Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»
Інститут електроенергетики
(інститут)
факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня ___магістра___
(бакалавра, спеціаліста, магістра)

Студентки _____Підгорної Катерини Дмитрівни_____
(ПІБ)
академічної групи _____126м-22з-1_____
(шифр)
спеціальності _____126 «Інформаційні системи та технології»_____
(код і назва спеціальності)
за освітньо-професійною програмою _____
_____«Інформаційні системи та технології»_____
(офіційна назва)
на тему _____Розробка системи керування ділянкою виготовлення_____
_____профнастилу з оптимальним розкроєм рулону на мові C#_____
(назва за наказом ректора)

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	Олевський В.І.			
розділів				

Рецензент	Журба А.О.			

Нормоконтролер	Коротенко Г.М.			

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій та комп'ютерної інженерії
(повна назва)
_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

«___»_____20___ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр
(бакалавра, спеціаліста, магістра)

студентці Підгорній К.Д. **академічної групи** 126м-22з-1
(прізвище та ініціали) (шифр)
спеціальності 126 «Інформаційні системи та технології»
за освітньою-професійною програмою «Інформаційні системи та технології»
на тему Розробка системи керування ділянкою виготовлення профнастилу з оптимальним розкроєм рулону на мові C#,
затверджену наказом ректора НТУ «Дніпровська політехніка» від 09.10.23 № 1228-С

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз стану області рішення задачі	09.10.2023 – 25.10.2023
Розділ 2	Проєктування інформаційної системи	26.10.2023 – 15.11.2023
Розділ 3	Програмна реалізація інформаційної системи	16.11.2023 – 08.12.2023

Завдання видано _____ _____
(підпис керівника) (прізвище, ініціали)
Дата видачі _____
Дата подання до екзаменаційної комісії _____
Прийнято до виконання _____ Підгорна К.Д.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 81 стор., 32 рис., 3 додатків, 40 джерел.

Об'єкт дослідження процеси автоматизації виробничої ділянки з виготовлення заготівок з оптимальним розкроєм рулону.

Предмет дослідження: система управління процесом керування ділянкою виготовлення заготівок.

Мета магістерської роботи: розробка інформаційної системи керування ділянкою нарізання металу на смуги з оптимальним розкроєм рулону на основі сучасних методів оптимізації та інформаційних технологій, що забезпечить ефективне використання матеріалів.

У вступній частині роботи висвітлено суть проблеми та сформульовано постановку завдань для подальшого дослідження. В першому розділі наведено опис предметної області та основні відомості про існуючі інформаційні системи оптимального розкрою матеріалу. У другому розділі наведено аналіз методів для вирішення поставленої задачі, описано методику її вирішення, представлено архітектуру, функціональну модель, проєктування бази даних та інтерфейс користувача системи. У третьому розділі виконано проєктування, реалізацію та тестування автоматизованої інформаційної системи керування виробничою ділянкою з виготовлення профнастилу.

Наукова новизна отриманих результатів полягає в застосуванні комбінаторних методів у поєднанні з симплекс-методом та методом Гоморі для вирішення задачі цілочислового програмування.

Практичне значення отриманих результатів полягає в створенні автоматизованої інформаційної системи розрахунку оптимального плану розкрою для виробничої ділянки з виготовленню заготівок, що сприяє підвищенню ефективності діяльності підприємства.

ІНФОРМАЦІЙНА СИСТЕМА, СИМПЛЕКС-МЕТОД, МЕТОД ГОМОРІ, ОПТИМАЛЬНИЙ РОЗКРІЙ, ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, C#.

ABSTRACT

Explanatory note: 81 pages, 32 figures, 3 applications, 40 sources.

Object of research: automation processes in the manufacturing area of producing blanks with optimal roll cutting

Subject of research: management system for the metal cutting area with optimal roll cutting based on modern optimization methods and information technologies, ensuring efficient material utilization.

In the introduction, the essence of the problem is highlighted, and tasks are formulated for further research. The first chapter provides a description of the subject area and basic information about existing information systems for material optimization. The second chapter includes an analysis of methods for solving the stated problem, a description of the methodology for its solution, the presentation of the system's architecture, functional model, database design, and user interface. The third chapter covers the design, implementation, and testing of an automated information management system for the production area of profiled sheets.

The scientific novelty of the obtained results lies in the application of combinatorial methods combined with the simplex method and the Gomory method to solve integer programming problems.

The practical significance of the results obtained is reflected in the creation of an automated information system for calculating the optimal cutting plan for the production area, contributing to the increased efficiency of the enterprise's activities."

INFORMATION SYSTEM, SIMPLEX METHOD, GOMORY METHOD, OPTIMAL CUTTING, OBJECT-ORIENTED PROGRAMMING, C#.

ЗМІСТ

Вступ 6
Розділ 1 Аналіз стану області рішення задачі 9
1.1 Опис предметної області 9
1.2 Огляд існуючих інформаційних систем оптимального розкрою матеріалу 12
1.3 Постановка завдання на розробку систему 21
Розділ 2 Проєктування інформаційної системи 25
2.1 Аналіз методів для вирішення поставленої задачі 25
2.2 Опис методики для вирішення задачі розрахунку оптимального розкрою рулонного матеріалу 36
2.3 Архітектура інформаційної системи 40
2.4 Функціональна модель проектованої системи 41
2.5 Проєктування бази даних інформаційної системи 51
2.6 Розробка інтерфейсу користувача системи 57
Розділ 3 Програмна реалізація інформаційної системи 63
3.1 Елементи програмної реалізації проєктованої системи 63
3.2 Тестування інформаційної системи 66
Висновки 73
Список використаних джерел 75
Додаток А 79
Додаток Б 81

ВСТУП

Ефективне використання сировини є важливою проблемою. для різних промислових галузей. Забезпечення ефективності управління матеріальними ресурсами є ключовим фактором для підвищення продуктивності виробництва та загальної діяльності підприємства. Розв'язання цього завдання напряму впливає на вартість продукції. Відходи виробництва становлять Суттєву частину вартості, тож мінімізація відходів є пріоритетним завданням.

Один із важливих факторів, що сприяє мінімізації відходів виробництва, — це раціональний розкрій матеріалів. У сучасних умовах ефективного використання матеріалів неможливо досягти без автоматизації процесу розкрою. Це обумовлено потребою в зменшенні матеріальних і часових витрат, пов'язаних із виготовленням продукції, та сприяє підвищенню коефіцієнта розкрою, забезпечуючи оптимальність витрат ресурсів.

Актуальність дослідження обумовлена тим, що ефективність процесу виконання замовлень з розкрою заготовок залежить від різноманіття складу замовлень (кількість та довжина деталей) та наявності в наявності заготовок різної довжини для розрізання. Навіть при ідентичних замовленнях із попередніми, зміна наявного матеріалу різної довжини може суттєво впливати на план розкрою. При зростанні потоку замовлень, ручне планування розкрою стає надто складним і не відповідає ефективності процесу, а існуюче програмне забезпечення здебільшого перевантажено або обмежено функціоналом, має високу вартість, потребує витрат на супровід. Розвиток технологій інженерного проектування викликає збільшений інтерес до вирішення проблем у цьому напрямі.

Представлена автоматизована інформаційна система, в якій розглядається у цій роботі, може ефективно вирішувати завдання оптимального розкрою смуг матеріалу заданої довжини. Це завдання є широко використовуваним у промисловості, де регулярно постають виклики щодо оптимізації процесів розкрою та упаковки матеріалів. Розглянутий алгоритм

може бути включений у практичні розрахунки та інтегрований в автоматизовані системи проектування та управління.

Задачі оптимального розкрою викликають значний інтерес, переважно через їхню велику практичну важливість. Ці завдання часто постають у виробництвах, де ефективне використання ресурсів є ключовим фактором для зниження собівартості виробленої продукції.

Зв’язок роботи з науковими програмами, планами, темами. Кваліфікаційна робота виконана згідно з планом наукових досліджень кафедри інформаційних технологій та комп’ютерної інженерії НТУ «ДП» як складова частина науково-дослідницької роботи на тему «Моделі й інформаційні технології обробки та аналізу даних в складних комп'ютерних системах і мережах» (державний реєстраційний номер 0121U114523).

Мета і завдання дослідження. Метою даної кваліфікаційної роботи магістра є розробка інформаційної системи керування ділянкою нарізання металу на смуги з оптимальним розкроєм рулону на основі сучасних методів оптимізації та інформаційних технологій, що забезпечить ефективне використання матеріалів.

Для досягнення поставленої мети треба вирішити наступні завдання:

– дослідити існуючі інформаційні системи оптимального розкрою матеріалу;

– проаналізувати наявні методи, підходи та алгоритми для вирішення задач розкрою та упаковки;

– представити алгоритм розв’язання задачі оптимального розкрою рулону, заснованому на використанні методів математичного програмування;

– розробка інтерфейсу користувача відповідної інформаційної системи прийняття рішень;

– проєктування бази даних інформаційної системи;

– виконання програмної реалізації спроєктованої системи;

– виконання тестування інформаційної системи.

Об'єкт дослідження – процеси автоматизації виробничої ділянки з виготовлення заготівок з оптимальним розкроєм рулону.

Предметом дослідження виступає система управління процесом керування ділянкою виготовлення заготівок.

Методи дослідження – методи оптимізації: методи комбінаторної оптимізації, симплекс-метод та метод Гоморі – для побудови математичної моделі оптимального розкрою, інформаційні технології: .NET, СУБД MS SQL Server – при розробці інформаційної системи.

Наукова новизна отриманих результатів полягає в розробці інформаційної системи на основі точних комбінаторних методів у поєднанні з симплекс-методом та методом Гоморі. Використання такого підходу надає можливість автоматизувати процес визначення можливих способів розрізу заготовки і виконати розрахунок оптимального плану, мінімізуючи при цьому залишки матеріалів.

Практичне значення отриманих результатів полягає в створенні автоматизованої інформаційної системи розрахунку оптимального плану розкрою для виробничої ділянки з виготовлення заготівок, що сприяє підвищенню ефективності діяльності підприємства, швидкості розрахунків та зменшенню відходів.

Апробація результатів дослідження. Основні положення дослідження були оприлюднені на Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих учених «Сучасні інформаційні технології: теорія, практика, перспективи» (м. Дніпро, 2023 р.).

Публікації. За результатами дослідження опубліковано 1 роботу [7].

РОЗДІЛ 1
АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ

1.1. Опис предметної області

Зростання інтенсивності виробництва вимагає якісних сучасних підходів для вирішення поставлених задач на кожному етапі виробничого процесу. Одним із ключових завдань, що стоять перед керівниками виробничих підприємств, є оптимальне використання матеріальних ресурсів. Значимість даної задачі підтверджується тим, що рішення, прийняті керівництвом підприємства у процесі планування та реалізації виробництва, впливають на основні параметри виробничої системи (обсяг виробництва у фізичному та грошовому вираженні, питомі витрати на технології, амортизація матеріальних та нематеріальних активів, операційний прибуток) та сприяють формуванню необхідних споживчих властивостей продукції [1]. Також слід зазначити, що швидкі темпи технологічного розвитку, вимоги ефективності та конкурентоспроможності зумовлюють необхідність переходу виробничих підприємств на автоматизовані системи управління [2, 3]. Ці системи дозволяють оптимізувати виробничі процеси, скорочувати витрати і підвищувати якість продукції, що стає важливим фактором в економічному оточенні, що швидко змінюється.

Сучасне автоматизоване виробництво - це складна система, яка вимагає застосування ефективних методів автоматизації управління на всіх рівнях функціонування [4, 5]. З швидким прогресом у сфері інформаційних технологій, зміни у підходах до вирішення наукових та виробничих завдань стають необхідні. Автоматизація всіх етапів виробничого процесу, включаючи проектування, технологічну підготовку та управління, визнається важливим методом прискорення науково-технічного прогресу та підвищення ефективності виробництва [6].

Зі зростанням складності виробничих об'єктів і зі скороченням термінів проектування та підготовки виробництва через жорстку конкуренцію, виникає потреба в ефективних рішеннях. Інтенсифікація роботи в галузі проектування та технологічної підготовки можлива лише завдяки використанню комп'ютерних технологій та систем автоматизованого управління [7].

Застосування автоматизованих систем управління, таких як системи автоматизованого проектування (САПР) та автоматизована система технологічної підготовки виробництва (АСТПВ), дозволяє вирішувати завдання управління, проектування та виробництва, забезпечуючи збереження конкурентоспроможності продукції, що випускається [8]. Розвиток таких систем з урахуванням сучасних інформаційних технологій дозволяє досягти найбільш ефективного управління та зберегти конкурентоспроможність вироблених виробів.

На промислових підприємствах, ефективне зниження витрат на сировину та матеріали можливе за рахунок зменшення кількості утворених відходів під час виробництва [9]. Зокрема, у системі планування виробництва металевих заготівок для виготовлення різноманітних продуктів, таких як профнастил, добірні елементи вентиляційних коробів, водостічні системи, будівельні маяки та штампована оцинкований посуд, оптимізація процесу розкрою стає ключовим чинником серед численних факторів [10, 11].

Проблема оптимального розкрою привернула значну увагу дослідників з часу її початкової появи наприкінці 40-х років, коли завдання було вперше сформульовано та вирішено. Різні аспекти оптимізації виробництва та операцій, які можуть бути пов'язані з проблемою автоматизації процесу розкрою матеріалу на виробничих підприємствах розкривали вчені в своїх наукових роботах. Джордж Немхаузер (George Nemhauser) – відомий своїми дослідженнями в галузі оптимізації та теорії прийняття рішень, досліджував задачі оптимального розкрою та комбінаторної оптимізації [12]. Ральф Гоморі (Ralph Gomory) – відомий своїми роботами в галузі дослідження операцій та управління виробничими процесами. [13]. Дональд Кнут (Donald Knuth) – його

роботи в галузі комп'ютерних алгоритмів та досліджень операцій [14] корисні для розуміння алгоритмічних підходів до завдань оптимального розкрою. Джордж Данціг (George Dantzig) – відомий своїми працями у галузі дослідження операцій, включаючи моделювання та оптимізацію виробничих процесів [15].

Дослідження вчених та їх наукові праці охоплюють різноманітні аспекти оптимізації виробництва, включаючи різні методи, алгоритми та стратегії, що можуть бути використані для автоматизації процесу розкрою матеріалу на промислових підприємствах.

У сучасний період вирішення традиційних задач оптимального розкрою в умовах виробництва стикається із рядом викликів. Серед них — необхідність використання цілісних значень для розв'язання, забезпечення наявності оптимального рішення при будь-яких вихідних умовах та інші вимоги, які визначаються особливостями роботи конкретних підприємств [16].

На основі вищевказаного, розв'язання завдання розробки системи управління ділянкою поперечної нарізки плаского прокату в рулонах з оптимальним розкроєм на виробничих підприємствах України стає особливо важливим у поточний період з кількох причин:

- ефективне використання ресурсів – в умовах нестабільності ринку та обмеженої доступності сировини під час пандемії та військових конфліктів оптимізація розкрою матеріалу стає критично важливою. Це дозволяє використовувати матеріали найбільш ефективно та мінімізувати втрати;

- скорочення витрат та економія бюджету – оптимізація процесу розкрою рулону дозволить знизити витрати на придбання сировини та матеріалів, що особливо важливо у періоди економічних криз та нестабільності, спричинених пандемією та військовими діями;

- підвищення конкурентоспроможності – ефективний розкрій матеріалу дає підприємству можливість бути більш конкурентоспроможним, оскільки це покращує виробничі показники та якість продукції, що особливо важливо в умовах обмеженої доступності ресурсів.

- гнучкість виробництва – автоматизовані системи управління ділянкою нарізки виробів дозволяють швидко адаптуватися до умов виробництва, що змінюються, в періоди кризи, що особливо цінно в умовах пандемії та військових дій;

- оптимізація трудовитрат та виробничих процесів – автоматизовані системи управління дозволяють скоротити час на підготовку матеріалу до виробництва, що підвищує ефективність праці та загальну продуктивність.

Враховуючи економічну нестабільність, обмеженість постачання матеріалів та необхідність забезпечення ефективного виробництва в умовах післяпандемії та воєнних дій, розробка системи управління ділянкою нарізки плаского прокату з оптимальним розкроєм стає важливим кроком для забезпечення стійкості та ефективності виробництва підприємств в Україні.

1.2. Огляд існуючих інформаційних систем оптимального розкрою матеріалу

Огляд існуючих інформаційних систем за досліджуваною тематикою є важливим етапом перед розробкою нового продукту, оскільки він надає інформацію та враження, які можна використовувати для покращення та оптимізації власного програмного рішення. Таке дослідження виконується з кількох причин:

- вивчення найкращих практик – огляд існуючих систем дозволяє розробникам вивчити та оцінити найкращі практики, які вже застосовані в інших програмних рішеннях. Це дозволяє уникнути повторення помилок та використати оптимальні рішення;

- визначення функціональності – огляд систем допомагає визначити функціональність, яка може бути корисною або необхідною для цільової аудиторії. Він сприяє розумінню вимог користувачів та можливостей, які вони вже використовують у подібних системах;

- аналіз недоліків – оцінка існуючих систем дозволяє виявити їхні недоліки та обмеження. Це надає можливість розробникам підходити до розробки власного продукту з урахуванням уникнення аналогічних проблем та недоліків;

- інновації та удосконалення – огляд інших систем може надихнути розробників на впровадження новаторських ідей та покращення. Спостереження за тим, як інші системи вирішують проблеми, може сприяти розвитку більш ефективних та інноваційних рішень;

- орієнтація на потреби користувачів – зрозуміння того, як існуючі системи взаємодіють з користувачами, дозволяє розробникам краще орієнтуватися на потреби кінцевих користувачів. Це важливо для створення програмного продукту, який буде відповідати очікуванням та вимогам користувачів.

Огляд існуючих інформаційних систем перед розробкою нового продукту важливий для вивчення найкращих практик, визначення функціональності, аналізу недоліків, вдосконалення та інновацій. Цей процес допомагає розробникам уникнути помилок, враховувати потреби користувачів і створювати ефективні та інноваційні рішення.

На сучасному ринку інформаційних технологій з представлених інформаційних систем, спрямованих на оптимізацію розкрою листових матеріалів можна виділити наступні.

OptCut - це програмне забезпечення, призначене для оптимального розкрою листового матеріалу, такого як метал, деревина, пластик тощо [17]. OptCut має інтуїтивний інтерфейс та можливості налаштування параметрів для максимізації використання матеріалу та мінімізації відходів.

Система OptCut пропонує наступні функції розкрою листового матеріалу:

- розраховує оптимальний розклад – повертає найбільш ефективний розклад деталей на листі матеріалу з метою мінімізації відходів та оптимального використання ресурсів;

- використовує різні конфігурації параметрів – надає можливість налаштовувати різноманітні параметри, такі як розмір листа, кількість деталей, необхідність урахування обмежень, що допомагає враховувати конкретні вимоги виробництва;

- виконує візуалізацію результатів – надає інструменти для візуалізації розкладу деталей на матеріалі, що допомагає наочно оцінити оптимальний розріз;

- забезпечує автоматизацію процесу – забезпечує швидкий та автоматизований розрахунок оптимального розкладу, що спрощує процес планування виробництва та зменшує час на розрахунки;

- забезпечує ефективне управління матеріалами – допомагає оптимізувати використання матеріалів, мінімізуючи втрати та забезпечуючи ефективне використання ресурсів.

Інтерфейс додатку OptCut представлено на рисунку 1.1.

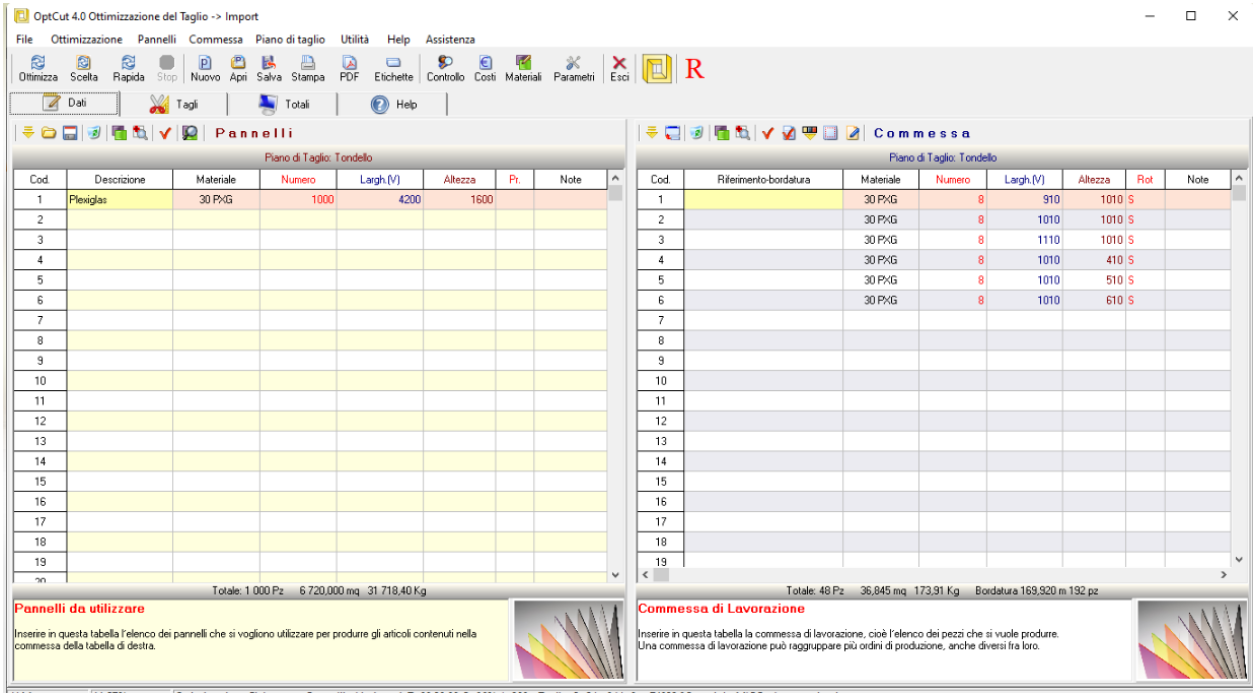


Рис. 1.1. Інтерфейс програми OptCut [17]

Програми CutLogic 1D та CutLogic 2D спрощують процес оптимізації розкрою матеріалу [18, 19], що дозволяє ефективніше використовувати ресурси та зменшити витрати.

CutLogic 1D – це програмне забезпечення для оптимізації розкрою пруткового матеріалу (наприклад, прутки, труби, профілі) [18]. Воно допомагає мінімізувати відходи матеріалу та оптимізувати використання пруткових виробів. Основні можливості CutLogic 1D включають:

- автоматизоване створення оптимальних планів розкрою для пруткового матеріалу;

- можливість роботи з різними типами матеріалу та формами (круглі, квадратні, прямокутні, труби тощо);

- врахування обмежень та умов розкрою;

- зручний інтерфейс користувача для введення параметрів та відстеження результатів.

Інтерфейси додатків CutLogic 1D та CutLogic 2D представлені відповідно на рисунках 1.2 та 1.3.

CutLogic 2D є розширенням CutLogic 1D і призначене для оптимізації розкрою листового матеріалу [19]. Воно розв'язує задачі, пов'язані з мінімізацією відходів при розкрої аркушів матеріалу. Основні функції CutLogic 2D включають:

- підтримку розрахунків для двох типів різання – гільйотинне (від краю до краю) та негільйотинне різання;

- можливість виконання кількох етапів різання для оптимального розкладу деталей;

- настроювання орієнтації першого зрізу – дозволено встановлювати вертикальну або горизонтальну орієнтацію першого зрізу для оптимального розкладу;

- можливість групової оптимізації – система забезпечує можливість групової оптимізації для знаходження оптимального плану розкладу для певних груп деталей.

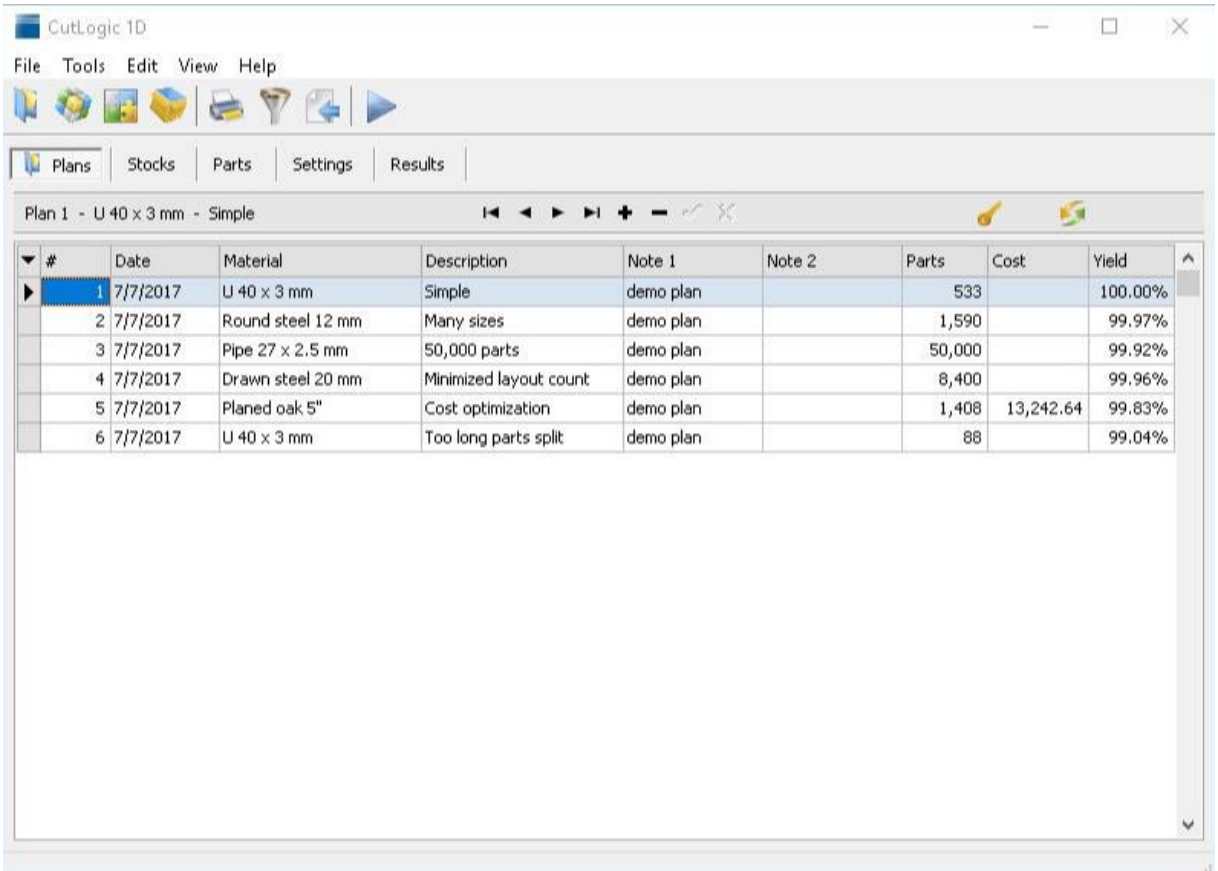


Рис. 1.2. Інтерфейс програми CutLogic 1D [18]

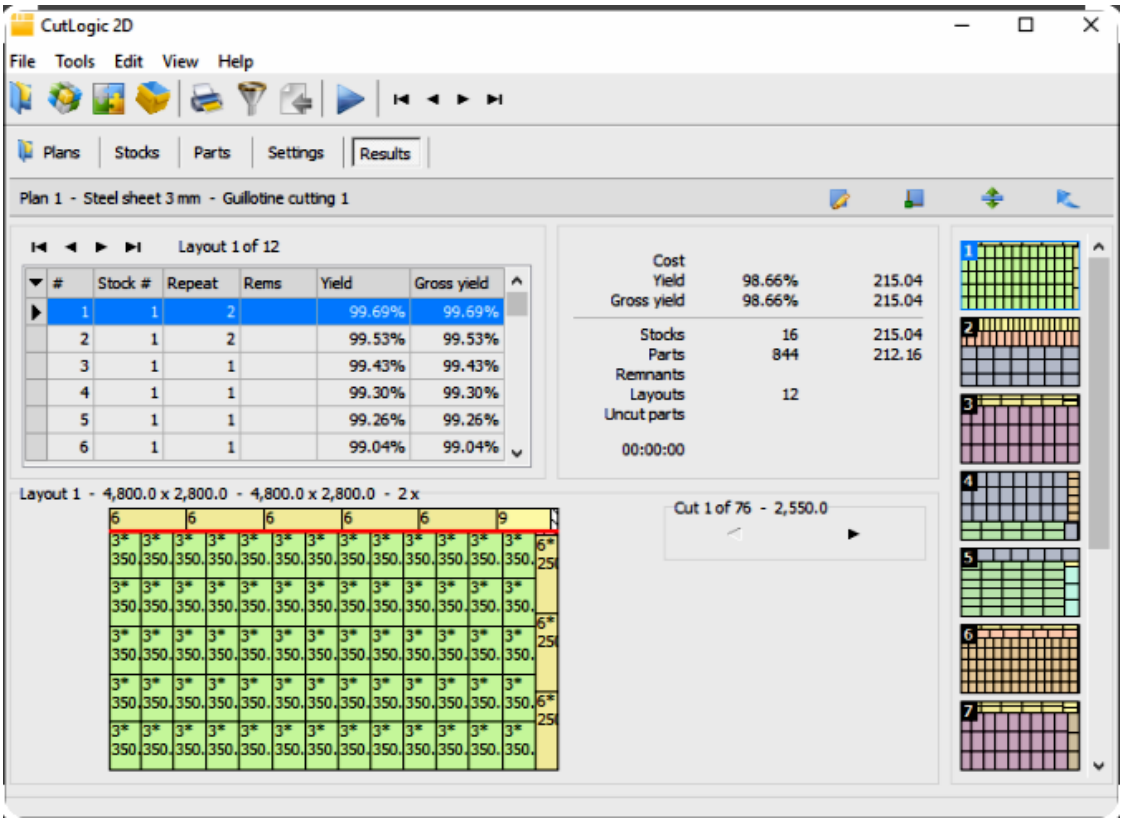


Рис. 1.3. Інтерфейс програми CutLogic 2D [19]

Обидві програми спрощують процес оптимізації розкрою матеріалу, що дозволяє ефективніше використовувати ресурси та зменшити витрати.

Real Cut 1D — це комп'ютерна програма, яка використовується для отримання оптимальних конструкцій різання одновимірних частин, які можуть мати різні кути на 90 градусів на своїх кінцях [20].

Real Cut 1D можна використовувати для різання лінійних деталей, таких як прутки, труби, балки, труби, сталеві стрижні, профілі даху, балюстради, огорожі, дерев'яні рами, алюмінієві профілі, профілі з ПВХ, нейлонові профілі, мостові рейки, універсальні колони, вікна. рами тощо. Real Cut 1D створює оптимальні рішення, використовуючи мінімальну кількість матеріалів.

Інтерфейс додатку Real Cut 1D представлено на рисунку 1.4.

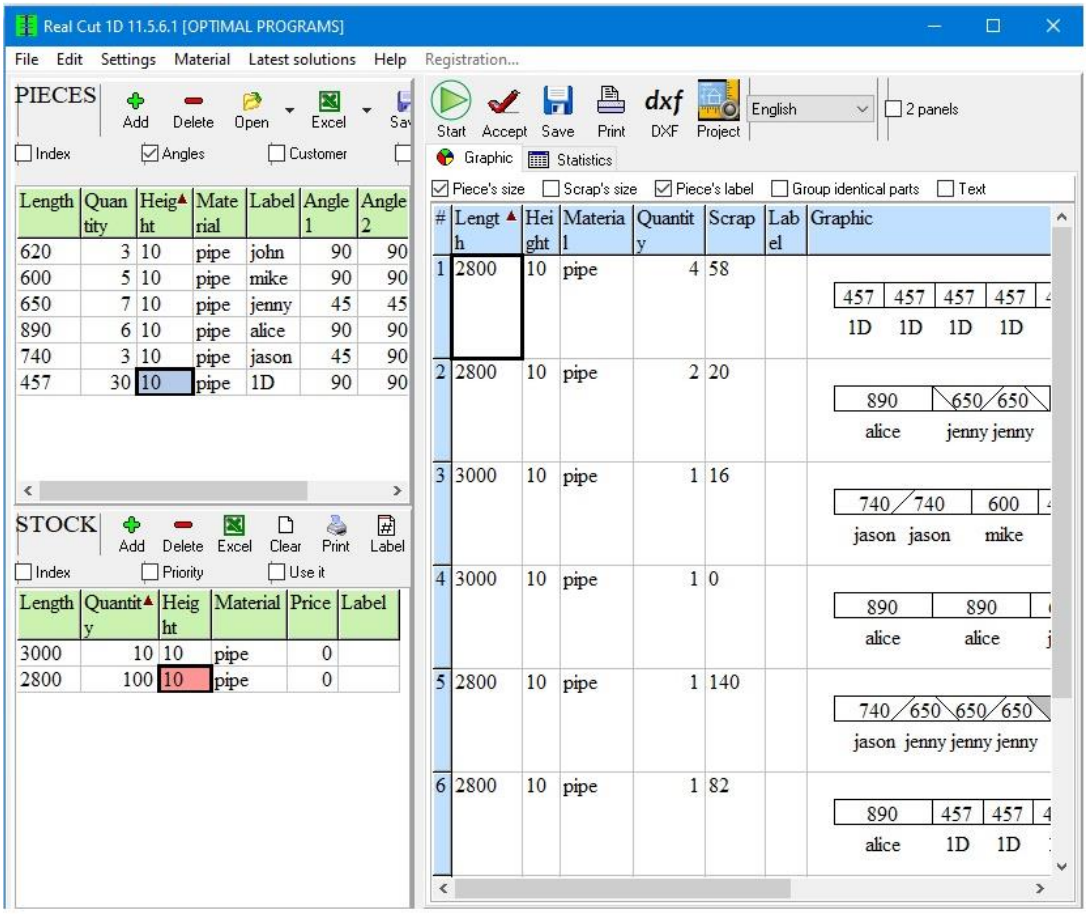


Рис. 1.4. Інтерфейс програми Real Cut 1D [20]

CutMaster 2D – це програмне забезпечення пропонується для оптимізації різання для професійних та домашніх майстерень з метою підвищення продуктивність та рівеня обслуговування, використовуючи новітні технології оптимізації [21].

CutMaster 2D – це пакет програмного забезпечення для прямокутного розкрою. Воно дозволяє максимально ефективно використовувати матеріал за рахунок створення оптимізованих схем розкрою з використанням передових алгоритмів. Адаптований для використання в листовій, меблевій, скляній та подібних галузях промисловості.

Інтерфейс додатку CutMaster 2D представлено на рис. 1.5.

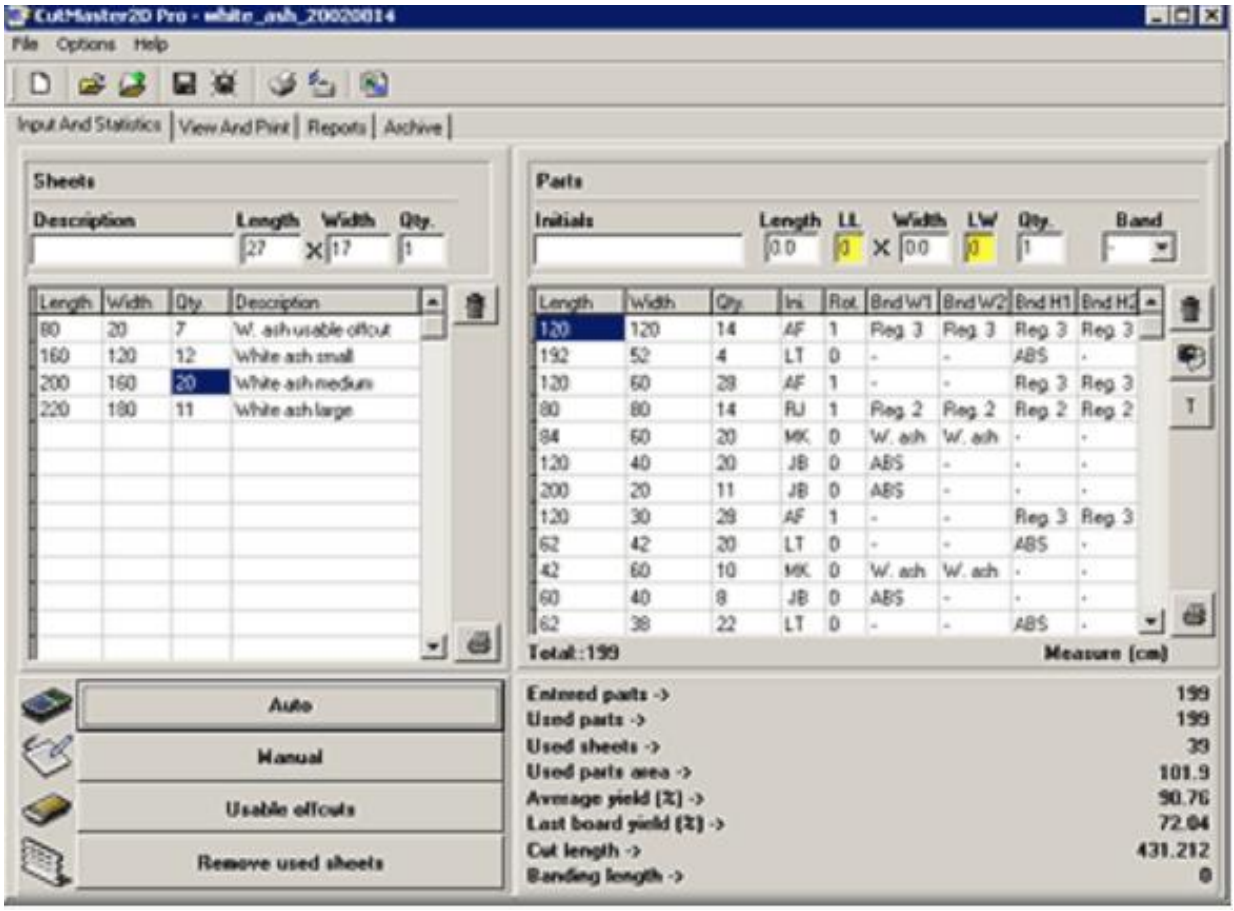


Рис. 1.5. Інтерфейс програми CutMaster 2D [21]

CutMaster 2D пропонується у двох редакціях - CutMaster 2D Lite v1.5.3 та CutMaster 2D Pro v1.5.3. Перший пакет оптимізований для використання в

домашніх майстернях, якість оптимізації така сама, як і у версії Pro, але має обмежений функціонал. Другий пакет виробники пропонують використовувати у професійному середовищі. У таблиці 1.1 наведено функціональні можливості обох пакетів програмного забезпечення.

Таблиця 1.1

Порівняльна таблиця функціональних можливостей CutMaster 2D Lite v1.5.3 та CutMaster 2D Pro v1.5.3

	CutMaster 2D Lite v1.5.3	CutMaster 2D Pro v1.5.3
Скорочення відходів та вартості панелей за рахунок високого виходу матеріалу та повторного використання відходів.	+	+
Розрахунок кромки.	-	+
Управління напрямом зерна, яке налаштовується для кожної деталі у списку.	+	+
Імпорт даних про деталі та панелі з файлів Excel або XML.	-	+
Експорт даних у файли AutoCAD DXF та XML.	-	+
Докладний звіт, що показує список використаних панелей та деталей, розміри та статистику компонування.	-	+
Маркування деталей із зазначенням типу кромки на розрахунковому макеті.	+	+
Просте введення даних за допомогою клавіатури, файлів даних Microsoft Excel, XML або CutMaster 2D.	+	+
Друк етикеток, що настроюється.	+	+

CutMaster 2D може працювати як із автоматизованим обладнанням для різання, так і без нього.

Огляд існуючих інформаційних систем автоматизації процесу розкрою матеріалів виявив різноманітні підходи та рішення, спрямовані на вдосконалення ефективності та точності розкрою. Серед них виділяються певні тенденції, такі як використання передових алгоритмів оптимізації, інтеграція з сучасним обладнанням для різання та автоматизованими системами управління виробництвом. Такі вимагають стрімких розвинень відповідно до потреб сучасних промислових підприємств, сприяючи ефективнішому використанню матеріальних ресурсів та зменшенню витрат у виробничих процесах.

Незважаючи на те, що розробники, які підтримують свій продукт, час від часу можуть виправляти та оновлювати програми та додатки, наведені інформаційні системи мають певні вади:

- обмеження функціоналу: деякі програми можуть мати обмежений функціонал порівняно з більш складними або професійними аналогами;

- перевантаження функціоналу: інформаційні системи для автоматизації процесу розкрою матеріалів можуть бути надмірно навантажені різноманітними функціями та можливостями. Це може впливати на ефективність та швидкодію системи, особливо при обробці великого обсягу даних;

- складність використання: для деяких користувачів може знадобитися певний час, щоб зрозуміти та вивчити усі можливості програми;

- неактуальність інтерфейсу: деякі програми можуть не мати сучасного інтерфейсу або можуть виглядати застаріло;

- вартість: висока вартість професійних або розширених версій програм та супутніх витрат може стати перешкодою для широкого впровадження автоматизованих систем у виробничих підприємствах. Великі витрати на обладнання та програмне забезпечення можуть бути важким фінансовим

тягарем, особливо для невеликих підприємств чи тих, що опинилися в умовах обмежених бюджетних ресурсів.

З огляду на вищезазначене, можна визначити, що розробка та впровадження інформаційних систем для розкрою матеріалів є затребуваною та актуальною задачею з урахуванням всіх аспектів функціональних вимог підприємства або ділянки виробництва та зокрема оцінюванням співвідношення між потребами, вартістю та очікуваними вигодами.

1.3. Постановка завдання на розробку систему

Під час виробництва на підприємстві, при обробці замовлень, виникає необхідність вирішення задачі автоматизації розрахунку оптимального розкрою листів металу заданих розмірів на прямокутні заготовки. Це завдання передбачає визначення оптимального розміщення на листі зазначених замовлень без перекриття, з метою подальшого ефективного використання гільйотини з мінімальними витратами матеріалу. Гільйотинний розкрій в даному випадку описується як послідовність наскрізних розрізів, паралельних кромкам матеріалу [22].

На сьогодні та в передбачуваному майбутньому в різних галузях виробництва виникатимуть та залишатимуться актуальними завдання ресурсозбереження та енергоефективності [23], пов'язані з проблемами розкрою та упаковки (компонування) [24, 25]. Ці завдання охоплюють:

- ефективне розміщення матеріалу для вирізання заготовок різної конфігурації, що має вирішальне значення в виробництві у різних секторах промисловості, таких як машинобудівна, авіаційна, суднобудівна, текстильна, шкіряна, деревообробна, меблева та інші;

- складання вантажів у різні типи контейнерів, розробка генеральних планів для промислових підприємств, розташування двигунів та радіоелементів на платах тощо;

- визначення розподілу ресурсів, включаючи обсяг пам'яті обчислювальних машин та визначення областей лісу для вирубки чи посадки.

Всі ці завдання відносяться до проблеми оптимізації геометричного моделювання, яка передбачає оптимізацію розташування об'єктів відповідного виду в заданих областях.

Як зазначалося раніше, задача розкрою матеріалу — це задача оптимізації, яка полягає у визначенні найбільш ефективного способу розрізання матеріалу (наприклад, листа або рулону) для отримання заданих деталей з мінімальними втратами матеріалу. Ця задача є важливою у виробництві та дизайні (розташуванні елементів) для економії матеріальних ресурсів [26, 27].

Отже, метою роботи є розробка інформаційної системи керування ділянкою нарізання металу на смуги з оптимальним розкроєм рулону на основі сучасних методів оптимізації та інформаційних технологій, що забезпечить ефективне використання матеріалів.

Для досягнення поставленої мети треба вирішити наступні завдання:

– дослідити існуючі інформаційні системи оптимального розкрою матеріалу;

– проаналізувати наявні методи, підходи та алгоритми для вирішення задач розкрою та упаковки;

– представити алгоритм розв’язання задачі оптимального розкрою рулону, заснованому на використанні методів математичного програмування;

– розробка інтерфейсу користувача відповідної інформаційної системи прийняття рішень;

– проєктування бази даних інформаційної системи;

– виконання програмної реалізації спроєктованої системи;

виконання тестування інформаційної системи.

Для розробки інформаційної системи оптимального розкрою матеріалу пропонується обрати мову програмування C# [28], що може бути обґрунтовано рядом факторів:

- широкі можливості розробки. C# є мовою програмування, яка підтримує різноманітні парадигми програмування, включаючи об'єктно-орієнтований, процедурний та функціональний підходи. Це дає розробникам широкі можливості вибору та ефективне використання різних концепцій при реалізації програмного забезпечення;

- інтеграція з іншими технологіями Microsoft. C# розроблена корпорацією Microsoft, тому вона має сильну інтеграцію з іншими технологіями та інструментами, такими як .NET Framework і Visual Studio. Це полегшує створення високоефективних та сучасних додатків;

- багатопоточність та паралельна обробка. C# надає зручність у роботі з багатозадачністю та можливість легко використовувати багатопоточність для оптимізації виконання програми. Це корисно для оптимального розкрою матеріалу, де може виникати необхідність обробки великої кількості даних паралельно;

- безпека та стабільність. C# має вбудовані механізми безпеки, такі як система типізації, яка дозволяє виявляти помилки на етапі компіляції, що сприяє створенню більш стабільного та безпечного програмного коду;

- широкі можливості для розробки інтерфейсів. Для створення інтерфейсу користувача для системи оптимального розкрою матеріалу можна використовувати різні технології та бібліотеки, доступні для мови C#. Це дозволяє розробникам створювати інтуїтивно зрозумілі та ефективні інтерфейси;

- активна спільнота та документація. C# має велику та активну спільноту розробників, а також широку документацію, що сприяє вирішенню проблем та отриманню підтримки у випадку необхідності;

- швидкість та продуктивність. C# володіє високою швидкістю виконання, що робить її відмінним вибором для розробки продуктивних та еф

ективних додатків. Із застосуванням оптимізованого коду, написаного на #C, можна досягти високої продуктивності при розрахунках та обробці даних, що може бути важливим для систем оптимального розкрою матеріалу.

Враховуючи ці фактори, можна зазначити, що використання мови програмування C# є обґрунтованим вибором для розробки інформаційної системи оптимального розкрою матеріалу, оскільки це дозволить забезпечити ефективність, безпеку, швидкість та легку інтеграцію з іншими технологіями Microsoft.

РОЗДІЛ 2
ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Аналіз методів для вирішення поставленої задачі

Серед існуючих методів вирішення задачі розкрою матеріалу можна виділити наступні:

- жадібний метод (Greedy Algorithm);
- динамічне програмування (Dynamic Programming);
- метод лінійного програмування (Linear Programming);
- метаевристики (Metaheuristics);
- точні алгоритми (Exact Algorithms);
- методи штучного інтелекту.

Розглянемо кожний метод більш детально.

Жадібний метод (Greedy Algorithm) – за цим методом деталі розміщуються на матеріалі в порядку їхнього вибору, враховуючи доступну площу та мінімізуючи втрати [29]. Це алгоритм, який приймає локально оптимальні рішення на кожному кроці в надії досягти глобально оптимального рішення. У випадку задачі розкрою матеріалу, жадібний метод допомагає вибирати деталі для розміщення на матеріалі так, щоб на кожному кроці максимізувати використання доступної площі та мінімізувати втрати матеріалу.

Основні кроки жадібного методу для задачі розкрою матеріалу можуть бути такими:

1. Вибір деталі. Обирається наступна деталь для розміщення на матеріалі. Вибір може базуватися на різних критеріях, таких як розмір деталі, пріоритет виробництва, чи інші фактори.

2. Розміщення на матеріалі. Обрана деталь розміщується на доступному матеріалі, враховуючи його розміри та форму, таким чином, щоб максимізувати використання площі.

3. Оновлення доступного матеріалу. Розміщену деталь видаляємо зі списку доступного матеріалу або враховуємо її в подальших обчисленнях.

4. Повторення: Процес вибору та розміщення деталей повторюється до тих пір, поки всі деталі не будуть розміщені на матеріалі.

Жадібний метод є евристичним підходом, де на кожному кроці вибирається локально оптимальний варіант з метою досягнення глобальної оптимізації. Розглянемо приклад задачі розкрою з використанням жадібного методу.

Приклад - задача розкрою листа. Параметри:

W та L - ширина та довжина матеріального листа;

n - кількість деталей;

w_i та l_i - ширина та довжина кожної деталі;

Жадібний алгоритм:

Крок 1 – відсортувати деталі за одним з критеріїв (наприклад, за зростанням ширини або довжини).

Крок 2 – розмістити деталі на матеріальному листі, вибираючи їх у відсортованому порядку. Додати нову деталь, якщо вона поміщається на залишковому просторі.

Математична постановка:

- цільова функція: максимізувати кількість розміщених деталей.

- обмеження: деталі повинні поміщатися на матеріальний лист без перекривання.

Алгоритм:

- на кожному кроці вибрати деталь, яка виглядає найбільш оптимально з точки зору максимізації використання матеріалу.

- розмістити вибрану деталь на матеріальному листі.

- повторювати цей процес, доки є можливість розмістити нові деталі.

Цей жадібний метод може допомогти знайти швидке та ефективне рішення для задачі розкрою, хоча воно може не завжди бути глобально оптимальним.

Отже, жадібний метод добре підходить для простих задач та може привести до швидших рішень, але не завжди гарантує оптимальність результату. Важливо враховувати специфіку задачі та вибирати критерії вибору деталей, щоб досягти певних цілей оптимізації.

Динамічне програмування (Dynamic Programming) застосовується для вирішення задач розкрою з обмеженими розмірами, де використовується оптимальна підструктура [29, 30]. Цей метод дозволяє ефективно розв'язувати задачі, які можуть бути розділені на менші підзадачі, і вирішувати їх знизу вгору, зберігаючи та використовуючи проміжні результати для запобігання зайвого повторення обчислень.

Основні кроки методу динамічного програмування у вирішенні задачі розкрою:

Формулювання оптимальної підструктури: Розкривна задача ділиться на менші підзадачі, розв'язання яких також призводить до оптимального рішення загальної задачі.

Створення таблиці: Створюється таблиця для зберігання інтермедійних результатів. Розмір таблиці визначається обсягом можливих варіантів розкрою та обмеженнями розмірів.

Заповнення таблиці: Проходженням по таблиці здійснюється обчислення оптимальних значень для кожної підзадачі. Результати зберігаються у таблиці для подальшого використання.

Визначення оптимального рішення: За допомогою таблиці визначається оптимальний варіант розкрою, який задовольняє умовам задачі.

Нехай задача розкрою полягає в розкритті прямокутних деталей з матеріального листа з мінімізацією втрат матеріалу. Припустимо, що маємо матеріальний лист фіксованого розміру і декілька деталей, кожна з яких має власні ширину і довжину.

1. Параметри:

W та L - ширина та довжина матеріального листа;

n - кількість деталей;

w_i та l_i - ширина та довжина кожної деталі.

2. Змінні:

x_i та y_i - координати верхнього лівого кута кожної деталі на матеріальному листі.

3. Цільова функція:

$$\min \sum_{i=1}^{n} (W - x_i) \cdot (L - y) \quad (2.1)$$

4. Обмеження:

- деталі повинні поміщатися на матеріальний лист, тобто для всіх i:

$$(x_i + w_i) \leq W \quad (2.2)$$

$$(y_i + l_i) \leq L \quad (2.3);$$

- деталі не повинні перекриватися, тобто $(x_i + w_i) \leq x_j$ або $(x_j + w_j) \leq x_i$ або $(y_i + l_i) \leq y_j$ або $(y_j + l_j) \leq y_i$ для всіх i та j при $i \neq j$.

Ця математична постановка визначає оптимізаційну задачу розкрою матеріалу, де цільова функція виражає мінімізацію витрат матеріалу, а обмеження визначають коректне та неперекриваюче розміщення деталей на матеріальному листі. Розв'язок цієї задачі дозволяє знайти оптимальний спосіб розкрою для мінімізації втрат матеріалу.

Метод динамічного програмування дозволяє ефективно вирішувати складні задачі розкрою матеріалу, зокрема враховуючи обмежені розміри та оптимальну підструктуру, що допомагає уникнути зайвого перерахування та покращити ефективність розрахунків.

Метод лінійного програмування (Linear Programming) – математичний метод оптимізації [31, 32] для вирішення задачі розкрою, де визначаються обмеження та мінімізується функція втрат.

Приклад математичної постановки задачі лінійного програмування для задачі розкрою:

Уявімо, що маємо задачу розкрою, де є дві деталі (А та В) і матеріальний лист (М). Кожна деталь має власні розміри (довжина та ширина), і наша мета - розмістити ці деталі на матеріальному листі так, щоб мінімізувати втрати матеріалу.

1. Змінні рішення:

x_A - кількість деталей А на листі.

x_B - кількість деталей В на листі.

2. Цільова функція (мінімізація витрат матеріалу):

мінімізувати $f(x_A, x_B)$ = Витрати матеріалу

3. Обмеження:

- кількість деталей:

$x_A \geq 0$, $x_B \geq 0$

-розміщення на листі:

$a \cdot x_A + b \cdot x_B \leq L$,

де a, b - розміри деталей, L - розмір матеріального листа.

4. Перетворення задачі на лінійне програмування:

мінімізувати $f(x_A, x_B) = c_A \cdot x_A + c_B \cdot x_B$

за обмеженнями:

$a \cdot x_A + b \cdot x_B \leq L$

$x_A \geq 0$, $x_B \geq 0$,

де c_A,c_B - вартість одиниці площі деталей А та В відповідно.

Цей приклад дозволяє визначити оптимальну кількість деталей кожного типу на матеріальному листі так, щоб мінімізувати втрати матеріалу.

До переваг методу лінійного програмування в задачі розкрою відносять гнучкість - може легко враховувати різні обмеження та умови задачі розкрою, що робить його гнучким методом оптимізації та точність рішення - можна враховувати точні вимоги щодо розміщення деталей.

Також для розв'язання оптимізаційних задач лінійного програмування використовують симплекс-метод. Цей числовий алгоритм використовується для знаходження максимального чи мінімального значення лінійної цільової функції за умов обмежень у вигляді лінійних рівнянь та нерівностей [33]. Алгоритм працює шляхом переходу між вершинами випуклої оболонки, де розташовані допустимі точки, з метою знаходження оптимальної вершини, що відповідає максимальному чи мінімальному значенню функції.

Основні кроки симплекс-методу включають в себе вибір початкової опорної вершини, зміну опорної вершини для покращення значення цільової функції та перевірку умов зупинки для досягнення оптимального розв'язку. Метод широко використовується в області оптимізації та лінійного програмування.

До обмежень та недоліків можна віднести складність моделювання – побудова точної математичної моделі може виявитися непростою для складних задач розкрою та слід враховувати, що вирішення великих задач лінійного програмування може вимагати значних обчислювальних ресурсів.

Таким чином, метод лінійного програмування є потужним інструментом для оптимізації задач розкрою, але варто враховувати складність моделювання та обчислень при вирішенні великих задач.

Метаевристичні методи представляють собою високорівневі алгоритмічні стратегії, які використовуються для наближеного вирішення оптимізаційних задач у випадках, коли класичні методи можуть бути надто обтяжливими або неефективними [34]. Однією з таких задач є задача розкрою, яка виникає у виробничих процесах при необхідності оптимального розміщення елементів на матеріалі для максимізації використання ресурсів та мінімізації втрат матеріалу.

Еволюційні алгоритми, генетичні алгоритми та інші метаевристики є підходами, які створюють моделі процесів природного відбору та еволюції. для пошуку оптимальних або прийнятних рішень. Вони працюють шляхом

створення та модифікації популяцій можливих рішень шляхом імітації генетичних операцій, таких як схрещування, мутація та відбір.

У випадках, коли задача розкрою стикається зі складністю або великим обсягом даних, метаевристики можуть забезпечити ефективний метод наближеного розв'язання, хоча це може бути пов'язано зі збільшенням обчислювальних витрат.

Точні алгоритми, зокрема алгоритми гілок та меж, метод повного перебору спрямовані на досягнення точного оптимального рішення [29, 35] в задачі розкрою.

Розглянемо метод гілок та меж на конкретному прикладі.

Метод гілок та меж розпочинається зі створення початкового рішення, яке може бути необов'язково оптимальним. Потім це рішення розбивається на більше число менших задач, використовуючи стратегію гілкування. Для кожної нової задачі обчислюється верхня межа (верхня оцінка) та нижня межа (нижня оцінка) на оптимальне значення. Якщо верхня межа не менша за нижню, то можна виключити деякі гілки з подальших розглядів, що полегшує обчислення.

Процес гілкування та обчислення меж повторюється для кожної нової гілки до тих пір, поки не буде знайдено точне оптимальне рішення або досягнуто заданий критерій зупинки.

Задача розкрою може бути математично сформульована, використовуючи, наприклад, цільову функцію та обмеження. Ось приклад для задачі розкрою матеріалу на прямокутні заготовки.

Цільова функція: мінімізувати втрати матеріалу при розкрої, що може бути визначено як сума площ заготовок, які не використовуються, або як сума невикористаної площі матеріалу.

Обмеження:

- геометричні обмеження – розміри кожної заготовки та матеріалу повинні задовольняти геометричні вимоги;

- кількісні обмеження – кількість доступних заготовок та їхні розміри обмежені;

- орієнтаційні обмеження – можливість обертання заготовок і матеріалу з обмеженнями на кут.

Математична постановка. Нехай:

n - кількість заготовок;

m - кількість різних розмірів матеріалу;

x_{ij} - бінарна змінна, яка вказує, чи використовується заготовка i на матеріалі j;

a_i та b_i - довжина та ширина заготовки i;

c_j та d_j - довжина та ширина матеріалу j.

Цільова функція:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(a_i b_i - x_{ij} c_j d_j) \rightarrow \min \quad (2.4)$$

Обмеження:

$$\sum_{i}^{n} x_{ij} \leq 1, \quad \forall_j = 1,\ldots, m \quad (2.5)$$

$$\sum_{j}^{m} x_{ij} = 1, \quad \forall_j = 1,\ldots, n \quad (2.6)$$

$$x_{ij} \in \{0, 1\} \quad (2.7)$$

Ця постановка враховує, що кожна заготовка може бути використана лише один раз і що матеріал розрізається так, щоб мінімізувати втрати.

Щодо методу повного перебору - він дозволяє отримати точні результати. Це алгоритм, який розглядає всі можливі комбінації рішень задачі з метою знаходження оптимального або найкращого рішення [33, 36]. Цей метод використовується для точного вирішення задач, коли є необхідність розглянути всі можливі варіанти.

За методом повного перебору виконується генерація всіх можливих рішень, оцінюється кожне рішення, здійснюється вибір оптимального

рішення, виконується перевірка оптимальності та виводиться результат в якості відповіді.

Основною перевагою методу повного перебору є те, що він гарантує знаходження оптимального рішення, оскільки розглядає всі можливі варіанти.

Однак у нього також є недолік - велика обчислювальна складність. Цей метод може бути непрактичним для великих задач через велику кількість можливих комбінацій, що призводить до значного часового споживання.

У багатьох випадках, де обчислювальні витрати є критичним фактором, використання евристик або метаевристик може бути більш ефективним підходом для наближеного знаходження оптимального рішення.

Розглянемо математичну постановку задачі оптимального розкрою матеріалу і визначимо її для методу повного перебору.

Припустимо, є матеріальний рулон або лист і декілька деталей, які ми маємо розмістити на цьому матеріалі. Кожна деталь має свої розміри та кількість. Метою є розміщення цих деталей на матеріалі так, щоб мінімізувати відходи матеріалу.

Припустимо:

M - довжина матеріалу;

W_T – ширина матеріалу;

N - кількість різних типів деталей;

L_i, W_i - довжина та ширина деталі i;

q_i - кількість деталей i.

Можна визначити бінарні змінні xi, j, які вказують, чи використовується деталь i в розміщенні на матеріалі j.

Цільова функція (мінімізація відходів):

$$\sum_{i=1}^{N} \sum_{j=1}^{M} (L_i \cdot W_i - x_{i,j} \cdot L_i \cdot W_i). \quad (2.8)$$

Обмеження:

- область значень бінарних змінних:

$$x_{i,j} \in \{0,1\}, \forall_{i,j} \tag{2.9}$$

- кожна деталь повинна бути використана не більше одного разу:

$$\sum_{j=1}^{M} x_{i,j} \leq 1, \forall_i \tag{2.10}$$

- деталі не повинні виходити за межі матеріалу:

$$\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} \cdot l_i \leq M, \tag{2.11}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} \cdot W_i \leq W_T \tag{2.12}$$

Це базова постановка задачі розкрою, де ми намагаємося знайти оптимальний спосіб розміщення деталей на матеріалі для мінімізації відходів. Однак метод повного перебору може стати обтяжливим при збільшенні кількості деталей та розмірів матеріалу через експоненційний ріст кількості можливих комбінацій. Таким чином, використання ефективніших методів, таких як динамічне програмування чи метаевристики, може бути більш практичним для великих задач розкрою.

Методи штучного інтелекту (ШІ) в контексті задачі розкрою включають в себе застосування технік машинного навчання (МН) та нейронних мереж для автоматизації процесу оптимізації розкрою матеріалу [37, 38].

Машинне навчання (Machine Learning) – використання алгоритмів машинного навчання дозволяє створювати моделі, які можуть навчатися на прикладах і вирішувати завдання без явного програмування. У випадку задачі розкрою, можливо використовувати алгоритми класифікації або регресії для передбачення оптимального розташування та кількості розрізів.

Нейронні мережі (Neural Networks) – є потужним інструментом для моделювання нелінійних залежностей. В задачі розкрою, нейронні мережі можуть вивчати складні зв'язки між параметрами розміщення деталей на матеріалі та ефективністю розкрою.

Процес реалізації методів ШІ в задачі розкрою може виглядати наступним чином:

- збір даних – створення набору даних, що містить інформацію про розміри матеріалу, розміри та кількість деталей, а також ефективність різання в конкретних умовах;

- підготовка даних – обробка та аналіз даних, можливо, шкалювання розмірів, кодування категоріальних змінних, та інші операції для покращення якості вхідних даних;

- вибір моделі – вибір типу моделі машинного навчання або нейронної мережі, яка найкраще підходить для задачі;

- тренування моделі – використання навчального набору для навчання моделі, тобто налаштування параметрів моделі так, щоб вона максимально точно передбачала результати;

- тестування та валідація – перевірка ефективності моделі на тестовому наборі даних та оцінка її загальної придатності;

- впровадження – використання навченої моделі для прогнозування оптимального розташування та розрізу при нових вхідних даних.

Важливо відзначити, що успіх методів ШІ у задачі розкрою може залежати від якості та обсягу навчальних даних, а також від правильного вибору та конфігурації моделі.

Огляд існуючих методів і моделей для вирішення задачі оптимального розкрою заготовок важливий для розробки ефективного та оптимального алгоритму. Цей процес дозволяє оцінити сильні та слабкі сторони наявних рішень, вибрати найкращі підходи та методи, а також врахувати практичний досвід вирішення схожих завдань. Результатом є створення ефективного

алгоритму, який максимально задовольняє вимоги конкретної задачі розкрою заготовок.

Отже, кожен метод має свої переваги та недоліки, і вибір конкретного методу може залежати від розмірів задачі, обмежень, і вимог щодо точності рішення.

В даному випадку пропонується використовувати у комбінації метод Гоморі та симплекс-метод. Застосовуючи цей метод до задач оптимального розкрою матеріалу, можна визначити, які деталі і в яких кількостях слід вирізати з листа матеріалу так, щоб мінімізувати відходи. Це дозволить отримати точний результат, а кількість типорозмірів зазвичай обмежується трьома видами.

Отже в даному підрозділі проведено аналіз методів для вирішення задачі розрахунку оптимального розкрою рулонного матеріалу, розглянуто математичні постановки задач, запропоновано методику для вирішення поставленої задачі.

2.2. Опис методики для вирішення задачі розрахунку оптимального розкрою рулонного матеріалу

Задача розрахунку оптимального розкрою листового матеріалу полягає в максимізації використання доступного матеріалу при вирізанні конкретних деталей або заготовок із листа. Основні цілі включають мінімізацію відходів, оптимізацію розміщення деталей на листі, а також забезпечення ефективного використання матеріалу з урахуванням обмежень, таких як розміри листа та геометричні обмеження деталей. Розв'язок цієї задачі дозволяє економити матеріали та зменшувати витрати у виробництві. Така задача відноситься до задачі лінійного розкрою [39].

Задача лінійного розкрою полягає у вирішенні таких параметрів, як довжина вихідних заготовок ($L_з$), довжина деталей ($L_{ді}$), кількість деталей ($N_{ді}$) та довжина відходів (L_{oj}). При цьому важливо дотримуватися умов: $0 < L_{oj} <$

$L_{дmin}$, що означає, що довжина відходів повинна бути менше за найменшу деталь. Розміщення деталей в заготовці розпочинається з найбільших деталей і здійснюється в порядку зменшення їхньої кількості.

Для визначення довжини відходів (залишкових частин) після виконання процесу розкрою використовується така формула:

$$Lo_j = Lз — \Sigma(Lд_i * Nд_{ij}), \quad (2.13)$$

де

i – розміри деталей, починаючи від найбільшої до найменшої;

j – номер способу розкрою;

$Nд_{ij}$ – кількість деталей $Д_i$ в j-му способі розкрою.

Кількість деталей для кожного розміру, виготовлених за усіма можливими варіантами розкрою, обчислюється за допомогою наступної формули:

$$Nд_{iрозр} = \Sigma(Nд_{ij} * Nз_j) \quad (2.14)$$

Кількість деталей у визначеному плані розкрою має точно відповідати заданій кількості деталей.

Кількість необхідних заготовок для досягнення оптимального плану розкрою розраховується за допомогою наступної формули:

$$Nз_{розр} = \Sigma Nз_j \quad (2.15)$$

Сумарна довжина всіх потрібних заготовок для виконання лінійного розкрою всіх деталей розраховується за допомогою наступної формули:

$$Lз_{\Sigma} = Lз * Nз_{розр} \quad (2.16)$$

Сумарна довжина усіх відходів, отриманих під час реалізації виявленого плану розкрою, розраховується за допомогою наступної формули:

$$Lo_{\Sigma} = \Sigma(Lo_j * Nз_j) \quad (2.17)$$

Відносна частка втрат, виниклих у процесі реалізації оптимального лінійного розкрою, від загального обсягу використаного матеріалу, розраховується за допомогою такої формули:

$$Qo = Lo_{\Sigma} / Lз_{\Sigma} \quad (2.18)$$

Отже, формується оптимальний план розкрою, який визначає, скільки заготовок необхідно використовувати і які варіанти розкрою застосовувати, з метою отримання всіх необхідних деталей у визначеній кількості при мінімізації відходів.

У якості умов обмежень встановлюємо рівність між заданою кількістю деталей $N_{ді}$ і розрахунковою кількістю $N_{дірозр}$, а також вводимо змінні $N_{зj}$, що вказують на розрахункову кількість заготовок для різних варіантів розкрою. При цьому встановлюємо умову обмеження: ці значення повинні бути цілими числами

Отже, математичну модель нашої задачі лінійного розкрою можна представити у вигляді:

Визначити найменше значення функції:

$$\sum_{i=1}^{M} L_{ei} \times K_i \to \min, \quad (2.19)$$

за умов

$$K_i \geq 0, \quad K_i \in I, \quad \mathrm{i} = \overline{1,\mathrm{M}}, \quad (2.20)$$

$$\sum_{i=1}^{M} N_{Дij} \times K_i = N_j, \quad \overline{j=1,N}. \qquad (2.21)$$

Для вирішення цілочислових задач лінійного програмування (ЦЗЛП) використовують метод Гоморі. Задача лінійного цілочисельного програмування має зовнішній вигляд, що практично ідентичний задачі лінійного програмування, за винятком додаткового обмеження, яке обумовлене тим, що змінні у цій задачі можуть приймати тільки цілі значення. Основні кроки методу Гоморі для ЦЗЛП:

1. Початкова задача лінійного програмування – визначення математичної моделі задачі оптимізації з лінійною цільовою функцією та лінійними обмеженнями;

2. Знаходження початкового неперервного розв'язку лінійної програми, використовуючи метод симплексу чи інші методи;

3. Перевірка на цілочисловість – перевірка, чи отриманий початковий розв'язок є цілочисельним. Якщо так, завдання вирішено. Якщо ні, переходження до наступних кроків;

4. Генерація гоморівських обмежень – для кожної нецілочисельної змінної генеруються гоморівські обмеження, які виключають поточний розв'язок;

5. Оновлення задачі лінійного програмування – гоморівські обмеження додаються до початкової задачі і процес розв'язання лінійної програми повторюється;

6. Повторення – кроки 3–5 повторюються до тих пір, поки не буде знайдено цілочисельний розв'язок.

Головна ідея полягає в тому, що гоморівські обмеження додаються до лінійного програмування для вибору цілочислових розв'язків. Процес може бути ітеративно повторений, додаючи нові гоморівські обмеження до тих пір, поки не буде знайдено цілочисельний розв'язок або визначено, що оптимальний розв'язок не існує.

Метод Гоморі в поєднанні з симплекс-методом використовується для розв'язання цілочислових задач лінійного програмування (ЦЗЛП). Сполучення цих методів дозволяє ефективно враховувати цілочислові обмеження під час оптимізації.

Наведена модель повністю відповідає потребам підприємства. Розв’язання цієї задачі можна отримати за допомогою симплекс-методу (оптимізація) та методу Гоморі, що забезпечить отримання точних цілочисельних результатів.

В даному підрозділі описано методику для вирішення поставленої задачі розрахунку оптимального розкрою рулонного матеріалу, наведено математичну модель, визначено обмеження.

2.3. Архітектура інформаційної системи

Запропоновано створити інформаційну систему, процес розробки якої ґрунтується на використанні передових технологій обробки інформації, ефективному зберіганні даних та інструментах для розробки програмного забезпечення.

Можливості інформаційної системи формуються за допомогою її структури (рис. 2.1):

– взаємодія з користувачем;
– збереження вхідних даних та результатів розрахунків у базі даних;
– виконання обчислень на основі представленого алгоритму;
– представлення результатів варіантів оптимального розкрою.

Блок взаємодії користувача з системою відповідає за організацію операцій з перегляду, редагування та видалення інформації, а також за візуальне подання даних. Інтерфейс забезпечує взаємодію користувача з системою як при введенні даних, так і при отриманні результатів розрахунків або рекомендацій.

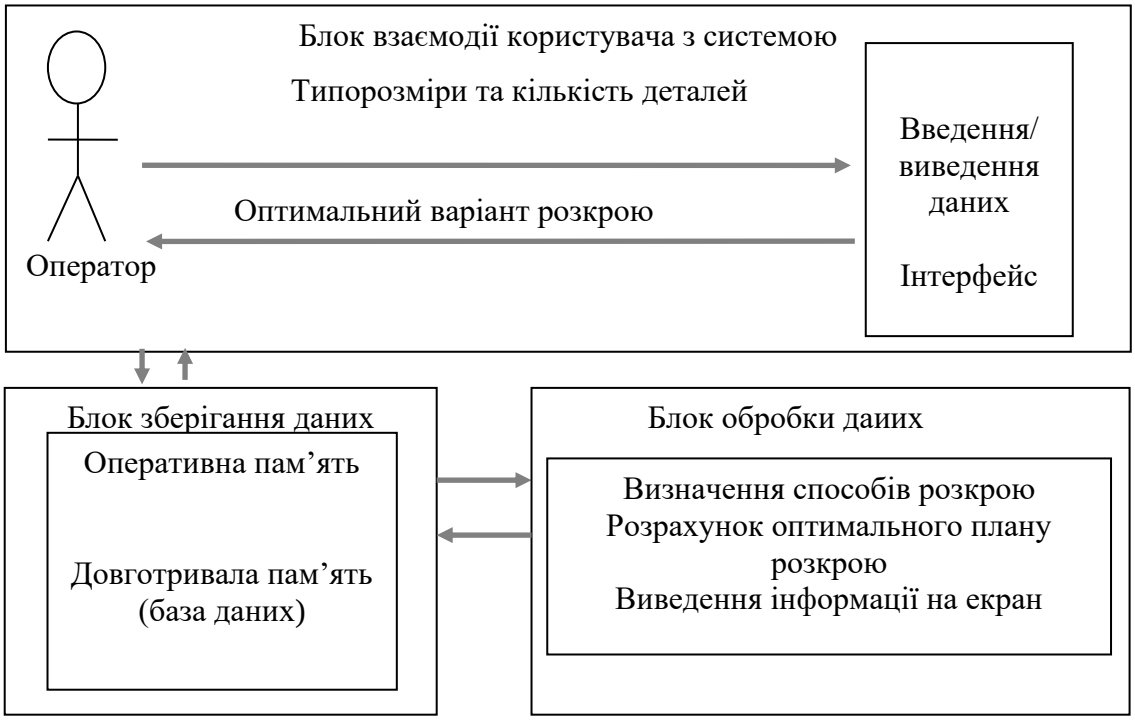


Рис. 2.1. Архітектура інформаційної системи розрахунку оптимального плану розкрою

У сегменті для зберігання даних використовується база даних MS SQL Server, де зберігається відповідні дані про замовників, типорозміри деталей, кількість деталей.

У блоці обробки інформації відбувається визначення способів розкрою матеріалу, розрахунок оптимального плану та відображення інформації на екрані.

2.4. Функціональна модель проектованої системи

Для створення та розробки системи використовується об'єктно-орієнтована технологія.

У основі об'єктно-орієнтованої методології програмування лежить підхід, в якому прикладна область представляється як сукупність взаємодіючих об'єктів, що обмінюються повідомленнями. Одиницею програмування в об'єктно-орієнтованому програмуванні (ООП) є клас, який

використовується для створення конкретних екземплярів об'єктів. Класи включають функції, що реалізують поведінку класу, та дані, які представляють атрибути класу.

У таблиці 2.1 представлено основні функції, які передбачається впровадити у проектованій системі. Де не лише визначено перелік цих функцій, але й розподілено їх за категоріями для зручності класифікації: очевидні (для користувачів), приховані (для розробників) та додаткові (необов'язкові).

Таблиця 2.1

Основні функції системи «Лінійний розкрій»

№з/п	Функції	Категорія
1.	Запуск системи	Очевидна
2.	Введення даних	Очевидна
3.	Корегування введеної інформації	Очевидна
4.	Видалення помилково введеної інформації	Очевидна
5.	Визначення способів розкрою	Прихована
6.	Визначення оптимального плану розкрою	Прихована
7.	Розрахунок ефективності отриманого плану розкрою	Прихована
8.	Відображення оптимального плану розкрою	Очевидна
9.	Збереження даних до бази даних	Прихована

Атрибути системи є функціональними характеристиками, такими як стиль інтерфейсу, відмовлення системи, час відгуку та інші. Ці атрибути можуть мати припустимий набір значень або обмеження, які характеризуються у таблиці 2.2.

Таблиця 2.2

Атрибути системи

Атрибут	Значення або обмеження
Час відгуку	Після запиту користувача інформація на екрані повинна з'явитися протягом 5с.
Стиль інтерфейсу	Діалогові вікна.
Відмовлення системи	Обробка інформації здійснюється протягом 24 годин, навіть при наявності перебоїв у мережі.
Платформа	Microsoft Windows 19

Автоматизована система розроблена як діалогова платформа для зручного використання кінцевим користувачем. Забезпечуючи виконання різноманітних завдань, ця система відповідає наступним вимогам:

- організація ефективного зберігання вхідних даних на машинному носієві;

- надання користувачеві допоміжної інформації через вивід на екран;

- захист від помилок, що можуть виникнути з боку користувача;

- формування та представлення результатів у формі запиту на екрані або їх друк у зручному для користувача форматі.

Система, яка в даний момент розробляється, має бути створена з урахуванням модульної структури, щоб надавати можливість розширення та розвитку в майбутньому.

Після визначення вимог до системи, слід ретельно розглянути всі можливі ситуації, на які програма повинна адекватно реагувати. Під ситуацією використання розуміється опис особливостей взаємодії між створеною системою та одним чи декількома суб'єктами, які взаємодіють із цією системою. Серед суб'єктів можуть бути користувачі, програми та пристрої, що взаємодіють із проектованою системою.

Однією з ситуацій є введення даних, де оператор розкрою вводить вихідні дані про замовлення з файлу або безпосередньо в програму. Система повинна надавати можливість додавання, зміни та видалення введеної інформації.

У ситуації розрахунку оптимального плану розкрою, оператор розкрою запускає процес обчислення оптимального плану відповідно до введеної інформації. Система генерує оптимальний план та відображає його на екрані. При необхідності отримані результати зберігаються у файлі. Якщо дані про замовлення вводилися вручну, вони та виведені результати зберігаються у файлі під назвою "Нове замовлення".

У ситуації змін у системі, системний програміст відповідає за підтримку системи, виправлення необхідних модулів та внесення змін до списку користувачів системи.

Розробка інформаційної системи проводиться з використанням уніфікованої мови моделювання UML, що застосовується в об'єктно-орієнтованому програмуванні.

Уніфікована мова моделювання (UML) є стандартною мовою для визначення, відображення, конструювання та документування систем, які використовують об'єктно-орієнтований підхід. UML дозволяє розробникам створювати структурні та поведінкові моделі програмних систем, щоб краще розуміти їх функції та взаємодії. Основні елементи та концепції UML включають діаграми, елементи, відносини, статус, компоненти.

Діаграми:

- структурні діаграми – описують структуру системи та її складові. До них входять класові діаграми, об'єктні діаграми, компонентні діаграми і діаграми розгортання;

- поведінкові діаграми – показують, як система взаємодіє та змінює свій стан. Це, наприклад, діаграми взаємодії (сеанси, колаборації), станові діаграми та діаграми діяльності.

Елементи:

- класи – визначають структуру об'єктів у системі та їхні взаємодії;

- об'єкти – конкретні екземпляри класів;

- інтерфейси – описують, які функції та методи може викликати об'єкт.

Відносини:

- асоціація – визначає зв'язок між об'єктами або класами;

- композиція та агрегація: – показують відносини між цілими та частинами.

Статус:

- діаграми станів – використовуються для моделювання різних станів об'єктів та їхніх переходів.

Компоненти:

- діаграми компонентів – показують фізичну організацію та взаємодію компонентів програмної системи.

UML дозволяє розробникам ефективно взаємодіяти та співпрацювати при створенні та розробці програмних систем. Вона надає стандартний набір інструментів для моделювання, що полегшує процес розробки та розуміння складних систем.

Опис усіх перерахованих ситуацій, що забезпечують виконання функціональних вимог, надає можливість визначити основні прецеденти (варіанти використання), які представлено в таблиці 2.3.

Використання діаграм прецедентів є важливою частиною процесу моделювання систем в інженерії програмного забезпечення та об'єктно-орієнтованому проектуванні. Діаграми прецедентів надають зручний спосіб визначення та відображення взаємодії між користувачами системи та її функціональністю.

Таблиця 2.3

Перелік користувачів системи й варіантів використання

Користувач	Найменування прецеденту
Оператор розкрою	Запуск системи Введення даних Визначення способів розкрою Розрахунок оптимального плану розкрою
Начальник виробництва	Запуск системи Розрахунок оптимального плану розкрою
Системний програміст	Зміна списку користувачів і інших змін у системі.

Ось декілька ключових аспектів використання діаграм прецедентів:

- визначення функціональності системи. Діаграми прецедентів дозволяють ідентифікувати ключові функції системи та їхні взаємодії з користувачами. Кожен прецедент представляє конкретну функцію або операцію, яку система може виконувати;

- опис сценаріїв взаємодії. Діаграми прецедентів надають можливість деталізованого опису сценаріїв взаємодії між користувачами та системою. Кожен прецедент може мати свій власний набір кроків чи дій, що розкриває, як система реагує на конкретний запит чи подію;

- визначення ролей користувачів. Діаграми прецедентів дозволяють ідентифікувати ролі різних користувачів системи та їх участь у викликах прецедентів. Це сприяє уточненню вимог щодо прав доступу та функціональності для різних користувачів;

- сприяння комунікації з зацікавленими сторонами. Діаграми прецедентів можуть служити ефективним інструментом комунікації між розробниками, клієнтами та іншими зацікавленими сторонами. Ці діаграми

допомагають візуалізувати функціональність та взаємодію системи на високому рівні абстракції;

- підтримка аналізу вимог. Діаграми прецедентів можуть бути використані для аналізу та формалізації вимог до системи. Вони допомагають зрозуміти, як система буде використовуватися та які функції їй потрібно виконувати;

- база для дизайну системи. Діаграми прецедентів можуть слугувати основою для подальшого проектування системи, визначаючи ключові компоненти, їхні взаємозв'язки та взаємодію.

Загалом, діаграми прецедентів є потужним інструментом для визначення та моделювання взаємодії системи з її користувачами та іншими системами. Вони сприяють розумінню вимог та допомагають розробникам створювати ефективний та функціональний дизайн системи, що відповідає потребам користувачів та бізнес-вимогам. Крім того, вони сприяють уникненню непорозумінь та невірного сприйняття вимог між розробниками та клієнтами, що може призвести до більш успішного впровадження проекту.

Додатково, діаграми прецедентів є ефективним інструментом для визначення ключових сценаріїв взаємодії, враховуючи основні взаємодії між акторами та системою. Вони дозволяють легко визначити функціональні обов'язки кожного актора та визначити, як вони взаємодіють з системою в конкретних ситуаціях. Це сприяє точнішому визначенню потреб користувачів і покращує здатність системи задовольняти ці потреби на різних етапах роботи. Також, використання діаграм прецедентів сприяє легкому виявленню можливих вдосконалень та оптимізацій у взаємодії між різними компонентами системи.

На підставі наведеного в таблиці 2.3 переліку користувачів системи й варіантів використання будується діаграма прецедентів (рисунок 2.2).

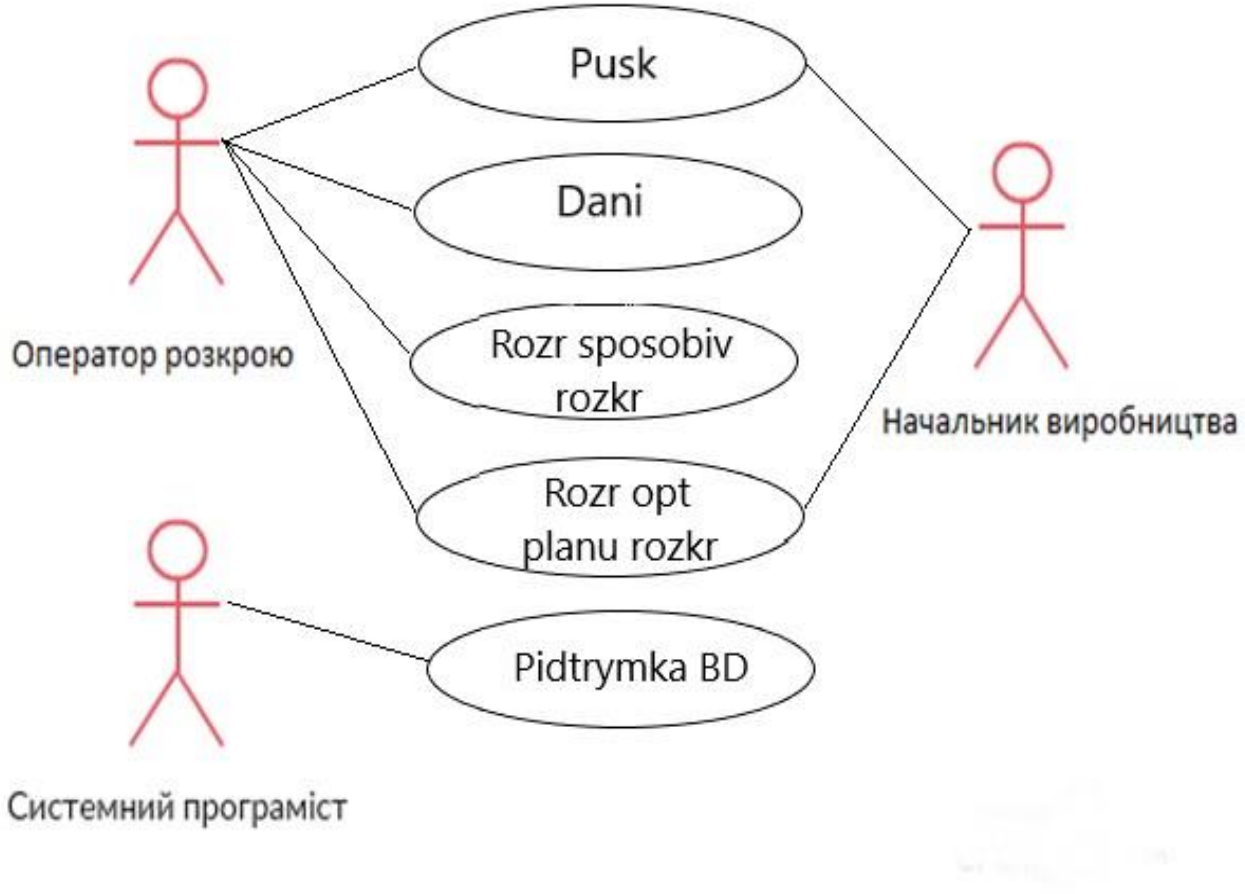


Рис. 2.2. Діаграма варіантів використання системи «Лінійний розкрій»

При проектуванні системи «Лінійний розкрій» створюється концептуальна модель – теоретичний концепт, що визначає будову системи, характеристики її компонентів та взаємозв'язки причинно-наслідкового характеру. Концептуальна модель експериментальної системи зображена на рисунку 2.3.

Функціонування системи здійснюється на персональному комп’ютері, встановленому на робочому місці фахівця.

Декомпозиція процесу оптимізації розкрою наведена на рисунку 2.4.

Рис. 2.3. Концептуальна модель процесу розрахунку оптимального розкрою

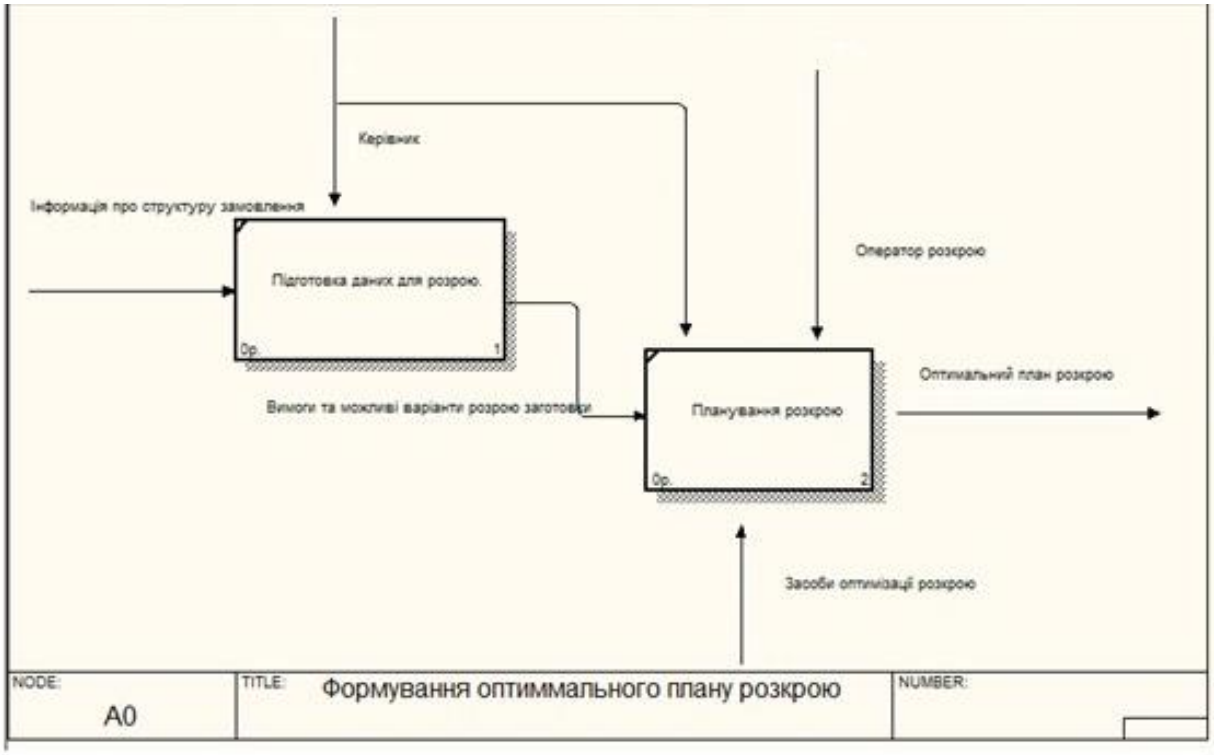


Рис. 2.4. Декомпозиція процесу розрахунку оптимального розкрою

Для візуалізації та моделювання послідовності дій в системі було побудовано діаграму діяльності (рис. 2.5). Діаграми діяльності (Activity Diagrams) є важливим інструментом в інженерії програмного забезпечення та моделюванні систем. Вони дозволяють візуалізувати послідовність операцій, дій та взаємодій між об'єктами в системі.

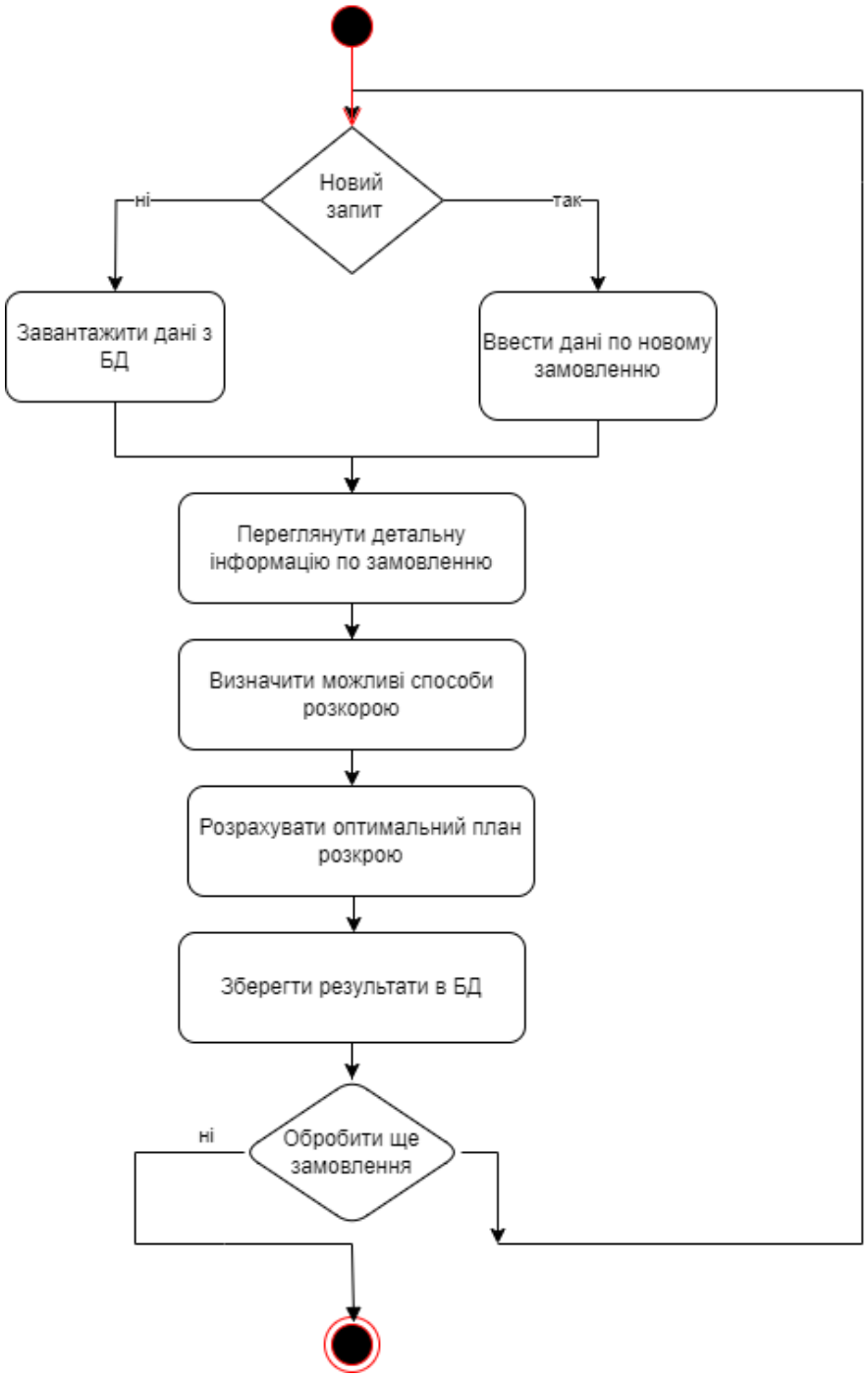


Рис. 2.5. Діаграма діяльності проєктованої інформаційної системи

Діаграми діяльності дозволяють вам точно визначити порядок виконання різних дій у системі. Це особливо корисно для розуміння та моделювання бізнес-процесів чи алгоритмів. Вони дозволяють вам відобразити, як об'єкти взаємодіють між собою в рамках конкретного процесу чи функції. Це допомагає уникнути непорозумінь та забезпечує консенсус між розробниками, аналітиками та клієнтами. Діаграми діяльності часто використовуються для моделювання бізнес-процесів, що дозволяє команді краще розуміти, як функції системи взаємодіють в бізнес-середовищі. Вони допомагають визначити, як система повинна вести себе в конкретних умовах, що сприяє створенню більш повного та зрозумілого опису функціональності. При використанні діаграм діяльності можна виявити можливі проблеми або архітектурні недоліки ще на етапі проектування, що полегшує виправлення помилок та внесення виправлень.

Отже, побудова діаграм діяльності надає можливість чітко визначити порядок виконання операцій та взаємодій між об'єктами. Це допомагає не лише уточнити бізнес-процеси та алгоритми, але і визначити, як об'єкти взаємодіють під час конкретного процесу. Діаграми діяльності також використовуються для моделювання бізнес-процесів, опису бізнес-логіки та визначення поведінки системи в конкретних сценаріях. При їх використанні можна виявити можливі проблеми та недоліки на ранніх етапах розробки, сприяючи покращенню функціональності та уникненню потенційних проблем.

Отже, у цьому розділі визначено ролі інформаційної системи, і з метою зрозумілості функціонування системи було створено діаграму варіантів використання.

2.5. Проєктування бази даних інформаційної системи

Коректність структури бази даних забезпечується за допомогою різноманітних механізмів і правил, які дотримуються під час проєктування,

реалізації та обслуговування бази даних. Основні аспекти, які гарантують коректність структури бази даних, включають:

- нормалізація даних – забезпечує уникнення аномалій даних та дублювання інформації, допомагає розподілити дані між різними таблицями так, щоб кожна з них зберігала інформацію про конкретний аспект домену;

- первинні та зовнішні ключі – використовуються для встановлення зв'язків між таблицями та забезпечення їх консистентності, захищають від появи записів з некоректними або відсутніми посиланнями;

- унікальність даних – забезпечуються визначенням унікальних обмежень для певних полів або комбінацій полів, інші обмеження, такі як перевірка, допомагають уникнути некоректних значень;

- транзакції – застосування концепції транзакцій дозволяє виконувати групу операцій як єдиний атомарний блок. Якщо операції транзакції виконані повністю, база даних перебуває в консистентному стані; в іншому випадку, виконані зміни скасовуються;

- валідація даних – механізми валідації, такі як обмеження, допомагають перевіряти, чи задовольняють дані певним критеріям перед їх додаванням чи зміною в базі даних, використовуються тригери, обмеження та перевірки для впровадження цих правил;

- автоматизована генерація коду та документації – використання CASE-засобів (Computer-Aided Software Engineering) для створення коду бази даних та автоматичної генерації документації. Зменшує ймовірність помилок та сприяє однорідності структури.

Ці підходи та механізми спільно забезпечують коректність структури бази даних та допомагають уникнути проблем, пов'язаних з некоректною архітектурою чи вмістом бази даних.

Нормалізація даних є процесом організації реляційної бази даних для зменшення дублювання даних та забезпечення узгодженості, цілісності та оптимальної ефективності в межах бази даних. Вона базується на теорії

реляційних баз даних і введена Едгаром Коддом [40]. Основні принципи та етапи нормалізації даних включають:

- перша нормальна форма (1НФ);
- друга нормальна форма (2НФ);
- третя нормальна форма (3НФ);
- звичайна форма Бойса-Кодда (BCNF або 3,5NF);
- четверта нормальна форма (4НФ);
- п'ята нормальна форма (5НФ);
- шоста нормальна форма (6НФ).

Необхідність нормалізації визначається конкретними вимогами та характером даних в конкретній інформаційній системі. В деяких випадках повна нормалізація може призвести до втрати деякої продуктивності, тому процес нормалізації повинен бути ретельно збалансований враховуючи конкретні потреби та умови використання бази даних, тому на практиці часто використовуються перші три форми нормалізації (1НФ, 2НФ та 3НФ) [40].

Для виконання нормалізації даних визначено наступні сутності:

- клієнти;
- замовлення;
- деталі;
- план деталей;
- рулони;
- результати.

Для першої нормальної форми було усунуто повторювані групи даних, і введений первинний ключ для гарантії унікальності кожного рядка.

Для другої нормальної форми була врахована залежність кожного атрибута сутності від усього ключа.

У процесі приведення до третьої нормальної форми перевірено відсутність неключових полів, які залежать від інших неключових полів, та наявність похідних полів.

Зазначено атрибути до кожної сутності, встановлено ключові поля, і при створенні таблиць вказано властивості кожного поля (рис. 2.6-2.11).

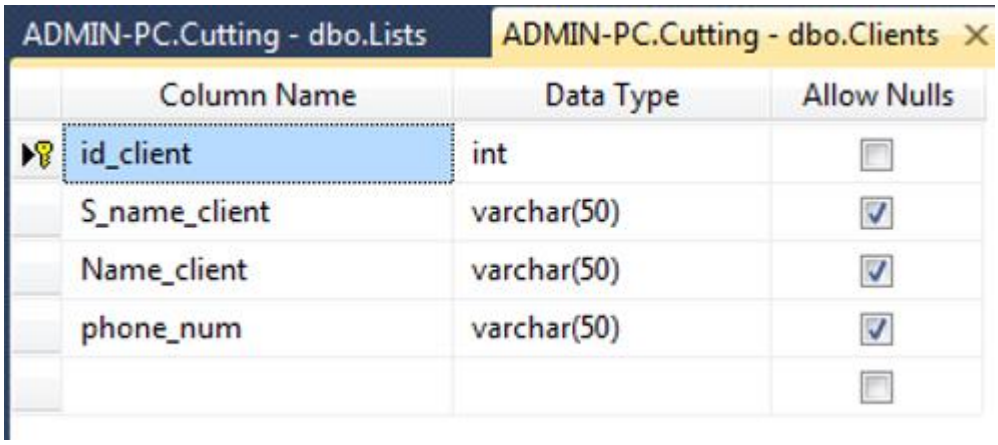
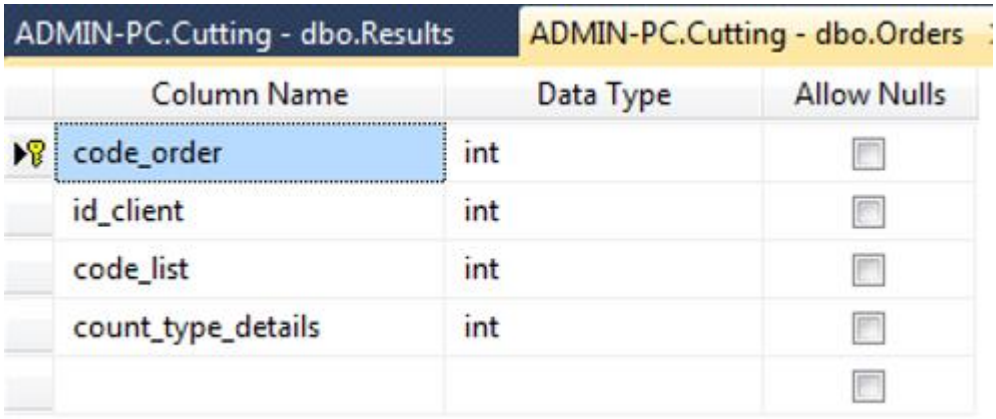
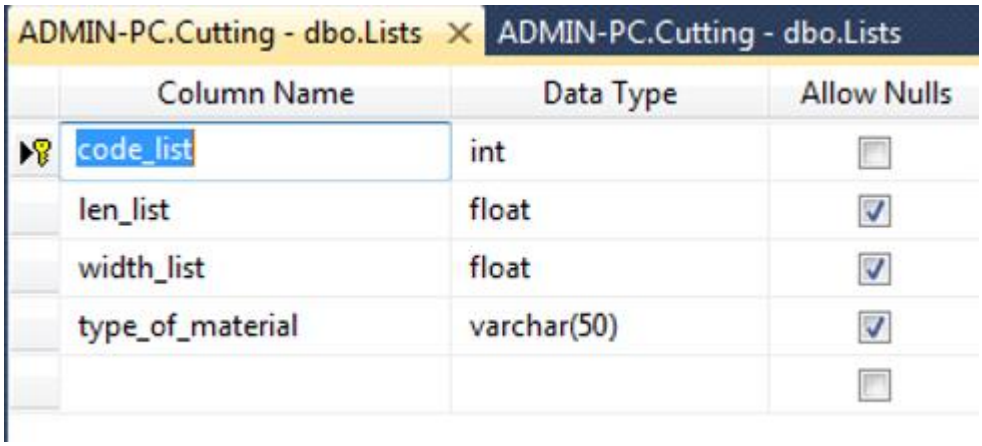


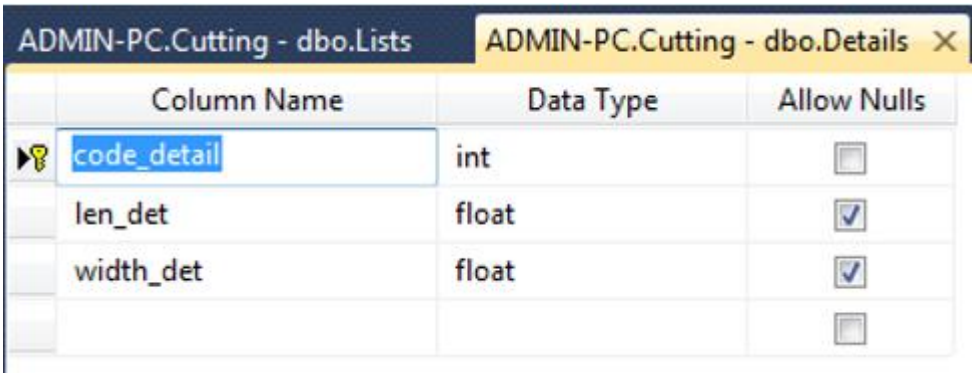
Рис. 2.6. Конструктор таблиці Clients



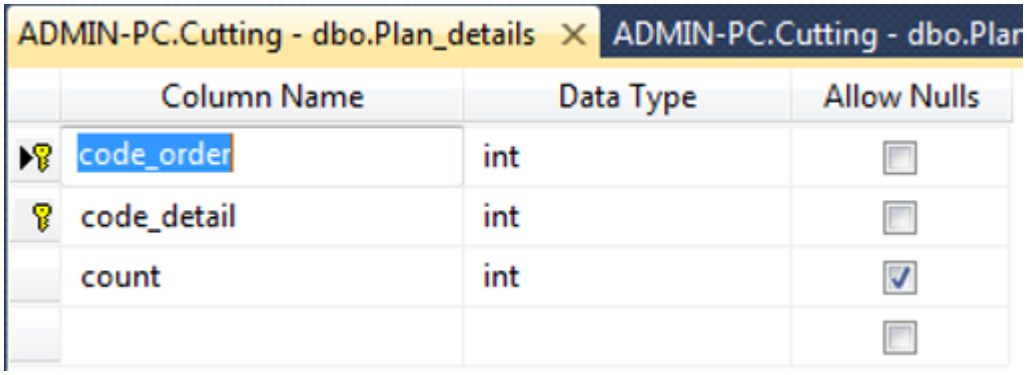
2.7. Конструктор таблиці Orders



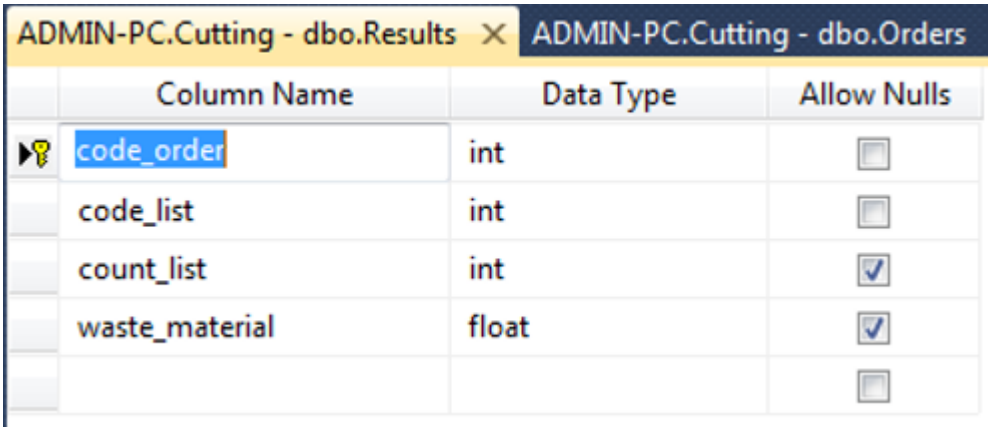
2.8. Конструктор таблиці Lists



2.9. Конструктор таблиці Details



2.10. Конструктор таблиці Orders



2.11. Конструктор таблиці Results

На рисунку 2.12 представлена структура нормалізованої бази даних з вказівкою первинних ключів та зв'язків між таблицями. Діаграма включає в себе взаємозв'язки "один-до-одного" і "один-до-багатьох".

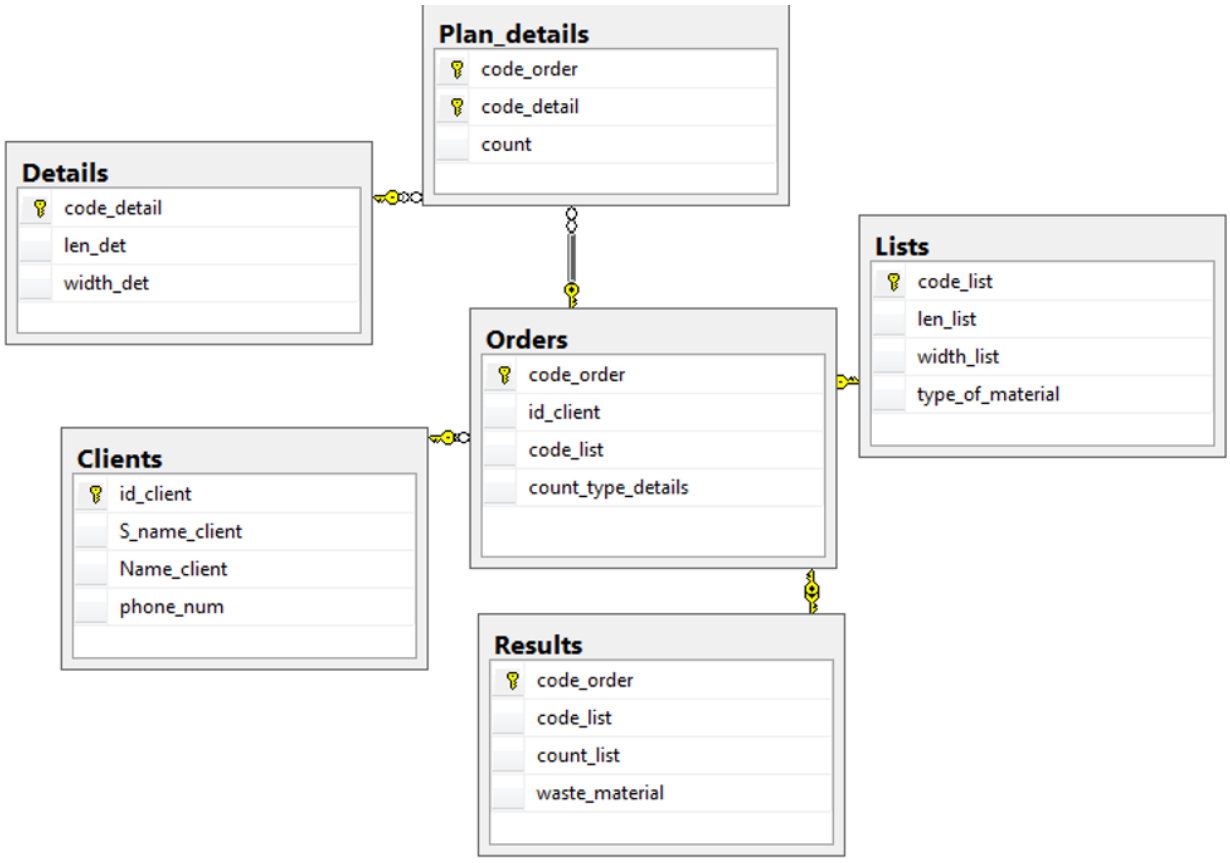


Рис. 2.12. Схема даних

Система дозволяє зберігати результати до бази даних, що забезпечує надійне зберігання і доступ до важливої інформації. Як зазначалося вище, база даних використовує реляційну модель, що сприяє ефективній організації даних і забезпечує легкий доступ до потрібної інформації.

Крім того, система забезпечує можливість проводити аналіз та вибірковий доступ до даних, що сприяє швидкій обробці і використанню інформації для подальших розрахунків та аналізу. Результати можуть бути використані для генерації звітів, статистичного аналізу та прийняття управлінських рішень.

Крім того, система автоматично стежить за інтегритетом даних і забезпечує їх консистентність. Це гарантує, що інформація, що зберігається в базі даних, є достовірною та актуальною на будь-який момент часу.

Загалом, система здатна не лише ефективно зберігати результати, але й в подальшому надавати користувачам потужні інструменти для управління та використання цих даних у найбільш продуктивний спосіб.

Отже, в даному підрозділі при створенні структури бази даних було застосовано процедуру нормалізації, що включає в себе видалення зайвих даних та визначення функціональних залежностей.

2.6. Розробка інтерфейсу користувача системи

Інтерфейс користувача був створений за допомогою платформи WS Forms (WinForms) - це технологія для розробки графічних інтерфейсів користувача (GUI) для десктопних застосунків під операційною системою Windows. Ця технологія була представлена Microsoft і є частиною .NET Framework.

Windows Forms надає широкий набір вбудованих елементів керування, таких як кнопки, тексти, вікна, списки, таблиці і багато інших. Ці елементи дозволяють створювати різноманітні та інтерактивні інтерфейси. Розробники можуть призначати обробники подій для реагування на різні події, такі як натискання кнопки, зміна значення текстового поля тощо. Windows Forms використовує об'єктно-орієнтовану модель програмування, де різні елементи і дії взаємодіють як об'єкти, що спрощує структуру коду та робить його більш зрозумілим і обслуговуваним. Розробники можуть використовувати різні стратегії розташування елементів на формі, такі як таблиці, які дозволяють ефективно організовувати вигляд форми. Windows Forms дозволяє працювати з графікою, малювати фігури, зображення та використовувати графічні ефекти. Розробники можуть взаємодіяти з подіями мережі, такими як натискання клавіш, переміщення миші, що дозволяє створювати додатки, які реагують на користувацькі взаємодії. Windows Forms може використовуватися з різними мовами програмування, такими як C#, VB.NET, іншими .NET-сумісними мовами.

Отже, Windows Forms був одним із попередніх підходів до створення десктопних додатків на платформі .NET, і, хоча зараз існують інші технології, такі як WPF та UWP, Windows Forms залишається популярним вибором для певних сценаріїв розробки.

Функціонування програми "Лінійний розкрій" розпочинається при її запуску, при цьому користувач автоматично переходить на основне вікно програми (рис. 2.13). Користувач натискає кнопку «Заповнити дані», де зчитується вхідна інформація з бази даних (рис. 2.14).

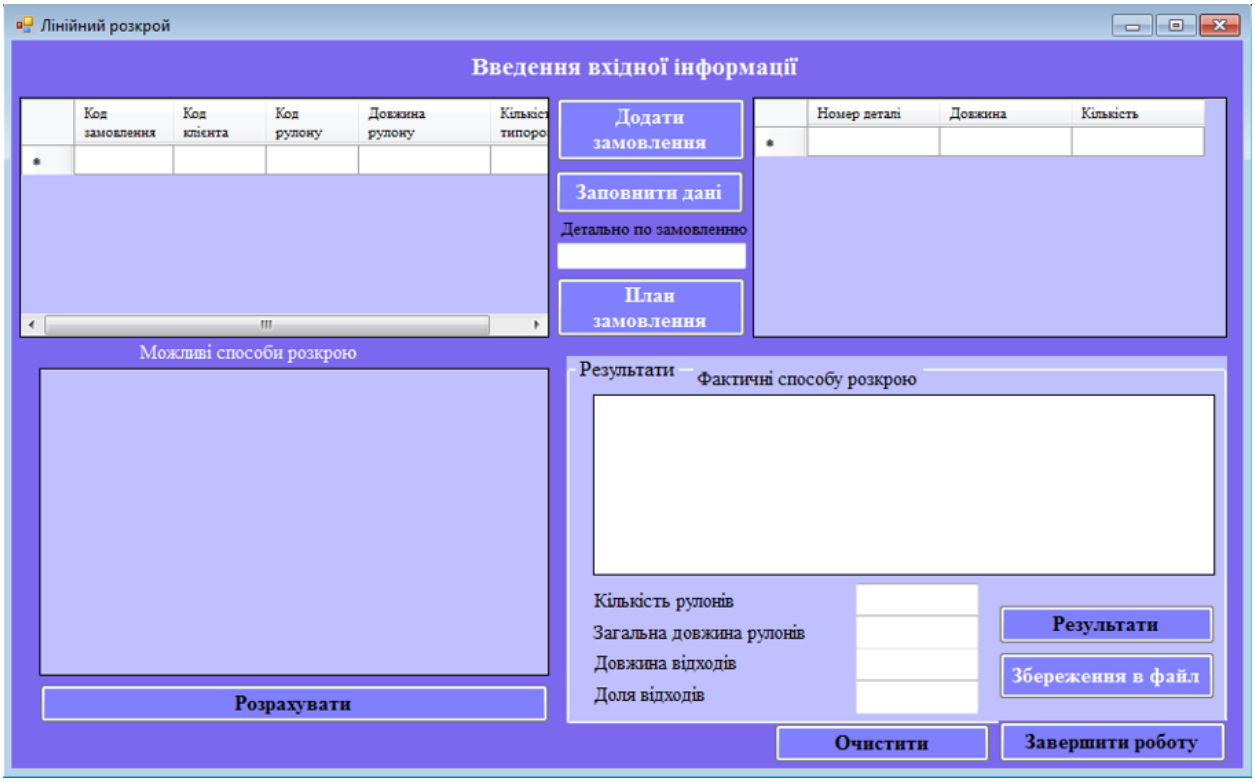
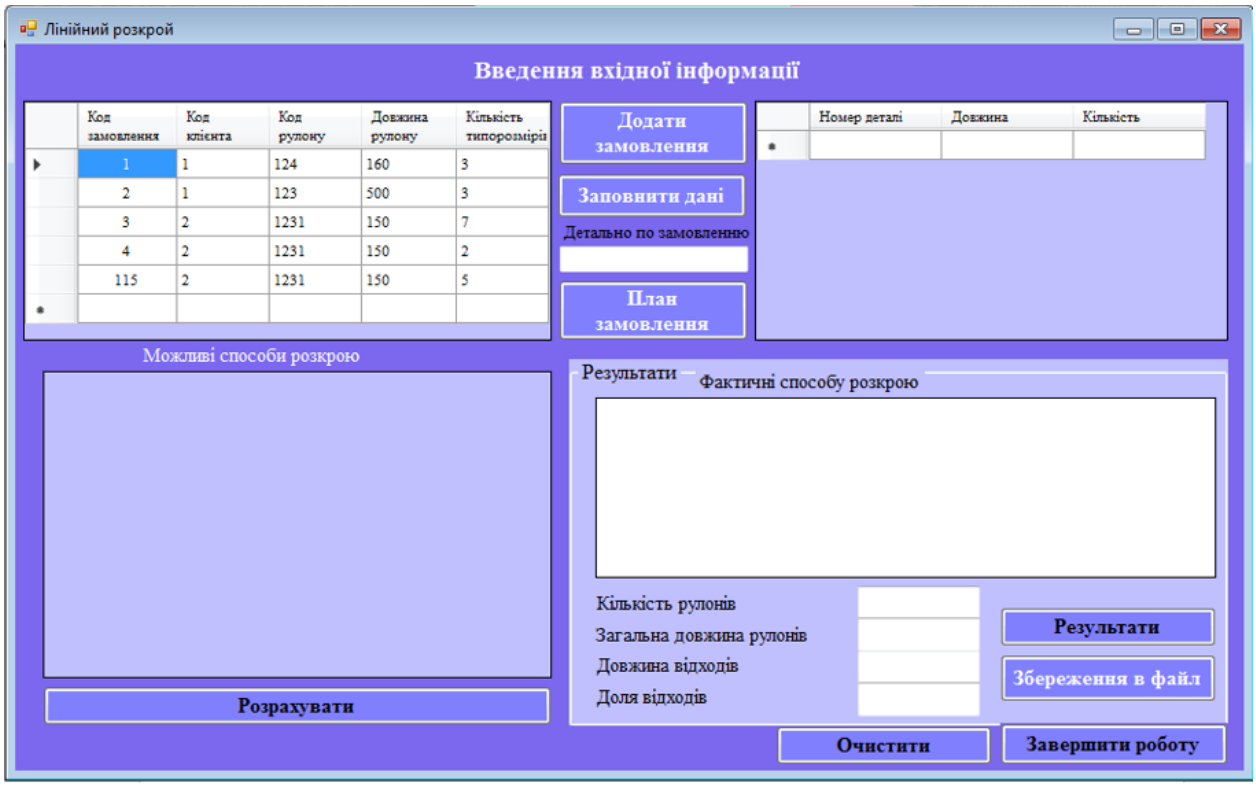
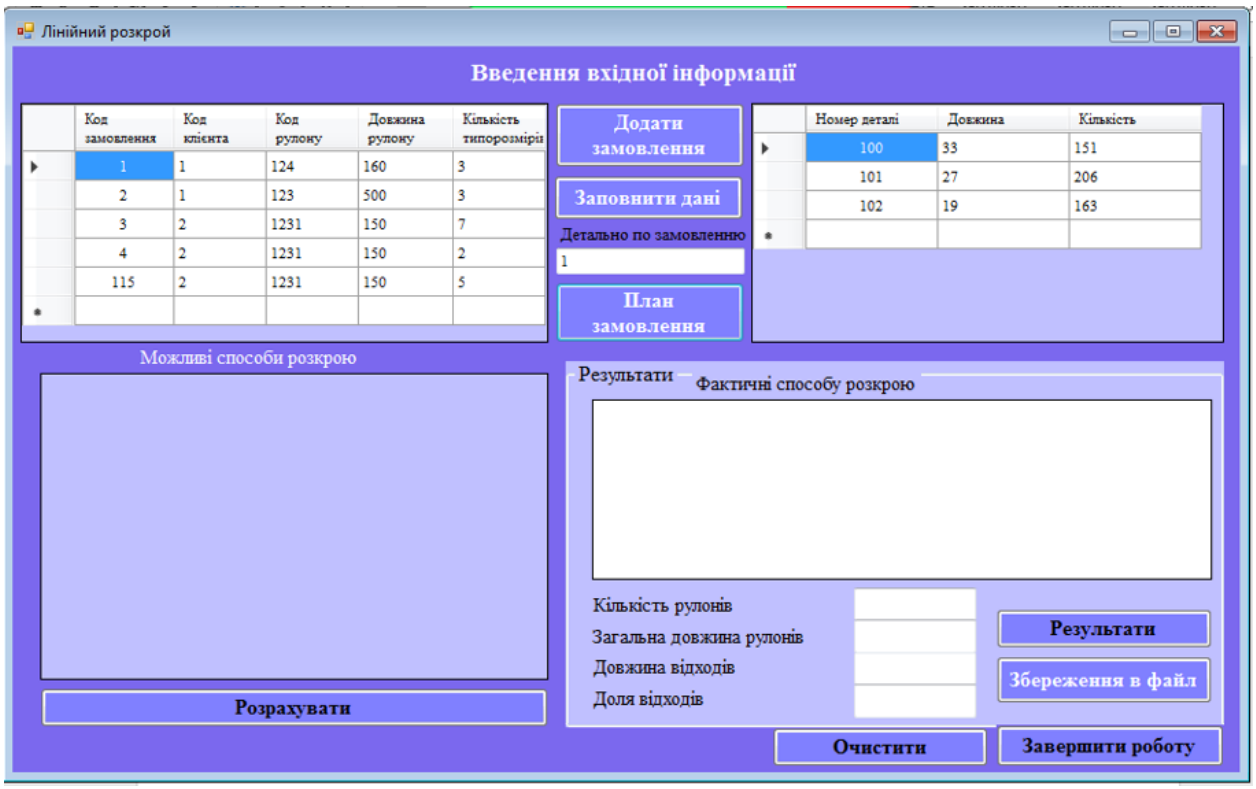


Рис. 2.13. Інтерфейс проектованої системи

Для відображення детальної інформації за певним замовленням необхідно ввести номер замовлення у відповідну строку – «Детально по замовленню» та натиснути кнопку «План замовлення». Система виводить інформацію з бази даних у таблицю з інформацією по номеру деталі, довжині та кількості (рис. 2.15).

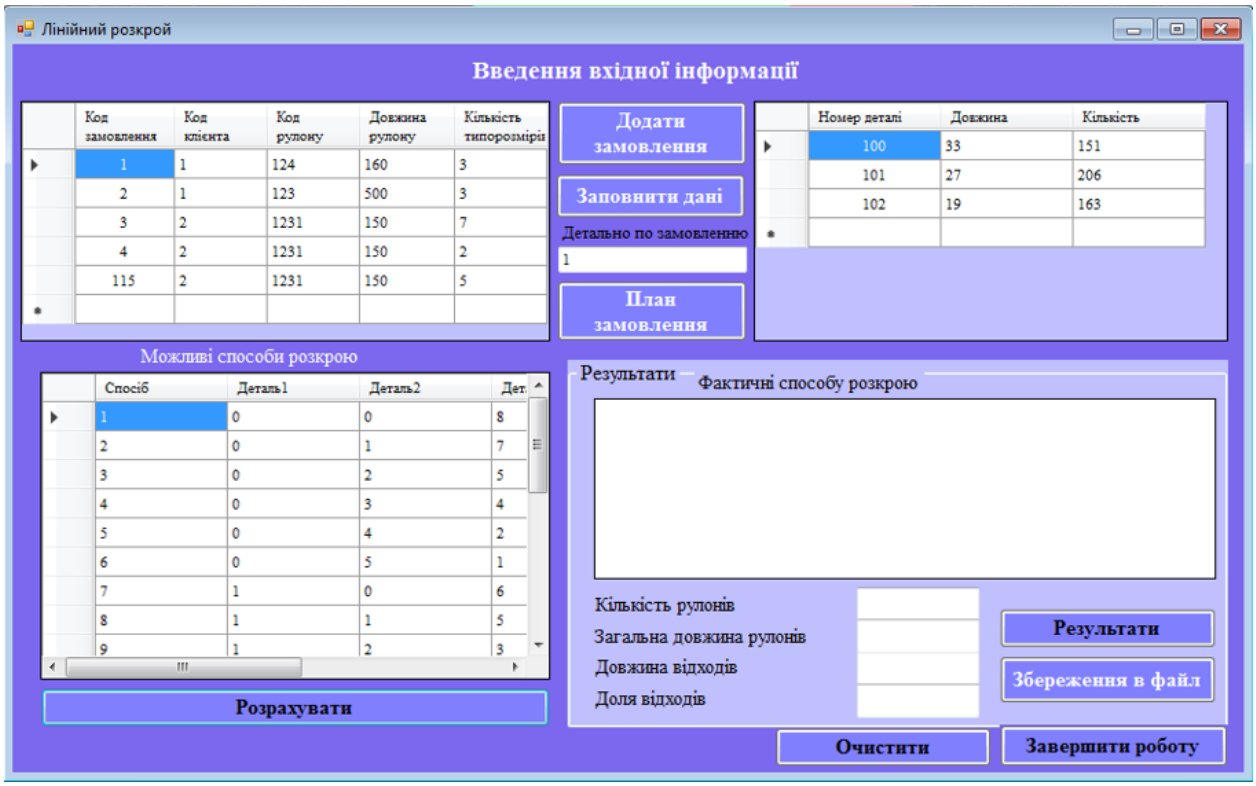


2.14. Вхідна інформація – кнопка «Заповнити дані»



2.15. Інформація по замовленню – кнопка «План замовлення»

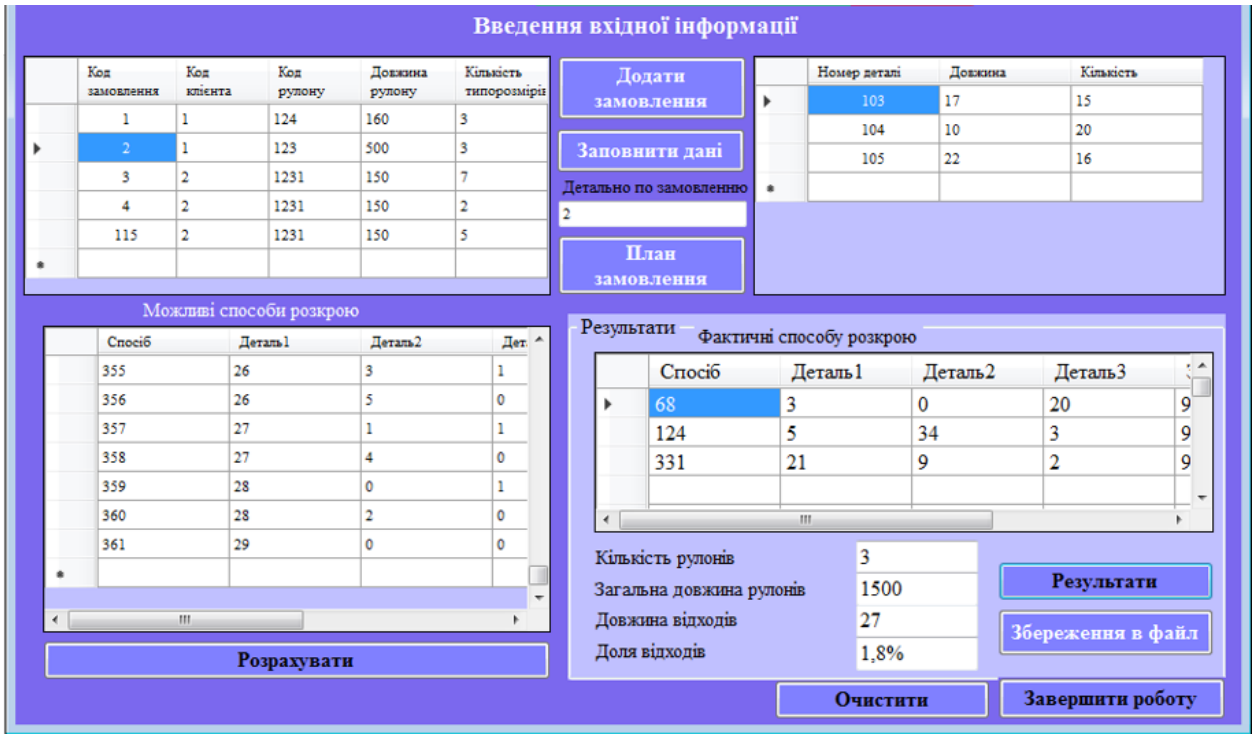
Для розрахунку можливих способів розкрою необхідно натиснути кнопку «Розрахувати», інформаційна система виконує розрахунок та відображає дані у відповідній таблиці (рис. 2.16).



2.16. Розрахунок можливих способів розкрою – кнопка «Розрахувати»

Далі виконується вибірка з можливих способів розкрою – натиснути кнопку «Результати» (рис. 2.17). У таблиці «Фактичні способи розкрою» відображається план розкрою за замовленням. Також будуть наводяться підсумкові дані за цим планом розкрою – сумарна кількість листів, загальна довжина листів, довжина та доля відходів.

Наприкінці роботи з замовленням можна зберегти інформацію до бази даних, очистити дані. Після завершення використання програми "Лінійний розкрій" користувач має можливість завершити роботу, натиснувши відповідну кнопку «Завершити роботу».



2.17. Фактичні способи розкрою – кнопка «Результати»

Для забезпечення ефективності, зручності та коректності використання інформаційної системи оптимізації розкрою матеріалу висуваються певні вимоги до інтерфейсу. Вони можуть включати:

- інтуїтивний інтерфейс для забезпечення простоти та легкості в користуванні для всіх категорій користувачів, а також інтуїтивне розташування елементів інтерфейсу;

- можливість введення даних, наприклад, введення вхідних даних, таких як розміри матеріалу та деталей, кількість деталей тощо;

- можливість редагування параметрів, наприклад, можливості зміни вхідних даних та параметрів оптимізації для проведення різних сценаріїв;

- зручність аналізу – надання засобів для аналізу результатів оптимізації та порівняння різних варіантів розкладу;

- повідомлення та статуси – система повинна надавати зрозумілі повідомлення та статуси під час роботи алгоритму оптимізації;

- сумісність та доступність – забезпечення сумісності інтерфейсу з різними пристроями та доступність для користувачів з обмеженими можливостями.

Отже, виходячи з вищезазначеним, можна зробити висновок, що дизайн інтерфейсу системи вирізняється своєю простотою та легкістю в використанні для всіх користувачів, забезпечуючи зручність і ефективність взаємодії з програмою.

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Елементи програмної реалізації проєктованої системи

Інформаційну систему «Лінійний розкрій» для розрахунку оптимального розкрою матеріалу було реалізовано на мові програмування C# у середовищі Microsoft Visual Studio.

Як вже зазначалося (підрозділ 2.6), мова програмування C# є потужною, сучасною та об'єктно-орієнтованою мовою програмування, що була розроблена для платформи .NET. Вона надає широкі можливості для створення різноманітних програм, включаючи десктопні, веб-додатки та служби. C# є основною мовою для .NET, що дозволяє легко взаємодіяти з іншими частинами фреймворку [28].

.NET є компромісним фреймворком, що надає широкі можливості для розробки, виконання та взаємодії програм. Він підтримує багатоплатформенність, що робить його ідеальним вибором для розробки різноманітних додатків. Є багатофункціональним - має велику кількість бібліотек і інструментів, що спрощують розробку та поліпшують продуктивність.

Для ІС було розроблено базу даних за допомогою системи управління базами даних (СУБД) MS SQL Server, яка володіє потужними можливостями управління даними та є надійним інструментом для зберігання та обробки інформації. Забезпечує підтримку транзакцій, що робить його ефективним для додатків, де важлива консистентність даних, добре інтегрується з C# та .NET, що дозволяє легко взаємодіяти з базою даних з додатків.

Об'єднання C#, .NET та SQL Server створює потужну інфраструктуру для розробки та реалізації проєктованої системи. C# надає ефективний засіб програмування, .NET робить розробку більш зручною та багатофункціональною, а SQL Server гарантує надійне зберігання та

управління даними. Цей стек технологій ідеально підходить для створення високоефективних, надійних та розширюваних програмних рішень.

Фрагменти програмного коду розробленої інформаційної системи наведено на рисунках 3.1-3.4.

```
8       public class Simplex
9       {
10          //source - симплекс-таблиця без базисних змінних
11          double[,] table; //симплекс-таблиця
12
13          int m, n;
14
15          List<int> basis; //список базисних змінних
16
            ссылка: 1
17          public Simplex(double[,] source)
18          {
19              m = source.GetLength(0);
20              n = source.GetLength(1);
21              table = new double[m, n + m - 1];
22              basis = new List<int>();
23
24              for (int i = 0; i < m; i++)
25              {
26                  for (int j = 0; j < table.GetLength(1); j++)
27                  {
28                      if (j < n)
29                          table[i, j] = source[i, j];
30                      else
31                          table[i, j] = 0;
32                  }
33                  //коефіцієнт 1 перед базисною змінною в рядку
34                  if ((n + i) < table.GetLength(1))
35                  {
36                      table[i, n + i] = 1;
37                      basis.Add(n + i);
38                  }
39              }
40              n = table.GetLength(1);
41          }
42
43          //result - значення X
            ссылка: 1
44          public double[,] Calculate(double[] result)
45          {
```

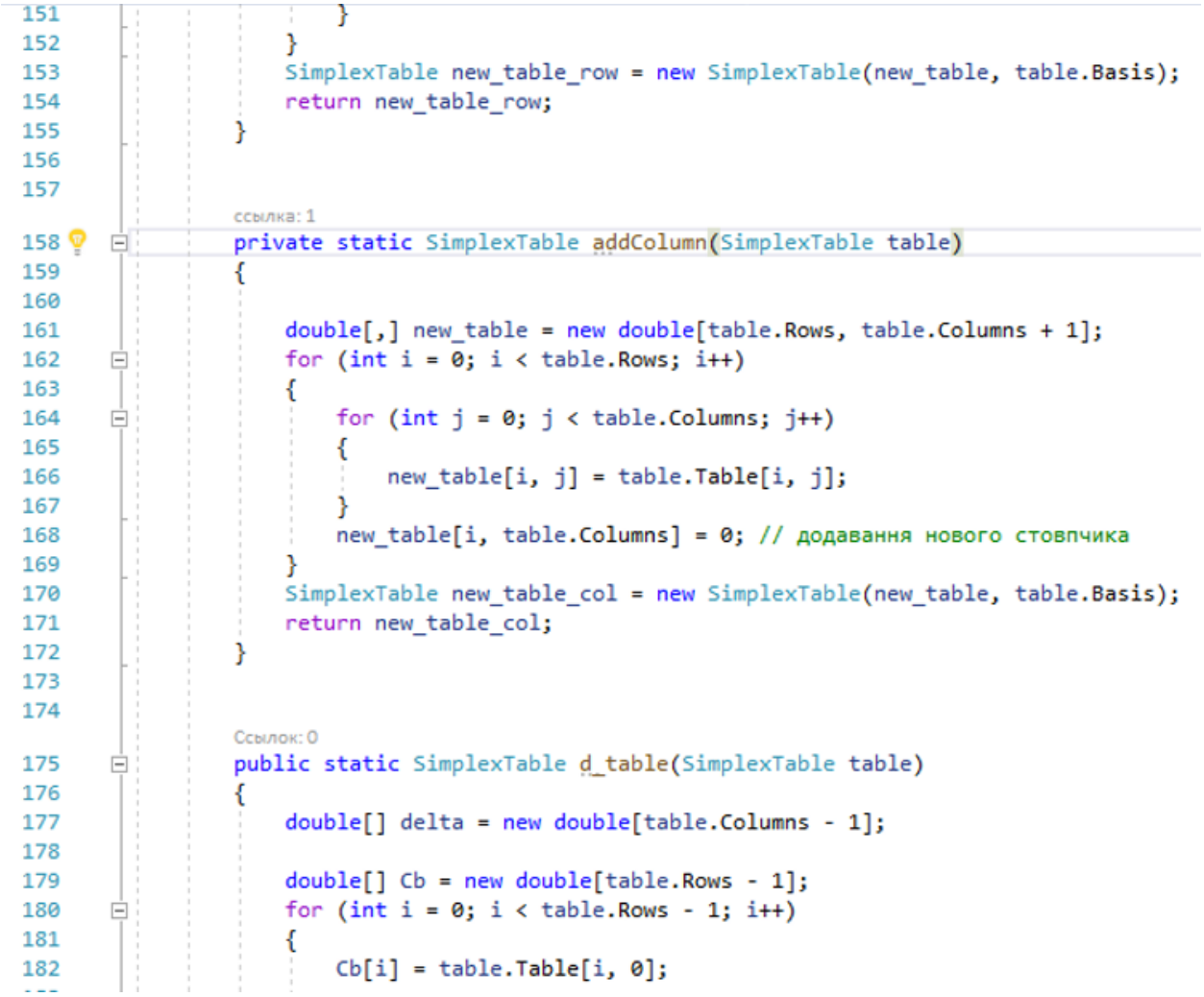
Рис. 3.1. Фрагмент коду класу Simplex

```
ссылка: 1
99      private int findMainCol()
100     {
101         int mainCol = 1;
102
103         for (int j = 2; j < n; j++)
104             if (table[m - 1, j] < table[m - 1, mainCol]) // знаходження головного стовпчика
105                 mainCol = j;
106
107         return mainCol;
108     }
109
ссылка: 1
110     private int findMainRow(int mainCol)
111     {
112         int mainRow = 0;
113
114         for (int i = 0; i < m - 1; i++)
115             if (table[i, mainCol] > 0)
116             {
117                 mainRow = i;
118                 break;
119             }
120
121         for (int i = mainRow + 1; i < m - 1; i++)
122             if ((table[i, mainCol] > 0) && ((table[i, 0] / table[i, mainCol]) < (table[mainRow, 0]
123                 mainRow = i;
124
125         return mainRow;
126     }
127
128
129 }
130
```

Рис. 3.2. Функції знаходження головного рядка і головного стовпчика в межах реалізації класу Simplex

```
135
136     // додавання нового рядка
        ссылка: 1
137     private static SimplexTable addRow(SimplexTable table)
138     {
139
140         double[,] new_table = new double[table.Rows + 1, table.Columns];
141         table.Basis.Add(table.Columns);
142         for (int i = 0; i < table.Rows; i++)
143         {
144             for (int j = 0; j < table.Columns; j++)
145             {
146                 if (i < table.Rows - 1)
147                     new_table[i, j] = table.Table[i, j];
148                 new_table[table.Rows - 1, j] = 0; // додавання нового рядка
149                 new_table[table.Rows, j] = table.Table[table.Rows - 1, j];
150
151             }
152         }
153         SimplexTable new_table_row = new SimplexTable(new_table, table.Basis);
154         return new_table_row;
155     }
156
157
        ссылка: 1
158     private static SimplexTable addColumn(SimplexTable table)
159     {
160
161         double[,] new_table = new double[table.Rows, table.Columns + 1];
162         for (int i = 0; i < table.Rows; i++)
163         {
164             for (int j = 0; j < table.Columns; j++)
165             {
166                 new_table[i, j] = table.Table[i, j];
```

Рис. 3.3. Фрагмент коду класу GomoryMethod – функція додавання нового рядка



```
151                 }
152             }
153             SimplexTable new_table_row = new SimplexTable(new_table, table.Basis);
154             return new_table_row;
155         }
156
157
                ссылка: 1
158         private static SimplexTable addColumn(SimplexTable table)
159         {
160
161             double[,] new_table = new double[table.Rows, table.Columns + 1];
162             for (int i = 0; i < table.Rows; i++)
163             {
164                 for (int j = 0; j < table.Columns; j++)
165                 {
166                     new_table[i, j] = table.Table[i, j];
167                 }
168                 new_table[i, table.Columns] = 0; // додавання нового стовпчика
169             }
170             SimplexTable new_table_col = new SimplexTable(new_table, table.Basis);
171             return new_table_col;
172         }
173
174
                Ссылок: 0
175         public static SimplexTable d_table(SimplexTable table)
176         {
177             double[] delta = new double[table.Columns - 1];
178
179             double[] Cb = new double[table.Rows - 1];
180             for (int i = 0; i < table.Rows - 1; i++)
181             {
182                 Cb[i] = table.Table[i, 0];
```

Рис. 3.4. Фрагмент коду класу Gomory Method – функція додавання нового стовпчика

Отже, у даному підрозділі наведено елементи програмної реалізації розробленої системи.

3.2. Тестування інформаційної системи

Тестування інформаційної системи «Лінійний розкрій» щодо функціональності може включати різні аспекти та сценарії для перевірки правильності роботи системи.

Наведемо розширений опис тестування з урахуванням натискання кнопки "План замовлення" з різними вхідними даними.

Позитивне тестування:

- сценарій: введення правильного номера замовлення та натискання кнопки "План замовлення";

- очікуваний результат: система має вивести план замовлення для введеного номера без помилок.

Негативне тестування (введення нечислового значення, введення замовлення якого не існує):

- сценарій: введення нечислових значень (наприклад, тексту) як номера замовлення, введення замовлення якого не існує та натискання кнопки "План замовлення";

- очікуваний результат: система має вивести вікно повідомлення про необхідність введення коректного коду замовлення (рис. 3.5).

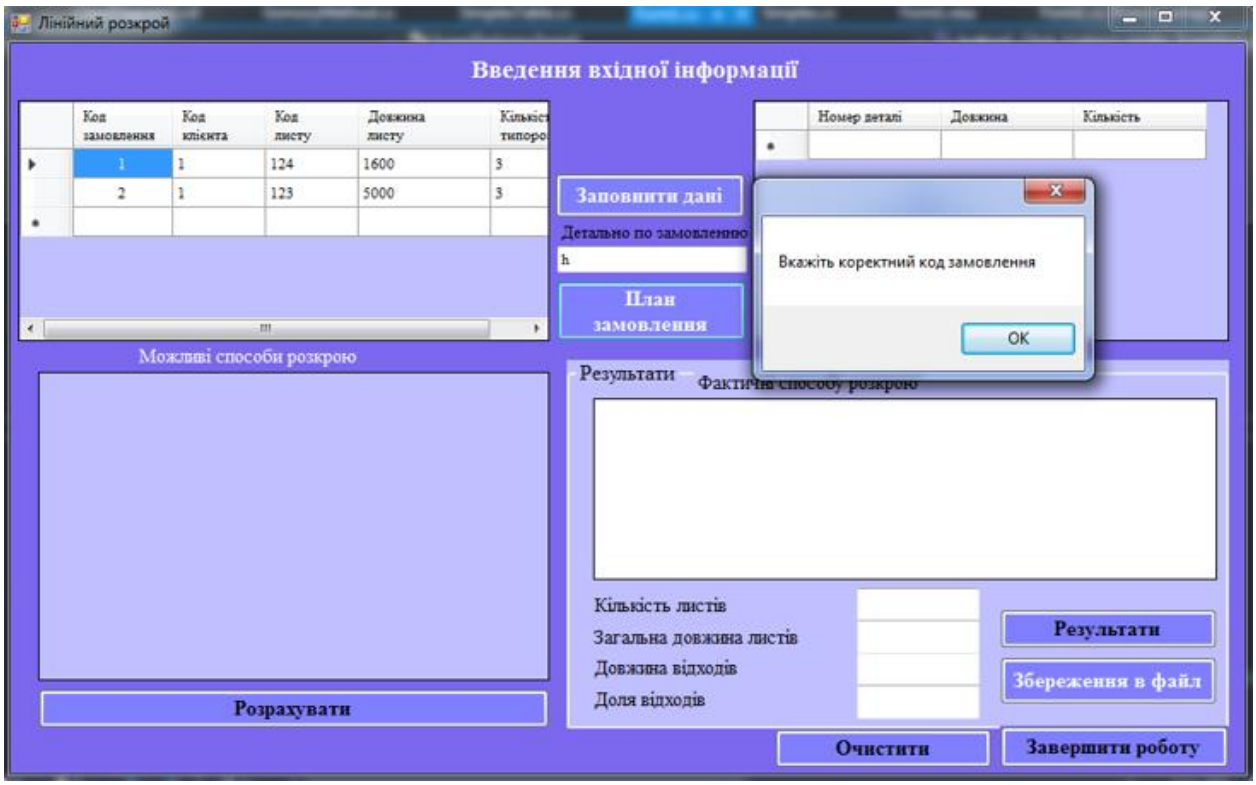


Рис. 3.5. Результат відповіді системи під час роботи за негативним сценарієм кнопки "План замовлення"

Негативне тестування (порожнє введення):

- сценарій: натискання кнопки "План замовлення" без введення номера замовлення;

Очікуваний результат: система має вивести вікно повідомлення про необхідність введення номера замовлення (рис. 3.6).

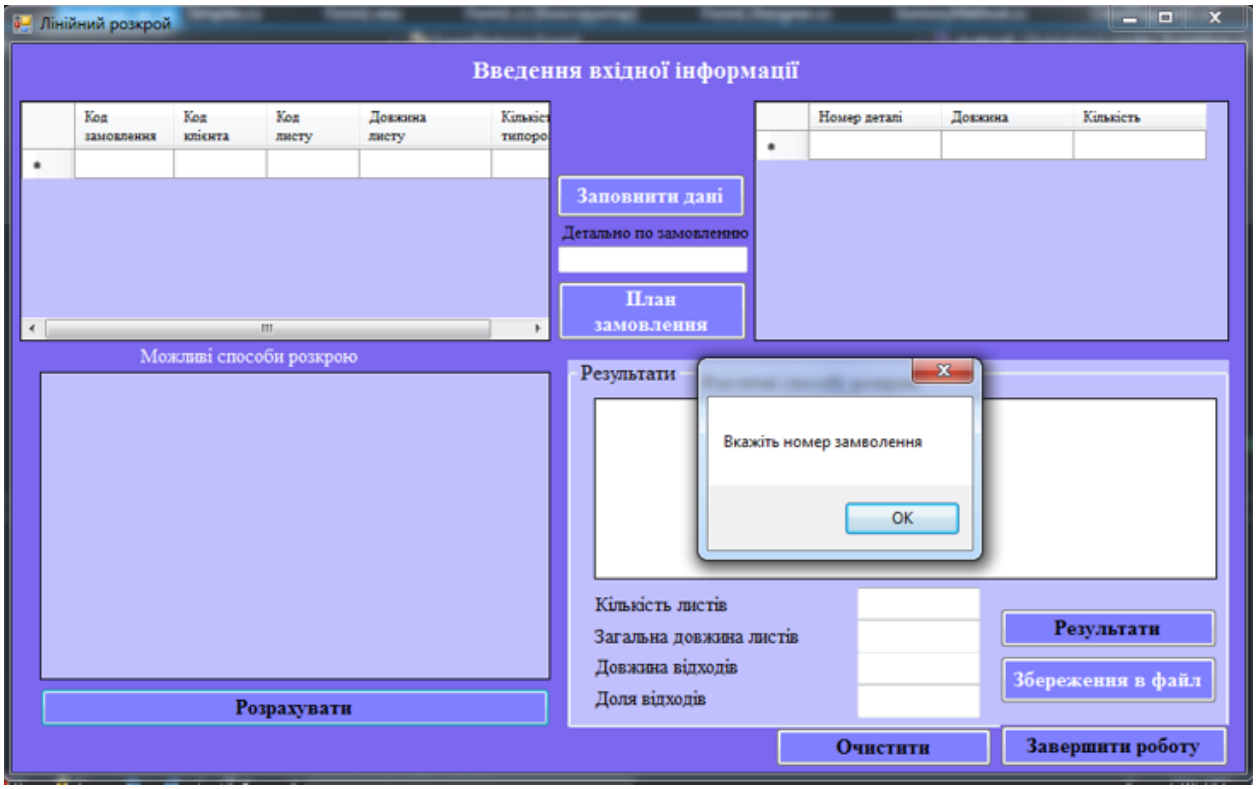


Рис. 3.6. Результат відповіді системи під час роботи за негативним сценарієм кнопки "План замовлення"

Наведемо розширений опис тестування з урахуванням натискання кнопки "Розрахувати" без введення коду замовлення з бази даних.

Позитивне тестування:

- сценарій: натискання кнопки "Розрахувати" за наявним планом замовлення;

- очікуваний результат: система має виконати обчислення та навести результати можливих способів розкрою.

Негативне тестування:

- сценарій: натискання кнопки "Розрахувати" без попереднього виведення плану замовлення;

- очікуваний результат: система має вивести вікно повідомлення про необхідність виконання запиту до бази даних щодо плану замовлення (рис. 3.7).

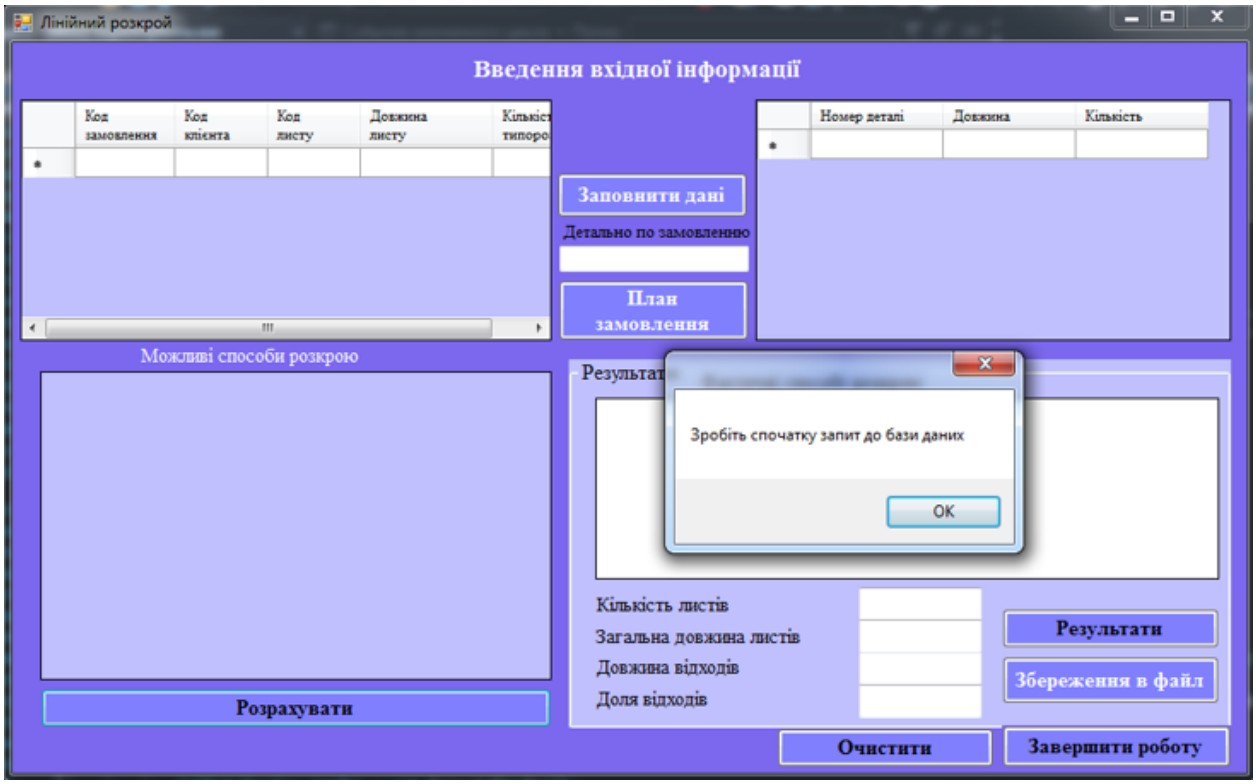


Рис. 3.7. Результат відповіді системи під час роботи за негативним сценарієм кнопки "Розрахувати"

Наведемо розширений опис тестування з урахуванням натискання кнопки "Результати" без визначення можливих способів розкрою.

Позитивне тестування:

- сценарій: натискання кнопки "Результати" з попереднім розрахунком можливих способів розкрою;

- очікуваний результат: система має відобразити фактичний план розкрою, а також навести підсумкові розрахунки за цим планом розкрою – сумарну кількість листів, загальну довжину листів, довжину та долю відходів.

Негативне тестування:

- сценарій: сценарій: натискання кнопки "Результати" без визначення можливих способів розкрою;

- очікуваний результат: система має вивести вікно повідомлення про необхідність обчислення можливих способів розкрою (рис. 3.8).

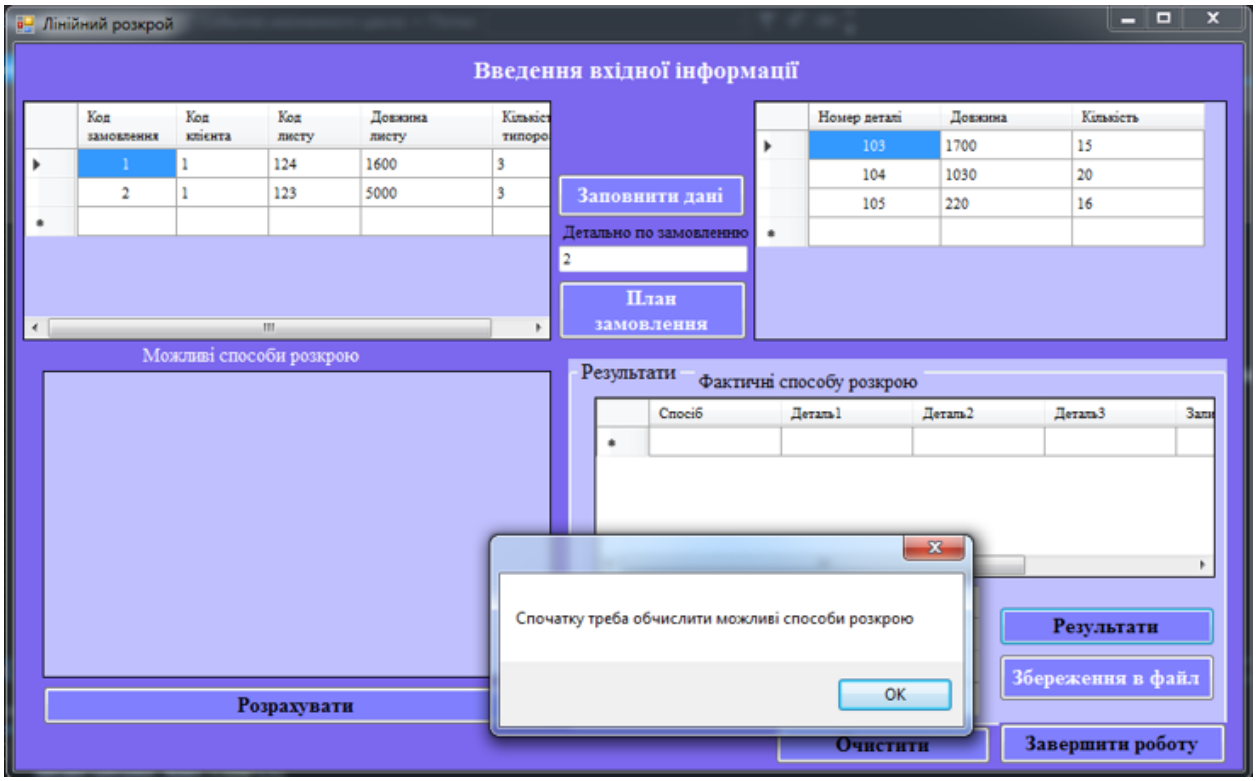


Рис. 3.8. Результат відповіді системи під час роботи за негативним сценарієм кнопки "Результати"

Наведемо розширений опис тестування з урахуванням натискання кнопки "Збереження даних" без отримання результатів розрахунку або повторним натисканням.

Позитивне тестування:

- сценарій: натискання кнопки "Збереження даних" з виконанням остаточних розрахунків;

- очікуваний результат: система зберігає результати розрахунків в до бази даних.

Негативне тестування (без отримання результатів розрахунку):

- сценарій: натискання кнопки "Збереження даних" без виконання остаточних розрахунків;

- очікуваний результат: система має вивести вікно повідомлення про необхідність виконання обчислення (рис. 3.9).

Негативне тестування (повторне натискання):

- сценарій: повторне натискання кнопки "Збереження даних";

- очікуваний результат: система має вивести вікно повідомлення про те, що результати для цього розкрою вже збережені (рис. 3.10).

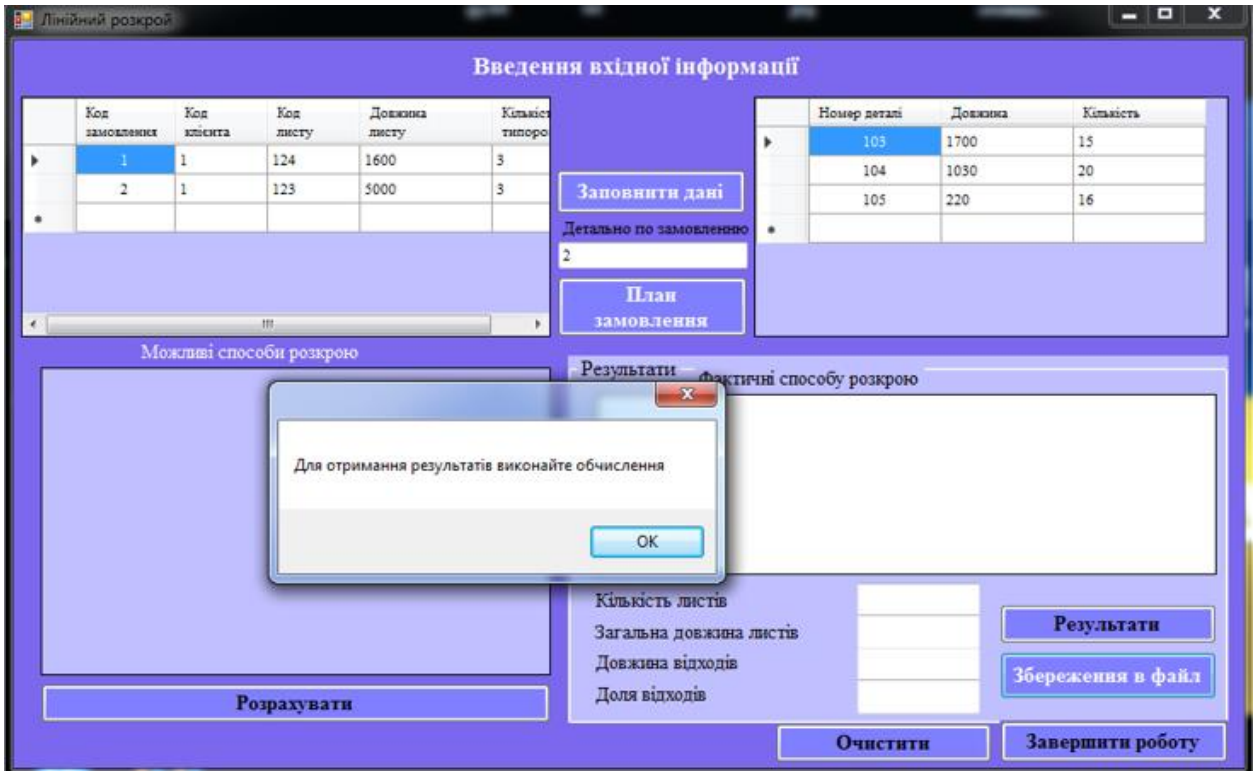


Рис. 3.9. Результат відповіді системи під час роботи за негативним сценарієм кнопки "Результати" (без отримання результатів розрахунку)

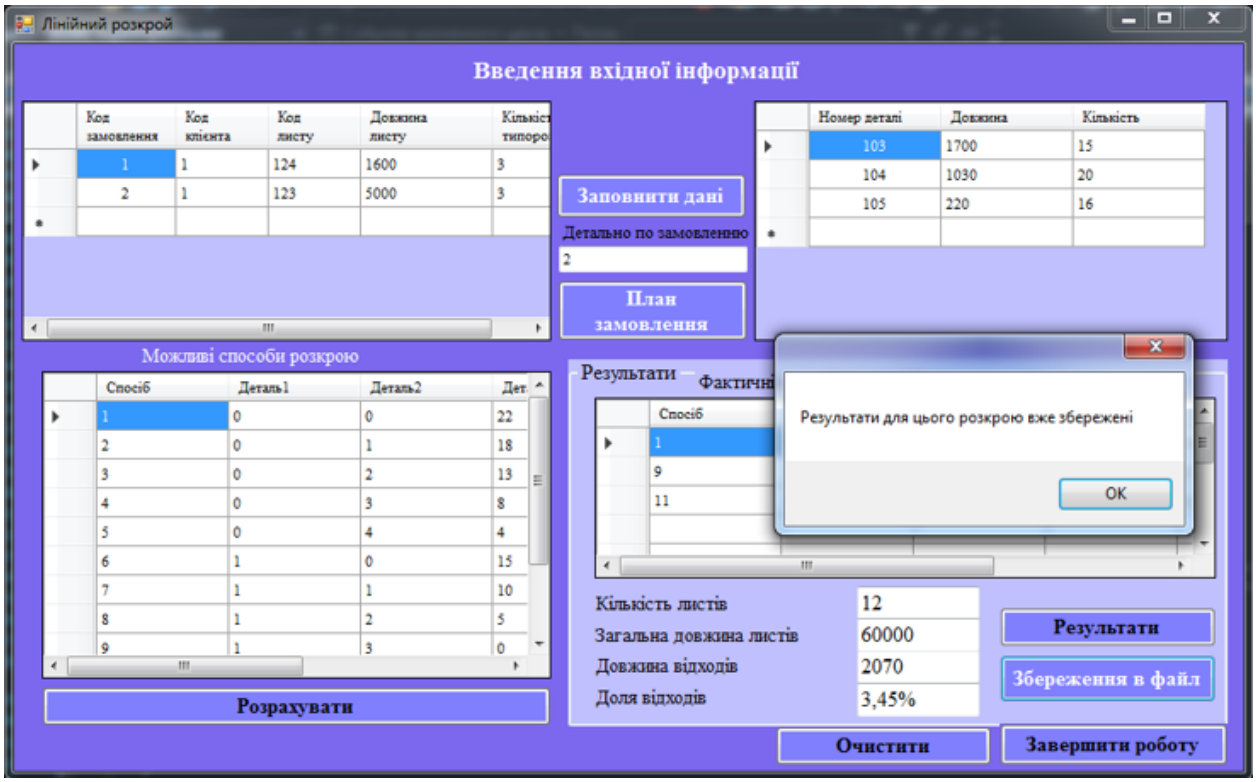


Рис. 3.10. Результат відповіді системи під час роботи за негативним сценарієм кнопки "Результати" (повторне збереження даних)

У даному підрозділі було наведено результати тестування інформаційної системи «Лінійний розкрій». Загальний висновок полягає в тому, що система виявилася стійкою до різноманітних вхідних даних та сценаріїв використання. Позитивні та негативні тестові сценарії дозволили виявити, що система правильно обробляє коректні введення, а також надійно виявляє та коректно відображає помилки.

Важливо відзначити, що інтерфейс користувача виявився зручним та інтуїтивно зрозумілим, що полегшує взаємодію користувача з системою. Успішне збереження стану та коректне повідомлення про результати дій свідчать про високу якість розробленої інформаційної системи. Такий підхід до тестування гарантує ефективність та надійність функціоналу інформаційної системи "Лінійний розкрій".

ВИСНОВКИ

В даній кваліфікаційній роботі магістра відповідно до поставленої мети було створено інформаційну систему на основі платформи .NET для керування ділянкою нарізання металу на смуги з оптимальним розкроєм рулону, що забезпечить ефективне використання матеріалів.

Для досягнення поставленої мети було проведено аналіз існуючих інформаційних систем оптимального розкрою, вивчено особливості та можливості, проаналізовано недоліки, вдосконалення та інновації.

Було проведено огляд існуючих методів і моделей для вирішення задачі оптимального розкрою заготовок та запропоновано ефективний та оптимальний алгоритм з використанням комбінаторних методів у поєднанні з симплекс-методом та методом Гоморі.

Представлено структуру інформаційної системи, описано її функціональну модель, розроблено базу даних, а також створено простий та зручний інтерфейс для користувача

Розроблено інформаційну систему керування ділянкою виготовлення заготівок з оптимальним розкроєм рулону на мові C# у середовищі Microsoft Visual Studio 2019 з використанням Microsoft .NET Framework 4.8. Розроблено базу даних в середовищі MS SQL Server.

Було проведено тестування програмного продукту за різними сценаріями, підтверджено коректність та ефективність роботи інформаційної системи.

Основні переваги інформаційної системи – простий і зручний інтерфейс, низька вартість, як самого продукту, так і його наступних можливих модифікацій. Перелік функцій може бути розширено, але і на даному етапі програмний продукт являє собою цілісну інформаційну систему, яка допоможе керівникам виробничих ділянок забезпечити більш ефективну роботу підприємства.

Висновки отриманих результатів даної роботи вказують на успішне вирішення поставлених завдань та досягнення основної мети. Розроблений алгоритм з оптимізації плану розкрою матеріалу виявився ефективним, забезпечуючи зменшення витрат та мінімізацію відходів на виробництві. Інформаційна система, побудована на основі сучасних методів оптимізації та інформаційних технологій, продемонструвала свою ефективність у практичному застосуванні. Отримані результати відкривають перспективи для подальших досліджень у сфері оптимального використання матеріальних ресурсів та автоматизації процесів на виробничих підприємствах. Таким чином, можна визначити, що мету досягнуто, а отримані результати мають позитивний внесок у відповідну наукову галузь.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іванець О.О. Концептуальні підходи до управління потенціалом розвитку підприємства. Вісник НТУ «ХПІ». Серія: Актуальні проблеми управління та фінансово-господарської діяльності підприємства. Харків : НТУ «ХПІ», 2013. № 52 (1025). С. 62–69.

2. Ващенко А.А. Організація виробництва в умовах мінливого внутрішнього та зовнішнього середовища функціонування машинобудівних підприємств. *Ефективна економіка*. 2015. № 3. URL: http://nbuv.gov.ua/UJRN/efek_2015_3_79 (дата звернення: 29.11.2023).

3. Метеленко Н.Г. Внутрішній господарський механізм ефективного функціонування промислових підприємств: теорія, практика, перспективи: монографія. Запоріжжя : КПУ, 2010. 480 с.

4. Гончаренко Б.М. Автоматизація виробничих процесів: Підручник. Національний ун-т харч. технологій. К.: НУХТ, 2013. 796 с.

5. Невлюдов І.Ш. Бережна М. А. Комп'ютерні технології автоматизованого виробництва : навч. посібник. Харків: Компанія СМІТ, 2007. 368 с.

6. Mikell P. Groover Fundamentals of Modern Manufacturing: Materials, Processes, and Systems. 7th ed. Wiley, 2020. 816 p.

7. Підгорна К.Д., Підгорний В.О. Особливості автоматизації ділянки виготовлення заготівок з оптимальним розкроєм рулону. *Сучасні інформаційні технології: теорія, практика, перспективи* : збірник тез доповідей Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих учених, Дніпро, 4-5 грудня 2023 р. Дніпро: УДУНТ, 2023. С. 82–83.

8. Трегуб В. Г. Проектування систем автоматизації : навч. посібник. К.: Видавництво Ліра-К, 2014. 344 с.

9. Ладанюк А. П. Архангельська К. С., Власенко Л. О., Ладанюк А. П. Теорія автоматичного керування технологічними об'єктами : навч. посібник. К.: НУХТ, 2006. 139 с.

10. Волонтир Л.О, Потапова Н.А., Ушкаленко І.М., Чіков І.А. Оптимізаційні методи та моделі в підприємницькій діяльності: навчальний посібник. Вінницький національний аграрний університет. Вінниця: ВНАУ, 2020. 334 с.

11. Wascher G., Haussner H., Schumann H. An improved typology of cutting and packing problems. Working Paper at Faculty of Economics and Management. Magdeburg : Otto von Guericke University. 2005. № 24. P. 1109-1130.

12. Nemhauser G., Wolsey L. *Integer and combinatorial optimization*: John Wiley and Sons, New York, 1988. 776 p.

13. Gomory R.E. Outline of an algorithm for integer solutions to linear programs, Bulletin of the American Mathematical Society 64 (1958). P. 275–278.

14. Donald E. Knuth Art of Computer Programming, The: Sorting and Searching, Volume 3 2nd ed. Addison-Wesley Professional. 1998. 800 p.

15. G. B. Dantzig, A. Orden and P. Wolfe, "The Generalized Simplex Method for Minimizing a Linear form under Linear Inequality Constraints," Pacific Journal of Mathematics, Vol. 5, No. 2, 1955. P. 183-195.

16. Пустюльга С.І. Пуць В.С., Самостян В.Р. Удосконалення технологічного процесу розкрою плит ДСП для виробництва меблів: монографія, Луцьк : ІВВ Луцького НТУ, 2020. 124 с.

17. Cut Optimization OptCut. *Microsoft Apps*. URL: https://apps.microsoft.com/detail/XPDP6H3395J0RV?hl=en-US&gl=US (Last accessed: 29.11.2023).

18. CutLogic 1D Overview. *TMachines* URL: https://www.tmachines.com/cutlogic-1d/ (Last accessed: 29.11.2023).

19. CutLogic 2D Overview. *TMachines* URL: https://www.tmachines.com/cutlogic-2d/ (Last accessed: 29.11.2023).

20. Real Cut 1D with angles. *Optimal Programs*. URL: https://www.optimalprograms.com/realcut1d.htm / (Last accessed: 29.11.2023)

21. CutMaster 2D. URL: https://www.cutmaster2d.com /(Last accessed: 29.11.2023)

22. Mukherjee I, Ray P. A review of optimization techniques in metal cutting processes. Computers and Industrial Engineering. 2006. Vol. 50(1). P.15–34.

23. Вовк В.М., Камінська Н.І. , Прийма С.С. Моделювання економічних процесів підприємства: монографія. Львів: ЛНУ імені Івана Франка, 2011. 448 с.

24. Pisinger D, Sigurd M. Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. *Informs Journal on Computing*, 2007. P. 36–51.

25. Dyckhoff, H., Finke. U. Cutting and Packing in Production and Distributing. Heidelberg : Physica Verlag, 1992. 248 p.

26. Faroe O., Pisinger D., Zachariasen M. Guided local search for the three-dimensional bin-packing problem. Informs journal on computing, 2003. №15. P. 267–283.

27. Maarouf W.F., Barbar A.M., Owayjan M.J. A New Heuristic Algorithm for the 3D Bin Packing Problem. Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering, 2008. P. 342–345.

28. Object-Oriented programming (C#) *Microsoft* URL: https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop/ (Last accessed: 29.11.2023)

29. Томас Г. Кормен Алгоритми доступно: монографія. Київ: «К. І. С.», 2021. 194 с.

30. Вітлінський В.В., Наконечний С.І., Терещенко Т.О. Математичне програмування: навч.-метод. посіб. К.: Вид-во КНЕУ, 2006. 246с.

31. Pinedo M. L. Theory, Algorithms, and Systems. Springer Science & Business Media. New York, 2012. 676 p.

32. Степанюк В.В. Методи математичного програмування. К.: Вища школа, 1997. 272 с.

33. Вища та прикладна математика в економічних прикладах та задачах: практикум (друга частина): навч. посіб. Щетініна О.К., та ін. Київ: Київ. нац. торг.-екон. ун-т, 2019. 416 с.

34. Білоусова С. В., Ковальчук Т. В. Економіко-математичне моделювання: компендіум і практикум: навч. посіб. Київ: Київ. нац. торг.-екон. ун-т, 2018. 524 с.

35. Федоренко І. К. Дослідження операцій в економіці: підручник. Київ: Знання, 2012. 401 с.

36. Філатова Л.Д. Економіко – математичні методи та моделі: Опорний конспект лекцій для самостійного вивчення дисципліни. Харків: ХНМІ ДВНЗ «УБС», 2018. 120 с.

37. Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach 4rd Edition, - Upper Saddle River, NJ : Prentice Hall, 2021. 1166 p.

38. Шаповал Н.В. Методи та системи штучного інтелекту комп'ютерний практикум: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2022. 45с. URL: https://ela.kpi.ua/bitstream/123456789/57162/1/Shapoval_SMShI.pdf (Дата звернення: 29.11.2023).

39. Підгорна К.Д. Дослідження проблем планування розкрою матеріалів // Проблеми економіки та політичної економії. – 2018. – №1. – С. 296–305.

40. Куваєв Я. Нормалізація відношень при проектуванні БД URL:https://rdb.dp.ua/uk/chapter_03 (Дата звернення: 29.11.2023).

ДОДАТОК А

В І Д Г У К

на кваліфікаційну роботу магістра на тему:

«Розробка системи керування ділянкою виготовлення профнастилу з оптимальним розкроєм рулону на мові C#»

студентки групи 126м-22з-1 Підгорної Катерини Дмитрівни.

Метою даної кваліфікаційної роботи є розробка інформаційної системи керування ділянкою нарізання металу на смуги з оптимальним розкроєм рулону на основі сучасних методів оптимізації та інформаційних технологій, що забезпечить ефективне використання матеріалів.

Дана тема актуальна через потребу в ефективному використанні ресурсів та мінімізацію втрат при виробництві. Оптимізація плану розкрою матеріалів дозволяє значно знизити витрати, забезпечуючи економію матеріалу та оптимізовані розрахункові процеси.

Тема кваліфікаційної роботи прямо відноситься до сфери діяльності фахівців галузі знань 12 "Інформаційні технології" на спеціальності 126 "Інформаційні системи та технології". Ця галузь передбачає розробку та операційне управління різноманітними інформаційними системами та відповідним програмним забезпеченням.

Оригінальність технологічних рішень полягає в використанні точних комбінаторних методів у поєднанні з симплекс-методом та методом Гоморі.

Практичне значення результатів кваліфікаційної роботи полягає в розробці інформаційної системи, що дозволяє виконувати оптимальний розрахунок розкрою листового матеріалу для керування виробничою ділянкою підприємства.

Оформлення графічних та текстових матеріалів у пояснювальній записці кваліфікаційної роботи відповідає стандарту і не має відхилень.

Ступінь самостійності у виконанні кваліфікаційної роботи повністю відповідає другому освітньо-кваліфікаційному рівню вищої освіти, а саме, магістра.

У дослідженні детально розглянуто сучасний стан ринку програмних додатків, спрямованих на оптимізацію розкрою листових матеріалів відповідного рівня.

Деякі дискусійні положення та несуттєві недоліки пов’язані з наступними моментами:

– в роботі інформаційної системи не передбачено авторизацію користувача;

– є обмеження по кількості типорозмірів, що обумовлено специфікою діяльності підприємства;

– для введення певних нових даних є необхідність звернення до розробника.

Незважаючи на вищевказані зауваження, кваліфікаційна робота в цілому заслуговує оцінки «відмінно» а її виконавець, студенка Підгорна Катерина Дмитрівна, присвоєння їй кваліфікації магістр за спеціальністю 126 «Інформаційні системи та технології».

Керівник кваліфікаційної роботи,
професор кафедри ІТКІ, д-р техн. наук **В.І. Олевський**

ДОДАТОК Б

Рецензія

на кваліфікаційну роботу магістра на тему:
«Розробка системи керування ділянкою виготовлення профнастилу з оптимальним розкроєм рулону на мові C#»
студентки групи 126м-22з-1 Підгорної Катерини Дмитрівни

Збільшення продуктивності виробництва вимагає впровадження якісних та сучасних підходів для вирішення завдань на кожному етапі виробничого процесу. Зменшення втрат матеріалу завдяки оптимізації розкрою є критичним для підприємств, оскільки це призводить до значних економічних вигод через зниження витрат та підвищення ефективності виробництва. Саме тому розробка інформаційної системи керування ділянкою нарізання металу на смуги з оптимальним розкроєм рулону на основі сучасних методів оптимізації та інформаційних технологій є актуальною задачею.

В рецензованій кваліфікаційній роботі було розроблено інформаційну систему розрахунку оптимального розкрою листового матеріалу на мові C# у середовищі Microsoft Visual Studio. Система підключена до бази даних для зберігання інформації.

Використовувані технології для розробки та забезпечення функціонування аналогічних програм є не тільки евристичним елементом, але й прямо пов'язані з предметною областю діяльності фахівця, що спеціалізується в галузі знань 12 "Інформаційні технології", на спеціальності 126 "Інформаційні системи та технології".

Студентка Підгорна К.Д. досить добре розібралася в методиках, що стосуються алгоритмів оптимізації, підходах та рішеннях, спрямованих на вдосконалення ефективності та точності розкрою. За результатами виконання роботи було розроблено інформаційну систему оптимального розкрою.

До недоліків роботи слід віднести:

- інформаційна система передбачає фіксовані списки замовників, які потрібно додавати через базу даних розробнику;
- відсутність можливості автентифікації користувача в інформаційній системі.

Проте, створену інформаційну систему можна розглядати як завершений етап виконаної кваліфікаційної роботи.

Враховуючи вищевикладене, можна зробити висновок, що подані матеріали повністю відповідають вимогам до кваліфікаційних робіт другого рівня вищої освіти, а саме ступеня магістра. З огляду на широкий спектр розроблених компонентів, які забезпечують створення цієї кваліфікаційної роботи, вона в цілому заслуговує на оцінку «відмінно» а її виконавець, студенка Підгорна К.Д., присвоєння їй кваліфікації магістр за спеціальністю 126 «Інформаційні системи та технології».

Рецензент, доцент кафедри інформаційних систем і технологій УДУНТ, к.т.н **Анна ЖУРБА**