

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)
Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

Студента Нагорного Максима Вікторовича
академічної групи 124-20-1
спеціальності 124 Системний аналіз
на тему: «Оптимальне завантаження транспортних засобів на основі інтелектуальної системи підтримки прийняття рішень»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>к. т. н. доц. Желдак Т.А.</i>			
розділів:				
Інформаційно-аналітичний	<i>к. т. н. доц. Желдак Т.А.</i>			
Спеціальний розділ	<i>к. т. н. доц. Желдак Т.А.</i>			
Рецензент	<i>д.т.н., проф. Алексєєв М.А.</i>			
Нормоконтролер	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			

Дніпро
2024

ЗАТВЕРДЖЕНО:
завідувач кафедри
Системного аналізу та управління
(повна назва)
к.т.н., доц. Желдак Т.А
(підпис) (прізвище, ініціали)
« ____ » _____ 20__ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

студенту Нагорному М.В. академічної групи 124-20-1
спеціальності: 124 Системний аналіз
на тему «Оптимальне завантаження транспортних засобів на основі інтелектуальної системи підтримки прийняття рішень»

затверджену наказом ректора НТУ «Дніпровська політехніка»
від _____

Розділ	Зміст завдання	Термін виконання
1. Інформаційно-аналітичний розділ	<i>Дослідити метод глобальної оптимізації Tabu Search, та методи оптимізації. На основі їх аналізу виділити основні компоненти й принципи побудови інтелектуальної системи оптимізації задачі завантаження транспортних засобів.</i>	
2. Спеціальний розділ	<i>Розробити й реалізувати інтелектуальну систему, що знаходить глобальні екстремуми задачі оптимального завантаження продукції. Протестувати алгоритм для тестових даних. Вирішити практичну задачу оптимізації при зміні налаштувань розробленого алгоритму пошуку із заборонами.</i>	

Завдання видано _____ доц. Желдак Т.А
(підпис) (прізвище, ініціали)

Дата видачі: _____

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Нагорний М.В.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 109 с., 12 рис., 10 табл., 12 джерел, 8 додатків.

Об'єктом дослідження є процес виконання замовлень товарів в умовах підприємства ТОВ «РУШ».

Практична цінність роботи полягає в тому, що на основі даної роботи можна прийняти правильне рішення при виборі способу розподілу замовленої продукції та її оптимального завантаження у транспортні засоби, використовуючи розроблений алгоритм пошуку із заборонами, що дає кращі показники за часом, ніж обраний альтернативний метод оптимізації.

Метою даної кваліфікаційної роботи є підвищення ефективності розподілення ресурсів.

В результаті роботи розроблено програмний продукт, який дозволяє розв'язувати задачі оптимізації за допомогою жадного алгоритму, метода «гілок та меж» і пошуку Tabu. Результати розрахунків подано у вигляді таблиць та графіків. Наведено детальний опис структури програми та проект по впровадженню та використанню програмного продукту. Отримані результати вказують на доцільність використання розробленого алгоритму для отримання високоякісних розв'язків розглянутої задачі.

Практична цінність роботи полягає в тому, що на основі даної роботи можна прийняти правильне рішення при виборі способу розподілу замовленої продукції та її оптимального завантаження у транспортні засоби, використовуючи розроблений алгоритм пошуку із заборонами, що дає кращі показники за часом, ніж обраний альтернативний метод оптимізації.

Ключові слова: ГЛОБАЛЬНА ОПТИМІЗАЦІЯ, ЛОКАЛЬНИЙ ПОШУК, ГЛОВЕР, ОДИНИЧНЕ ЗАВДАННЯ, ПРИНЦИП ПОШУКУ ІЗ ЗАБОРОНАМИ, МЕХАНІЗМ КОРОТКОЇ ТА ДОВГОЇ ПАМ'ЯТІ

THE ABSTRACT

Explanatory note: 109 pages, 12 figures, 10 tables, 12 sources, 8 addition.

The object of the research is the process of fulfilling orders for goods at the enterprise LLC "PYIII".

The subject of the research is the process of optimizing order distribution to reduce the number of pallets used.

The purpose of this qualification work is to improve the efficiency of resource distribution.

Results of the work: A software product has been developed that solves optimization problems using a greedy algorithm, the branch and bound method, and tabu search. The calculation results are presented in the form of tables and graphs. A detailed description of the program structure and the project for the implementation and use of the software product is provided. The obtained results indicate the feasibility of using the developed algorithm to achieve high-quality solutions to the considered problem.

Practical value of the work: Based on this work, the correct decision can be made when choosing the method of distributing ordered products and their optimal loading into vehicles using the developed tabu search algorithm, which provides better time performance than the chosen alternative optimization method.

Keywords: GLOBAL OPTIMIZATION, LOCAL SEARCH, GLOVER, SINGLE TASK, TABU SEARCH PRINCIPLE, SHORT-TERM AND LONG-TERM MEMORY MECHANISM.

ЗМІСТ

ВСТУП.....	7
1. ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ.....	9
1.1 Опис підприємства.....	9
1.2 Характеристика методів рішення задач оптимізації.....	14
1.3 Жадібні алгоритми.....	17
1.4. Метод гілок і меж.....	19
1.5 Планування завдань.....	21
1.6 Постановка задачі.....	25
2 СПЕЦІАЛЬНИЙ РОЗДІЛ.....	30
2.1 Алгоритм пошуку Tabu.....	30
2.2 Дослідження розв'язання поставленої задачі різними методами.....	42
2.2.1 Розробка жадібного алгоритму.....	44
2.2.2 Розробка алгоритму методу гілок та меж.....	44
2.2.3 Розробка алгоритму пошуку із заборонами.....	45
2.3 Порівняння розроблених та протестованих методів вирішення задачі оптимізації.....	45
2.3.1 Метод Зворотного Усунення.....	74
2.3.2 Відставання Переміщень.....	76
2.3.3 Метод Забороненого Циклу.....	77
2.3.4 Умовний Імовірнісний Метод.....	81
2.4. Оптимальне налаштування роботи алгоритму Tabu Search.....	50
3 ПРОЕКТНИЙ РОЗДІЛ.	
3.1 Актуальність розробки проектів, інновацій та бізнес-планування.	

Сучасні системи розробки проектів.....	52
3.2 Розробка проекту. Перелік етапів, виконавців та затрат проекту.....	65
3.3 Побудова діаграми Ганта і можлива оптимізація проекту.....	69
3.4 Окупність проекту та висновки.....	71
ВИСНОВКИ.....	73
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ:.....	74
ДОДАТОК А.....	76
ДОДАТОК Б.....	77
ДОДАТОК В.....	78
ДОДАТОК Г.....	79
ДОДАТОК Д.....	87
ДОДАТОК Е.....	89
ДОДАТОК Є.....	92
ДОДАТОК Ж.....	96

ВСТУП

Актуальність викликана основними тенденціями розвитку логістики, до яких відносяться наступні чинники:

1. Зростання витрат на перевезення. Традиційні методи розподілу стали дорожчими у зв'язку із зростанням товарних цін і інфляцією;
2. Досягнення межі ефективності виробництва. Стає все важчим досягти істотного зниження виробничих витрат, тому що з виробництва вже «зняті всі сливки». З іншого боку, логістика залишається областю, де ще зберігаються значні потенційні можливості скорочення витрат фірми;
3. Комп'ютерні технології. Розвиток комп'ютерних технологій дозволяє здійснити на практиці концепцію логістики.

Задача полягає в дослідженні алгоритмів для розв'язання задачі оптимізації завантаження транспортних засобів замовленою продукцією побутової хімії для подальшої поставки на точки та сформульована у наступному вигляді:

- 1) Задані та сформульовані замовлення дистрибуторів;
- 2) Дистрибутори замовляють лише присутню на складі імпортера продукцію (обговорюється з адміністратором департаменту продаж);
- 3) Із замовлень вилучені акційні позиції та набори;
- 4) Задані характеристики для кожного виду продукції;
- 5) Задана місткість палет, на які завантажують замовлену продукцію а також зазначено;

Необхідно знайти

- Оптимальне завантаження палети;
- Оптимальну кількість завантажених палет для вдоволення потреб дистрибуторів.

Зважаючи на таку постановку проблеми дана задача відноситься до класу задач комбінаторної оптимізації і полягає в наступному:

1. Розробити та реалізувати алгоритм пошуку із заборонами із використанням структур короткої та довгої пам'яті та декілька альтернативних алгоритмів;

2. Створити програмний продукт, призначений для розв'язання оптимізаційної задачі з використанням розроблених алгоритмів, дослідити ефективність їх застосування;

3. Використовуючи розроблене програмне середовище провести дослідження впливу значень змінних параметрів алгоритму пошуку Tabu на якість отриманих розв'язків для тестових задач різної розмірності.

4. Зробити висновки щодо доцільності використання запропонованого методу Tabu Search.

Наукова новизна отриманих в роботі результатів полягає в наступному:

- Вперше застосовано алгоритм пошуку із заборонами для задачі оптимального завантаження транспортних засобів;

- Досліджено налагодження запропонованого алгоритму з метою покращення показників якості його роботи;

- Отримані оптимальні значення налаштувань в межах припустимого часу розрахунків.

1. ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Опис підприємства

Лінія магазинів EVA, утворена у 2002 р. - найбільша національна мережа магазинів краси та здоров'я, що пропонує широкий асортимент косметики, парфумерії, аксесуарів, засобів для догляду за малюками та побутових товарів відомих світових і власних брендів (загалом понад 30000 асортиментних позицій). Понад 8 мільйонів осіб є користувачами програми лояльності «EVA МОЗАІКА».

Принципи роботи:

- Партнерський стиль ведення бізнесу;
- Професійний підхід у всіх проявах діяльності компанії;
- Чесність, порядність, відданість.

Бачення: Підвищення ефективності основної діяльності компанії, шляхом партнерського стилю ведення бізнесу, за рахунок професійного підходу і високоефективних проектів.

Ціль: Покращення стану компанії, не дивлячись на першість на ринку України.

Стратегічні цілі:

- Створення найбільш повного і збалансованого асортиментного портфеля повною мірою задовольняючої потреби, як торговельних точок, так і кінцевого споживача;
- Покращення партнерських відносин з постачальниками і клієнтами;
- Розвиток та підвищення кваліфікації сильної команди професіоналів;
- Диверсифікованість і розширення діяльності, нові ринки і продукти.

Структура компанії «EVA» включає:

1. Генеральний відділ (Відділ стратегічного планування; Відділ Внутрішнього аудиту)
2. Фінансовий відділ (Бухгалтерія; Фінансовий аналіз та планування; Управління ризиками; Казначейство)
3. Операційний відділ (Логістика та склади; Закупівлі; Виробництво; Управління ланцюгом постачання)
4. Відділ маркетингу (Маркетингові дослідження; Реклама та PR; Цифровий маркетинг; Управління брендом)
5. Відділ продажу (Відділ роздрібних продажів; Відділ корпоративних продажів; Відділ обслуговування клієнтів; Управління каналами дистрибуції)
6. Відділ інформаційних технологій (Розробка та підтримка ПЗ; Інфраструктура та мережі; Кібербезпека; Управління даними)
7. Відділ персоналу (Підбір та адаптація персоналу; Навчання та розвиток; Управління кадровим резервом; Корпоративна культура та мотивація)
8. Юридичний відділ (Відділ комплаєнсу; Управління договорами; Захист прав інтелектуальної власності)
9. Відділ досліджень та розвитку (R&D) (Інновації та технології; Розробка нових продуктів; Вдосконалення процесів)
10. Відділ охорони праці та безпеки (Безпека на робочому місці; Екологічна безпека; Планування аварійних ситуацій)

Наразі компанія має понад 1 000 фірмових магазинів по всій Україні, а також власний інтернет-магазин EVA.UA. Штат її співробітників налічує близько 13 тисяч осіб. EVA дотримується найвищих стандартів роботи з клієнтами та високого рівня корпоративної культури.

У структуру відділу продажів інтегрований відділ по роботі з VIP-клієнтами. Уся територія України покривається продажами різної продукції з загального портфеля брендів. Дистрибуція продукції здійснюється нашою компанією на 70% території України, відповідно до територіального поділу (рис. 1.1).



Рисунок 1.1 – Територіальний поділ дистрибуції.

1.2 Характеристика методів рішення задач оптимізації.

Під загальною задачею оптимізації розуміють наступну задачу: задана множина X і функція $f(x)$, визначена на X , для якої необхідний визначити її максимальне або мінімальне значення. Ця задача може бути записана так:

$$f(x) \rightarrow \min(\max), x \in X. \quad (1.1)$$

Функція $f(x)$ називається цільовою функцією, X - припустимою безліччю. Задача (1.1) називається задачею безумовної оптимізації, якщо $X = E_n$, тобто якщо вона має вигляд $f(x) \rightarrow \min(\max), x \in E_n$.

Задача (1.1) називається задачею умовної оптимізації, якщо X - власна підмножина простору E_n , тобто $X \subset E_n$.

Окремий клас умовних задач оптимізації складають задачі математичного програмування, що поділяються на задачі лінійного програмування (цільова функція й обмеження лінійні) і задачі нелінійного програмування.

Загальною задачею лінійного програмування називається задача, що складається у визначенні максимального (мінімального) значення функції:

$$F = \sum_{j=1}^n c_j x_j \quad (1.2)$$

при умовах:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i (i = \overline{1, k}) \quad (1.3)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i (i = \overline{k+1, m}) \quad (1.4)$$

$$\text{та } x_{ij} \geq 0 (j = \overline{1, l}, l \leq n), \quad (1.5)$$

де a_{ij} , b_i , c_j - задані постійні величини і $k \leq m$. Функція (1.1) - цільова функція, (2)-(4) - обмеження задачі.

При рішенні конкретної задачі оптимізації дослідник насамперед повинний вибрати математичний метод, що приводив би до кінцевих результатів з найменшими витратами на обчислення або ж давав можливість одержати найбільший обсяг інформації про шукане рішення. Вибір того або іншого методу в значній мірі визначається постановкою оптимальної задачі, а також використовуваною математичною моделлю об'єкта оптимізації.

В даний час для рішення оптимальних задач застосовують в основному наступні методи:

- методи дослідження функцій класичного аналізу;
- методи, засновані на використанні невизначених множників Лагранжа;
- варіаційне вирахування;
- динамічне програмування;
 - принцип максимуму;
- лінійне програмування;
- нелінійне програмування.

Останнім часом розроблений і успішно застосовується для рішення визначеного класу задач метод геометричного програмування.

Як правило, не можна рекомендувати який-небудь один метод, якому можна використовувати для рішення усіх без винятку задач, що виникають на практиці. Одні методи в цьому відношенні є більш загальними, інші - менш загальними. Нарешті, повну групу методів (методи дослідження функцій класичного аналізу, метод множників Лагранжа, методи нелінійного програмування) на визначених етапах рішення оптимальної задачі можна застосовувати в сполученні з іншими методами, наприклад динамічним програмуванням або принципом максимуму.

Відзначимо також, що деякі методи спеціально розроблені або щонайкраще підходять для рішення оптимальних задач з математичними моделями визначеного виду. Так, математичний апарат лінійного програмування, спеціально створений для рішення задач з лінійними критеріями оптимальності і лінійних обмежень на перемінні і дозволяє

вирішувати більшість задач, сформульованих у такій постановці. Так само і геометричне програмування призначене для рішення оптимальних задач, у яких критерій оптимальності й обмеження представляються спеціального виду функціями позиномами.

Динамічне програмування гарне пристосовано для рішення задач оптимізації многостадійних процесів, особливо тих, у яких стан кожної стадії характеризується відносно невеликим числом перемінні стани. Однак при наявності значного числа цих перемінних, тобто при високій розмірності кожної стадії, застосування методу динамічного програмування важко внаслідок обмежені швидкодії й обсягу пам'яті обчислювальних машин.

Мабуть, найкращим шляхом при виборі методу оптимізації, найбільш придатного для рішення відповідної задачі, варто визнати дослідження можливостей і досвіду застосування різних методів оптимізації.

1.3 Жадібні алгоритми.

Алгоритми, призначені для рішення задач оптимізації, звичайно являють собою послідовність кроків, на кожному з яких надається деяка множина виборів. Визначення найкращого вибору, керуючись принципами динамічного програмування, у багатьох задачах оптимізації нагадує стрілянину з гармати по горобцях; іншими словами, для цих задач краще підходять більш прості й ефективні алгоритми. У жадібному алгоритмі (greedy algorithm) завжди робиться вибір, що здається найкращим у даний момент - тобто виробляється локально оптимальний вибір у надії, що він приведе до оптимального рішення глобальної задачі.

Жадібні алгоритми не завжди приводять до оптимального рішення, але в багатьох задачах вони дають потрібний результат і добре підходять для досить широкого класу задач.

Жадібний алгоритм дозволяє одержати оптимальне рішення задачі шляхом здійснення ряду виборів. У кожній крапці ухвалення рішення в алгоритмі

робиться вибір, що у даний момент виглядає найкращим . Ця евристична стратегія не завжди дає оптимальне рішення, але все-таки рішення може виявитися й оптимальним, у чому ми змогли переконатися на прикладі задачі про вибір процесів. У дійсному розділі обговорюються деякі загальні властивості жадібних методів.

Процес розробки жадібного алгоритму, розглянутий у розділі 1.3, трохи складніше, ніж звичайно. Були пройдені перераховані нижче етапи:

- а) Визначена оптимальна підструктура задачі;
- б) Розроблене рекурсивне рішення;
- в) Доведено, що на будь-якому етапі рекурсії один з оптимальних виборів є жадібним. З цього випливає, що завжди можна робити жадібний вибір;
- г) Показано, що усі виникаючі в результаті жадібного вибору підзадачі, крім однієї, - порожні;
- д) Розроблений рекурсивний алгоритм, що реалізує жадібну стратегію;
- е) Рекурсивний алгоритм перетворений в ітеративний;

При виконанні цих етапів розглянуто, як динамічне програмування є основою для жадібного алгоритму. Однак звичайно на практиці при розробці жадібного алгоритму ці етапи спрощуються. Необхідно розробити підструктуру так, щоб у результаті жадібного вибору залишалася тільки одна підзадача, що підлягає оптимальному рішення. Наприклад, у задачі про вибір процесів спочатку визначаються підзадачі S_{ij} , у яких змінюються обидва індекси, - i і j . Потім ми з'ясували, що якщо завжди робиться жадібний вибір, ті підзавдання можна було б обмежити видом $S_{i,n+1}$.

Можна запропонувати альтернативний підхід, у якому оптимальна підструктура пристосовувалася б спеціально для жадібного вибору - тобто другий індекс можна було б опустити і визначити підзадачі у виді $S_i = \{a_k \in S : f_i \leq s_k\}$. Потім можна було б довести, що жадібний вибір (процес, що закінчується першим у задачі S_i) у сполученні з оптимальним рішенням для множини S_m інших сумісних між собою процесів приводить до оптимального

рішення задачі S_i . Узагальнюючи сказане, опишемо процес розробки жадібних алгоритмів у виді послідовності перерахованих нижче етапів:

1. Привести задачу оптимізації до виду, коли після зробленого вибору залишається вирішити тільки одну підзадачу;
2. Довести, що завжди існує таке оптимальне рішення вихідної задачі, яке можна одержати шляхом жадібного вибору, так що такий вибір завжди допустимо;
3. Показати, що після жадібного вибору залишається підзадача, що володіє тим властивістю, що об'єднання оптимального рішення підзадачі зі зробленим жадібним вибором приводить до оптимального рішення вихідної задачі.

Описаний вище спрощений процес буде використовуватися далі. Проте, помітимо, що в основі кожного жадібного алгоритму майже завжди знаходиться більш складне рішення в стилі динамічного програмування.

Як визначити, чи здатний жадібний алгоритм вирішити задачу оптимізації, що стоїть перед нами? Загального шляху тут немає, однак можна виділити дві основні складові: властивості жадібного вибору й оптимальну підструктуру. Якщо задача володіє двома цими властивостями, то можемо сказати, що для неї можна розробити жадібний алгоритм.

1.4. Метод гілок і меж.

Метод був вперше запропонований Ленд і Дойг в 1960 р. для вирішення завдань лінійного програмування, він відноситься до групи комбінаторних методів дискретного програмування і є одним з найбільш поширених методів цієї групи. Комбінаторні методи виходять з кінцівки числа допустимих планів завдання і заміняють повний перебір всіх планів їх частковим направленим перебором. Комбінаторні методи в значно меншому ступені схильні в процесі обчислень до впливу помилок округлення, тому є

переважнішими в порівнянні з методами відсікання. Метод гілок і меж - один з найбільш ефективних методів рішення задач комбінаторного типу.

Перейдемо до викладу суті методу. Для цього розглянемо загальне завдання дискретного програмування:

$$\max Z = f(x) \quad (1.6)$$

$$x \in \Omega, \quad (1.7)$$

де Ω - кінцевамножина допустимих планів.

Знаходимо верхню межу (оцінку) функції $f(x)$, $x \in \Omega$, тобто таке число $\phi_0(\Omega)$, що для будь-яких $x \in \Omega$: $f(x) \leq \phi_0(\Omega)$. Якщо при цьому вдається знайти такий план задачі (1.6) - (1.7), для якого виконується рівність $f(x_0) = \phi_0(\Omega)$, то x_0 - оптимальний план задачі (1.6) - (1.7).

Якщо оптимальний план не знайдений, то деяким способом розбиваємо множину Ω на кінцеве число непересічних підмножин Ω_r^1 :

$\Omega = \cup_{r=1}^{r_1} \Omega_r^1, \cap_{r=1}^{r_1} \Omega_r^1 = \emptyset$ і знаходимо для кожної з цих підмножин верхню межу $\phi_1(\Omega_r^1)$ ($r = \overline{1, r_1}$). Якщо ж такий план не знайдений, то вибираємо підмножину Ω_r^1 з найбільшою верхньою межею (перспективна підмножина) і розбиваємо її на декілька непересічних підмножин Ω_s^2 ($s = \overline{1, s_1}$). Для кожної нової підмножини знаходимо верхню межу $\phi(\Omega_s^2)$. Якщо буде знайдений такий план x_k^2 , що $f(x_k^2) = \phi(\Omega_k^2) \geq \phi(\Omega_s^2)$, то x_k^2 - оптимальний план завдання. Якщо оптимальний план не знайдено, то подальшому розгалуженню піддаємо підмножину з найбільшою верхньою межею, і т.д. Процес триває до отримання оптимального плану. Способи галуження і знаходження верхніх меж вибираються для кожного конкретного завдання дискретного програмування. Процес супроводжується побудовою дерева галуження (рис. 1.2).

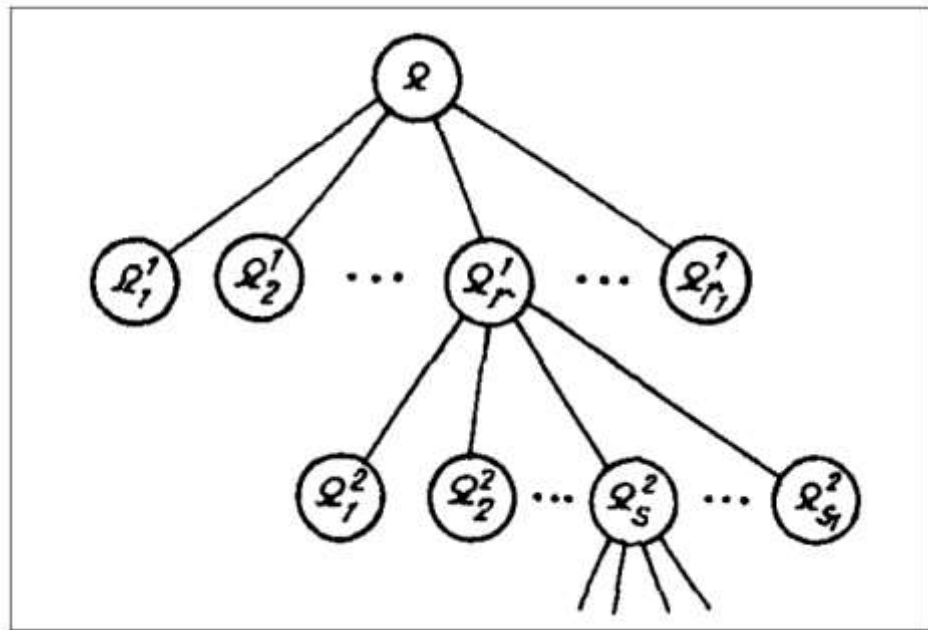


Рисунок 1.2 – Дерево галуження на основі методу гілок та меж.

1.5 Планування завдань.

Цікава задача, яку можна вирішити за допомогою матроїдів, - задача по складанню оптимального розкладу одиничних завдань, що виконуються на одному процесорі. Кожне завдання характеризується кінцевим терміном виконання, а також штрафом при пропуску цього терміну. Ця задача здається складною, однак вона на подив просто вирішується за допомогою жадібного алгоритму.

Одиничне завдання (unit-time job) - це завдання (наприклад, комп'ютерна програма), для виконання якого потрібен одиничний інтервал часу. Якщо мається кінцева множина S таких завдань, розклад (schedule) для цієї множини являє собою перестановку елементів множини S , що визначає порядок їхнього виконання. Перше завдання в розкладі починається в нульовий момент часу і закінчується в момент часу 1, друге завдання починається в момент часу 1 і закінчується в момент часу 2 і т.д.

Вхідні дані в задачі по плануванню на одному процесорі одиничних завдань, що характеризуються кінцевим терміном виконання і штрафом, мають такий вигляд:

- множина $S = \{a_1, a_2, \dots, a_n\}$, що складається з p одиничних завдань;
- множина d_1, d_2, \dots, d_n кінцевих термінів виконання, представлених цілими числами $1 \leq d_i \leq n$; передбачається, що завдання a_i , повинне завершитися до моменту часу d_i ;
- множина з p ненегативних ваг або штрафних сум w_1, w_2, \dots, w_n ; якщо завдання a_i , не буде виконано до моменту часу d_i , вилучається штраф w_i , якщо це завдання буде виконано в термін, штрафні санкції не застосовуються.

Для множини завдань S потрібно знайти розклад, мінімізующее сумарний штраф, що накладається за всі прострочені завдання.

Розглянемо довільний розклад. Говорять, що завдання в ньому прострочене, якщо воно завершується пізніше кінцевого терміну виконання. У протилежному випадку завдання своєчасне. Довільний розклад завжди можна привести до виду з першочерговими своєчасними завданнями (early-first form), коли своєчасні завдання виконуються перед простроченими. Щоб продемонструвати це, помітимо, що якщо яке-небудь своєчасне завдання a_i , впливає після деякого простроченого завдання a_j , те завдання a_i і a_j можна поміняти місцями, причому завдання a_i все рівно залишиться своєчасним, а завдання a_j - простроченим.

Аналогічно, справедливе твердження, відповідно до якого довільний розклад можна привести до канонічного виду (canonical form), у якому своєчасні завдання передують простроченим і розташовані в порядку монотонного зростання кінцевих термінів виконання. Для цього спочатку приведемо розклад до виду з першочерговими своєчасними завданнями. Після цього, доти поки в розкладі будуть матися своєчасні завдання a_i і a_j , що закінчуються в моменти часу k і $k + 1$ відповідно, але

при цьому $d_j < d_i$, ми будемо змінювати їх місцями. Оскільки завдання a_j до перестановки було своєчасним, $k + 1 \leq d_j$. Таким чином, $k + 1 < d_i$, і завдання a_i , залишається своєчасним і після перестановки. Завдання a_j зрушується на більш ранній час, тому після перестановки воно теж залишається своєчасним.

Приведені вище міркування дозволяють звести пошук оптимального розкладу до визначення множини A , що складає зі своєчасних завдань в оптимальному розкладі. Як тільки така множина буде визначено, можна буде створити фактичний розклад, включивши в нього елементи множини A в порядку монотонного зростання моментів їхнього закінчення, а потім - перелічивши прострочені завдання $(S - A)$ у довільному порядку. У такий спосіб буде отриманий канонічно упорядкований оптимальний розклад.

Говорять, що множина завдань A незалежна, якщо для нього існує розклад, у якому відсутні прострочені завдання. Очевидно, що множина своєчасних завдань розкладу утворить незалежну множина завдань. Позначимо через T сімейство всіх незалежних безлічей завдань.

Розглянемо задачу, що складається у визначенні того, чи є задана множина завдань A незалежним. Позначимо через $N_t(A)$ кількість завдань множини A , кінцевий термін виконання яких дорівнює t або настає раніш (величина t може приймати значення $0, 1, 2, \dots, n$). Помітимо, що $N_0(A) = 0$ для будь-якої множини A .

Лема 1.1 – Для будь-якої множини завдань A сформульовані нижче твердження еквівалентні.

1. Множина A незалежна.
2. Для всіх $t = 0, 1, 2, \dots, n$ виконуються нерівності $N_t(A) \leq t$.
3. Якщо в розкладі завдання з множини A розташовані в порядку монотонного зростання кінцевих термінів виконання, то жодне з них не є простроченим.

Доведення. Очевидно, що якщо для деякого t $N_t(A) > t$, те неможливо скласти розклад таким чином, щоб у множини A не виявилось прострочених

завдань, оскільки до настання моменту t залишається більш t незавершених завдань. Таким чином, твердження (1) припускає виконання твердження (2). Якщо виконується твердження (2), то i -й один по одному термін завершення завдання не перевищує i , так що при розміщенні завдань у цьому порядку всі терміни будуть дотримані. Нарешті, із твердження (3) тривіальним образом впливає справедливість твердження (1).

За допомогою властивості (2) леми 2.9 легко визначити, чи є незалежним задана множина завдань.

Задача по мінімізації суми штрафів за прострочені завдання - це та ж саме, що задача по максимізації суми штрафів, які удалося уникнути завдяки своєчасному виконанню завдань. Таким чином, приведена нижче теорема гарантує, що за допомогою жадібного алгоритму можна знайти незалежна множина завдань A з максимальною сумою штрафів.

Теорема 1.2 - Якщо S - множина одиничних завдань з кінцевим терміном виконання, а I - сімейство всіх незалежних множин завдань, те відповідна система (S, I) - матроїд.

Доведення. Зрозуміло, що будь-яка підмножина незалежної множини завдань теж незалежна. Щоб довести, що виконується властивість заміни, припустимо, що B і A - незалежні множини завдань, і що $|B| > |A|$. Нехай k - найбільше t , таке що $N_t(B) \leq N_t(A)$ (таке значення t існує, оскільки $N_0(A) = N_0(B) = 0$). Так як $N_n(B) = |B|$ і $N_n(A) = |A|$, але $|B| > |A|$, виходить, що $k < n$, і для всіх j у діапазоні $k + 1 \leq j \leq n$ повинне виконуватися співвідношення $N_j(B) > N_j(A)$. Таким чином, B множини B утримується більше завдань з кінцевим терміном виконання $k + 1$, чим у множини A . Нехай a_i - завдання з множини $B - A$ з кінцевим терміном виконання $k + 1$, і нехай $A' = A \cup \{a_i\}$.

Тепер за допомогою другої властивості леми 16.12 покажемо, що множина A' повинна бути незалежним. Оскільки множина A незалежна, для будь-якого $0 \leq t \leq k$ виконується співвідношення $N_t(A') = N_t(A) \leq t$. Для $k < t \leq n$, оскільки B - незалежна множина, маємо $N_t(A') \leq N_t(B) \leq t$. Отже, множина A незалежна, що і завершує доказ того, що (S, I) - матроїд.

За допомогою теореми 2.8 можна сформулювати жадібний алгоритм, що дозволяє знайти незалежна множина завдань A з максимальною вагою. Після цього можна буде створити оптимальний розклад, у якому елементи множини A будуть відігравати роль своєчасних завдань. Цей метод дає ефективний алгоритм планування одиничних завдань з кінцевим терміном виконання і штрафом для одного процесора. Час роботи цього алгоритму, у якому використовується процедура GREEDY, дорівнює $O(n^2)$, оскільки кожна з $PRO(n)$ перевірок незалежності, що виконуються в цьому алгоритмі, вимагає часу $PRO(n)$.

У табл. 2.2 приводиться приклад задачі по плануванню на одному процесорі одиничних завдань з кінцевим терміном виконання і штрафом. У цьому прикладі жадібний алгоритм вибирає завдання a_1, a_2, a_3 і a_4 , потім відкидає завдання a_5 і a_6 , і нарешті вибирає завдання a_7 . У результаті виходить оптимальне рас писання $(a_2, a_4, a_1, a_3, a_7, a_5, a_6)$, якому відповідає загальна сума штрафу, рівна $w_5 + w_6 = 50$.

Таблиця 1.1 – Приклад задачі по плануванню одиничних завдань з кінцевим терміном виконання і штрафом на одному процесорі.

a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

1.6 Постановка задачі.

Для початку опишемо процес поступання замовлень від дистрибуторів. Згідно прописаної і затвердженої маршрутизації дистрибуторів у центральний офіс ТОВ “РУШ” щодня надходять замовлення в установленням бланку замовлення (Додаток Г). У замовленні присутні близько 300 назв продукції, із зазначеними вагою БРУТТО кожної позиції, кількістю штук у ящику і ціною товару. Товарознавець дистрибутора здійснює замовлення шляхом заповнення кількості замовлених ящиків в установленому бланку замовлення,

а також погоджує замовлення в онлайн режимі з адміністратором відділу продаж центрального офісу ТОВ “РУШ” (уточнення наявності замовленої продукції на складі імпортера, добавка або скорочення замовлення згідно плану закупівель дистрибутора, що встановлюється щомісяця).

Далі замовлення узгоджується (або відхиляється) директорами всіх департаментів і після підписання директором відділу продажів, надходить у відділ логістики, а далі на склад для завантаження транспортних засобів і постачання на дистрибуторів у зазначений термін (табл. 1.4).

Таблиця 1.2 – Зони і терміни доставки.

Області України	№ Зони	Термін доставки
Закарпатська	11, 12	72
Запорізька	1-5, 7	24-72
Вінницька	5-9, 11	48-72
Волинська	10, 11	72
Дніпропетровська	1-3, 5	24-72
Житомирська	5-7	48
Івано-Франківська	8, 10, 11	48-72
Кіровоградська	2-5, 7-9	24-72
Київська	5, 6, 8	24-48
Львівська	8,10	48-72
Миколаївська	3, 7, 8, 12	24-72
Одеська	4, 7-9	48-72
Полтавська	2, 3, 5-7	24-72
Рівненська	8, 10-12	48-72
Сумська	5-8	48-72
Тернопільська	8, 10, 11	48-72
Харківська	5, 6, 8	48-72
Херсонська	3, 4, 7-9	24-72
Хмельницька	7, 8, 11, 12	48-72
Черкаська	4-7	24-48
Чернівецька	11, 12	72
Чернігівська	5-7	24-48

24 - один робочий день
48 - два робочих дня
96 - чотири робочих дня

Основним завданням роботи з оптимізація завантаження транспортних, а саме раціональне завантаження палет з метою економії ресурсів підприємства згідно з існуючими тарифами перевезень (Табл. 1.5).

Таблиця 1.3 – Тарифи на транспортування палетованого вантажу

Зона доставки палет	Дніпро Тернівка Запоріжжя	Комсомольськ Кривий ріг	Кременчук Токмак	Кіровоград
	1	2	3	4
1	124,98	147,32	162,12	200,49
2	102,22	122,76	135,10	167,08
3	92,00	110,49	121,59	147,90
4	85,31	102,55	113,04	137,43
5	80,83	97,25	107,37	130,48
6	75,19	92,13	101,87	123,74
7	70,84	88,33	97,81	118,75
8	66,77	84,65	93,87	113,92
9	62,96	81,08	90,04	109,22
10	58,55	76,54	85,11	103,19
Зона доставки палет	Харків Київ Чернігів	Суми Житомир Бердичів Черкаси	Миколаїв Одеса Вінниця Конотоп	Червоноград Херсон Першотравенськ Львів
	5	6	7	8
1	226,85	257,05	300,96	354,88
2	187,34	214,21	246,24	290,31
3	165,69	192,79	221,62	261,28
4	153,84	178,94	205,57	242,31
5	145,93	169,70	194,83	229,62
6	138,29	160,76	181,34	213,66
7	132,61	154,12	170,96	201,37
8	127,12	147,70	161,23	189,86
9	121,79	141,48	152,10	179,06
10	114,99	133,55	141,50	166,54

**Продовження таблиці 1.5 – Тарифи на транспортування палетованого
вантажу**

Зона доставки палет		Тернопіль	Хмельницький
		Нововолинськ	Рівне
	Новоукраїнськ	Калуш	Луцьк
	Каховка	Івано- Франківськ	Чернівці
	9	10	11
1	373,34	375,49	395,52
2	305,36	310,12	326,73
3	274,82	276,68	289,12
4	254,83	256,76	268,54
5	241,45	243,46	254,85
6	224,60	230,61	241,60
7	211,61	221,05	231,77
8	199,47	210,08	222,25
9	188,08	198,08	211,42
10	174,89	184,19	198,26

Необхідно якомога раціональніше завантажити замовленою продукцією кожен палету, при цьому мінімізуючи кількість завантажених палет для замовлення дистрибутора. Тобто математична модель опису існуючої задачі матиме вигляд:

$$F(x) = \sum_{j=1}^N \left| \left(\sum_{i=1}^n c_i x_{ij} - 800 \right) \right| \rightarrow \min$$

$$N \rightarrow \min$$

$$\sum_{j=1}^N x_{ij} = a_i, i = \overline{1, n} \quad x_{ij} \geq 0 - \text{цїле,}$$

де

n – кількість назв продукції в замовленні;

N – кількість завантажених палет;

a_i – кількість замовлених ящиків i -го виду продукції;

$c_i = m_i n_i$ – вага одного ящика i -го виду продукції;

m_i – вага одиниці i -го виду продукції;

n_i – кількість одиниць i -го виду продукції в одному ящику;

2 СПЕЦІАЛЬНИЙ РОЗДІЛ.

2.1 Алгоритм пошуку Tabu.

Цей евристичний метод пошуку був запропонований Фредом Гловером в 1986 році для рішення різних комбінаторних проблем, залишається актуальним і в наш час. Світ наукової літератури налічує більше ста документів, присвячених методу Пошуку Табу (TS). У ряді випадків методи, побудовані на базі методу пошуку табу, дозволяли знайти рішення дуже близькі до оптимального й ставляться до числа найбільш ефективних, якщо не кращих для рішення складних проблем. TS надзвичайно популярний серед тих, хто зацікавлений у знаходженні ефективних рішень для великих комбінаторних проблем.

Виходячи з деяких своїх попередніх робіт, Фред Гловер запропонував в 1986 році новий підхід, який він назвав Пошук Табу. Цей метод дозволяв класичному методу LS переходити місцеві. У дійсності, багато елементів цього першого TS і деякі елементи пізніх TS розробок, були уведені Гловером в 1977 році, у тому числі короткострокова пам'ять – для того, щоб не допустити відкату останніх змін, і довгострокова пам'яті – розширити простір пошуку. Основний принцип TS складається виконанні LS, а коли цей алгоритм зустрічає місцевий оптимум дозволяти не поліпшуючі ходи. Відкат до раніше знайдених рішень не дозволяє робити пам'ять, що називається списком табу, у якій відбувається запис недавньої історії пошуку. Це є ключова ідея методу пошуку табу, що може бути пов'язана з концепціями штучного інтелекту.

Таким чином алгоритм пошуку із заборонами є розширенням класичних методів локального пошуку. Насправді, пошук із заборонами може бути розглянутий як просте поєднання локального пошуку із механізмами короткої пам'яті. Отже як і в методах локального пошуку основними елементами розв'язання задачі стають винайдення околу точки та функції фітнесу.

Простір пошуку в LS і TS методах - це простір всіх можливих рішень, які можуть бути отримані в ході пошуку. Наприклад, в CVRP простір пошуку може бути простим набором можливих рішень цієї проблеми, коли кожній точці в просторі пошуку відповідає набір маршрутів автотранспортних засобів задовольняючим всім зазначеним обмеженням. В цьому випадку визначення пошуку, представляється цілком природним, але це не завжди буває так.

Схоже визначення має окіл рішення кожній ітерації LS або TS, локальні перетворення, які можуть бути застосовані до поточного рішення, позначається S . Дане перетворення визначає ряд сусідніх рішень простору пошуку, позначається $N(S)$ (S окіл). Взагалі кажучи, $N(S)$ є підмножиною простору пошуку визначається: $N(S) =$ (рішення, отримані шляхом застосування одного локального переходу S).

Околом для завдання CVRP є перехід на кожній ітерації одного клієнта з поточного маршруту. Важлива особливість цих локальних структур полягає в тому, яким образом здійснюється вставка: можна скористатися випадковим вибором включення або виключення в найкращому положенні цільового шляху; поперемінно можна використовувати більше складні схеми включення, що передбачають часткову повторну оптимізацію цільового маршруту як GENI вставки (див. Жандро, Герца й Лапорте, 1994 рік). Перш ніж приступитися до подальшого, важливо підкреслити, що хоча ми говоримо, що ці структури зв'язані околом переходу для окремого клієнта, вони повинні містити всі можливі шляхи конфігурацій, які можуть бути отримані з поточного рішення шляхом переходу кожного клієнта.

Більш складні структури для визначення околу рішення задачі CVRP, наприклад, λ - обмін Османа (1993), дозволяють одночасно переміщення клієнтів у різних маршрутах і параметрів клієнтів між маршрутами. В Rego and Roucairol (1996), ходи визначаються виключенням замкнутого ланцюга, що є послідовністю скоординованих рухів клієнтів з одного маршруту на

інший. Структури околів пов'язані із заміною послідовностей ряду клієнтів між маршрутами. Такого роду околи, рідко використовуються для CVRP.

Як вже було згадано вище, заборони використовуються для попередження зациклення при подоланні локального оптимуму з використанням кроків, що не поліпшують поточний розв'язок. Основна ідея полягає в тому, що коли така ситуація виникає, потрібно щось вдіяти для того, щоб не допустити повернення процесу пошуку до розв'язків, отриманих на попередніх ітераціях. Це досягається накладенням заборон (tabus) на проведення кроків, які змінюють на протилежний ефект від останніх здійснених кроків.

Заборони зберігаються в структурах механізму короткої пам'яті процесу пошуку (так звані списки заборон), причому звичайно записується постійна та в достатній мірі обмежена частка інформації. В будь-якому випадку існує декілька можливостей збереження такої інформації. По-перше, можна зберігати повністю отримані розв'язки, що, звичайно, потребує багато ресурсів обчислювальної системи та робить не вигідним в сенсі обчислювальної складності процес перевірки, чи є потенційний крок забороненим. Тому раціональніше використовувати коротку пам'ять для запису інформації про найпростіші перетворення над поточним розв'язком та забороняти зворотні перетворення. Стосовно попереднього прикладу зрозуміло, що достатньо зберігати лише номер пункту можливого розташування підприємства, який був виключений з множини активних пунктів.

Стандартні списки заборон представлені як циклічні списки фіксованої довжини. Але такі списки не завжди можуть запобігти зацикленню, тому деякі автори пропонують змінювати довжину списку заборон протягом виконання пошуку. Іншим рішенням може бути випадкова генерація значення терміну заборони на кожному кроці, але такий метод вимагає дещо іншої схеми зберігання інформації про заборони.

Але механізм короткої пам'яті, використаний в пошуку із заборонами, також має свої недоліки. Наприклад, виконання деякого привабливого з точки зору змінення значення цільової функції перетворення буде заборонено на попередніх ітераціях навіть за відсутності загрози зациклення, або заборони можуть призвести до повільного розвитку процесу пошуку. В цьому випадку необхідним є використання таких особливостей обчислювального процесу, які б дозволяли анулювати встановлені раніше заборони, тобто треба встановити такі умови (*aspiration criteria*), при виконанні яких потрібно здійснювати наступний крок, навіть якщо він є забороненим.

Стратегія подолання заборон, що є найбільш простою та найчастіше вживаною (вона зустрічається майже в усіх реалізаціях алгоритму забороненого пошуку) полягає в наступному: якщо деякий крок, який є забороненим, генерує розв'язок з найкращим за весь час пошуку значенням цільової функції і цей розв'язок не був відвіданий раніше, то такий крок буде здійснений незважаючи на заборону. Більш ускладнені стратегії подолання заборон також було запропоновано та успішно реалізовано, але вони рідко використовуються. Ключова ідея цього механізму полягає в тому, що, якщо процесу пошуку не загрожує зациклення, заборонами можна знехтувати.

Стандартний алгоритм локального спуску зупиняється при досягненні локального оптимуму, тобто, якщо в околі поточного розв'язку немає елемента, що покращує значення цільової функції. На кожній ітерації перехід до наступного розв'язку відбувається заміною в поточному розв'язку одного (для околів *Swap*, *LK1*, *FM*, *FM1*) або декількох (окіл $k - Swap$) елементів на інші. Різницю між значенням цільової функції f в поточному розв'язку S та в будь-якому елементі S' околу $N(S)$ цього розв'язку будемо називати вигодою $adv(S, S')$ від здійснення кроку з S до S' , тобто

$$adv(S, S') = f(S) - f(S') \quad (2.1)$$

Таким чином стандартний алгоритм локального спуску дозволяє лише здійснення кроків, вигода від яких є в загальному випадку невід'ємною.

Можна змінити правило заміщення і дозволити алгоритму здійснювати кроки, в результаті яких значення цільової функції погіршується, обираючи серед усіх можливих кроків такий, що робить це мінімальним чином. Здійснення такого кроку дозволяє продовжити пошук після потрапляння до локального оптимуму. Може трапитися так, що на наступній ітерації ми знову потрапимо в той самий локальний оптимум (оскільки в околі поточного розв'язку вже буде допустимий розв'язок, здійснення кроку до якого покращує значення цільової функції), а це призведе до зациклення процесу пошуку (рис. 2.1).

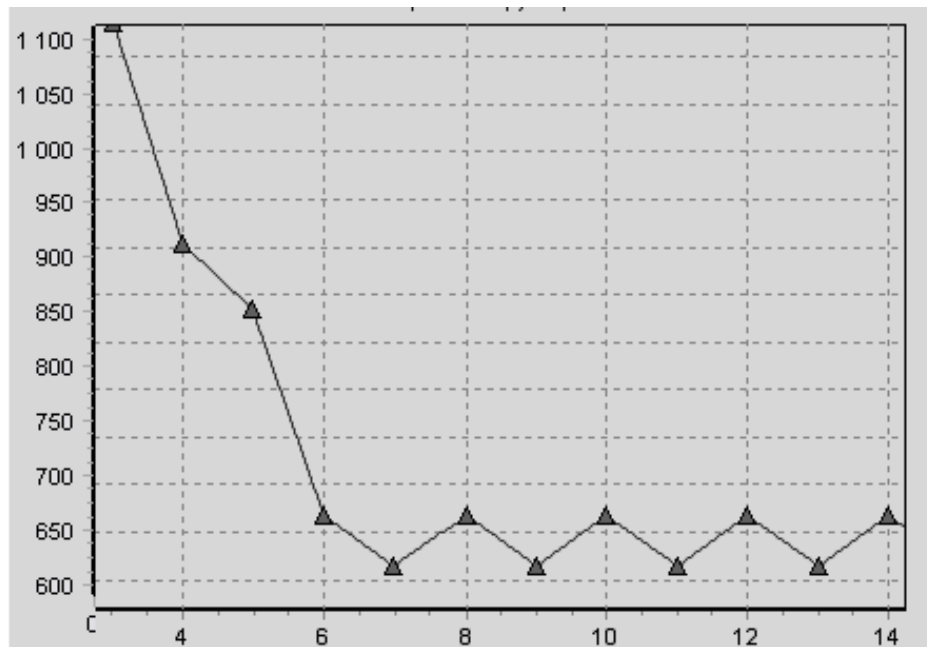


Рисунок 2.1 – Графік цільової функції в залежності від номеру ітерації.

В алгоритмі пошуку із заборонами дозволено робити кроки, вигода від здійснення яких є від'ємною. Для подолання зациклення використовується механізм короткої пам'яті, який не дозволяє повернення до отриманих на попередніх кроках розв'язків протягом певного часу. Для цього організуються списки заборон, в яких зберігаються дані про елементи, що були видалені з поточного розв'язку або додані до нього при виконанні чергового кроку. На них буде накладено заборону відповідно на додавання

до нового розв'язку та на видалення з поточного на певну кількість ітерацій. Цей параметр називається терміном перебування під заборонаю (tabu tenure) або довжиною короткої пам'яті, він задається користувачем перед початком обчислень та його значення є сталим протягом всього процесу пошуку.

Але при деяких значеннях довжини короткої пам'яті алгоритм пошуку із заборонами завжди буде досліджувати лише певний окіл локального оптимуму, із визначеною періодичністю повертаючись до нього. Графік цільової функції у такому випадку набуває вигляду, зображеного на рис. 2.2, що вказує на недоцільність використання алгоритму пошуку Tabu із застосуванням лише механізму короткої пам'яті, оскільки у цьому випадку може досліджуватися досить обмежена область простору пошуку.

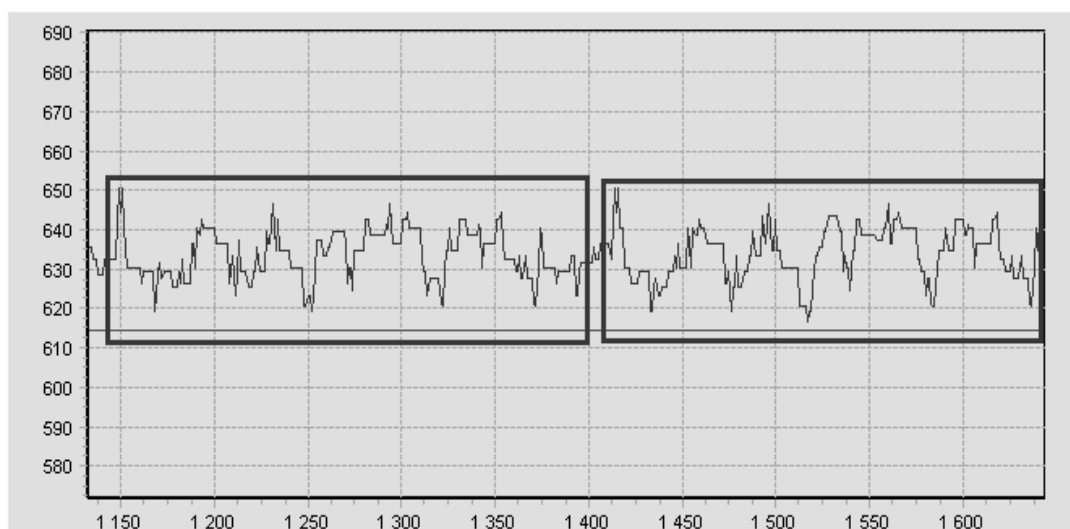


Рисунок 2.2 – Графік цільової функції при деяких значеннях довжини короткої пам'яті.

З вигляду цільової функції на рис. 2.2. випливає, що в процесі обчислень алгоритм пошуку із заборонами відвідує деякі розв'язки декілька разів, тому доцільно штучно зменшити вигоду від здійснення кроку до розв'язку, вже відвіданого раніше, щоб стимулювати перехід алгоритму до деякого нового розв'язку. Безперервна диверсифікація реалізується за допомогою впливу на обчислення вигоди від здійснення кроку з поточного

розв'язку S до деякого елементу S' околу $N(S)$. Цей метод базується на використанні механізмів довгої пам'яті, яка зберігає певну інформацію про всю попередню історію пошуку.

Отже для реалізації алгоритму пошуку з заборонами в контексті вирішення задачі оптимізації необхідно сформулювати механізми короткої та довгої пам'яті. Побудова початкового розв'язку, околу, визначення значення цільової функції залишаються аналогічними алгоритму локального пошуку.

Механізм короткої пам'яті (рис. 2.3) необхідний для запобігання зациклення локального пошуку при дозволі переходів, що не поліпшують цільову функцію.

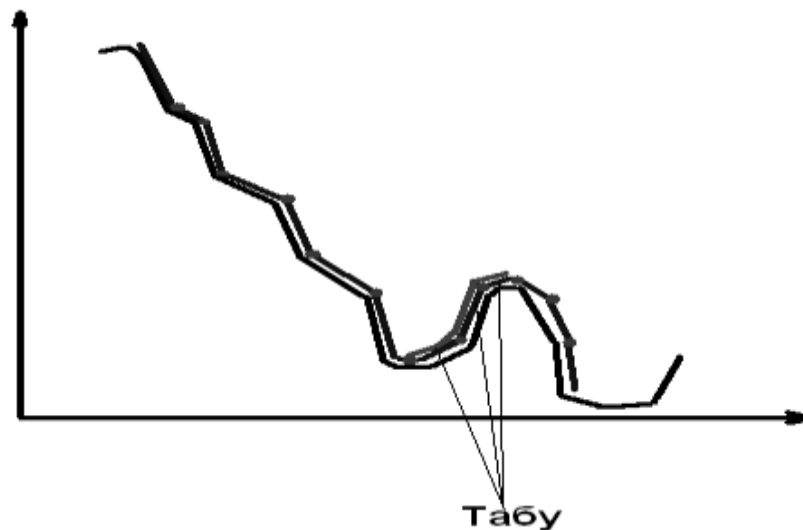


Рисунок 2.3 – Механізм короткої пам'яті.

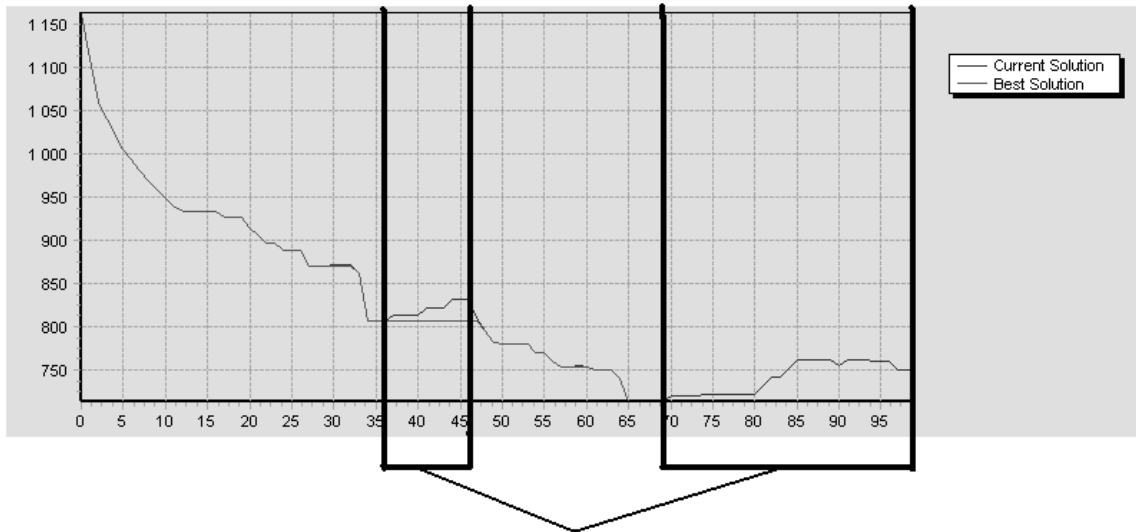
Для побудови механізму короткої пам'яті скористаємося варіантом моделі задачі, що зображений на Рис.2.3. В цьому випадку коротка пам'ять буде представлена у вигляді вектору з цілими числами p_{ijk} . Умова

$$p_{ijk} = q > 0 \quad (2.2)$$

означає те, що значення розв'язку x_{ijk} не може змінюватися протягом q кроків алгоритму.

На кожному кроці значення кожного табу зменшується на одиниці. Це означає, що через певну кількість кроків зміна значення x_{ijk} вже не буде заборонене, що забезпечує більш повний обхід області пошуку.

Початкове значення кількості кроків табу визначається першим параметром алгоритму.



Області дії короткої пам'яті
Рисунок 2.4 – Побудова механізму довготривалої пам'яті

Механізм довготривалої пам'яті необхідний для запобігання зацікнення в більш довготривалих термінах. Коротка пам'ять може запобігти повернення в межах одного локального оптимуму, та вона має занадто мало інформації для визначення повернення розв'язку після проходження декількох локальних оптимумів (рис. 2.4):

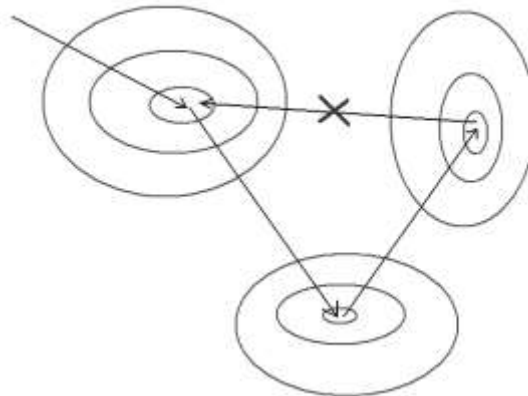


Рисунок 2.5. – Зацікнення між декількома оптимумами.

Зацікнення такого роду дозволяє подолати механізм довготривалої пам'яті. Він побудований на принципі штрафування зміни значення x_{ijk} .

Отже, якщо ми в точку, то зміна значення штрафується на сталу величину. Значення цієї величини є другим параметром роботи алгоритму.

Вибір цього параметру досить суттєво впливає на роботу алгоритму. Мале значення може призвести до досить довгого зациклення алгоритму між локальними оптимумами або в межах одного екстремуму. Вибір великого значення призводить до зменшення покриття області пошуку шляхом відкидання ліпших рішень задля неповторюваних.

Роботу механізму довгострокової пам'яті можна побачити на рис. 2.6:

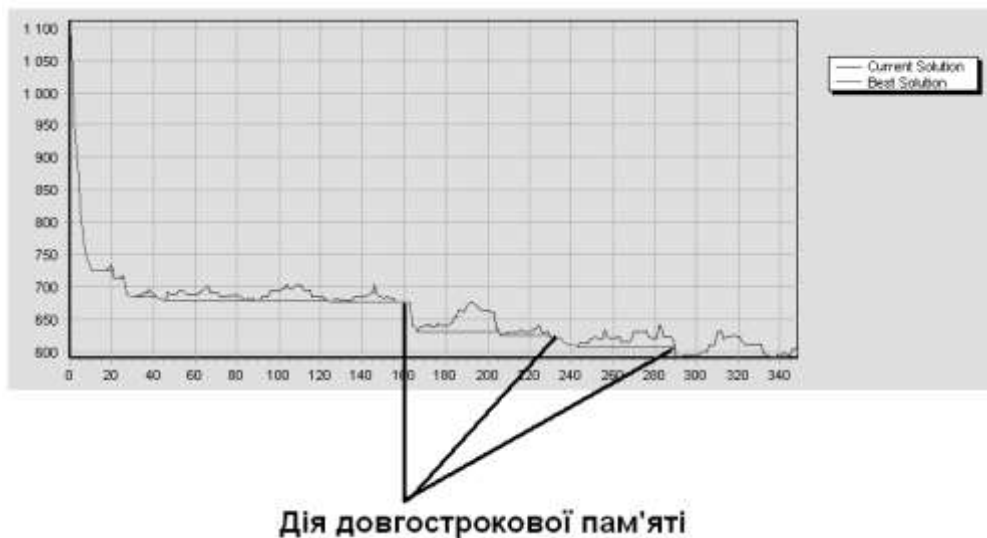


Рисунок 2.6 – Робота механізму довгострокової пам'яті.

Вибір параметрів методу пошуку з заборонами є нетривіальною задачею. Під кожную конкретну ситуацію існує пара параметрів, за яких для отримання розв'язку, близького до оптимального, знадобиться досить короткий час.

Вибір параметрів короткої та довготривалої пам'яті взаємопов'язані. При виборі малого значення короткої пам'яті значення довгої має бути збільшено, оскільки в цьому разі дуже ймовірно довге зациклення алгоритму в межах декількох локальних оптимумів (а скоріше за все в межах одного екстремуму). Та велике значення другого параметру може погіршити якість роботи алгоритму знизивши покриття області пошуку.

Велике значення параметру короткої пам'яті може заблокувати перехід алгоритму в досить перспективні області і не дозволити відшукати глобальний оптимум задачі.

Стратегію подолання заборон призначено для надання дозволу алгоритму пошуку за певних умов здійснювати заборонені кроки. Такою умовою є можливість отримання значення цільової функції, яке виявиться найкращим за весь час обчислень. Нехай на деякій ітерації f^* є мінімальним значенням цільової функції на поточний момент. Якщо перехід від поточного розв'язку S до деякого елемента S' околу $N(S)$ на цій ітерації є забороненим у відповідності з даними структур короткої пам'яті, але $f(S') < f^*$, то такий крок може бути здійсненим. Наведемо схему роботи алгоритму пошуку табу для вирішення оптимізаційної задачі.

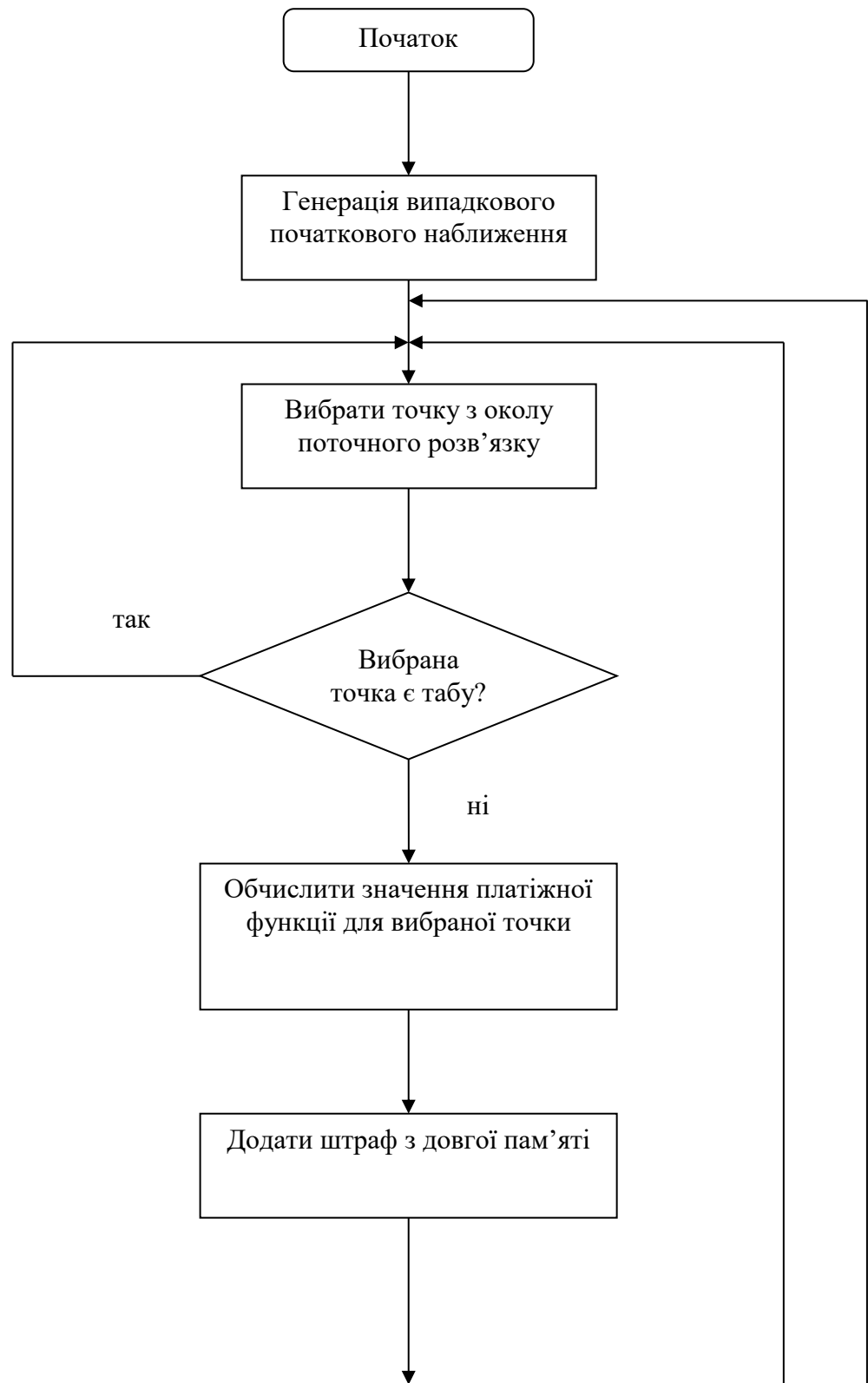
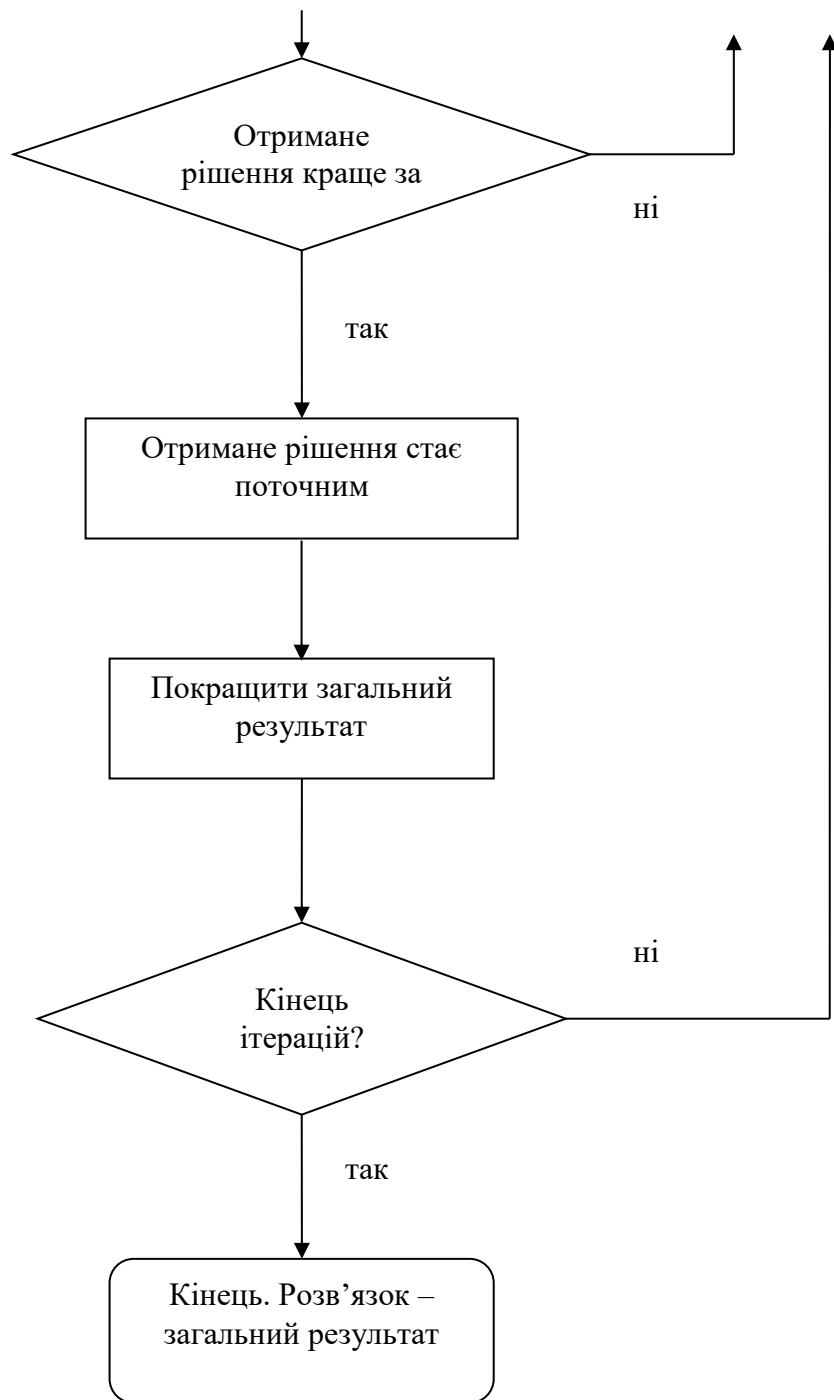


Рисунок 2.7 – Схема роботи алгоритму Tabu Search.



Продовження рисунку 2.7 – Схема роботи алгоритму Tabu Search.

2.2 Дослідження розв'язання поставленої задачі різними методами.

Дослідивши основні методи оптимізації, що найбільш точно підходять для вирішення поставленої задачі, обрані жадібний алгоритм і метод гілок та меж, як альтернативні методи розробленому пошуку із заборонами.

Як середовище розробки було обрано Python, як інтегроване середовище розробки — Visual Studio, тобто інтегрований набір інструментів, що полегшують і значно прискорюють процес розробки готового додатку.

За допомогою Python можна створювати додатки практично для будь-якої області сучасних комп'ютерних технологій: бізнес-додатки, ігри, мультимедіа, бази даних. При цьому додатки можуть бути як простими, так і дуже складними, залежно від поставленої задачі.

Слід зазначити, що для всіх обраних методів необхідно виконати попередній етап обчислень, що полягає у розробці механізму розподілу великого замовлення однієї позиції (більше ніж 800 кг) шляхом додавання в замовлення фіктивних позицій по 800 кг. Алгоритм полягає в наступному:

- Знаходимо SKU, замовлення якої перевищує 800 кг;
- Ділимо дане замовлення на m повних замовлень по 800 кг і додаємо m фіктивних SKU в кінець замовлення;
- В початковому замовленні позиції маємо залишок від ділення на 800.

2.2.1 Розробка жадібного алгоритму.

Розроблено типовий жадібний алгоритм, здатний шукати глобальні оптимуми функцій. Алгоритм працює за схемою:

- Знаходимо максимальну вагу замовлення однієї позиції;
- Завантажуємо першу палету: $Sum = max$;
- Серед інших замовлених позицій знаходимо з максимальною вагою таку, що менше, ніж вільне місце в першій палеті: $if\ max <\ razn\ then\ Sum = Sum + max$
- Переходимо до наступної ітерації поки сума замовлення не буде дорівнювати сумі всіх завантажених палет: $i = i+1$
- Переходимо до оформлення наступного заказу.

Лістинг програми вищезазначеного алгоритму міститься в Додатку Д.

2.2.2 Розробка алгоритму методу гілок та меж.

Алгоритм полягає в наступному:

- Впорядковуємо в напрямку спадання всі замовлені позиції;
- З розрахунків поступово вилучаємо максимально заповнені позиції:
 - а) Спочатку ті, що дорівнюють 800 кг, таким чином заповнюємо перші m палет;
 - б) Далі ті, що не вміщують одну мінімально замовлену позицію з тих, що залишилися, заповнюючи наступні $(m + 1) \dots (m + n)$ палети;
 - в) Потім вилучаємо ті, що не вміщують дві мінімально замовлені позиції з тих, що залишилися і т. д.;
- Між кроками всі замовлені позиції, що залишились невилученими впорядковуємо в напрямку спадання.
- Повторюємо вищезазначені пункти поки сума замовлення не буде дорівнювати 0;
- Переходимо до оформлення наступного заказу.

Лістинг програми вищезазначеного алгоритму міститься в Додатку Е.

2.2.3 Розробка алгоритму пошуку із заборонами.

Розроблено алгоритм пошуку із заборонами. Дія алгоритму починається з пунктів:

- випадкового завантаження першої палети будь-якою із зазначеної продукції в замовленні дистрибутора;
- вищезазначене завантаження приймаємо за 1й локальний оптимум;
- відображуємо наявність чи відсутність і-го виду продукції на палеті за допомогою бінарного вектора:

$$H_{ij} = \begin{cases} 1, & \text{якщо } i\text{-й вид продукції завантажено на } j\text{-у палету} \\ 0, & \end{cases}$$

Де H_{ij} показує стан замовленої продукції.

- змінюємо значення вектору стану продукції на фліп-окіл (видаляємо останній покладений вид продукції з палети), отримали наступне оптимальне значення стану;
- значення короткої пам'яті встановлюємо 20, а довгої за допомогою штрафної функції Shtraf:

$$\text{Sum} = \text{Sum} * (1 - \text{Shtraf} / 30)$$

- кількість ітерацій встановлюємо 100;

Лістинг програми вищезазначеного алгоритму міститься в Додатку Є.

2.3 Порівняння розроблених та протестованих методів вирішення задачі оптимізації.

Отримали результати роботи розробленого жадібного алгоритму, методу гілок і меж та алгоритму пошуку із заборонами (Табл. 2.1):

Таблиця 2.1 – Порівняння результатів роботи алгоритмів.

№ п. п. замовлення	ВСЬОГО замовлення, кг	Кількість завантажених палет		
		Tabu Search	Метод гілок та меж	Жадібний алгоритм
1	73 816	93	93	98
2	93 361	119	118	126
3	46 011	58	58	60
4	34 750	45	44	46
5	92 194	116	116	124
6	89 547	113	113	119
7	42 803	54	54	56
8	20 152	27	27	28
9	82 516	104	104	109
10	26 354	34	34	35
11	95 290	120	120	126
12	65 653	86	83	86
13	91 323	115	115	120
14	87 316	110	110	114
15	29 587	38	38	39
16	95 673	121	121	124
17	60 342	77	77	80
18	47 880	61	61	64
19	92 011	116	116	121
20	57 254	73	72	75
21	88 273	112	111	116
22	42 565	54	54	56
23	85 593	108	108	112
24	84 136	106	106	112
25	21 233	27	27	27
26	84 277	107	106	112
27	93 409	118	118	124
28	90 565	114	114	121
29	33 211	42	42	44
30	94 649	120	119	124
31	85 837	108	108	113
32	61 015	78	78	87
33	28 659	39	39	41
34	84 670	107	107	110

Продовження таблиці 2.1 – Порівняння результатів роботи алгоритмів.

№ п. п. замовлення	ВСЬОГО замовлення, кг	Кількість завантажених палет		
		Tabu Search	Метод гілок та меж	Жадібний алгоритм
35	91 053	115	115	119
36	89 215	113	113	118
37	88 760	112	112	117
38	23 747	31	30	31
39	90 964	115	115	119
40	49 464	63	63	65
41	85 063	107	107	113
42	90 006	114	114	119
43	55 440	71	71	76
44	87 329	110	110	115
45	92 337	117	117	123
46	21 809	29	28	30
47	94 073	119	119	125
48	84 007	106	106	109
49	31 083	40	40	41
50	91 851	116	116	120

Із вищезазначеної таблиці видно, що методи гілок та меж і Tabu Search пропонують ефективно завантаження палет, так як їх кількість менша за розраховану жадібним методом (рис. 2.8). Різниця становить 4 палети що суттєво оптимізує завантаження замовлення, виходячи з того, що ми економимо 3200 кг. ваги продукції.

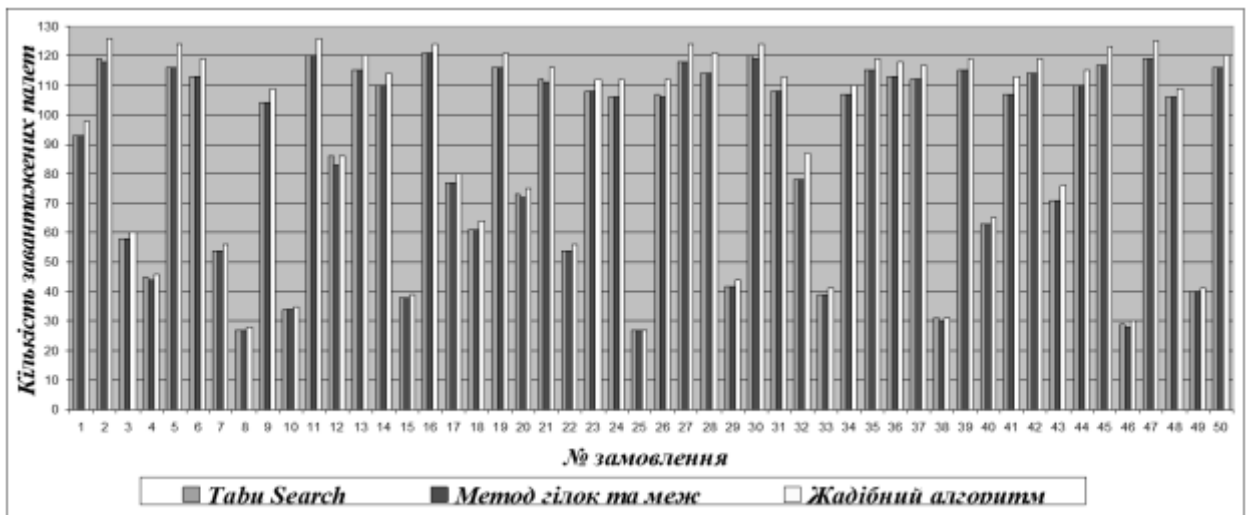


Рисунок 2.8 – Порівняння оптимальності розміщення замовленої продукції у транспортні засоби (для 50-ти замовлень).

Також можна спостерігати за ефективним завантаженням кожної і-ї палети (табл. 2.2) та на графіку спостереження за ходом завантаження всіма методами для 40-го замовлення, що вибране в якості прикладу для ілюстрації (Додаток Ж).

Таблиця 2.2 – Порівняння ефективності завантаження палет різними методами.

№ замов л.	ВСЬОГО замовлен ня, кг	Середнє вільне місце в палеті, кг			Середня завантаженість палети, %		
		Tabu Search	Метод гілок та меж	Жадібний алгоритм	Tabu Search	Метод гілок та меж	Жадібний алгоритм
1	73 816	6	6	47	99%	99%	94%
2	93 361	15	9	59	98%	99%	93%
3	46 011	7	7	33	99%	99%	96%
4	34 750	28	10	45	97%	99%	94%
5	92 194	5	5	56	99%	99%	93%
6	89 547	8	8	48	99%	99%	94%
7	42 803	7	7	36	99%	99%	96%
8	20 152	54	54	80	93%	93%	90%
9	82 516	7	7	43	99%	99%	95%
10	26 354	25	25	47	97%	97%	94%
11	95 290	6	6	44	99%	99%	95%
12	65 653	37	9	37	95%	99%	95%

**Продовження таблиці 2.2 – Порівняння ефективності завантаження палет
різними методами.**

13	91 323	6	6	39	99%	99%	95%
14	87 316	6	6	34	99%	99%	96%
15	29 587	21	21	41	97%	97%	95%
16	95 673	9	9	28	99%	99%	96%
17	60 342	16	16	46	98%	98%	94%
18	47 880	15	15	52	98%	98%	94%
19	92 011	7	7	40	99%	99%	95%
20	57 254	16	5	37	98%	99%	95%
21	88 273	12	5	39	99%	99%	95%
22	42 565	12	12	40	99%	99%	95%
23	85 593	7	7	36	99%	99%	96%
24	84 136	6	6	49	99%	99%	94%
25	21 233	14	14	14	98%	98%	98%
26	84 277	12	5	48	98%	99%	94%
27	93 409	8	8	47	99%	99%	94%
28	90 565	6	6	52	99%	99%	94%
29	33 211	9	9	45	99%	99%	94%
30	94 649	11	5	37	99%	99%	95%
31	85 837	5	5	40	99%	99%	95%
32	61 015	18	18	99	98%	98%	88%
33	28 659	65	65	101	92%	92%	87%
34	84 670	9	9	30	99%	99%	96%
35	91 053	8	8	35	99%	99%	96%
36	89 215	10	10	44	99%	99%	95%
37	88 760	7	7	41	99%	99%	95%
38	23 747	34	8	34	96%	99%	96%
39	90 964	9	9	36	99%	99%	96%
40	49 464	15	15	39	98%	98%	95%
41	85 063	5	5	47	99%	99%	94%
42	90 006	10	10	44	99%	99%	95%
43	55 440	19	19	71	98%	98%	91%
44	87 329	6	6	41	99%	99%	95%
45	92 337	11	11	49	99%	99%	94%
46	21 809	48	21	73	94%	97%	91%
47	94 073	9	9	47	99%	99%	94%
48	84 007	7	7	29	99%	99%	96%
49	31 083	23	23	42	97%	97%	95%
50	91 851	8	8	35	99%	99%	96%

По результатам таблиці можна побачити, що при виконанні завантаження жадним методом ми отримуємо не повністю заповнені палети (6% вільного місця, 45 кг), а методи гілок та меж і Tabu Search дають досить кращі результати (1-2% вільного місця на палеті, 12-15 кг).

Таким чином виконуючи оцінку методів по отриманим і протестованим результатам можемо зробити висновки про оптимальність використання методів (табл. 2.3), а саме те, що методи гілок і меж та Tabu Search дають приблизно однакові результати, але значно кращі за результати, отримані при використанні жадного алгоритму. При цьому метод пошуку із заборонами має значно кращі показники за часом, ніж метод гілок і меж.

Таблиця 2.3 – Порівняння ефективності використання методів.

<i>Середня кількість завантажених палет</i>			<i>Середня завантаженість палети, %</i>			<i>Математичне очікування часу виконання розрахунків, с</i>		
<i>Tabu Search</i>	<i>Метод гілок та меж</i>	<i>Жадібний алгоритм</i>	<i>Tabu Search</i>	<i>Метод гілок та меж</i>	<i>Жадібний алгоритм</i>	<i>Tabu Search</i>	<i>Метод гілок та меж</i>	<i>Жадібний алгоритм</i>
88	88	92	98%	99%	94%	20,48	40,16	10,44

2.4. Оптимізація налаштування роботи алгоритму Tabu Search.

Серед критеріїв зміни налаштувань обрані:

- зміна кількості ітерацій (від 100 до 1000);
- зміна довжини короткострокової пам'яті (від 20 до 80);
- зміна величини штрафу (від $Shtraf / 50$ до $Shtraf / 30$).

При дослідженні даного питання, виявлено, що найсильніший і найефективніший вплив на хід розв'язання поставленої задачі оптимізації має зміна довжини короткострокової пам'яті (Рис. 2.9). Хоч кількість завантажених палет і не змінюється, але суттєво оптимізується ефективність їх завантаження, що добре видно на нижчезазначеному графіку:

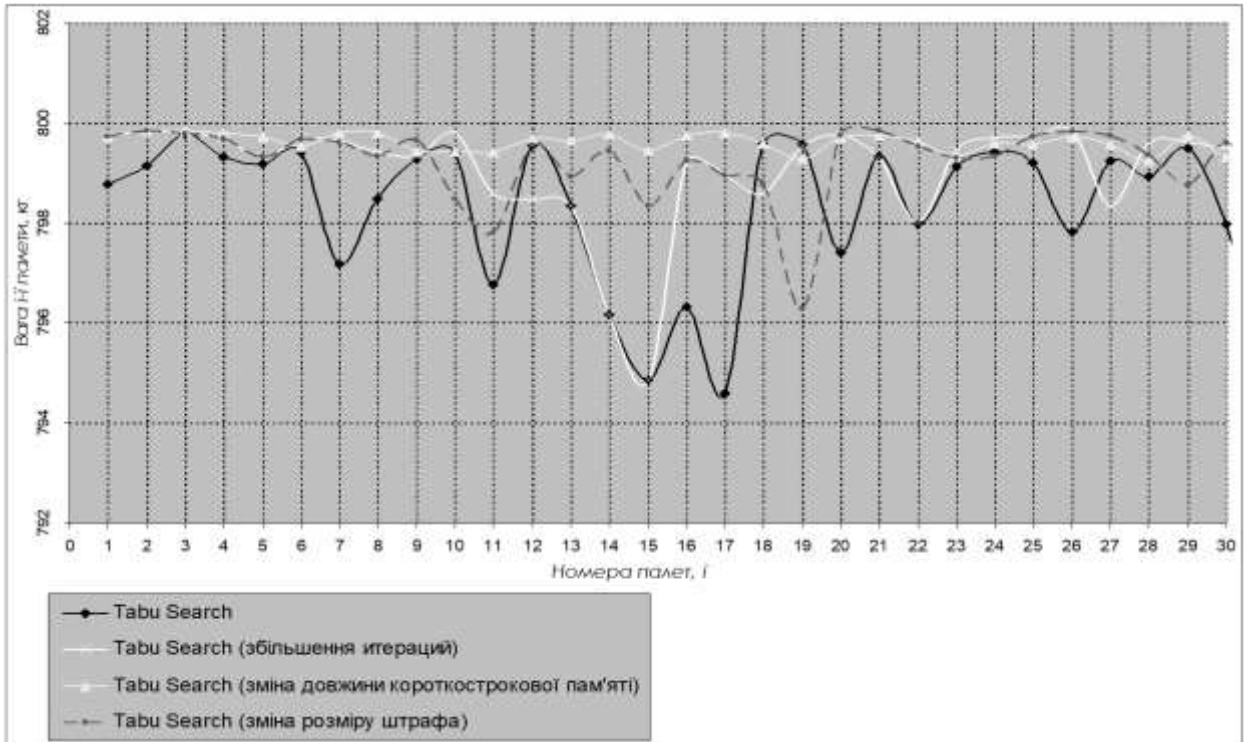


Рисунок 2.9 - Порівняння результатів роботи алгоритма Tabu Search, в залежності від змін налаштування.

3 ПРОЕКТНИЙ РОЗДІЛ.

3.1 Актуальність розробки проектів, інновацій та бізнес-планування. Сучасні системи розробки проектів

Останнім часом в багатьох компаніях росте потреба у веденні проектної діяльності. Це обумовлено об'єктивними причинами - впливом зовнішнього і внутрішнього середовища бізнесу. При цьому навіть в крупних проектний-орієнтованих компаніях можуть виникати складнощі з координуванням всіх проектних ресурсів компанії, визначенням і розробкою загальної методології проектного управління, централізованим моніторингом всіх процесів і навчанням співробітників.

Проекти здійснюються в повсякденному житті, в різних видах діяльності і мають між собою цілий ряд загальних ознак, що роблять їх проектами:

- 1 Направлені на досягнення конкретної мети;
- 2 Включають координоване виконання взаємозв'язаних дій;
- 3 Мають обмежену протяжність в часі, з певним початком і кінцем;
- 4 Вони певною мірою неповторювані і унікальні.

У загальному випадку, саме ці чотири характеристики відрізняють проекти від інших видів діяльності.

Проекти націлені на отримання певних результатів - іншими словами, вони направлені на досягнення мети. Саме ці цілі є рушійною силою проекту, і всі зусилля по його плануванню і реалізації робляться для того, щоб ці цілі були досягнуті. Проект зазвичай припускає цілий комплекс взаємозв'язаних цілей. Наприклад, основною метою проекту, пов'язаного з комп'ютерним програмним забезпеченням, може бути розробка інформаційної системи управління підприємством. Проміжними цілями можуть бути розробка бази даних, розробка математичного і програмного забезпечення, тестування системи.

У розробці бази даних, у свою чергу, також можуть бути виділені цілі нижчого рівня - розробка логічної структури бази даних, реалізація бази даних за допомогою СУБД, завантаження даних і так далі. Той факт, що проекти орієнтовані на досягнення мети, має величезний внутрішній сенс для управління ними. Перш за все, він припускає, що важливою межею управління проектами є точне визначення і формулювання цілей, починаючи з вищого рівня, а потім поступово опускаючись до найбільш деталізованих цілей і завдань. Крім того, звідси витікає, що проект можна розглядати як переслідування ретельно вибраних цілей, і що просування проекту вперед пов'язане з досягненням мети все більш високого рівня, поки нарешті не досягнута кінцева мета.

Проекти складні вже по самій своїй суті. Вони включають виконання численних взаємозв'язаних дій. В окремих випадках ці взаємозв'язки достатньо очевидні (наприклад, технологічні залежності), в інших випадках вони мають тоншу природу. Деякі проміжні завдання не можуть бути реалізовані, поки не завершені інші завдання; деякі завдання можуть здійснюватися тільки паралельно, і так далі. Якщо порушується синхронізація виконання різних завдань, проект може бути поставлений під загрозу. Якщо небагато задуматися над цією характеристикою проекту, стає очевидно що проект - це система, тобто ціле, таке, що складається з взаємозв'язаних частин, причому система динамічна, і, отже, що вимагає особливих підходів до управління.

Проекти виконуються протягом кінцевого періоду часу. Вони тимчасові. У них більш менш чітко виражені почало і кінець. Проект закінчується, коли досягнуті його основні цілі. Значна частина зусиль при роботі з проектом направлена саме на забезпечення того, щоб проект був завершений в намічений час. Для цього готуються графіки, що показують час почала і закінчення завдань, що входять в проект.

Відмінність проекту від виробничої системи полягає в тому, що проект є одноразовою, не циклічною діяльністю. Серійний же випуск продукції не

має наперед певного кінця в часі і залежить лише від наявності і величини попиту. Коли зникає попит, виробничий цикл кінчається. Виробничі цикли в чистому вигляді не є проектами. Проте, останнім часом проектний підхід все частіше застосовується і до процесів, орієнтованих на безперервне виробництво. Наприклад, проекти збільшення виробництва до вказаного рівня в перебігу певного періоду, виходячи із заданого бюджету, або виконання певних замовлень, що мають договірні терміни постачання. Проект як система діяльності існує рівно стільки часу, скільки його потрібно для отримання кінцевого результату. Концепція проекту, проте, не суперечить концепції фірми або підприємства і цілком сумісна з нею. Навпаки, проект часто стає основною формою діяльності фірми.

Проекти - заходи до певної міри неповторні і одноразові. Разом з тим, ступінь унікальності може сильно відрізнятись від одного проекту до іншого. Якщо ви займаєтеся будівництвом котеджів і зводите двадцятий по рахунку однотипний котедж, ступінь унікальності вашого проекту достатньо невеликий. Базові елементи цього будинку ідентичні елементам попередніх дев'ятнадцяти, які ви вже побудували. Основні ж джерела унікальності, проте, можуть бути закладені в специфіці конкретної виробничої ситуації - в розташуванні будинку і навколишнього ландшафту, в особливостях постачань матеріалів і що комплектують, в нових субпідрядниках. З іншого боку, якщо ви розробляєте унікальний прилад або технологію, ви, безумовно, маєте справу із завданням вельми унікальної. Ви робите те, що ніколи раніше не робилося. І оскільки минулий досвід може в даному випадку лише обмежено підказувати вам, чого можна чекати при виконанні проекту, він повний ризику і невизначеності.

Термін «проект» не є принципово новим для вітчизняної економіки. Під «проектом» було прийнято розуміти документально оформлений план споруди (конструкції). У англійській економічній літературі це поняття прийнято позначати терміном «design». Перехід до ринкових відносин привніс в економічне життя безліч нових термінів і понять. Термін «проект»

також отримав ширше тлумачення і за своїм змістом наблизився до англійського «project», який охоплює весь процес від появи ідеї, її розробки, реалізації до отримання результату. Таке тлумачення є декілька спрощеним. Точніше відображає зміст інвестиційного проекту наступне визначення:

Під проектом розуміється система сформульованих в його рамках цілей, створюваних або таких, що модернізуються для їх досягнення фізичних об'єктів (будівель, споруд, виробничих комплексів), технологічних процесів; технічній і організаційній документації для них, матеріальних, фінансових, трудових і інших ресурсів, а також управлінських рішень і заходів щодо їх виконання.

У ряді галузей, специфіка яких вимагає створення надскладних об'єктів, для досягнення поставленої мети потрібна сукупність самостійних проектів, які об'єднуються в програму.

Поняття «інвестиційний проект» також має два тлумачення: розглядається як діяльність, що припускає здійснення комплексу заходів, направлених на досягнення поставленої мети; та трактується як система організаційно-правових і розрахунковий-фінансових документів, що містять обґрунтування ефективності і можливості реалізації проекту, необхідних для здійснення комплексу заходів, направлених на досягнення поставленої мети.

Комерційна і підприємницька діяльність в середовищі науково-технічного комплексу стала одним з провідних напрямів в індустріально розвинених і постіндустріальних системах. У зв'язку з цим розробка бізнес-планів для інноваційних проектів (тобто розвитку науково-технічних комплексів, впровадження конкретних інноваційних рішень і т.п.) повинна спиратися на загальні методичні принципи. При цьому при підготовці бізнес-планів необхідно зважати на специфіку інноваційних проектів, їх високий економічний, інтелектуальний і соціальний потенціали.

Основним критерієм актуальності розробки бізнес-плану є рівень опрацювання наукової ідеї і її близькість до можливості комерційної реалізації. Якщо зазвичай об'єктом розробки бізнес-плану служить

конкретний інвестиційний проект або стратегія розвитку фірми, то в даному завданні таким об'єктом є один з процесів – науково-технічний прогрес, точніше, одна з складових цього процесу – конкретна інновація. У зв'язку з цим додатково виникає методологічно і технічно складна проблема виділення із загального потоку результатів і витрат тих з них, які безпосередньо пов'язані з даною інновацією.

Специфікою багатьох інноваційних проектів, що розробляються в даний час, є відсутність наперед певних споживачів. Пошук інвестора для венчурного проекту вимушує розробників особливо ретельно підходити до обґрунтування проекту.

Для інноваційних розробок, як правило, характерний високий ступінь ризику. Облік і ретельна оцінка чинників, здатних перешкодити здійсненню проекту, повинні бути здійснені в бізнес-плані, при підготовці відповідного розділу повинні використовуватися думки незалежних експертів в даній області. Можливе використання спеціалізованого комп'ютерного програмного модуля Projekt Risk.

При підготовці бізнес-плану інноваційного проекту слід звернути особливу увагу на патентну «чистоту» впроваджуваної розробки, описати перспективи її розвитку, зокрема можливість завоювання ринку спорідненої продукції або створення принципово нового товару (послуги).

Планування взагалі як найбільше досягнення людського розуму є практично природним і необхідним елементом поведінки людей і в побуті і на виробництві. Праця людини завжди припускає усвідомлену цілеспрямовану діяльність по досягненню бажаного результату, образ (план, проект) якого він ідеально має на самому початку роботи, у тому числі і в бізнесі. Тому планування можна розглядати як проектування процесу створення з безладу порядку і підвищення його ступеня, який зменшує невизначеність або ентропію в даному випадку системи підприємництва.

Наявність добре розробленого плану дозволяє активно розвивати підприємництво, привертати інвесторів, партнерів і кредитні ресурси. Він також дає можливість:

- Визначити шляхи і способи досягнення поставленої мети;
- Максимально використовувати конкурентні переваги підприємства;
- Запобігти помилковим діям;
- Відстежити нові тенденції в економіці, техніці і технології і використовувати в своїй діяльності;
- Довести і демонструвати обґрунтованість, надійність і реалізовується проекту;
- Пом'якшити вплив слабких сторін підприємства;
- Визначити потребу в капіталі і грошових коштах;
- Своєчасно прийняти захисні заходи проти різного роду ризиків;
- Повніше використовувати інновації в своїй діяльності;
- Об'єктивніше оцінювати результати виробничої і комерційної діяльності підприємства;
- Обґрунтувати економічну доцільність напряму розвитку підприємства (стратегія проекту).

Одночасно план є керівництвом до дії і виконання. У міру реалізації і зміни обставин план може уточнюватися шляхом коректувань відповідних показників.

Постійне приведення бізнес-плану у відповідність з новими умовами дає можливість використовувати його як інструмент оцінки практичних результатів роботи підприємства.

Розробка конкретного бізнес-плану в більшості випадків процес творчий, заснований на обліку не тільки загальних закономірностей бізнесу, але і специфічних умов, особистого досвіду і знань підприємця. Залежно від реальних обставин підготовка бізнес-плану може бути організована різним чином:

- 1 Самим підприємцем, що має відповідний досвід;
- 2 Командою підприємця, яка надалі реалізовуватиме проект;
- 3 Сторонньою спеціалізованою компанією за певну плату.

У кожному конкретному випадку будь-який план має свою специфіку як в предметі діяльності і порядку його здійснення, так і в якісному і особливо кількісному відношенні - цінах, нормах витрат ресурсів, ринках збуту і інших показниках.

У найзагальнішому випадку план - це образ чого-небудь, модель бажаного майбутнього або система заходів, направлена на досягнення поставленої мети і завдань. Бізнес-план, як один найбільш поширених в даний час видів планів, є:

- 1) Робочий інструмент підприємця для організацій своєї роботи;
- 2) Розгорнену програму (раціонально організованих заходів, дій) здійснення бізнес-проекту, що передбачає оцінку витрат і доходів;
- 3) Документ, що характеризує основні сторони діяльності і розвитку підприємства;
- 4) Результат дослідження і обґрунтування конкретного напрямку діяльності фірми на певному ринку.

Підприємство може мати одночасно декілька бізнес-планів, в яких ступінь деталізації обґрунтувань може бути різним. У малому підприємстві бізнес-план і план підприємства можуть співпадати і за об'ємом і за змістом.

Будь-який бізнес-план повинен давати переконливі відповіді для самого підприємця і його можливих партнерів, принаймні, на п'ять основних питань.

Бізнес-планування, як необхідний елемент управління виконує в системі підприємницької діяльності ряд найважливіших функцій, серед яких найбільше значення мають наступні:

- 1 Ініціація - активізація, стимулювання і мотивація позначених дій, проектів і операцій;

2 Прогнозування - передбачення і обґрунтування бажаного стану фірми в процесі аналізу і обліку сукупності чинників;

3 Оптимізація - забезпечення вибору допустимого і якнайкращого варіанту розвитку підприємства в конкретному соціально-економічному середовищі;

4 Координація і інтеграція - облік взаємозв'язку і взаємозалежності всіх структурних підрозділів компаній з орієнтацією їх на єдиний загальний результат;

5 Безпека управління - забезпечення інформацією про можливі ризики для своєчасного вживання попереджуючих заходів по зменшенню або запобіганню негативним наслідкам;

6 Впорядкування - створення єдиного загального порядку для успішної роботи і відповідальність;

7 Контроль - можливість оперативного відстеження виконання плану, виявлення помилок і можливого його коректування;

8 Виховання і навчання - сприятлива дія зразків раціонально спланованих дій на поведінку працівників і можливість навчання їх, у тому числі і на помилках;

9 Документування - представлення дій в документальній формі, що може бути доказом успішних або помилкових дій менеджерів фірми;

При розробці бізнес-планів необхідно дотримувати основоположні принципи планування, які створюють передумови для успішної діяльності підприємства в конкретному економічному середовищі: необхідність, безперервність, еластичність і гнучкість, єдність і повнота (системність), точність і деталізація, економічність, оптимальність, зв'язок рівнів управління, участь, холізм (поєднання координації і інтеграції).

Сьогодні на українському ринку автоматизованих систем управління проектами найбільш активні декілька гравців: лідер світового ринку ПО даного класу - Primavera Systems, компанія Microsoft з своїм пакетом Project і

вітчизняний розробник Spider Project Management Technologies, продукт якого відображає локальну специфіку.

Не дивлячись на різноманітність систем, представлених на ринку АСУПП, всі вони мають наступні складові:

- Засоби для календарно-мережевого планування;
- Засоби для вирішення приватних завдань (передпроектний аналіз, розробка бюджетів, аналіз ризиків, управління контрактами, часом, фінансами);
- Засоби для спрощеного доступу до проектних даних;
- Засоби для організації комунікацій;
- Засоби для інтеграції з іншими застосуваннями.

Засоби календарно-мережевого планування можуть бути представлені у вигляді модулів для складання розкладів і комплексних систем. Перші орієнтовані на тих керівників, яким час від часу доводиться планувати прості проекти. Це ПО дозволяє задавати взаємозв'язку між роботами, будувати діаграми Ганта, розраховувати критичний шлях, спрощено оцінювати завантаження ресурсів, вартість проекту і так далі. На жаль, локалізованих продуктів даного класу на вітчизняному ринку поки не представлено.

Комплексні системи коштують значно дорожче. І якщо попередній клас ПО орієнтований на менеджерів практично будь-якої компанії, то комплексні системи розраховані саме на проектний-орієнтованих компаній, що служать. Вони включають не тільки професійні інструменти для планування, аналізу і контролю над виконанням проектів, але і всі необхідні засоби для організації ефективних комунікацій між учасниками проектних команд і інтеграції з комплексними АСУП. Програмне забезпечення такого класу випускають компанії Artemis Management Systems, Primavera Systems, Welcom Software Technologies.

У фокусі нашої уваги - системи з цінової категорії \$300-800. Серед них програми як початкового рівня, в яких упор зроблений на легкість застосування, так і системи з розширеною функціональністю. Більшість з них

містять засоби для інтеграції з іншими застосуваннями і організації ефективних комунікацій в проектній команді: обмін інформацією по електронній пошті, видалений доступ через веб-браузер з можливістю оновлення даних, майстри для створення веб-звітів та інші функції. Як приклад таких систем можна назвати MS Project 2002 (Microsoft), SureTrak Project Manager (Primavera Systems), Spider Project Management Technologies (Spider) - всі вони дозволяють ефективно планувати проекти практично будь-якої складності. За допомогою АСУПП даної цінової категорії можна визначати етапи і список робіт проекту, створювати розклади, задавати календарні терміни виконання завдань і трудовитрати, призначати пріоритети, встановлювати взаємозв'язки, розраховувати критичні шляхи і так далі. Окрім цього, є можливість підтримувати множинне розбиття робіт на структурні частини, відповідні різним представленням проекту і що зважають на його специфіку. Кожний з пакетів має свої особливості в забезпеченні операцій планування. Так, інструменти Primavera дозволяють організувати більш ніж один зв'язок між двома роботами, тоді як MS Project і Spider - лише одну. Також продукти Primavera вимагають від менеджера створити структурну схему перш, ніж почати календарне планування, тоді як MS Project і Spider дозволяють отримати мережевий графік як побічний результат календарного планування. Spider при визначенні роботи разом із стандартними параметрами, прийнятими в західному світі (тривалість, трудовитрати, витрата ресурсів), надає можливість задати її об'єм в умовних одиницях (для організацій, які ведуть документацію за старими правилами, прийнятими за радянських часів).

У всіх системах організовані розвинені можливості оптимізації ходу виконання робіт за часом, бюджету і використанню ресурсів, зокрема трудових. І хоча подібна оптимізація передбачає активну самостійну корекцію завдань і призначень ресурсів, інструментальні засоби планування наочно показують, що підлягає оптимізації, а що ні. У результаті менеджерів залишається тільки перевірити ідеологічну можливість

оптимізації, і якщо це реально, внести відповідні зміни. За таким сценарієм діють пакети від Primavera і Microsoft. Вітчизняні ж розробники вважають, що ефективніша скрупульозна настройка майстра оптимізації з подальшим включенням його функціонування одним натисненням кнопки.

Хоча планування ризиків - процес, по суті, що мало оптимізується, непогані засоби по структуризації ризиків є у всіх проектних програмних модулях. Наприклад, непогані майстри, вбудовані в пакети вже на початковому етапі проектування, можуть підказати менеджерів, які ризики здатні зробити негативний вплив на проект. Так, дані системи забезпечують ідентифікацію потенційних ризиків в бюджетній сфері, ризиків щодо доставки ресурсів і інших. Крім того, ПО від Primavera і Microsoft дозволяє моделювати наслідки спрацьовування ризиків (вплив на календарний план, бюджет, витрати за проектом) за допомогою могутніх аналітичних засобів. Для цього застосовуються імітаційні сценарії «якщо.., то..», а також враховуються вірогідність ризику і фінансові наслідки його спрацьовування для кожної окремої роботи в проекті. На відміну від них додаткові можливості Spider в області планування ризиків зводяться до статистичного аналізу вірогідності закінчення проекту в межах встановлених календарних, бюджетних і ресурсних обмежень.

Що стосується планування забезпечення проекту ресурсами, в даній сфері описуване ПО вельми схоже. Всі АСУПП здатні вести планування призначень індивідуальних виконавців на роботи, облік неповного завантаження, планування витрат матеріальних ресурсів на окремих етапах.

У всіх трьох системах непогано розвинене планування витрат: надається можливість підтримки обліку вартості окремих робіт, часу роботи виконавців, використання одиниці матеріального ресурсу або години експлуатації устаткування, а також статті фіксованих витрат. До того ж інструменти від Primavera і Microsoft, на відміну від Spider, дозволяють враховувати окремо вартість урочної і наднормової роботи виконавців.

Одночасно Spider допускає використання фіксованих витрат, не пов'язаних з тривалістю і об'ємом роботи.

Що стосується моніторингу і аналізу виконання проекту, тут можливості описуваного ПО також знаходиться на високому рівні. Системи хороші для контролю і управління календарними термінами і витратою ресурсів. Крім того, вони дозволяють контролювати відхилення реального графіка проекту від базового, проводити аналіз освоєних об'ємів, а також створюють зрозумілі інформативні звіти.

Вельми важливим моментом в роботі сучасного АСУПП є функція колективного управління проектом. Наприклад, декілька менеджерів, ведучих справи проекту, повинні мати можливість не тільки паралельно працювати в розрахованому на багато користувачів режимі, але і одночасно пропонувати свої рішення на суд оптимізаційних модулів програми.

Технологічно програмний модуль проектної оптимізації розробки Spider, функціонуючи в мультикористувальному режимі, створює набір файлів, який розсилається по FTP всім менеджерам проекту. Таким чином, кожний з учасників робіт постійно знаходиться в курсі виконаного об'єму завдань. Аналогічно діють і ПО Primavera і MS Project. Вони використовують СУБД. Доступ до останніх здійснюється через глобальні і локальні мережі (правила доступу чітко регламентуються). Додаткова зручність MS Project - здатність підтримувати інтегрований документообіг в проектній групі, завдяки тісній інтеграції з серверними продуктами компанії.

Як видно, сьогодні робота проектного менеджера може бути достатня ефективно модернізована. При цьому на ринку співіснують продукти, що відповідають і світовим канонам в області ведення проектів (Primavera і MS Project), і радянським нормам проектування, які до цих пір активно використовуються окремими російськими підприємствами. В цьому випадку продукт вітчизняної компанії Spider Project Management Technologies представляється найбільш перспективним.

З іншого боку, в порівнянні з продуктами західних постачальників, де завдання етапів легко диференціюються між підрядчиками, Spider має значно менші можливості в області роботи над колективними проектами. Сильна сторона вітчизняної розробки - оптимізація проектів по наперед настроєному майстрові. Для компаній, що займаються, наприклад, тільки інтеграційною діяльністю і наладкою мереж, такий підхід виявиться вельми ефективним. Що стосується підприємств, які надають комплексні послуги, їм більше підійдуть Primavera або MS Project, що постійно вимагають від менеджера індивідуального підходу.

Одне з основних гідність MS Project - могутня підтримка місцевих партнерів Microsoft. Не варто забувати і про те, що MS Project тісно переплітається з офісними пакетами Microsoft, а також Exchange Server. MS Project гнучкий і може бути настроєний як за шаблоном, так і в режимі скрупульозного опрацювання можливостей оптимізації нового проекту.

Лідерське положення компанії Primavera на ринку відчувається досить-таки сильно. Її системи найбільш функціональні, проте мають і найвищу вартість на ринку, тому не популярні серед менеджерів, ведучих роботу над невеликими проектами. Пакет Primavera по-справжньому ефективний в тих проектах, де зайнято понад 350 учасників. У подібних випадках продукту Primavera рівних немає - широкі можливості по настройці оптимізації, а також безліч способів контролю роблять SureTrak Project Manager оптимальним вибором для крупних проектів з великим числом підрядчиків і тривалим терміном реалізації. Мабуть, найкращим чином Primavera SureTrak Project Manager може проявити себе в будівельно-монтажних і промислових організаціях.

В дипломній роботі розглядається проект розробки та впровадження нової системи підтримки прийняття рішень щодо завантаження транспортних засобів на підприємстві.

Ціль проекту – впровадження та використання ПЗ для розрахунків оптимального завантаження транспортних засобів ТОВ «EVA», з метою

ефективного використання ресурсів підприємства та скорочення загальних затрат.

3.2 Розробка проекту. Перелік етапів, виконавців та затрат проекту.

Формування переліку задач з урахуванням їх тривалості. Слід відзначити, що тривалість в даному розумінні вводиться в розрахунок на виконання кожної з задач 1 виконавцем. Пізніше, в ході управління ресурсами проекту, зокрема людськими, ці тривалості можуть бути змінені за рахунок колективного виконання робіт.

Після того, як визначено задачі, необхідно їх структурувати, тобто виділити головні етапи, задачі кожного з етапів та можливі підзадачі. Останній етап завершуємо задачею нульової довжини – віхою

Наступною дією є структурування історичної послідовності. Деякі роботи (задачі) не можуть бути розпочаті чи виконанні, допоки не виконані якісь інші. Саме зв'язки між попередніми і наступними задачами можуть показати критичний шлях проекту, його тривалість та загальну вартість. За допомогою вкладки «попередники» для задач встановлюються зв'язки, які об'єднують проект в одне ціле. В результаті маємо наступні етапи та задачі нашого проекту впровадження нового програмного забезпечення для ТОВ «EVA» (табл. 3.1):

Таблиця 3.1 – Структура етапів проекту та тривалість їх виконання.

№ п.п.	Назва задачі	Тривалість, дн.	Початок	Кінець	Попередники
1	Проект впровадження інтелектуальної системи автоматичного завантаження транспортних засобів	180	07.01.2024 9:00	15.09.2024 18:00	
2	Вибір проблеми	28	07.01.2024 9:00	13.02.2024 18:00	
3	Визначення критичної проблеми і задачі з тих, що існують в одному з департаментів підприємства	5	07.01.2024 9:00	13.01.2024 18:00	
4	Знаходження експерта для співробітництва при вирішенні проблеми, і призначення колективу розробників	4	14.01.2024 9:00	19.01.2024 18:00	3
5	Визначення попереднього підходу до вирішення проблеми	8	20.01.2024 9:00	29.01.2024 18:00	4;3
6	Аналіз витрат і прибутків від розробки проекту	7	30.01.2024 9:00	09.02.2024 18:00	5
7	Підготовка докладного плану розробки	4	10.02.2024 9:00	13.02.2024 18:00	6
8	Розробка прототипу експертної системи	141	16.02.2024 9:00	31.08.2024 18:00	
9	Ідентифікація проблеми (знаходження альтернативних методів оптимального рішення)	13	16.02.2024 9:00	04.03.2024 18:00	7;5
10	Добування і структуризація знань (отримання інформації про процес формування замовлень на підприємстві)	34	05.03.2024 9:00	21.04.2024 18:00	9
11	Формалізація (розробка бази знань в MS Excel для подальшої програмної реалізації СППР)	21	22.04.2024 9:00	20.05.2024 18:00	10
12	Реалізація (створення прототипу експертної системи на мові програмування Visual Basic)	57	21.05.2024 9:00	07.08.2024 18:00	11
13	Тестування роботи ІС й аналіз статистичних даних, отриманих при використанні альтернативних шляхів рішення проблеми	16	10.08.2024 9:00	31.08.2024 18:00	12
14	Розвиток прототипу до промислової ЕС (збільшення бази знань системи, додавання додаткових правил)	4	01.09.2024 9:00	04.09.2024 18:00	13
15	Оцінка системи (тестування відносно критеріїв ефективності)	14	10.08.2024 9:00	27.08.2024 18:00	13НН
16	Стиковка системи	7	07.09.2024 9:00	15.09.2024 18:00	
17	Забезпечення зв'язку ЕС з існуючими базами даних і іншими системами на підприємстві	3	07.09.2024 9:00	09.09.2024 18:00	15;14
18	Навчання персоналу по експлуатації і обслуговуванню ЕС	7	07.09.2024 9:00	15.09.2024 18:00	17НН
19	Закінчення робіт по проекту і подальша підтримка системи	0	15.09.2024 18:00	15.09.2024 18:00	17;18

В реалізації проекту приймає участь 6 виконавців (трудові ресурси), кожний з яких назначається для виконання окремих задач, а також один матеріальний ресурс – ПК для розробки та тестування ПЗ. Перелік трудових ресурсів, максимальне число одиниць їх використання, стандартні ставки та затрати представлені в табл. 3.2:

Таблиця 3.2 – Ресурси проекту.

№ п. п.	Назва ресурсу	Скороче на назва	Тип	Макс. одиниць, %	Стандарт на ставка, грн/год	Затрати на використання, грн
1	Директор відділу продаж	Д	Трудовий	100	57,00	0
2	Аналітик	А	Трудовий	100	14,00	0
3	Директор відділу фінансів	Д	Трудовий	100	57,00	0
4	Програміст	П	Трудовий	100	17,00	0
5	Адміністратор складу	А	Трудовий	100	9,00	0
6	Адміністратор відділу продаж	А	Трудовий	100	17,00	0
7	Настільний ПК	Н	Матеріальний			3000

Не слід забувати також про те, що всі працівники, що задіяні в проекті не є вивільненими працівниками, які на час виконання проекту полишають всі свої інші задачі. Тобто вони можуть використовуватися значно менше ніж на 100%. Типовою також є ситуація, коли основні виконавці – аналітик або програміст – повинні виконувати одночасно декілька задач. Необхідно розподіляти їх робочий час між задачами, бо інакше вони зможуть виконувати їх лише послідовно.

На кожному з етапів працівники отримують заробітну плату згідно з тарифами в таблиці 3.2. Витрати, які необхідні для реалізації проекту, наведені в таблиці 3.3, де розподілені по кожній із задач, що виконуються:

Таблиця 3.3 – Витрати на реалізацію етапів проекту.

№ п.п.	Назва задачі	Фіксовані витрати, грн	Загальні витрати, грн
1	Проект впровадження інтелектуальної системи автоматичного завантаження транспортних засобів	120,00	35 325,60
2	Вибір проблеми		6 416,00
3	Визначення критичної проблеми і задачі з тих, що існують в одному з департаментів підприємства		560,00
4	Знаходження експерта для співробітництва при вирішенні проблеми, і призначення колективу розробників		1 411,20
5	Визначення попереднього підходу до вирішення проблеми		1 171,20
6	Аналіз витрат і прибутків від розробки проекту		2 867,20
7	Підготовка докладного плану розробки		406,40
8	Розробка прототипу експертної системи		25 296,80
9	Ідентифікація проблеми (знаходження альтернативних методів оптимального рішення)		2 048,80
10	Добування і структуризація знань (отримання інформації про процес формування замовлень на підприємстві)		5 304,00
11	Формалізація (розробка бази знань в MS Excel для подальшої програмної реалізації СППР)		5 208,00

Таблиця 3.4 (продовження)

12	Реалізація (створення прототипу експертної системи на мові програмування Python)		7 752,00
13	Тестування роботи ІС й аналіз статистичних даних, отриманих при використанні альтернативних шляхів рішення проблеми		4 984,00
14	Розвиток прототипу до промислової ЕС (збільшення бази знань системи, додавання додаткових правил)		544,00
15	Оцінка системи (тестування відносно критеріїв ефективності)		1 036,00
16	Стиковка системи		1 912,80
17	Забезпечення зв'язку ЕС з існуючими базами даних і іншими системами на підприємстві		367,20
18	Навчання персоналу по експлуатації і обслуговуванню ЕС		1 545,60
19	Закінчення робіт по проекту і подальша підтримка системи		0,00

При створенні проекту значну роль займає планування витрат. В якості витрат можуть виступати стандартні ставки заробітної платні працівникам, ставки понаднормових, використання матеріальних ресурсів або визначення фіксованих витрат. В даній дипломній роботі для реалізації інвестиційного проекту було виділено 35 325,60 грн.

3.3 Побудова діаграми Ганта і можлива оптимізація проекту.

Іноді в проекті виникає ситуація, коли на певний ресурс призначається більше задач, ніж він в змозі забезпечити. В цьому випадку представлення діаграми Ганта (рис. 3.1) має ключове значення, оскільки саме вона демонструє послідовність виконання задач. Світлим кольором на діаграмі позначаються ті задачі, які складають критичний шлях – перелік робіт, тривалість яких не можна зменшити за існуючих трудових і матеріальних ресурсів.

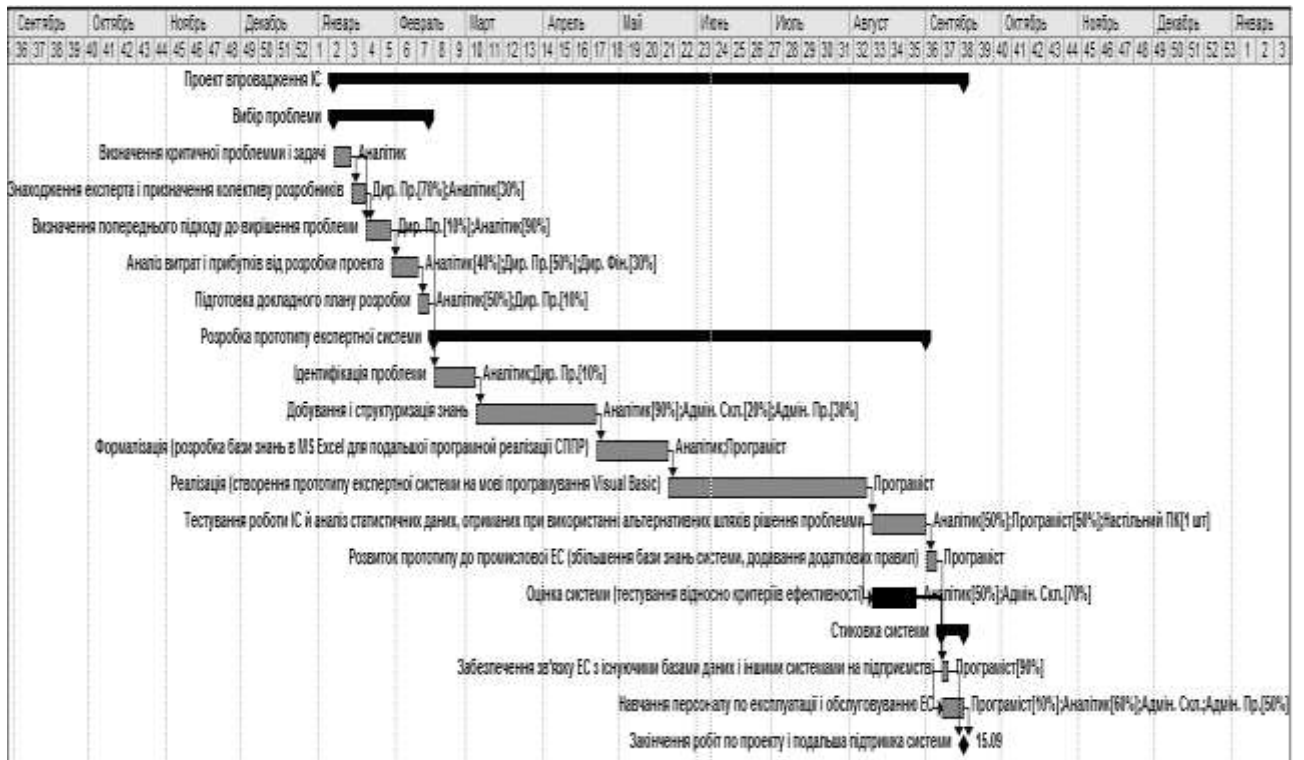


Рисунок 3.1 – Діаграма Ганта плану проекту розробки експертної системи.

Існує декілька методів вирішення задачі мінімізації критичного шляху проекту, серед них найчастіше використовують підхід зі збільшенням тривалості проекту або із застосуванням додаткових виробничих і трудових ресурсів.

3.4 Окупність проекту та висновки.

Ключовим критерієм привабливості інвестування в будь-якій сфері бізнесу є швидкість повернення вкладеного капіталу. Саме період окупності (Pay-Back Period) дозволяє інвесторові порівняти різні варіанти розвитку бізнесу і вибрати той проект, який найбільшою мірою відповідає його стратегії і фінансовим можливостям. Логічно припустити, що чим швидше проект окупить первинні витрати, тим раніше він почне приносити прибуток і тим більшою буде її сукупна величина.

Період окупності є термін, розрахований з дня початку реалізації проекту і здійснення перших інвестиційних витрат до моменту, коли різниця між накопиченою сумою чистого прибутку і об'ємом проведених інвестиційних витрат придбає позитивне значення, тобто буде досягнута точка беззбитковості. Тобто період окупності - час, протягом якого доходи від інвестицій досягли первинного вкладення капіталу в даний проект:

$$T_{ок} = \frac{\text{Затрати на проект/інвестиції}}{\text{Надходження грошових коштів за рік}}$$

Як зазначалося вище затрати на проект та його впровадження становлять 35 325,60 грн., а тривалість проекту – 180 робочих днів.

Розрахуємо надходження грошових коштів за рік використання впровадженого проекту. Для цього проведемо поетапний аналіз виконання замовлень ТОВ «EVA». За 1 робочий день у відділ продаж надходять і оформлюються 1-2 замовлення, з них виконується і відправляється для доставки за призначенням на дистрибуторів одне замовлення, що становить

приблизно 92 палети, тобто 7,1 завантажені машини. З використанням розробленої системи підтримки прийняття рішень отримуємо економію в 4 палети, тобто 0,3 вільного місця в машині (при вартості однієї машини 3900 грн). Таким чином ми економимо за день $3900 * 0,3 = 1170$ грн., а за рік: $1170 * 264 = 308\ 880$ грн.

Отже наш проект – окупний, так як період окупності становить:

$$T_{ок} = \frac{35325,60}{308880} = 0,11 \text{років} = 30,2 \text{дн.} = 1 \text{міс.}$$

ВИСНОВКИ

У ході виконання дипломної роботи було реалізовано програмно алгоритм Tabu Search, який показав непогані результати роботи і може бути застосований для пошуку рішень задач оптимізації.

Після виконання ряду тестів було визнано що алгоритм потребує більш чіткого встановлення параметрів короткострокової і довгострокової пам'яті. Результати виявили що алгоритм має великі відхилення якості пошуку оптимумів в залежності від деяких параметрів. За рахунок їх зміни було досягнуто зменшення похибки алгоритму, та збільшення швидкості пошуку.

Об'єктом дослідження виступав процес виконання замовлень на ТМВ в умовах підприємства ООО «EVA». Мету оптимізації розподілу замовлень задля зменшення кількості використаних палет досягнуто досить успішно.

В результаті роботи розроблено програмний продукт, який дозволяє розв'язувати оптимізації за допомогою жадного алгоритму, методу гілок та меж і пошуку Tabu. Результати розрахунків подано у вигляді таблиць та графіків. Наведено детальний опис структури програми та інструкцію по використанню програмного продукту. Отримані результати вказують на доцільність використання розробленого алгоритму для отримання високоякісних розв'язків розглянутої задачі.

Таким чином, можемо сказати, що на основі даної роботи можна прийняти правильне рішення при виборі способу розподілу замовленої продукції та її оптимального завантаження у транспортні засоби.

ВИКОРИСТАНІ ДЖЕРЕЛА:

1. Glover, F. and M. Laguna. Tabu Search. Kluwer, Norwell, MA. 1997.
2. Glover, F. "Tabu Search – Part I", ORSA Journal on Computing 1989.
3. Glover, F. "Tabu Search – Part II", ORSA Journal on Computing 1990.
4. Battiti R., Tecchiolli G. -The reactive tabu search.- preprint, Department of Mathematics, University of Trento, Trento, Italy (1992)
5. Costa D. -A Tabu Search Algorithm for Computing an Operational Time Table European Journal of Operational Research 76, (1994), pp. 98-110.
6. Faigle U., Kern W. -Some Convergence Results for Probabilistic Tabu Search – ORSA Journal on Computing 4, (1992), pp. 32-37.
7. Fox B.L. -Integrating and accelerating tabu search, simulated annealing and genetic algorithms - Annals of Operations Research 41, (1993) pp. 47-67.
8. Friden C. , Hertz A. , de Werra D. -STABULUS: a technique for finding stable sets in large graphs with tabu search - Computing 42, (1989) pp. 35-44.
9. Friden C. , Hertz A. , de Werra D. -TABARIS: an exact algorithm based on tabu search for finding a maximum independent set in a graph - Computers and Operations Research 17, (1990), pp. 437-445.
10. Gendreau M., Hertz A., Laporte G. -A Tabu Search Heuristic for the Vehicle Routing Problem- Management Science 40/10, (1994), pp. 1276-1290.
11. Glover F. -Future Paths for Integer Programming and Links to Artificial Intelligence- Computers and Operations Research 13, (1986), pp. 533-549.
12. Glover F. -Private communication - (1992).
13. Бекетова О.Н., Найденов В.И. Бізнес-план: теорія и практика. - М.: Альфа-Пресс, 2005, 271 с.
14. Бізнес-план. Методичні матеріали. / Р.Г. Маніловський, Л.С. Юлкіна, Н.А. Колеснікова. - М., 2005. - 254 с.
15. Бізнес-плани. Повне довідковий посібник / Під ред. И.М. Степнова. - М., 2007. - 240 с.: іл.
16. Аоки М. Введення в методи оптимізації. – М.: Наука, 1977.

- 17 Батищев Д.И. Методи оптимізаційного проектування. – М.: Радіо та зв'язок, 1984.
- 18 Дэгтярьов Ю.І. Методи оптимізації. – М.: Сов. Радіо, 1980.
- 19 Таха Х. Введення в дослідження операцій. – Кн.2. – М.: Мір, 1985.
- 20 Акуліч А. Математичне програмування.
- 21 Фундаментальні алгоритми. Алгоритми на графах: Пер. с англ./Роберт Седжвік. – СПб: ООО “ДіаСофтЮП”, 2003.
- 22 Гемінтерн В.И., Каган Б.М. Методи оптимального проектування – М Енергія, 1980 – 158 с.
- 23 Гольштейн Е.Г., Юдін Д.Б. Задачі лінійного програмування транспортного типу – М Наука, 1969 – 382 с.
- 24 Кузнецов А.В., Холод Н.І. Математичне програмування – Мн Вищ шк., 1984 – 221 с.
- 25 Кузнецов А.В., Холод Н.І., Костевіч Л.С. Посібник з розв'язку задач з математичного програмування – Мн Вищ шк., 1978 – 254 с.
26. Шевченко, Ю. О. (2022). Обробка і аналіз даних з використанням електронних таблиць. Частина II «Аналіз даних та макроси». <https://ir.nmu.org.ua/handle/123456789/162624>

Україна, 49127, м. Дніпро, вул. Шамільовська, 52
ТОВ "РУШ", тел 067 253 59 49

Обробка замовлень – Александр Фармило

тел. 097 123 45 67

email: schamil_tabu@gmail.com

Код пр.	Код внутр.	Название продукции	кол-во в ящ.	Зак., ящ	Вес шт. БРУТТО, кг	Вес, кг	Цена за шт.	Сумма заказа	№ палеты
С-500	CR11 272	SANINO з/п мин.+ фтор 50 мл	144	40	0,0956	550	1,88	-	
С-501	CR11 273	SANINO з/п мин.+ фтор 100 мл	48	40	0,1835	352	2,85	-	
С-501U	CR11 AC6	SANINO з/п мин.+ фтор 100+50 мл	48	40	0,264	506	2,85	-	
С-502A	CR11 064	SANINO з/п гель лед. свеж. 50 мл	144	40	0,0858	494	1,88	-	
С-730	CR11 066	SANINO з/п гель лед. свеж. 100 мл	48	40	0,1625	312	2,85	-	
С-503	CR11 922	SANINO з/п с травами 50 мл	144	40	0,0958	515	1,88	-	
С-620	CR11 777	SANINO з/п с травами 100 мл	48	40	0,183	315	2,85	-	
С-504	CR11 897	SANINO з/п отбеливающая 50 мл	144	40	0,0854	491	1,88	-	
С-619	CR11 776	SANINO з/п отбеливающая 100 мл	48	40	0,16	307	2,85	-	
С-619P	CR11 ACA	SANINO з/п отбеливающая 100 мл + 50 мл	48	40	0,229	439	2,85	-	
С-638	CR11 774	SANINO з/п Мультизащита 50 мл	72	40	0,086	247	2,24	-	
С-639	CR11 775	SANINO з/п Мультизащита 100 мл	48	40	0,163	312	3,74	-	
С-637	CR11 778	SANINO з/п apple 100 мл	48	40	0,182	349	2,85	-	
С-725	CR11 A42	SANINO Чувствительная з/п 50 мл	72	40	0,085	244	2,99	-	

C-726	CR11 A43	SANINO Чувствительная з/п 100 мл	48	40	0,158	303	4,5	-	
C-763	CR11 A98	SANINO Зеленый чай з/п 50 мл	72	40	0,095	273	1,5	-	
C-764	CR11 A99	SANINO Зеленый чай з/п 100 мл	48	40	0,179	343	2,48	-	
C-765	CR11 AA0	SANINO Прополис з/п 50 мл	72	40	0,095	273	1,5	-	
C-766	CR11 AA1	SANINO Прополис з/п 100 мл	48	40	0,179	343	2,48	-	
C-604	CR11 708	SANINO Vitamin з/п 50 мл	72	40	0,088	253	1,99	-	
C-603	CR11 707	SANINO Int Wh з/п 50 мл tropik	72	40	0,096	276	1,88	-	
	CR11 706	SANINO White з/п 50 мл apple	72	40	0,096	276	1,88	-	
C-607	CR11 AD6	SANINO White з/п 50 мл apple	72	40	0,096	276	2,85	-	
C-599	CR11 709	SANINO BSoda з/п 75 мл	48	40	0,125	240	4,88	-	
C-639U	CR11 A13	SANINO з/п Family, (Junior 75 ml+Multi 100 ml)	24	40	0,323	310	4,88	-	
C-640	CR11 779	SANINO з/п Детская 75 мл	48	40	0,125	240	2,99	-	
C-287L	CR11 548	ARKO Крем для бритья 65gr ментол	72	40	0,093	267	2,85	-	
	CR11 549	ARKO Крем для бритья 65gr прохладный	72	40	0,093	267	2,85	-	
	CR11 560	ARKO Крем для бритья 65gr нормаль	72	40	0,093	267	2,85	-	
	CR11 561	ARKO Крем для бритья 65gr коммандо	72	40	0,093	267	2,85	-	
	CR11 562	ARKO Крем для бритья 65gr чувств.	72	40	0,093	267	2,85	-	
C-621A	CR11 A35	BLIX крем для бритья 65 мл свежесть	72	40	0,091	262	2,24	-	
	CR11 A34	BLIX крем для бритья 65 мл регуляр	72	40	0,091	262	2,24	-	
	CR11 A36	BLIX крем для бритья 65 мл спорт	72	40	0,091	262	2,24	-	
	CR11	BLIX крем для	72	40	0,091	262	2,24	-	

	A37	бритьёя 65 мл чувствительная						-	
С-49А	CR11 784	АРКО Пена для бритьёя 200 мл нормальн.	24	40	0,285	273	6,74	-	
	CR11 785	АРКО Пена для бритьёя 200 мл прохл.	24	40	0,285	273	6,74	-	
	CR11 786	АРКО Пена для бритьёя 200 мл влажн.	24	40	0,285	273	6,74	-	
	CR11 787	АРКО Пена для бритьёя 200 мл чувств.	24	40	0,285	273	6,74	-	
	CR11 788	АРКО Пена для бритьёя 200 мл коммандо	24	40	0,285	273	6,74	-	
	CR11 789	АРКО Пена для бритьёя 200 мл аква	24	40	0,285	273	6,74	-	
	CR11 790	АРКО Пена для бритьёя 200 мл цитрус	24	40	0,285	273	6,74	-	
	CR11 791	АРКО Пена для бритьёя 200 мл спорт	24	40	0,285	273	6,74	-	
С-389	CR11 714	АРКО Пена для бритьёя 100 мл прохл.	24	40	0,145	139	4,73	-	
	CR11 715	АРКО Пена для бритьёя 100 мл чувств.	24	40	0,145	139	4,73	-	
С-396-А	CR11 882	АРКО гель для бритьёя 200 мл. регуляр	24	40	0,305	292	11,99	-	
	CR11 883	АРКО гель для бритьёя 200 мл. чувствительный	24	40	0,305	292	11,99	-	
	CR11 884	АРКО гель для бритьёя 200 мл. прохладный	24	40	0,305	292	11,99	-	
	CR11 885	АРКО гель для бритьёя 200 мл. коммандо	24	40	0,305	292	11,99	-	
	CR11 886	АРКО гель для бритьёя 200 мл. увлажняющий	24	40	0,305	292	11,99	-	
С-691	CR11 949	АРКО пена для бритьёя 250 ml прохлада	24	40	0,3446	330	6,74	-	

	CR11 AAB	ARCO пена для бритья 250 ml аква	24	40	0,3446	330	6,74	-	
	CR11 AAA	ARCO пена для бритья 250 ml регуляр	24	40	0,3446	330	6,74	-	
	CR11 AAC	ARCO пена для бритья 250 ml цитрус	24	40	0,3446	330	6,74	-	
	CR11 950	ARCO пена для бритья 250 ml увлажн.	24	40	0,3446	330	6,74	-	
	CR11 951	ARCO пена для бритья 250 ml чувствит.	24	40	0,3446	330	6,74	-	
	CR11 AAD	ARCO пена для бритья 250 ml спорт	24	40	0,3446	330	6,74	-	
	CR11 952	ARCO пена для бритья 250 ml коммандо	24	40	0,3446	330	6,74	-	
C- 692	CR11 974	ARCO гель д/б 250 мл. регуляр	24	40	0,3458	331	11,99	-	
	CR11 975	ARCO гель д/б 250 мл. прохл.	24	40	0,3458	331	11,99	-	
	CR11 976	ARCO гель д/б 250 мл. чувст.	24	40	0,3458	331	11,99	-	
	CR11 977	ARCO гель д/б 250 мл. комм.	24	40	0,3458	331	11,99	-	
	CR11 978	ARCO гель д/б 250 мл. увл.	24	40	0,3458	331	11,99	-	
C- 560	CR11 716	Arco гель для бритья 60 ml аква	72	40	0,089	256	2,31	-	
	CR11 900	Arco гель для бритья 60 ml спорт	72	40	0,089	256	2,31	-	
C- 345L	CR11 639	ARCO крем после бритья 50 ml прохл	72	40	0,071	204	2,85	-	
	CR11 640	ARCO крем после бритья 50 ml аква	72	40	0,071	204	2,85	-	
	CR11 641	ARCO крем после бритья 50 ml коммандо	72	40	0,071	204	2,85	-	
	CR11 792	ARCO крем после бритья 50 ml спорт	72	40	0,071	204	2,85	-	
	CR11 793	ARCO крем после бритья 50 ml цитрус	72	40	0,071	204	2,85	-	
C- 61B	CR11 488	ARCO бальзам после бритья 150	24	40	0,071	68	10,69	-	

		мл прохладный							
	CR11 736	ARKO бальзам после бритья 150 мл аква	24	40	0,179	171	10,69	-	
	CR11 652	ARKO бальзам после бритья 150 мл commando	24	40	0,179	171	10,69	-	
C- 61R	CR11 A69	ARKO бальзам после бритья 100 мл аква	24	40	0,125	120	8,24	-	
	CR11 A70	ARKO бальзам после бритья 100 мл прохлада	24	40	0,125	120	8,24	-	
	CR11 A71	ARKO бальзам после бритья 100 мл командо	24	40	0,125	120	8,24	-	
C- 116	CR11 311	ARKO лосьон после бритья 250 мл прохладный	24	40	0,278	266	14,25	-	
	CR11 820	ARKO лосьон после бритья 250 мл аква	24	40	0,278	266	14,25	-	
	CR11 821	ARKO лосьон после бритья 250 мл спорт	24	40	0,278	266	14,25	-	
	CR11 831	ARKO лосьон после бритья 250 мл цитрус	24	40	0,278	266	14,25	-	
	CR11 297	ARKO лосьон после бритья 250 мл командо	24	40	0,278	266	14,25	-	
C- 644	CR11 850	ARKO лосьон после бритья 100 мл командо	24	40	0,2854	273	12,74	-	
	CR11 851	ARKO лосьон после бритья 100 мл спорт	24	40	0,2854	273	12,74	-	
	CR11 852	ARKO лосьон после бритья 100 мл цитрус	24	40	0,2854	273	12,74	-	
	CR11 853	ARKO лосьон после бритья 100 мл прохлада	24	40	0,2854	273	12,74	-	
	CR11 854	ARKO лосьон после бритья 100 мл аква	24	40	0,2854	273	12,74	-	
C- 321	CR11 319	FAX крем для бритья 65 гр. регуляр	72	40	0,097	279	2,08	-	
	CR11	FAX крем для	72	40	0,097	279	2,08		

	321	бритья 65 гр. чувствительный						-	
	CR11 409	FAX крем для бритья 65 гр. свежесть	72	40	0,097	279	2,08	-	
	CR11 410	FAX крем для бритья 65 гр. спорт	72	40	0,097	279	2,08	-	
C- 555	CR11 544	Арко гель после бритья 150 мл аква	24	40	0,177	169	10,69	-	
	CR11 824	Арко гель после бритья 150 мл спорт	24	40	0,177	169	10,69	-	
C- 555R	CR11 A93	Арко гель после бритья 100 мл Аква	24	40	0,125	120	8,24	-	
	CR11 A92	Арко гель после бритья 100 мл Спорт	24	40	0,125	120	8,24	-	
C- 650C	CR11 857	Арко дезодор. 150 мл. прохлада	12	40	0,1668	80	6,37	-	
	CR11 858	Арко дезодор. 150 мл. аква	12	40	0,1668	80	6,37	-	
	CR11 859	Арко дезодор. 150 мл. спорт	12	40	0,1668	80	6,37	-	
	CR11 860	Арко дезодор. 150 мл. коммандо	12	40	0,1668	80	6,37	-	
	CR11 861	Арко дезодор. 150 мл. цитрус	12	40	0,1668	80	6,37	-	
	CR11 862	Арко дезодор. 150 мл. форза	12	40	0,1668	80	6,37	-	
C- 633	CR11 825	Duru Агота жидк мыло 400 мл имбирь	12	40	0,483	231	4,49	-	
	CR11 826	Duru Агота жидк мыло 400 мл кислород	12	40	0,483	231	4,49	-	
	CR11 827	Duru Агота жидк мыло 400 мл лаванда	12	40	0,483	231	4,49	-	
	CR11 828	Duru Агота жидк мыло 400 мл бамбук	12	40	0,483	231	4,49	-	
C- 635	CR11 842	Duru Саге жидк мыло 400 мл кокос	12	40	0,4983	239	4,49	-	
	CR11 843	Duru Саге жидк мыло 400 мл миндаль	12	40	0,4983	239	4,49	-	
	CR11 844	Duru Саге жидк мыло 400 мл мед	12	40	0,4983	239	4,49	-	
	CR11	Duru Саге жидк	12	40	0,4983	239	4,49		

	845	мыло 400 мл молоко						-	
С-515	CR11 517	Duru гель для душа 250 мл бамбук	12	40	0,302	144	5,24	-	
	CR11 516	Duru гель для душа 250 мл минерал	12	40	0,302	144	5,24	-	
	CR11 518	Duru гель для душа 250 мл энергетический	12	40	0,302	144	5,24	-	
	CR11 519	Duru гель для душа 250 мл расслабляющий	12	40	0,302	144	5,24	-	
С-641	CR11 846	Duru Care гель для душа 250 мл кокос	12	40	0,3	144	5,24	-	
	CR11 847	Duru Care гель для душа 250 мл миндаль	12	40	0,3	144	5,24	-	
	CR11 848	Duru Care гель для душа 250 мл мед	12	40	0,3	144	5,24	-	
	CR11 849	Duru Care гель для душа 250 мл молоко	12	40	0,3	144	5,24	-	
С-651	CR11 895	Arko гель для душа 250 мл цитрус	24	40	0,3	288	6,75	-	
	CR11 896	Arko гель для душа 250 мл спорт	24	40	0,3	288	6,75	-	
S-515Y	CR11 A17	Duru Moods гель д/д 250 мл Energy энергия и бодрость	12	40	0,326	156	6,75	-	
	CR11 A18	Duru Moods гель д/д 250 мл Desire страсть и желание	12	40	0,326	156	6,75	-	
	CR11 A19	Duru Moods гель д/д 250 мл Luxury гламур и люкс	12	40	0,326	156	6,75	-	
	CR11 A20	Duru Moods гель д/д 250 мл Tranquility спок. и наслажд.	12	40	0,326	156	6,75	-	
S-515P	CR11 A80	Duru Moods гель д/д 250 мл Энергия (Energy) + мочалка	12	40	0,354	169	6,75	-	
	CR11 A81	Duru Moods гель д/д 250 мл Желание (Desire) + мочалка	12	40	0,354	169	6,75	-	
	CR11 A82	Duru Moods гель д/д 250 мл Роскошь (Luxury) +	12	40	0,354	169	6,75	-	

		мочалка							
	CR11 A83	Duru Moods гель д/д 250 мл Расслабление (Tranquility) + моч.	12	40	0,354	169	6,75	-	
S- 515S	CR11 A84	Duru Moods гель д/д 250 мл Энергия (Energy) + мыло	12	40	0,421	202	6,75	-	
	CR11 A85	Duru Moods гель д/д 250 мл Желание (Desire) + мыло	12	40	0,421	202	6,75	-	
	CR11 A86	Duru Moods гель д/д 250 мл Роскошь (Luxury) + мыло	12	40	0,421	202	6,75	-	
	CR11 A87	Duru Moods гель д/д 250 мл Расслабление (Tranquility) + мыло	12	40	0,421	202	6,75	-	
S- 641Y	CR11 A21	Duru Looks гель д/д 250 мл Nourishing пит. (лесной орех)	12	40	0,317	152	6,75	-	
	CR11 A22	Duru Looks гель д/д 250 мл Exfoliating шелк, укрепл. (гранат)	12	40	0,317	152	6,75	-	
	CR11 A23	Duru Looks гель д/д 250 мл Refreshing освежающий (грейпфрут)	12	40	0,317	152	6,75	-	
	CR11 A24	Duru Looks гель д/д 250 мл Repairing восстанавл. (клубника)	12	40	0,317	152	6,75	-	
S- 641P	CR11 A76	Duru Looks гель д/д 250 мл Питающий (Nourishing) + моч.	12	40	0,354	169	6,75	-	
	CR11 A77	Duru Looks гель д/д 250 мл Шелк, укрепляющий (Exfoliating) + моч.	12	40	0,354	169	6,75	-	
	CR11 A78	Duru Looks гель д/д 250 мл Освежающий (Refreshing) + моч.	12	40	0,354	169	6,75	-	
	CR11	Duru Looks гель	12	40	0,354	169	6,75	-	

	A79	д/д 250 мл Восстанавливающ ий (Repairing) + моч.							-	
S-588	CR11 A46	DURU Looks ж/мыло 300 мл Nourishing питающий (лесной орех)	12	40	0,383	183	5,48		-	
	CR11 A47	DURU Looks ж/мыло 300 мл Moisturising увл. (персик)	12	40	0,383	183	5,48		-	
	CR11 A48	DURU Looks ж/мыло 300 мл Silkening/Exfoliating обновл. (гранат)	12	40	0,383	183	5,48		-	
	CR11 A49	DURU Looks ж/мыло 300 мл Refreshing освежающее (грейпфрут)	12	40	0,383	183	5,48		-	
S-589	CR11 A50	DURU Moods ж/мыло 300 мл Energy-энергия	12	40	0,383	183	5,48		-	
	CR11 A51	DURU Moods ж/мыло 300 мл Desire-желание	12	40	0,383	183	5,48		-	
	CR11 A52	DURU Moods ж/мыло 300 мл Nature-природа	12	40	0,383	183	5,48		-	
	CR11 A53	DURU Moods ж/мыло 300 мл Tranquility- расслабление	12	40	0,383	183	5,48		-	
S-425A	CR11 460	DURU DEO 3*100G\BAR морск.	24	40	0,3292	316	4,88		-	
	CR11 461	DURU DEO 3*100G\BAR тропик	24	40	0,3292	316	4,88		-	
S-558	CR11 741	DURU экопак 4*100 весна	24	40	0,4154	398	5,78		-	
	CR11 742	DURU экопак 4*100 осень	24	40	0,4154	398	5,78		-	
	CR11 743	DURU экопак 4*100 лето	24	40	0,4154	398	5,78		-	
	CR11 744	DURU экопак 4*100 зима	24	40	0,4154	398	5,78		-	
S-261	CR11 041	DURU 1+1 экопак 4/100gr мор.мин.	24	40	0,414	397	5,78		-	

	CR11 042	DURU 1+1 экопак 4/100gr зел. чай	24	40	0,414	397	5,78	-	
	CR11 043	DURU 1+1 экопак 4/100gr лотос	24	40	0,414	397	5,78	-	
	CR11 044	DURU 1+1 экопак 4/100gr лимон	24	40	0,414	397	5,78	-	
S- 562A	CR11 834	DURU 1+1 инд. 100 gr лотос	72	40	0,1049	302	1,58	-	
	CR11 835	DURU 1+1 инд. 100 gr зел. чай	72	40	0,1049	302	1,58	-	
	CR11 836	DURU 1+1 инд. 100 gr мор. минералы	72	40	0,1049	302	1,58	-	
S- 128 E	CR11 863	FAX 3*125 планшет яблоко	20	40	0,4195	335	3,97	-	
	CR11 868	FAX 3*125 планшет ландыш	20	40	0,4195	335	3,97	-	
	CR11 865	FAX 3*125 планшет роза	20	40	0,4195	335	3,97	-	
	CR11 864	FAX 3*125 планшет лимон	20	40	0,4195	335	3,97	-	
	CR11 866	FAX 3*125 планшет лаванда	20	40	0,4195	335	3,97	-	
	CR11 867	FAX 3*125 планшет персик	20	40	0,4195	335	3,97	-	
	CR11 870	FAX 3*125 планшет магнолия	20	40	0,4195	335	3,97	-	
	CR11 869	FAX 3*125 планшет сирень	20	40	0,4195	335	3,97	-	
S- 186 E	CR11 871	FAX 5*75 экопак яблоко	24	40	0,3925	376	3,67	-	
	CR11 872	FAX 5*75 экопак лимон	24	40	0,3925	376	3,67	-	
	CR11 873	FAX 5*75 экопак роза	24	40	0,3925	376	3,67	-	
	CR11 874	FAX 5*75 экопак лаванда	24	40	0,3925	376	3,67	-	
	CR11 876	FAX 5*75 экопак ландыш	24	40	0,3925	376	3,67	-	
	CR11 877	FAX 5*75 экопак магнолия	24	40	0,3925	376	3,67	-	
	CR11 875	FAX 5*75 экопак сирень	24	40	0,3925	376	3,67	-	
S- 586R	CR11 A62	DURU Looks 100 гр Nourishing питающий (лесной орех)	48	40	0,114	218	1,73	-	
	CR11 A63	DURU Looks 100 гр Exfoliating шелковое прикосновение	48	40	0,114	218	1,73	-	

		(Гранат)							
	CR11 A64	DURU Looks 100 гр Refreshing освежающее (Грейпфрут)	48	40	0,114	218	1,73	-	
S- 587R	CR11 A65	DURU Moods 100 гр Energy-энергия	48	40	0,11	211	1,73	-	
	CR11 A67	DURU Moods 100 гр Luxury гламур и люкс	48	40	0,11	211	1,73	-	
	CR11 A66	DURU Moods 100 гр Desire страсть и желание	48	40	0,11	211	1,73	-	
	CR11 A68	DURU Moods 100 гр Tranquility- расслабление	48	40	0,11	211	1,73	-	
S- 80C	CR11 481	DURU Lady 100gr мед.	48	40	0,111	213	1,73	-	
	CR11 482	DURU Lady 100gr фрукт.	48	40	0,111	213	1,73	-	
	CR11 483	DURU Lady 100gr вит.Е	48	40	0,111	213	1,73	-	
	CR11 484	DURU Lady 100gr крем	48	40	0,111	213	1,73	-	
S- 417	CR11 425	Savoy э/п 5*60 яблоко	24	40	0,313	300	2,66	-	
	CR11 426	Savoy э/п 5*60 лимон	24	40	0,313	300	2,66	-	
	CR11 427	Savoy э/п 5*60 роза	24	40	0,313	300	2,66	-	
S- 421D	CR11 563	ARKO baby 100g	96	40	0,109	418	1,5	-	
S- 431A	CR11 725	DURU Vital 5*75g экопак аква	24	40	0,396	380	4,73	-	
	CR11 726	DURU Vital 5*75g экопак лес	24	40	0,396	380	4,73	-	
	CR11 727	DURU Vital 5*75g экопак закат	24	40	0,396	380	4,73	-	
S- 432	CR11 671	DURU Vital 3*125g планш. аква	20	40	0,415	332	4,4	-	
	CR11 572	DURU Vital 3*125g планш. лес	20	40	0,415	332	4,4	-	
	CR11 673	DURU Vital 3*125g планш. закат	20	40	0,415	332	4,4	-	
S- 302A	CR11 A01	DURU банное 200 г мед	48	40	0,2083	399	2,43	-	
	CR11 A02	DURU банное 200 г огурец	48	40	0,2083	399	2,43	-	
	CR11 A03	DURU банное 200 г мол.протеин	48	40	0,2083	399	2,43	-	

	CR11 A04	DURU банное 200 г миндаль	48	40	0,2083	399	2,43	-	
S- 218A	CR11 717	DURU NATURAL 3x100гр планш. олив.	24	40	0,354	339	5,4	-	
	CR11 719	DURU NATURAL 3x100гр планш. минд.	24	40	0,354	339	5,4	-	
	CR11 720	DURU NATURAL 3x100гр планш черешня	24	40	0,354	339	5,4	-	
S- 415A	CR11 594	DURU NATURAL 100гр олив.масло	72	40	0,111	319	1,54	-	
	CR11 581	DURU NATURAL 100гр минд.масло	72	40	0,111	319	1,54	-	
	CR11 592	DURU NATURAL 100гр черешня	72	40	0,111	319	1,54	-	
S- 442	CR11 533	DURU Арома инд 100г лаванда	48	40	0,1104	211	1,25	-	
	CR11 534	DURU Арома инд 100г имбирь	48	40	0,1104	211	1,25	-	
	CR11 822	DURU Арома инд 100г бамбук	48	40	0,1104	211	1,25	-	
	CR11 823	DURU Арома инд 100г кислород	48	40	0,1104	211	1,25	-	
S- 557	CR11 732	DURU Fruity 5*75 э/п арбуз	24	40	0,394	378	4,88	-	
	CR11 733	DURU Fruity 5*75 э/п виноград	24	40	0,394	378	4,88	-	
	CR11 734	DURU Fruity 5*75 э/п мандарин	24	40	0,394	378	4,88	-	
	CR11 735	DURU Fruity 5*75 э/п персик	24	40	0,394	378	4,88	-	
S- 444A	CR11 728	DURU Milk 5*75г экопак мол+крем	24	40	0,396	380	4,5	-	
	CR11 729	DURU Milk 5*75г экопак мол+миндаль	24	40	0,396	380	4,5	-	
	CR11 730	DURU Milk 5*75г экопак мол+пшеница	24	40	0,396	380	4,5	-	
	CR11 731	DURU Milk 5*75г экопак мол+мед	24	40	0,396	380	4,5	-	
S- 444R	CR11 057	DURU Milk 6*75г э/п кр.	24	40	0,479	459	4,5	-	
	CR11 063	DURU Milk 6*75г э/п мед.	24	40	0,479	459	4,5	-	
	CR11 058	DURU Milk 6*75г э/п мин.	24	40	0,479	459	4,5	-	
S-	CR11	DURU Fruity 75 gr	72	40	0,0859	247	1,12		

570	887	арбуз						-	
	CR11 888	DURU Fruity 75 gr виноград	72	40	0,0859	247	1,12	-	
	CR11 889	DURU Fruity 75 gr мандарин	72	40	0,0859	247	1,12	-	
	CR11 890	DURU Fruity 75 gr персик	72	40	0,0859	247	1,12	-	
S- 553E	CR11 954	Фах инд 75г яблоко	72	40	0,0799	230	0,86	-	
	CR11 955	Фах инд 75г роза	72	40	0,0799	230	0,86	-	
	CR11 956	Фах инд 75г лимон	72	40	0,0799	230	0,86	-	
	CR11 957	Фах инд 75г лаванда	72	40	0,0799	230	0,86	-	
	CR11 958	Фах инд 75г персик	72	40	0,0799	230	0,86	-	
	CR11 959	Фах инд 75г сирень	72	40	0,0799	230	0,86	-	
	CR11 960	Фах инд 75г магнолия	72	40	0,0799	230	0,86	-	
	CR11 961	Фах инд 75г ландыш	72	40	0,0799	230	0,86	-	
S- 60A	CR11 801	Duru Fresh 4*125г п/э океан	16	40	0,55	352	7,35	-	
	CR11 802	Duru Fresh 4*125г п/э цветы	16	40	0,55	352	7,35	-	
	CR11 803	Duru Fresh 4*125г п/э фрукты	16	40	0,55	352	7,35	-	
S- 138 E	CR11 688	DURU хозяйств. 4*150 белое	21	40	0,621	521	4,88	-	
	CR11 893	DURU хозяйств. 4*150 роза	21	40	0,621	521	4,88	-	
	CR11 894	DURU хозяйств. 4*150 лаванда	21	40	0,621	521	4,88	-	
	CR11 891	DURU хозяйств. 4*150 яблоко	21	40	0,621	521	4,88	-	
S- 556	CR11 781	DURU хозяйств. White 2*125 п/э станд	42	40	0,258	433	2,24	-	
S- 255	CR11 230	Хозяйств. стандартное 200гр. Инд.	48	40	0,205	393	1,2	-	
s-7	CR11 195	DURU хозяйств. 4*250 G	12	40	1,03	494	7,5	-	
S- 139	CR11 677	DURU хозяйств. 4*200 G	18	40	0,824	593	6,37	-	

ИТОГО ТОНН

0

Итого

-

	ИТОГО заказано ящИКОВ

		o		
0				

```
import openpyxl

# Открываем существующую рабочую книгу
wb = openpyxl.load_workbook('in.xlsx')

# Проверка существования листа "Размещение"
if "Размещение" in wb.sheetnames:
    ws_placement = wb["Размещение"]
else:
    raise ValueError("Лист 'Размещение' не найден в рабочей книге")

for lis in range(2, 52):
    ws = wb.worksheets[lis - 1]
    i = 19

    # находим количество позиций в заказе
    while ws.cell(row=i, column=3).value is not None:
        i += 1
    kol = i - 1

    # находим количество дополнительных позиций и размещаем их на первые
    палеты
    for schet in range(19, kol + 1):
        value7 = ws.cell(row=schet, column=7).value
        value16 = value7 // 800 if value7 is not None else 0
        ws.cell(row=schet, column=16, value=value16)

        if value16 != 0:
            ws.cell(row=schet, column=12, value=value7 - 800 * value16)
        else:
            ws.cell(row=schet, column=12, value=value7)

    summdop = 0
    for j in range(19, kol + 1):
        cell_value = ws.cell(row=j, column=16).value
        if cell_value is not None:
            summdop += cell_value

    for d in range(1, summdop + 1):
        ws_placement.cell(row=lis + 2, column=d + 2, value=800)
```

```

Zak = 1
while Zak != 0:
    S = 0
    Sum = S

    for x in range(19, kol + 1):
        razn = 800 - S
        value12 = ws.cell(row=x, column=12).value

        if value12 is not None and value12 != 0:
            if value12 <= razn:
                S += value12
                ws.cell(row=x, column=11, value=1 + summdop)
                ws.cell(row=x, column=12, value=0)

    Zak = sum(ws.cell(row=i, column=12).value or 0 for i in range(19, kol + 1))

    ws_placement.cell(row=lis + 2, column=3 + summdop, value=S)
    summdop += 1

    kolpalet = ws.cell(row=19, column=11).value or 0
    for i in range(19, kol + 1):
        cell_value = ws.cell(row=i + 1, column=11).value
        if cell_value is not None and cell_value >= kolpalet:
            kolpalet = cell_value

    ws_placement.cell(row=lis + 2, column=154, value=kolpalet)

# Сохраняем рабочую книгу в формате .xlsx
wb.save('in.xlsx')

```

```
import openpyxl

def get_numeric_value(cell_value):
    """Функция для получения числового значения из ячейки."""
    if cell_value is None:
        return 0
    try:
        return float(cell_value)
    except ValueError:
        return 0

# Открываем существующую рабочую книгу
wb = openpyxl.load_workbook('in.xlsx')

# Проверка существования листа "Размещение"
if "Размещение" in wb.sheetnames:
    ws_placement = wb["Размещение"]
else:
    raise ValueError("Лист 'Размещение' не найден в рабочей книге")

for lis in range(1, 52):
    ws = wb.worksheets[lis - 1]
    i = 4

    # находим количество позиций в заказе
    while ws.cell(row=i, column=3).value is not None:
        i += 1
    kol = i - 1

    # находим количество дополнительных позиций и размещаем их на первые
    # палеты
    for schet in range(4, kol + 1):
        value7 = get_numeric_value(ws.cell(row=schet, column=7).value)
        value16 = int(value7 // 800)
        ws.cell(row=schet, column=16, value=value16)

        if value16 != 0:
            ws.cell(row=schet, column=12, value=value7 - 800 * value16)
        else:
            ws.cell(row=schet, column=12, value=value7)

    summdop = 0
```

```

for j in range(4, kol + 1):
    cell_value = get_numeric_value(ws.cell(row=j, column=16).value)
    summdop += cell_value

summdop = int(summdop) # Приводим summdop к целому числу

for d in range(1, summdop + 1):
    ws_placement.cell(row=lis + 2, column=d + 2, value=800)

# загрузка палет
while get_numeric_value(ws.cell(row=321, column=12).value) != 0:
    S = 0
    max1 = 0
    max3 = 100000
    for i in range(4, kol + 1):
        if get_numeric_value(ws.cell(row=i, column=12).value) > max1:
            max1 = get_numeric_value(ws.cell(row=i, column=12).value)
            nom = i

    ws.cell(row=nom, column=12, value=0)
    ws.cell(row=nom, column=11, value=summdop + 1)

    while True:
        S = max1
        max2 = 0
        for k in range(4, kol + 1):
            if get_numeric_value(ws.cell(row=k, column=12).value) > max2:
                if get_numeric_value(ws.cell(row=k, column=12).value) < max3:
                    max2 = get_numeric_value(ws.cell(row=k, column=12).value)
                    nom = k

        S += max2

    if S < 800:
        if max2 != 0:
            max1 = S
            ws.cell(row=nom, column=12, value=0)
            ws.cell(row=nom, column=11, value=summdop + 1)

        razn = 800 - S
        minim = 100000

        for i in range(4, kol + 1):
            cell_value = get_numeric_value(ws.cell(row=i, column=12).value)

```



```
if cell_value != 0 and cell_value < minim:  
    minim = cell_value
```

```
if minim <= razn:
```

```
    continue
```

```
else:
```

```
    ws_placement.cell(row=2 + lis, column=3 + summdop, value=S)
```

```
    summdop += 1
```

```
    break
```

```
if S > 800:
```

```
    max3 = max2
```

```
    continue
```

```
if S == 800:
```

```
    ws.cell(row=nom, column=12, value=0)
```

```
    ws_placement.cell(row=2 + lis, column=3 + summdop, value=S)
```

```
    summdop += 1
```

```
    break
```

```
# Сохраняем рабочую книгу в формате .xlsx  
wb.save('in.xlsx')
```

```

import openpyxl

def to_number(value):
    try:
        return float(value)
    except (TypeError, ValueError):
        return 0

def process_excel(file_path):
    wb = openpyxl.load_workbook(file_path)

    for lis in [42]:
        sheet = wb.worksheets[lis - 1] # Assuming 0-based indexing for sheets

        i = 19
        MSum = -1000000

        while sheet.cell(row=i, column=3).value is not None:
            i += 1

        kol = i - 1

        for i in range(19, kol + 1):
            sheet.cell(row=i, column=16, value=0) # short
            sheet.cell(row=i, column=17, value=0) # long
            sheet.cell(row=i, column=18, value=0) # iter
            sheet.cell(row=i, column=19, value=0) # best
            sheet.cell(row=i, column=20, value=0) # botb

            if to_number(sheet.cell(row=i, column=5).value) == 0:
                sheet.cell(row=i, column=18, value=200) # iter
                sheet.cell(row=i, column=19, value=200) # best
                sheet.cell(row=i, column=20, value=200) # botb

        for schet in range(19, kol + 1):
            cell_val = to_number(sheet.cell(row=schet, column=7).value)
            sheet.cell(row=schet, column=15, value=cell_val // 800)
            if sheet.cell(row=schet, column=15).value != 0:
                sheet.cell(row=schet, column=12, value=cell_val - 800 *
sheet.cell(row=schet, column=15).value)
            else:
                sheet.cell(row=schet, column=12, value=cell_val)

```

```
summdop = sum(to_number(sheet.cell(row=j, column=15).value) for j in
range(19, kol + 1))
```

```
for d in range(1, int(summdop) + 1):
    wb['Размещение'].cell(row=lis + 2, column=d + 2, value=800)
```

```
sheet = wb.worksheets[lis - 1]
```

```
i = 18
```

```
pak = int(summdop)
```

```
while to_number(sheet.cell(row=19, column=20).value) == 0:
```

```
    pak += 1
```

```
    sheet.cell(row=17, column=20, value=pak)
```

```
    ind = 0
```

```
    MegaSum = 0
```

```
    oldsum = 0
```

```
for iter in range(1, 101):
```

```
    MSum = -1000000
```

```
    sheet.cell(row=17, column=17, value=iter)
```

```
for i in range(19, kol + 1):
```

```
    if to_number(sheet.cell(row=i, column=18).value) == 0:
```

```
        if to_number(sheet.cell(row=i, column=16).value) != 0:
```

```
            sheet.cell(row=i, column=16, value=sheet.cell(row=i,
column=16).value - 1)
```

```
        else:
```

```
            sheet.cell(row=i, column=18, value=pak)
```

```
            Sum = sum(to_number(sheet.cell(row=j, column=12).value) for j
in range(19, kol + 1) if to_number(sheet.cell(row=j, column=18).value) == pak)
```

```
            Shtraf = sum(to_number(sheet.cell(row=j, column=17).value) for
j in range(19, kol + 1) if to_number(sheet.cell(row=j, column=18).value) == pak)
```

```
            if Sum <= 800:
```

```
                Sum *= (1 - Shtraf / 50)
```

```
                Shtraf = 0
```

```
            if Sum > MSum:
```

```
                MSum = Sum
```

```
                ind = i
```

```
            sheet.cell(row=i, column=18, value=0)
```

```
            sheet.cell(row=ind, column=18, value=pak)
```

```
            if oldsum == MSum:
```

```

sheet.cell(row=ind, column=16, value=20)
    sheet.cell(row=ind, column=17, value=sheet.cell(row=ind,
column=17).value + 1)
    sheet.cell(row=ind, column=18, value=0)

    dead = 19
    while to_number(sheet.cell(row=dead, column=18).value) != pak:
        dead += 1
        if dead > 261:
            ind = 0
            Count = sum(1 for i in range(19, 262) if
to_number(sheet.cell(row=i, column=18).value) == 0 and
to_number(sheet.cell(row=i, column=16).value) == 0)
            if Count == 0:
                break
            else:
                ind = dead
        if MegaSum < MSum:
            for k in range(19, kol + 1):
                sheet.cell(row=k, column=20, value=pak if
to_number(sheet.cell(row=k, column=18).value) == 0 else 0)
            MegaSum = MSum
            sheet.cell(row=17, column=18, value=MegaSum)
            if 800 - MegaSum < 0.5:
                break

    oldsum = MSum

    for k in range(19, kol + 1):
        sheet.cell(row=k, column=16, value=0)
        sheet.cell(row=k, column=17, value=0)

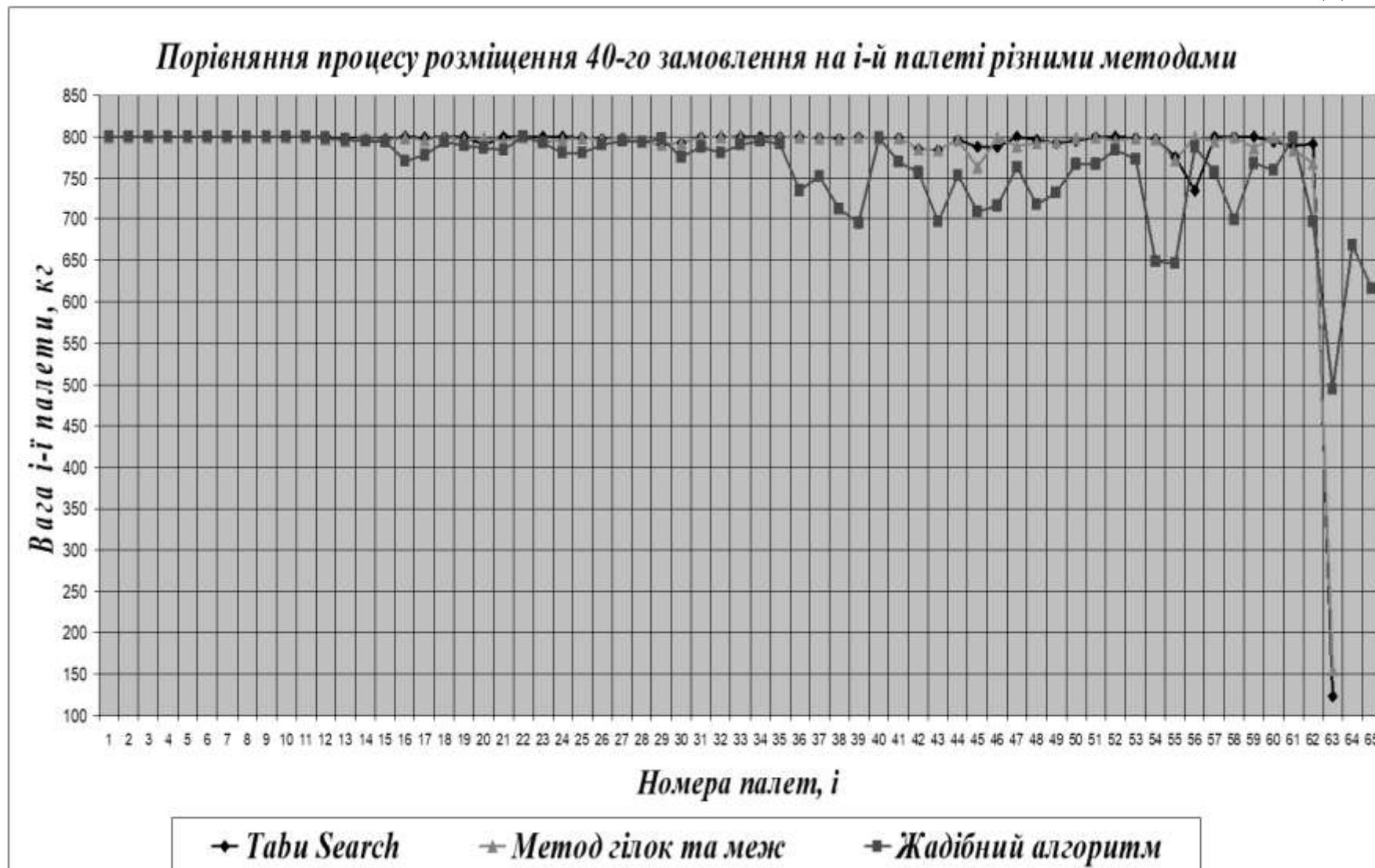
    if MegaSum == 0:
        continue

    wb['Размещение'].cell(row=2 + lis, column=2 + pak, value=MegaSum)
    for k in range(19, kol + 1):
        sheet.cell(row=k, column=18, value=sheet.cell(row=k,
column=20).value if to_number(sheet.cell(row=k, column=20).value) != 0 else 0)

    wb.save(file_path)

# Example usage
process_excel('in.xlsx')

```



Відгук

на кваліфікаційну роботу бакалавра

студента(ки) групи 124 – XX – X
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: _____

Обсяг кваліфікаційної роботи _____ стор.
Мета кваліфікаційної роботи: _____

Актуальність теми _____

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки _____

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра.
Оригінальність наукових рішень полягає в _____

Практичне значення результатів кваліфікаційної роботи полягає в _____

Висновки підтверджують можливість використання результатів роботи в _____

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі (*в разі невідповідності – вказати*)
У роботі відзначено такі недоліки: _____

Кваліфікаційна робота в цілому заслуговує оцінки: _____

З урахуванням висловлених зауважень автор (не) заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Керівник кваліфікаційної роботи бакалавра,
науковий ступінь, вчене звання, посада _____ / ППБ

Шаблон оформлення рецензії на кваліфікаційну роботу

Рецензія

на кваліфікаційну роботу бакалавра

студента групи 124 – 20 – 1

спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи:

Обсяг кваліфікаційної роботи:

Висновок про відповідність кваліфікаційної роботи завданню та освітньо-професійній програмі спеціальності

Загальна характеристика кваліфікаційної роботи, ступінь використання

нормативно-методичної літератури та передового досвіду

Позитивні сторони кваліфікаційної роботи:

Основні недоліки кваліфікаційної роботи:

Кваліфікаційна робота в цілому заслуговує оцінки:

З урахуванням висловлених зауважень автор (не) заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Рецензент,
науковий ступінь, вчене звання, посада _____ / ПІБ

Додаток Ж:

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки		
1									
2					Документація				
3									
4	САУ.КР.2024.ЗЗ.ПЗ				Пояснювальна записка	№1	Формат А4		
5									
6					Демонстраційний матеріал	№2	Презентація на CD-R		
7									
8					Копія роботи	1	Диск CD-R		
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
					САУ.КР.УУ.ЗЗ.ДА.ПЗ.				
Змін.	Аркуш	№ докум.	Підпис	Дата					
Розроб.		ПІБ				Літ.	Аркуш	Аркушів	

К. розд.	ПІБ		
Керівн.	ПІБ		
Н.контр.	ПІБ		
Зав. каф.	ПІБ		

**Матеріали
кваліфікаційної
роботи**

НТУ «ДП», 12; 124-18-1				