer NOT NULL ,  
Тип_продукції integer NULL  
)  
go
```

```
ALTER TABLE Продукція  
ADD PRIMARY KEY CLUSTERED (Код_продукції ASC)  
go
```

```
CREATE TABLE Рецептатура_виготовлення  
(  
Код_рецептури integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,  
Кількість_сировини integer NULL ,  
Код_сировини integer NOT NULL ,  
Код_продукції integer NOT NULL ,  
Номер_рецептури integer NULL  
)  
go
```

```
ALTER TABLE Рецепттура_виготовлення
ADD PRIMARY KEY CLUSTERED (Код_рецептури ASC)
go
```

```
CREATE TABLE Сировина
(
Код_сировини      integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Вид_сировини     varchar(50) NULL ,
Технологія_зберігання varchar(100) NULL
)
go
```

```
ALTER TABLE Сировина
ADD PRIMARY KEY CLUSTERED (Код_сировини ASC)
go
```

```
CREATE TABLE Технологічне_обладнання
(
Код_обладнання   integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Назва_обладнання varchar(100) NULL
)
go
```

```
ALTER TABLE Технологічне_обладнання
ADD PRIMARY KEY CLUSTERED (Код_обладнання ASC)
go
```

```
CREATE TABLE Технологічний_рух_продукції
(
```

```

Технологічний_етап varchar(100) NULL ,
Код_техн_руху_продукції integer IDENTITY ( 1,1 ) NOT FOR REPLICATION ,
Код_продукції integer NOT NULL ,
Код_обладнання integer NOT NULL
)
go

```

```

ALTER TABLE Технологічний_рух_продукції
ADD PRIMARY KEY CLUSTERED (Код_техн_руху_продукції ASC)
go

```

```

CREATE VIEW

```

```

Змінний_журнал(Дата_формування,Дата_виконання,Зміна,Назва_цеху,Обсяги_виго
товлення_по_факту,Назва_виду_продукції,Тип_продукції)

```

```

AS

```

```

SELECT

```

```

Змінне_завдання.Дата_формування,Змінне_завдання.Дата_виконання,Змінне_завдан
ня.Зміна,Змінне_завдання.Назва_цеху,Змінне_завдання.Обсяги_виготовлення_по_ф
акту,Вид_продукції.Назва_виду_продукції,Продукція.Тип_продукції

```

```

FROM Змінне_завдання,Продукція,Вид_продукції

```

```

go

```

```

CREATE VIEW

```

```

Обсяги_передачі_готової_продукції(Дата_відвантаження,Кількість_відвантаженої_п
родукції,Назва_продукції,Тип_продукції,Зміна,Назва_цеху,Назва_виду_продукції)

```

```

AS

```

```

SELECT

```

```

Акт_передачі_на_склад.Дата_відвантаження,Акт_передачі_на_склад.Кількість_відв
антаженої_продукції,Продукція.Назва_продукції,Продукція.Тип_продукції,Змінне_з
авдання.Зміна,Змінне_завдання.Назва_цеху,Вид_продукції.Назва_виду_продукції

```

```
FROM Вид_продукції,Продукція,Акт_передачі_на_склад,Змінне_завдання  
go
```

```
ALTER TABLE Акт_передачі_на_склад  
ADD FOREIGN KEY (Код_змінного_завдання) REFERENCES  
Змінне_завдання(Код_змінного_завдання)  
go
```

```
ALTER TABLE Всі_замовлення  
ADD FOREIGN KEY (Код_продукції) REFERENCES  
Продукція(Код_продукції)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Залишок_сировини  
ADD FOREIGN KEY (Код_сировини) REFERENCES  
Сировина(Код_сировини)  
go
```

```
ALTER TABLE Змінне_завдання  
ADD FOREIGN KEY (Код_плану_виробництва) REFERENCES  
План_виробництва(Код_плану_виробництва)  
go
```

```
ALTER TABLE Змінне_завдання  
ADD FOREIGN KEY (Код_продукції) REFERENCES  
Продукція(Код_продукції)  
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE План_виробництва
```

```
ADD FOREIGN KEY (Код_продукції) REFERENCES
```

```
Продукція(Код_продукції)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE Повернута_сировина
```

```
ADD FOREIGN KEY (Код_сировини) REFERENCES
```

```
Сировина(Код_сировини)
```

```
go
```

```
ALTER TABLE Прийнята_сировина
```

```
ADD FOREIGN KEY (Код_сировини) REFERENCES
```

```
Сировина(Код_сировини)
```

```
go
```

```
ALTER TABLE Продукція
```

```
ADD FOREIGN KEY (Код_виду_продукції) REFERENCES
```

```
Вид_продукції(Код_виду_продукції)
```

```
go
```

```
ALTER TABLE Рецепт_виготовлення
```

```
ADD FOREIGN KEY (Код_сировини) REFERENCES
```

```
Сировина(Код_сировини)
```

```
go
```

```
ALTER TABLE Рецепттура_виготовлення  
ADD FOREIGN KEY (Код_продукції) REFERENCES  
Продукція(Код_продукції)  
go
```

```
ALTER TABLE Технологічний_рух_продукції  
ADD FOREIGN KEY (Код_продукції) REFERENCES  
Продукція(Код_продукції)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```

```
ALTER TABLE Технологічний_рух_продукції  
ADD FOREIGN KEY (Код_обладнання) REFERENCES  
Технологічне_обладнання(Код_обладнання)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
go
```