



ДИПЛОМНА РОБОТА МАГІСТРА



ДРОГОБИЧ – 2024

ДВНЗ «Донецький національний технічний університет»
Факультет машинобудування, електроінженерії та хімічних технологій
Кафедра електричної інженерії

«До захисту допущено»

Завідувач кафедри

 (підпис) О. КОЛЛАРОВ
 (ініціали, прізвище)
 «__» _____ 2024 р.

Кваліфікаційна робота магістра

на тему Оптимізація структури електричної розподільної мережі на базі еволюційних методів

Виконав студент 2 курсу, групи ЕЛКм-23
 (шифр групи)

спеціальності підготовки 141 «Електроенергетика, електротехніка та електромеханіка»
 (шифр і назва спеціальності підготовки)

Іван КОРСУН
 (ім'я та прізвище)

 (підпис)

Керівник зав. каф., к. т. н., доц., О. КОЛЛАРОВ
 (посада, науковий ступінь, вчене звання, ініціали, прізвище)

 (підпис)

Рецензент _____
 (посада, науковий ступінь, вчене звання, ініціали, прізвище)

 (підпис)

Нормоконтроль:

Засвідчую, що у цій випускній кваліфікаційній роботі немає запозичень з праць інших авторів без відповідних посилань.

 (підпис) **Е. НЕМЦЕВ**
16.12.2024
 (дата)

 (підпис)
16.12.2024
 (дата)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ ВИЩИЙ
НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

Факультет машинобудування, електроінженерії та хімічних технологій
Кафедра електричної інженерії
Освітній ступінь: магістр
Спеціальність: електроенергетика, електротехніка та електромеханіка

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ (О. КОЛЛАРОВ)

« » _____ 2024 р.

**ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Івану КОРСУНУ

(ім'я та прізвище)

1. Тема роботи: Оптимізація структури електричної розподільної мережі на базі еволюційних методів.

керівник роботи Олександр КОЛЛАРОВ, канд. техн. наук, доцент
(ім'я та прізвище, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від _____ № _____

2. Строк подання студентом роботи 17 грудня 2024 року

3. Вихідні дані до роботи: Система що проєктується має 5 джерел живлення потужністю (S_i): 50, 100, 150, 200 і 300 [у.о.п.] та 7 споживачів потужністю (S_j): 30, 60, 90, 120, 150, 170, 180 [у.о.п.]. Питомі витрати на передачу потужності по електричних мережах між i -им джерелом та j -им споживачем визначаються за формулою: $C_{ij} = 1 + 1/(|\sin(S_i) \times \cos(S_j)|)$ [у.о.в./у.о.п.], де: у.о.п. – умовна одиниця потужності, а у.о.в. – умовна одиниця вартості. Первинна схема електричної мережі представляє собою усі можливі комбінації з'єднання джерел живлення і споживачів електричної енергії.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз методів оптимізації електричних розподільних мереж. Особливості застосування еволюційних методів в задачах оптимізації електричних мереж.

2. Розв'язання поставленого завдання шляхом застосування методу найменшої

питомої вартості із покращенням результату через розподільний метод.

3. Розв’язання поставленого завдання через застосування еволюційних методів.

4. Аналіз отриманих, шляхом математичного моделювання, результатів оптимізації електричної розподільної мережі. Висновки по пунктах завдання.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень, якщо передбачається)

Вісім-десять слайдів презентаційного матеріалу, орієнтовно.

6. Консультанти розділів роботи

Розділ	Ініціали, прізвище та посада консультанта	Підпис, дата	
		завдан ня видав	завдан ня прийня в
Розділи 1 – 4	О. КОЛЛАРОВ, зав. каф.		
Нормоконтроль	Е. НЕМЦЕВ, ст. викл. каф.		

7. Дата видачі завдання 30 вересня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Розділ 1	30.09.24 – 13.10.24	
2.	Розділ 2	14.10.24 – 10.11.24	
3.	Розділ 3	11.11.24 – 11.12.24	
4.	Розділ 4	12.12.24 – 17.12.24	

Студент _____
(підпис)

Іван КОРСУН
(ім’я та прізвище)

Керівник роботи _____
(підпис)

Олександр КОЛЛАРОВ
(ім’я та прізвище)

АНОТАЦІЯ

Іван КОРСУН. Оптимізація структури електричної розподільної мережі на базі еволюційних методів / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 141 Електроенергетика, Електротехніка та Електромеханіка.– ДВНЗ ДонНТУ, Дрогобич, 2024.

Дипломна робота складається з вступу, основної частини, яка включає чотири розділи, висновків, списку застосованих джерел, та додатків.

У першому розділі розглянуто поняття транспортної задачі, оптимізації та методи його досягнення.

У другому розділі виконано розрахунок виконано розрахунок оптимізації з'єднань класичним методом та отримано схему мережі, а також виявлено можливість зниження вартості пересилання.

У третьому розділі розглянуто поняття генетичного алгоритму транспортної задачі, та принцип його роботи в середовищі MATLAB.

У четвертому розділі було здійснено розрахунок транспортної задачі та оптимізації електричної системи на базі генетичних алгоритмів та отримано нову схему електричної мережі з найменшими витратами.

Ключові слова: ТРАНСПОРТНА ЗАДАЧА, ГЕНЕТИЧНИЙ АЛГОРИТМ, РОЗПОДІЛЬНИЙ МЕТОД, БАЛАНС ПОТУЖНОСТІ.

ANNOTATION

Ivan KORSUN. Optimization of the structure of the electric distribution network based on evolutionary methods / Graduation qualification work for obtaining the degree of "master" in the specialty 141 Electric power engineering, Electrotechnics and Electromechanics.– State Higher Education Establishment of DonNTU, Drohobych, 2024.

The diploma work consists of an introduction, the main part, which includes four sections, conclusions, a list of used sources, and appendices.

The first section considers the concept of the transport problem, optimization and methods for achieving it.

In the second section, the calculation of the optimization of connections by the classical method is performed and a network diagram is obtained, and the possibility of reducing the cost of transmission is identified.

In the third section, the concept of the genetic algorithm of the transport problem is considered, and the principle of its operation in the MATLAB environment is considered.

In the fourth section, the calculation of the transport problem and optimization of the electric system based on genetic algorithms is performed and a new scheme of the electric network with the lowest costs is obtained.

Keywords: TRANSPORTATION PROBLEM, GENETIC ALGORITHM, DISTRIBUTION METHOD, POWER BALANCE.

ЗМІСТ

ВСТУП	8
1 ЗАГАЛЬНІ ПОНЯТТЯ ОПТИМІЗАЦІЇ СТРУКТУРИ ЕЛЕКТРИЧНОЇ РОЗПОДІЛЬНОЇ МЕРЕЖІ	9
1.1 Поняття оптимізації	9
1.2 Транспортна задача	14
1.3 Висновки за розділом.....	16
2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ	17
2.1 Оптимізація схеми з'єднань за допомогою класичного методу	17
2.2 Оптимізація з'єднань за допомогою розподільного методу	28
2.3 Висновки за другим розділом.....	32
3 ГЕНЕТИЧНИЙ АЛГОРИТМ ТРАНСПОРТНОЇ ЗАДАЧІ.....	33
3.2 Особливості генетичних алгоритмів	36
3.3 Робота з генетичним алгоритмом в середовищі MATLAB.....	41
3.4 Висновки за третім розділом	49
4 РОЗВ'ЯЗАННЯ ЗАДАЧІ ОПТИМІЗАЦІЇ ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ У СЕРЕДОВИЩІ MATLAB.....	50
4.1 Оптимізація результатів розрахунку класичного методу за допомогою MATLAB	50
4.2 Висновки за четвертим розділом.....	59
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	61
ДОДАТОК А – ГРАФІЧНА ЧАСТИНА.....	64
ДОДАТОК Б – Скрипт роботи генетичного алгоритму	67

ДОДАТОК В – Скрипт простої функції мінімізації	69
ДОДАТОК Г – Перелік зауважень нормоконтролера до дипломної роботи	71

ВСТУП

Найбільш універсальним та важливим видом енергії, що використовує людство є електрична. Широке використання у побуті, промисловості, транспорті пояснюється можливістю її перетворення на інші види енергії – теплову, механічну тощо, а також доступністю її довгострокового запасання.

Однак широке розповсюдження накладає певні ризики – у випадку припинення електропостачання з будь якої причини (стихійні лиха, людський фактор) може призвести до небажаних наслідків, в тому числі до колапсу важливих систем. Тому важливою задачею є оптимізація структури електричних мереж таким чином, щоб водночас відповідати критеріям надійності та зниження вартості передачі електроенергії.

Мета роботи – знайти ефективний метод з'єднання електричної мережі з найменшою вартістю передачі.

Завдання роботи:

- перерахувати методи оптимізації структури електричних розподільчих мереж,
- розрахувати умовну вартість передачі електроенергії класичним методом,
- створити математичну модель на базі генетичних алгоритмів,
- провести порівняння створеної математичної моделі на базі еволюційних методів з розрахунком, зробленим класичним методом.

Об'єкт досліджень – розподільні електричні мережі

Предмет досліджень – способи та методи оптимізації вартості передачі електричної енергії.

1 ЗАГАЛЬНІ ПОНЯТТЯ ОПТИМІЗАЦІЇ СТРУКТУРИ ЕЛЕКТРИЧНОЇ РОЗПОДІЛЬНОЇ МЕРЕЖІ

1.1 Поняття оптимізації

Оптимізацією називають процес пошуку найкращого з усіх можливих варіантів об'єкту критеріїв ефективності, що обумовлюють вибір параметрів, таких як конструктивних, енергетичних чи технологічних, що в свою чергу дають оптимальні або близькі до таких значень критеріїв оптимальності.

Поняття оптимальності означає такий варіант розв'язку завдань чи проблем котрий найбільш відповідає висунутим вимогам та умовам, та є собою найбільш кращий з усіх можливих. Перебіг оптимізації являється основою діяльності великої кількості галузей науки і техніки, в тому числі енергетичної частини промислового господарства. Вони ґрунтуються на теоріях оптимізації, що в своєму принципі складає собою сполучення математичних методів та насамперед, фундаментального та прикладного характеру. Ці методи, в тому числі велика кількість чисельних націлені на пошук та уподібнення експериментальних різновидів рішення досить різних задач, функціональних залежностей і так далі[1].

Звідси виходить, що основними завданнями оптимізації є знаходження таких факторів і визначення їх величин, при яких показники ефективності досліджуваного об'єкта досягнуть екстремального (максимального або мінімального) значення. До об'єктів оптимізації відносять план та розподілення електропостачання, структури виробництва електроенергії, режим роботи та параметри електрообладнання, електричних станцій та підстанцій, електричних мереж, і багатьох інших частин енергосистеми. Об'єкти оптимізації поділяють на наступні групи:

- об'єкти проектування,
- об'єкти розробки,
- об'єкти планування,

— об'єкти керування.

Важливою частиною при цьому є оптимізаційна модель, дослідження якої може дати можливість оволодіти основними закономірностями функціонування реальних об'єктів енергетичної системи, підібрати найбільш кращі способи керування та його застосування. В такому випадку отримані вирішення мають бути такими, щоб мати або забезпечувати суттєві переваги над іншими аналогічними рішеннями.

Для розв'язування задач з оптимізації та правильності в послідовності етапів, формалізація задач з оптимізації повинна включати наступні питання:

- постановка задачі,
- обирання критерію оптимізації,
- обирання керуючих (інша назва - змінних) факторів,
- записування функції мети,
- виділення множин допустимих значень змінних,
- остаточний аналіз змінних.

Оптимізаційна модель, що вирішує велику кількість різних технічно-економічних проблем в галузі електроенергетики, базується на розв'язуванні задач, що пов'язані з процесом пошуку максимумів чи мінімумів деякої функції. Встановлення оптимізаційної задачі в принципі є перекладенням на математичний вигляд висунутих вимог найкращого функціонування досліджуваної системи.

У більшості випадків постановка оптимізаційної задачі включає два математичні об'єкти, а саме множини допустимих розв'язків та цільову функцію, що є математичною моделлю об'єкту дослідження. В такому випадку перед науковцем виникає задача обирання показника функціонування об'єкта (критерій оптимальності, параметри оптимізації), а також формулювання задач на з'ясування максимуму або мінімуму цього критерію із врахуванням необхідних для умови обмежень.

1.2 Основні методи оптимізації енергетичних систем

Для оптимізації функціонування енергетичної системи застосовуються аналітичні а також чисельні методи. Перерахуємо основні:

- метод покрокової оптимізації - цей метод базується на покроковому відборі найбільш оптимальних значень змінних у системі, з ціллю досягнення мінімуму чи навпаки максимуму цільової функції, кожен крок включає аналіз на чутливість цільової функції та коригування за необхідністю значення змінних з метою отримання найбільш кращого результату,

- метод динамічного програмування - він слугує для оптимізації режиму роботи систем шляхом розділення розв'язуваної задачі на частини (називають підзадачами) та розв'язання їх послідовно, названий метод дозволяє в ході дослідження знайти глобальні оптимуми для заданого режиму роботи енергетичної системи шляхом розв'язування менших за об'ємом,

- метод оптимізаційних задач,

- метод нелінійного програмування – це метод оптимізації енергетичних систем, застосовується для розв'язування більш складних задач (у порівнянні з покроковим та динамічними методами), даним методом користується для знаходження оптимальних режимів роботи енергетичних систем з урахуванням різноманітних обмежень і цілей. Основним параметром, що відрізняє його від лінійного та нелінійного програмування є такий факт, що нелінійне програмування допускає враховувати нелінійні залежності поміж змінними, що в свою чергу впливає на функцію витрат чи цільову функцію. Нелінійне програмування може бути використане для оптимального управління режимами роботи електричних мереж, включаючи оптимальне розподілення навантаження, визначення оптимального розміщення та розміру компенсуючих пристроїв, управління режимами переключення та багато іншого. Для розв'язання задач

нелінійного програмування в електроенергетиці використовуються різні методи, такі як методи градієнтного спуску, методи ітераційної релаксації, генетичні алгоритми та інші,

— метод табу-пошуку. Названий метод використовують для розв'язування задачі комбінаторної оптимізації. Даний ітеративний алгоритм пошуку характеризується використанням гнучкої пам'яті. Основною особливістю методу табу-пошуку є зберігання та використання інформації про попередньо зроблені розв'язання та використання їх для уникання повторного переходу до розв'язків що вже були. Інформація щодо заборонених розв'язків зберігається у списку та включає заборонені кроки, що не можуть виконуватися протягом певного заданого періоду часу. Під час табу-пошуку виключається відбір початкового розв'язку, створюється початковий список табу, та проводиться пошук оптимальніших розв'язків, проходить оновлення списку табу та оцінюється найбільш кращий з усіх знайдених розв'язків. В енергетичних системах, методи табу-пошуку використовуються для розв'язання задач економічної диспетчеризації, грамотного розподілу ресурсів, оптимізація структури та параметрів розподільчих мереж, оптимізація режимів роботи системи та інших. Спосіб дає змогу досліднику знаходити найбільш ефективні рішення з урахуванням обмежень за надійністю та витратами режимні параметри енергетичних систем,

— еволюційний метод розрахунку на базі генетичних алгоритмів. У зв'язку з зростанням обчислювальних можливостей отримав високий розвиток метод інтелектуального пошуку оптимального розв'язання та нечіткої множини. Цей метод відрізняється великою універсальністю а також адаптивністю до задачі різних типів, наприклад для випадку розв'язання задачі економічної диспетчеризації та оптимізації потоків електроенергії в енергосистемах,

— еволюційні алгоритми. Вони також мають значну різницю у порівнянні зі звичайними методами оптимізації. Для застосовування не потрібно щоб цільова функція і обмеження були диференційованими по незалежних змінних задачі. Для

розв'язання достатньо володіти можливістю швидко і також однозначно оцінити оптимальність стану енергетичної системи після коригування незалежних змінних. Еволюційний метод, маючи у складі еволюційне програмування та генетичний алгоритм, відносять до методів штучного інтелекту для цілей оптимізації, що в свою чергу заснованими на процесах природного відбору.

Застосування оптимізаційних нейронних мереж у минулому мало місце для розв'язування задачі лінійного програмування. Однак надалі оптимізаційні нейронні мережі були адаптовані для розв'язування нелінійної задачі. В свою чергу система дає змогу застосувати нейронну мережу в задачі оптимізації електроенергетичної системи. Застосування нейронних мереж значно відрізняється від традиційних методів оптимізації, адже названі мережі перетворюють вирішення задачі оптимізації на пошуки точки рівноваги чи рівноважний стан нелінійної динамічної системи і проводять змінення критерію оптимальності на енергетичні функції для динамічної системи. Нейронні мережі виявили застосування для розв'язання задач економічної диспетчеризації енергосистеми. Важливим рисою нейронних мереж є здатність адаптування до змінення умов і вихідних даних розрахунку, що властиво для оптимізаційних задач енергетичних систем. Нейронна мережа здатна самостійно коригувати свої параметри в процесі роботи, для досягнення оптимального рішення оптимізаційної задачі.

В процесі оптимізації енергетичних систем еволюційний алгоритм має застосування для розв'язування різноманітних оптимізаційних задач, наприклад:

- економічна диспетчеризація,
- планування розвитку електромереж,
- розподіл потоків енергії та інші.

Еволюційний алгоритм є досить ефективним під час розв'язку задачі з дуже великою кількістю змінних, нелінійним обмеженням і багатьма локальними мінімумами. Названий алгоритм забезпечує пошук оптимального рішення, що

враховують економічний, екологічний, технічний та критерій, сприяючи покращенню стабільності а також ефективності енергосистеми, що досліджується.

1.2 Транспортна задача

Транспортна задача – це така задача, в якій цільовою функцією є пошук шляхів мінімізації довжини розподільчих електричних мереж, зниження вартості передачі електроенергії від джерел до споживачів, оптимізування витрат на схему електричної мережі [2].

Застосуванням транспортної задачі є система електропостачання, в якій задано кількість вузлів джерел живлення (позначено i ; $i= 1,2 \dots n$) та кількості вузлів споживання (позначено j ; $j= 1,2 \dots m$).

Потужність джерела становить A_i , в свою чергу потужність кожного споживача позначено як B_j . Потужність в задачі позначається в умовних одиницях (у.о.п.).

Схема, взаємне розміщення джерел живлення та споживачів а також вартість передачі одиниць потужності від джерел живлення i до споживача j має позначення C_{ij} (у.о.п./у.о.в). В свою чергу загальна кількість можливих варіантів з'єднань становить $n \cdot m$.

Потужності що передаються по названих мережах являють собою шукані змінні X_{ij} , а їх кількість становить $n \cdot m$. Загальні витрати в електричній мережі дорівнює сумі добутку питомих потужностей на величину потужностей що передаються від джерел живлення до споживачів. За вказаних умов функція має такий вигляд:

$$\sum_{i=1}^n \cdot \sum_{j=1}^m C_{ij} \cdot X_{ij} \rightarrow \min, \quad (1.1)$$

Для кожного i джерела сума потужностей що надходять по лініях до споживачів, складає потужність A_i джерела:

$$\sum_{j=1}^n x_{ij} = A_i, \quad (1.2)$$

При розв'язання транспортної задачі складають табличну форму запису. В цьому випадку обмеження записуються у вигляді транспортної матриці розмірністю $n \cdot m$ (в нашому випадку $5 \cdot 7$). Кожний i й рядок відповідає балансу потужності i го джерела живлення, а кожний j й стовпчик – рівнянню балансу споживача. В свою чергу для кожного j споживача сума потужностей що надходять від всіх джерел що надходять від всіх джерел живлення B_j :

$$\sum_{i=1}^m x_{ij} = B_j \quad (1.3)$$

Наведені формули є балансом потужності з граничними умовами:

$$X_{ij} \geq 0 ; i = 1, 2 \dots n ; j = 1, 2 \dots m \quad (1.4)$$

Алгоритм транспортної задачі виглядає наступним чином. Допустимий розв'язок отримується за методом мінімальної питомої вартості. В транспортній матриці обираємо клітинку з найменшим C_{ij} . Якщо таких значень декілька то можна взяти будь – яку. У цю клітинку як базисну змінну заноситься найменша з величин A_i або B_j . Після цього береться друге найменше число C_{ij} , та у обрану клітинку вставляється різниця між A_i і отриманим значенням X_{ij} . Інші

незаповнені клітини у рядку встановлюються зі значенням $X_{ij} = 0$. Наступний рядок заповнюється аналогічно. У випадку заповнення стовпця в подальшому його можна ігнорувати. Подальше заповнення аналогічно, з врахуванням балансу по A_i та B_j .

1.3 Висновки за розділом

Еволюційні методи та методи на базі генетичних алгоритмів дозволяють знаходити оптимальні розв'язання у великих за об'ємом задачах, що мають складні обмеження, нелінійності, та задачі з багатьма змінними, за досить невелику кількість часу. Крім того, названі методи не мають обмежень у диференційованості постановки задач, а також здатні бути адаптовані до різних типів задач і критеріїв оптимальності.

Недоліками еволюційних методів розрахунку є потреба у застосування значних обчислювальних потужностей, та великі обчислювальні витрати під час аналізу задач.

В свою чергу, класичні методи не потребують великих обчислювальних потужностей, але придатні лише для порівняно простих задач. Варто відмітити обмеженість даного методу у випадку об'ємних і більш складних задач. У складних випадках потребується значні витрати часу порівняно з еволюційними методами.

2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ

2.1 Оптимізація схеми з'єднань за допомогою класичного метода

За умовами задачі немає обмежень на спосіб та кількість з'єднань, тому первинна схема буде виглядати наступним чином згідно рис.2.1.

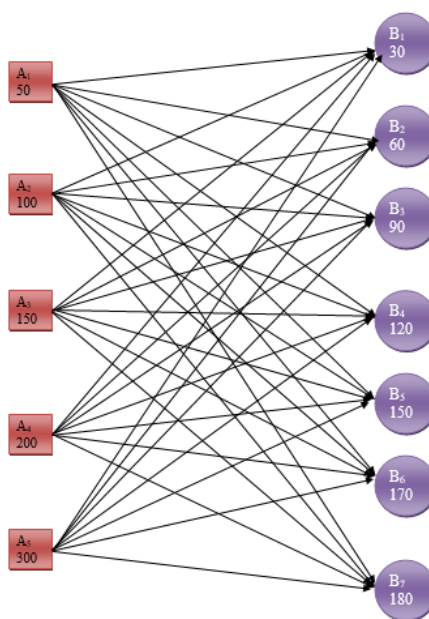


Рисунок 2.1 – Первинна схема з'єднань

Визначаємо вартість передачі всіх можливих варіантів з'єднання. Питомі витрати на передачу потужності визначаються за формулою:

$$C_{ij} = 1 + 1/(|\sin(S_i) \cdot \cos(S_j)|) \quad (2.1)$$

де S_i - потужність джерела живлення, у.о.п

S_j - потужність споживача, у.о.в.

Слід пам'ятати, що значення в формулі обираються в радіанах згідно даних табл 2.1 та табл 2.2.

Таблиця 2.1 – Вхідні дані потужності джерел

№	S_i , у.о.п
1	50
2	100
3	150
4	200
5	300

Таблиця 2.2 – Вхідні дані потужності споживачів

№	S_j , у.о.п
1	30
2	60
3	90
4	120
5	150
6	170
7	180

Звідси розрахуємо витрати потужності на всіх варіантах з'єднань, та згрупуємо отримані дані в табл. 2.3:

$$C_{11} = 1 + \frac{1}{|\sin 50 \cdot \cos 30|} = 25,7 [\text{у. о. в.} / \text{у. о. п.}]$$

Таблиця 2.3 – Коефіцієнти вартості пересилання потужності

	A_1 50	A_2 100	A_3 150	A_4 200	A_5 300
B_1 30	25,709	13,803	10,069	8,423	7,485
B_2 60	5,002	3,074	2,469	2,202	2,050
B_3 90	10	5	4	4	3
B_4 120	5,681	3,426	2,718	2,406	2,229
B_5 150	6,451	3,824	3,000	2,638	2,430
B_6 170	5,063	3,105	2,491	2,221	2,066
B_7 180	7,369	4,300	3,337	2,913	2,671

Складаємо рівняння балансу:

$$A_1 = x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17}$$

$$A_2 = x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27}$$

$$A_3 = x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37}$$

$$A_4 = x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47}$$

$$A_5 = x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57}$$

$$B_1 = x_{11} + x_{21} + x_{31} + x_{41} + x_{51}$$

$$B_2 = x_{21} + x_{22} + x_{32} + x_{42} + x_{52}$$

$$B_3 = x_{13} + x_{23} + x_{33} + x_{43} + x_{53}$$

$$B_4 = x_{14} + x_{24} + x_{34} + x_{44} + x_{54}$$

$$B_5 = x_{15} + x_{25} + x_{35} + x_{45} + x_{55}$$

$$B_6 = x_{16} + x_{26} + x_{36} + x_{46} + x_{56}$$

$$B_7 = x_{17} + x_{27} + x_{37} + x_{47} + x_{57}$$

За умовами задачі, змінна x_{ij} не може приймати від'ємне значення, але може дорівнювати нулю, або деякому додатному значенню.

$$x_{11} \geq 0, x_{12} \geq 0, x_{13} \geq 0, x_{14} \geq 0, x_{15} \geq 0, x_{16} \geq 0, x_{17} \geq 0, x_{21} \geq 0, x_{22} \geq 0, x_{23} \geq 0,$$

$$x_{24} \geq 0, x_{25} \geq 0, x_{26} \geq 0, x_{27} \geq 0, x_{31} \geq 0, x_{32} \geq 0, x_{33} \geq 0, x_{34} \geq 0, x_{35} \geq 0, x_{36} \geq 0,$$

$$x_{37} \geq 0, x_{41} \geq 0, x_{42} \geq 0, x_{43} \geq 0, x_{44} \geq 0, x_{45} \geq 0, x_{46} \geq 0, x_{47} \geq 0, x_{51} \geq 0, x_{52} \geq 0, x_{53} \geq 0,$$

$$x_{54} \geq 0, x_{55} \geq 0, x_{56} \geq 0, x_{57} \geq 0$$

Тепер складаємо транспортну матрицю, яку показано в табл.2.4.

Таблиця 2.4 – Початковий варіант транспортної матриці

x11 c11=25,709	x12 c12=5,002	x13 c13=10	x14 c14=5,681	x15 c15=6,451	x16 c16=5,063	x17 c17=7,369	A1=50
x21 c21=13,803	x22 c22=3,074	x23 c23=5	x24 c24=3,426	x25 c25=3,824	x26 c26=3,105	x27 c27=4,3	A2=100
x31 c31=10,069	x32 c32=2,469	x33 c33=4	x34 c34=2,718	x35 c35=3,0	x36 c36=2,491	x37 c37=3,337	A3=150
x41 c31=8,423	x42 c42=2,202	x43 c43=4	x44 c44=2,406	x45 c45=2,638	x46 c46=2,221	x47 c47=2,913	A4=200
x51 c51=7,485	x52 c52=2,050	x53 c53=3	x54 c54=2,229	x55 c55=2,43	x56 c56=2,066	x57 c57=2,671	A5=300
B1=30	B2=60	B3=90	B4=120	B5=150	B6=170	B7=180	C=0

Починаємо розрахунок з обирання найменшого C_{ij} в матриці. В даному випадку $C_{52} = 2,050$. Порівнюємо значення потужності споживача та джерела,

найменше з двох значень буде $B_2 = 60$, тому обираємо значення $x_{52} = 60$. Після цього на A_5 залишиться 240 у.о.п., отримане заносимо до табл. 2.5.

Вартість пересилання

$$C = 60 \cdot 2,050 = 151,2 \text{ [у. о. в. \setminus у. о. п.]}$$

Таблиця 2.5 – Розрахунок першого рядка

$x_{11}=0$ $c_{11}=25,709$	$x_{12}=0$ $c_{12}=5$	$x_{13}=0$ $c_{13}=10$	$x_{14}=0$ $c_{14}=5,681$	$x_{15}=0$ $c_{15}=6,451$	$x_{16}=0$ $c_{16}=5,063$	$x_{17}=0$ $c_{17}=7,369$	$A_1=50$
x_{21} $c_{21}=13,803$	$x_{22}=0$ $c_{22}=3,074$	x_{23} $c_{23}=5$	x_{24} $c_{24}=3,426$	x_{25} $c_{25}=3,824$	x_{26} $c_{26}=3,105$	x_{27} $c_{27}=4,3$	$A_2=100$
x_{31} $c_{31}=10,06$	$x_{32}=0$ $c_{32}=2,469$	x_{33} $c_{33}=4$	x_{34} $c_{34}=2,718$	x_{35} $c_{35}=3,0$	x_{36} $c_{36}=2,491$	x_{37} $c_{37}=3,337$	$A_3=150$
x_{41} $c_{41}=8,423$	$x_{42}=0$ $c_{42}=2,202$	x_{43} $c_{43}=4$	x_{44} $c_{44}=2,406$	x_{45} $c_{45}=2,638$	x_{46} $c_{46}=2,221$	x_{47} $c_{47}=2,913$	$A_4=200$
x_{51} $c_{51}=7,485$	$x_{52}=60$ $c_{52}=2,050$	x_{53} $c_{53}=3$	x_{54} $c_{54}=2,229$	x_{55} $c_{55}=2,43$	x_{56} $c_{56}=2,066$	x_{57} $c_{57}=2,671$	$A_5=240$
$B_1=30$	$B_2=60$	$B_3=90$	$B_4=120$	$B_5=150$	$B_6=170$	$B_7=180$	$C=250$

Незадіяні клітинки у рядках та стовпчика де вичерпано потужність джерела або споживача заповнюють нулями, що відповідають вільним змінним.

Далі наступне найменше значення $C_{56}=2,066$, з урахуванням залишку $A_5=240$ значення $x_{56}=170$, оскільки $B_2=170$ є найменшим значенням в табл. 2.6.

Залишок джерела:

$$A_5 = 240 - 170 = 70$$

Вартість пересилання

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 = 502,42 \text{ [у. о. в. \у. о. п.]}$$

Таблиця 2.6 – Розрахунок першого рядка

x11 c11=25,709	x12=0 c12=5,002	x13 c13=10	x14 c14=5,681	x15 c15=6,451	x16=0 c16=5,063	x17 c17=7,369	A1=50
x21 c21=13,803	x22=0 c22=3,074	x23 c23=5	x24 c24=3,426	x25 c25=3,824	x26=0 c26=3,105	x27 c27=4,3	A2=100
x31 c31=10,069	x32=0 c32=2,469	x33 c33=4	x34 c34=2,718	x35 c35=3	x36=0 c36=2,491	x37 c37=3,337	A3=150
x41 c41=8,42	x42=0 c42=2,202	x43 c43=4	x44 c44=2,406	x45 c45=2,638	x46=0 c46=2,221	x47 c47=2,913	A4=200
x51 c51=7,485	x52=60 c52=2,050	x53 c53=3	x54 c54=2,229	x55 c55=2,43	x56=170 c56=2,066	x57 c57=2,671	A5=70
B1=30	B2=00	B3=90	B4=120	B5=150	B6=00	B7=180	C=502,42

Наступне найменше значення $C_{54}=2,22$. Порівнюємо значення A_5 та B_7 бачимо що найменше значення приймає $A_5 = 70$, том $x_{54} = 70$, отримане занесемо в табл. 2.7.

Залишок B_4 :

$$B_4 = 120 - 70 = 50$$

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 = 689,32 \text{ [у. о. в. \у. о. п.]}$$

Таблиця 2.7 – Розрахунок першого рядка

x11 c11=25,709	x12=0 c12=5,002	x13 c13=10	x14 c14=5,681	x15 c15=6,451	x16=0 c16=5,063	x17 c17=7,369	A1=50
x21 c21=13,803	x22=0 c22=3,074	x23 c23=5	x24 c24=3,426	x25 c25=3,824	x26=0 c26=3,105	x27 c27=4,3	A2=100
x31 c31=10,069	x32=0 c32=2,469	x33 c33=4	x34 c34=2,718	x35 c35=3	x36=0 c36=2,491	x37 c37=3,337	A3=150
x41 c31=8,423	x42=0 c42=2,202	x43 c43=4	x44 c44=2,406	x45 c45=2,638	x46=0 c46=2,221	x47 c47=2,913	A4=200
x51=0 c51=7,485	x52=60 c52=2,050	x53=0 c53=3	x54=70 c54=2,229	x55=0 c55=2,43	x56=170 c56=2,066	x57=0 c57=2,671	A5=0
B1=30	B2=60	B3=90	B4=50	B5=150	B6=170	B7=180	C=657,8

В подальшому розрахунку найменшим буде $C_{44}=2,406$. З урахуванням залишку $B_4 = 50$, значення $x_{44} = 50$, а залишок A_4 :

$$A_4 = 200 - 50 = 150$$

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 + 50 \cdot 2,406 = 809,6$$

Отримане занесемо в табл. 2.8.

Таблиця 2.8 – Розрахунок другого рядка

x11 c11=25,709	x12=0 c12=5	x13 c13=10	x14 c14=5,681	x15 c15=6,451	x16=0 c16=5,063	x17 c17=7,369	A1=50
x21 c21=13,803	x22=0 c22=3,074	x23 c23=5	x24 c24=3,426	x25 c25=3,824	x26=0 c26=3,105	x27 c27=4,3	A2=100
x31 c31=10,069	x32=0 c32=2,469	x33 c23=4	x34 c34=2,718	x35 c35=3	x36=0 c36=2,491	x37 c37=3,337	A3=150
x41 c31=8,423	x42=0 c42=2,202	x43 c43=4	x44=50 c44=2,406	x45 c45=2,638	x46=0 c46=2,221	x47 c47=2,913	A4=150
x51=0 c51=7,485	x52=60 c52=2,050	x53=0 c53=3	x54=70 c54=2,229	x55=0 c55=2,43	x56=170 c56=2,06	x57=0 c57=2,67	A5=00
B1=30	B2=60	B3=90	B4=00	B5=150	B6=170	B7=180	C=809,6

Далі у матриці знаходимо мінімальне значення що становить $C_{45}=2,63$, A_4 та B_5 мають однакові значення в 150 у.о.п., тому $x_{45} = 150$, занесемо в табл. 2.9.

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 + 50 \cdot 2,406 + 150 \cdot 2,63 \\ = 1204,1 \text{ [у. о. в. \setminus у. о. п.]}$$

Шукаємо наступне значення найменшої вартості, ним буде $C_{37} = 3,337$. Найменшим значенням потужності ($A_3 = 150$; $B_7 = 180$) є A_3 , тому $x_{37} = 150$ згідно даних табл. 2.30. Залишок потужності на B_7 :

$$B_7 = 180 - 150 = 30$$

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 + 50 \cdot 2,406 + 150 \cdot 3,337 + 2,63 \cdot 150 \\ = 1703,6[\text{у. о. в.} \backslash \text{у. о. п.}]$$

Таблиця 2.9 – Розрахунок третього рядка

x11 c11=25,709	x12=0 c12=5	x13 c13=10	x14 =0 c14 =5,68	x15=0 c15=6,451	x16=0 c16=5,063	x17 c17=7,369	A1=50
x21 c21=13,803	x22=0 c22=3,074	x23 c23=5	x24=0 c24=3,426	x25=0 c25=3,824	x26=0 c26=3,105	x27 c27=4,3	A2=100
x31 c31=10,069	x32=0 c32=2,469	x33 c23=4	x34 =0 c34=2,718	x35=0 c35=3	x36=0 c36=2,491	x37 c37=3,337	A3=150
x41=0 c31=8,423	x42=0 c42=2,202	x43=0 c43=4	x44=50 c44=2,406	x45=150 c45=2,638	x46=0 c46=2,221	x47=0 c47=2,913	A4=00
x51=0 c51=7,485	x52=60 c52=2,050	x53=0 c53=3	x54 =70 c54=2,229	x55=0 c55=2,43	x56 =170 c56=2,066	x57=0 c57=2,671	A5=00
B1=30	B2=60	B3=90	B4=00	B5=00	B6=170	B7=180	C=1204,1

Таблиця 2.10- Розрахунок четвертого рядка

x11 = c11=25,7	x12 =0 c12=5	x13 c13=10	x14 =0 c14=5,681	x15=0 c15=6,451	x16=0 c16=5,063	x17 c17=7,369	A1=50
x21 c21=13,803	x22=0 c22=3,074	x23 c23=5	x24=0 c24=3,424	x25=0 c25=3,824	x26=0 c26=3,105	x27=30 c27=4,3	A2=100
x31 =0 c31=10,069	x32=0 c32=2,491	x33=0 c23=4	x34 =0 c34=2,718	x35=0 c35=3	x36=0 c36=2,491	x37 =150 c37=3,337	A3=00
x41=0 c31=8,423	x42=0 c42=2,202	x43 = 0 c43=4	x44=50 c44=2,406	x45=150 c45=2,638	x46=0 c46=2,221	x47 =0 c47=2,913	A4=00
x51=0 c51=7,485	x52=60 c52=2,050	x53=0 c53=3	x54 =70 c54=2,229	x55=0 c55=2,43	x56 =170 c56=2,066	x57 =0 c57=2,671	A5=00
B1=30	B2=00	B3=90	B4=00	B5=00	B6=00	B7=30	C=1703,6

Наступне мінімальне значення буде $C_{27} = 4,3$. Враховуючи що найменшим є значення $B_7 = 30$, приймаємо $x_{27} = 30$, та заносимо у табл. 2.10.

Залишок джерела A_2 :

$$A_2 = 100 - 30 = 70$$

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 + 50 \cdot 2,406 + 150 \cdot 3,337 + 2,63 \cdot 150 + 30 \cdot 4,3 = 1832,6 \text{ [у.о.в.\у.о.п.]}$$

Наступне мінімальне значення буде $C_{23} = 5$, найменше $A_2 = 70$, приймаємо $x_{23} = 70$ згідно даних табл. 2.11.

Залишок споживача B_3 :

$$B_3 = 90 - 70 = 20$$

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 + 50 \cdot 2,406 + 150 \cdot 3,337 + 2,63 \cdot 150 + 30 \cdot 4,3 + 70 \cdot 5 = 2182,6 \text{ [у.о.в.\у.о.п.]}$$

Таблиця 2.11 – Розрахунок четвертого рядка

x_{11} $c_{11}=25,709$	$x_{12}=0$ $c_{12}=5$	x_{13} $c_{13}=10$	$x_{14}=0$ $c_{14}=5,681$	$x_{15}=0$ $c_{15}=6,451$	$x_{16}=0$ $c_{16}=5,063$	$x_{17}=0$ $c_{17}=7,369$	$A_1=50$
$x_{21}=0$ $c_{21}=13,803$	$x_{22}=0$ $c_{22}=3,074$	$x_{23}=70$ $c_{23}=5$	$x_{24}=0$ $c_{24}=3,426$	$x_{25}=0$ $c_{25}=3,824$	$x_{26}=0$ $c_{26}=3,105$	$x_{27}=30$ $c_{27}=4,3$	$A_2=70$
$x_{31}=0$ $c_{31}=10,069$	$x_{32}=0$ $c_{32}=2,469$	$x_{33}=0$ $c_{33}=4$	$x_{34}=0$ $c_{34}=2,718$	$x_{35}=0$ $c_{35}=3$	$x_{36}=0$ $c_{36}=2,491$	$x_{37}=150$ $c_{37}=3,337$	$A_3=00$
$x_{41}=0$	$x_{42}=0$	$x_{43}=0$	$x_{44}=50$	$x_{45}=150$	$x_{46}=0$	$x_{47}=0$	$A_4=00$

c31=8,423	c42=2,202	c43=4	c44=2,406	c45=2,638	c46=2,221	c47=2,913	
x51=0	x52=60	x53=0	x54=70	x55=0	x56=170	x57=0	A5=00
c51=7,485	c52=2,050	c53=3	c54=2,229	c55=2,43	c56=2,066	c57=2,671	
B1=30	B2=00	B3=20	B4=00	B5=00	B6=00	B7=00	C=2182,6

Розраховуємо останній рядок. Почнемо з клітинки що має значення $C_{13} = 10$. Враховуємо що на споживача прийшло 70 у.о.п., тому нестача буде становити 20 у.о.п. Таке значення є найменшим, тому $X_{13} = 20$. В такому випадку остання незаповнена клітинка буде:

$$X_{11} = A_1 - X_{13} = 50 - 20 = 30$$

Вартість пересилання:

$$C = 60 \cdot 2,050 + 170 \cdot 2,066 + 70 \cdot 2,22 + 50 \cdot 2,406 + 150 \cdot 3,337 + 2,63 \cdot 150 + 30 \cdot 4,3 + 70 \cdot 5 + 20 \cdot 10 + 25,7 \cdot 30 = 3153,6 \text{ [у.о.в. \setminus у.о.п.]}$$

Розрахунок закінчено, всі дані розрахунку представлені в табл. 2.12.

Таблиця 2.12 – Результати розрахунку

x11=30	x12=0	x13=20	x14=0	x15=0	x16=0	x17=0	A1=50
c11=25,709	c12=5,002	c13=10	c14=5,681	c15=6,451	c16=5,016	c17=7,369	
x21=0	x22=0	x23=70	x24=0	x25=0	x26=0	x27=30	A2=100
c21=13,803	c22=3,074	c23=5	c24=3,426	c25=3,824	c26=3,105	c27=4,3	
x31=0	x32=0	x33=0	x34=0	x35=0	x36=0	x37=150	A3=150
c31=10,06	c32=2,469	c33=4	c34=2,718	c35=3	c36=2,491	c37=3,337	
x41=0	x42=0	x43=0	x44=50	x45=150	x46=0	x47=0	A4=200
c41=8,423	c42=2,202	c43=4	c44=2,406	c45=2,638	c46=2,221	c47=2,913	

$x_{51}=0$ $c_{51}=7,485$	$x_{52}=60$ $c_{52}=2,050$	$x_{53}=0$ $c_{53}=3$	$x_{54}=70$ $c_{54}=2,229$	$x_{55}=0$ $c_{55}=2,43$	$x_{56}=170$ $c_{56}=2,069$	$x_{57}=0$ $c_{57}=2,671$	$A_5=300$
$B_1=30$	$B_2=60$	$B_3=90$	$B_4=120$	$B_5=150$	$B_6=170$	$B_7=180$	$C=3153,6$

Схема з'єднань що отримана при розрахунку показана на рис. 2.2.

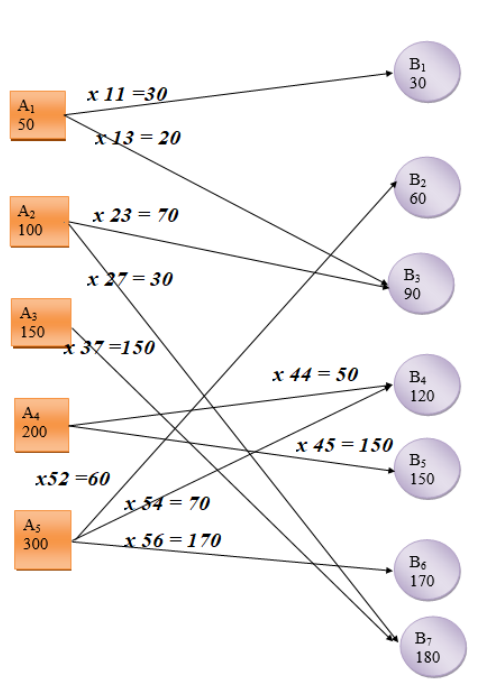


Рисунок 2.2 – Схема з'єднань з урахуванням отриманих результатів обчислень

2.2 Оптимізація з'єднань за допомогою розподільного методу

Цей метод застосовується для вдосконалення результатів розрахунку. Суть його в заміні вільних змінних. Проведемо розрахунок для уточнення розподільним методом. Довільно беремо клітинку x_{21} та встановлюємо його

значення +1. При цьому порушується баланс по стовпчику 1 та рядку 1. Для усунення розбалансу змінюємо на 1 значення сусідніх клітинок, встановивши $x_{21} = -29$ та $x_{13} = +21$. Після цього порушується баланс по рядку 2 та стовпчику 3. Для повернення балансу встановлюємо значення $x_{23} = -69$ та заносимо в табл. 2.13.

Зміна цільової функції при збільшенні змінної x_{21} становить:

$$\Delta C = C_{11} + C_{13} + C_{21} - C_{23} = -25,7 + 10 + 13,8 - 5 = -6,9 < 0$$

Таблиця 2.13 – Розрахунок розподільним методом рядка A_1

$x_{11} = -29$ $c_{11} = 25,709$	$x_{12} = 0$ $c_{12} = 5,002$	$x_{13} = +21$ $c_{13} = 10$	$x_{14} = 0$ $c_{14} = 5,681$	$x_{15} = 0$ $c_{15} = 6,451$	$x_{16} = 0$ $c_{16} = 5,016$	$x_{17} = 0$ $c_{17} = 7,369$	$A_1 = 50$
$x_{21} = +1$ $c_{21} = 13,803$	$x_{22} = 0$ $c_{22} = 3,074$	$x_{23} = -69$ $c_{23} = 5$	$x_{24} = 0$ $c_{24} = 3,426$	$x_{25} = 0$ $c_{25} = 3,824$	$x_{26} = 0$ $c_{26} = 3,105$	$x_{27} = 30$ $c_{27} = 4,3$	$A_2 = 100$
$x_{31} = 0$ $c_{31} = 10,069$	$x_{32} = 0$ $c_{32} = 2,469$	$x_{33} = 0$ $c_{33} = 4$	$x_{34} = 0$ $c_{34} = 2,718$	$x_{35} = 0$ $c_{35} = 3$	$x_{36} = 0$ $c_{36} = 2,491$	$x_{37} = 150$ $c_{37} = 3,337$	$A_3 = 150$
$x_{41} = 0$ $c_{41} = 8,423$	$x_{42} = 0$ $c_{42} = 2,202$	$x_{43} = 0$ $c_{43} = 4$	$x_{44} = 50$ $c_{44} = 2,406$	$x_{45} = 150$ $c_{45} = 2,638$	$x_{46} = 0$ $c_{46} = 2,221$	$x_{47} = 0$ $c_{47} = 2,913$	$A_4 = 200$
$x_{51} = 0$ $c_{51} = 7,485$	$x_{52} = 60$ $c_{52} = 2,050$	$x_{53} = 0$ $c_{53} = 3$	$x_{54} = 70$ $c_{54} = 2,229$	$x_{55} = 0$ $c_{55} = 2,430$	$x_{56} = 170$ $c_{56} = 2,069$	$x_{57} = 0$ $c_{57} = 2,671$	$A_5 = 300$
$B_1 = 30$	$B_2 = 60$	$B_3 = 90$	$B_4 = 120$	$B_5 = 150$	$B_6 = 170$	$B_7 = 180$	$C = 3153,6$

Отже, при збільшенні x_{21} значення цільової функції зменшується, що дає підставу перевести x_{21} у базис. Для цього значення що мають знак – зменшимо на 4, а + збільшимо на 4, це показано в табл. 2.14.

Таблиця 2.14 – Переведення змінних у базис у рядках A_1 та A_2

$x_{11} = 25$ $c_{11} = 25,709$	$x_{12} = 0$ $c_{12} = 5,002$	$x_{13} = 25$ $c_{13} = 10$	$x_{14} = 0$ $c_{14} = 5,681$	$x_{15} = 0$ $c_{15} = 6,451$	$x_{16} = 0$ $c_{16} = 5,016$	$x_{17} = 0$ $c_{17} = 7,369$	$A_1 = 50$
$x_{21} = 5$ $c_{21} = 13,803$	$x_{22} = 0$ $c_{22} = 3,074$	$x_{23} = 65$ $c_{23} = 5$	$x_{24} = 0$ $c_{24} = 3,426$	$x_{25} = 0$ $c_{25} = 3,824$	$x_{26} = 0$ $c_{26} = 3,105$	$x_{27} = 30$ $c_{27} = 4,3$	$A_2 = 100$
$x_{31} = 0$	$x_{32} = 0$	$x_{33} = 0$	$x_{34} = 0$	$x_{35} = 0$	$x_{36} = 0$	$x_{37} = 150$	$A_3 = 150$

c31=10,069	c32=2,469	c33=4	c34=2,718	c35=3	c36=2,491	c37=3,337	
x41=0	x42=0	x43=0	x44=50	x45=150	x46=0	x47=0	A4=200
c41=8,423	c42=2,202	c43=4	c44=2,406	c45=2,638	c46=2,221	c47=2,913	
x51=0	x52=60	x53=0	x54=70	x55=0	x56=170	x57=0	A5=300
c51=7,485	c52=2,050	c53=3	c54=2,229	c55=2,430	c56=2,069	c57=2,671	
B1=30	B2=60	B3=90	B4=120	B5=150	B6=170	B7=180	C=3153,6

Аналогічні дії можна виконати з x_{42} . При зміні значення на + 1 порушується баланс по 4 рядку та 2 стовпчику та четвертому стовпчику. Для його відновлення встановимо значення $x_{44} = -49$, $x_{52} = -59$, $x_{54} = +71$ згідно даних табл. 2.15.

Таблиця 2.15 – Балансування таблиці

x11=25	x12=0	x13=25	x14=0	x15=0	x16=0	x17=0	A1=50
c11=25,709	c12=5,002	c13=10	c14=5,681	c15=6,451	c16=5,016	c17=7,369	
x21=5	x22=0	x23=65	x24=0	x25=0	x26=0	x27=30	A2=100
c21=13,803	c22=3,074	c23=5	c24=3,426	c25=3,824	c26=3,105	c27=4,3	
x31=0	x32=0	x33=0	x34=0	x35=0	x36=0	x37=150	A3=150
c31=10,069	c32=2,469	c33=4	c34=2,718	c35=3	c36=2,491	c37=3,337	
x41=0	x42=+1	x43=0	x44=-49	x45=150	x46=0	x47=0	A4=200
c41=8,423	c42=2,202	c43=4	c44=2,406	c45=2,638	c46=2,221	c47=2,913	
x51=0	x52=-59	x53=0	x54=+71	x55=0	x56=170	x57=0	A5=300
c51=7,485	c52=2,050	c53=3	c54=2,229	c55=2,430	c56=2,069	c57=2,671	
B1=30	B2=60	B3=90	B4=120	B5=150	B6=170	B7=180	C=3153,6

Зміна цільової функції при збільшення змінної x_{42} становить:

$$\Delta C = C_{42} - C_{44} - C_{52} + C_{54} = 2,202 - 2,4 - 2,050 + 2,229 = -0,019$$

Робимо висновок, що при збільшенні x_{42} значення цільової функції зменшується, що дає підстави переводити x_{42} у базис. Як і в попередньому випадку, збільшуємо на 4 там де стоїть знак + та зменшуємо де -.

Результати розрахунку представлені в табл. 2.16.

Таблиця 2.16 – Результати розрахунку

x11 =25 c11=25,709	x12 =0 c12=5,002	x13=25 c13=10	x14=0 c14=5,681	x15=0 c15=6,451	x16=0 c16=5,016	x17=0 c17=7,369	A1=50
x21= 5 c21=13,803	x22 =0 c22=3,074	x23=65 c23=5	x24 =0 c24=3,426	x25 =0 c25=3,824	x26=0 c26=3,105	x27=30 c27=4,3	A2=100
x31=0 c31=10,069	x32=0 c32=2,469	x33=0 c33=4	x34=0 c34=2,718	x35=0 c35=3	x36=0 c36=2,491	x37=150 c37=3,337	A3=150
x41=0 c41=8,423	x42=5 c42=2,202	x43=0 c43=4	x44=45 c44=2,406	x45=150 c45=2,638	x46=0 c46=2,221	x47=0 c47=2,913	A4=200
x51 =0 c51= 7,485	x52=55 c52=2,050	x53=0 c53=3	x54=75 c54=2,229	x55=0 c55=2,430	x56=170 c56=2,069	x57 =0 c57=2,671	A5=300
B1=30	B2=60	B3=90	B4=120	B5=150	B6=170	B7=180	C=3153,6

Визначаємо вартість пересилання з урахуванням нового розв'язку:

$$C = 25,7 \cdot 25 + 10 \cdot 25 + 5 \cdot 13,8 + 65 \cdot 5 + 30 \cdot 4,3 + 150 \cdot 2,6 + 55 \cdot 2,050 + 75 \cdot 2,229 + 2,06 \cdot 170 = 3066$$

Оновлена схема з урахуванням показана на рис. 2.3.

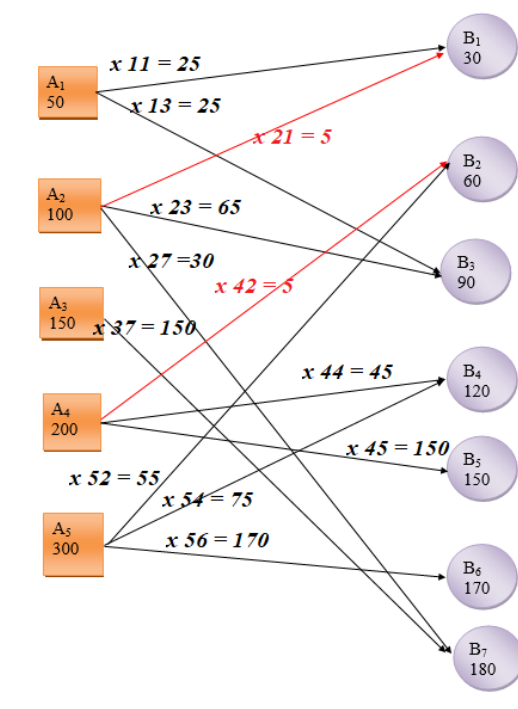


Рисунок 2.3 – Схема мережі з урахуванням змін що з'явилися після розрахунків розподільним методом

2.3 Висновки за другим розділом

В даному розділі було проведено розрахунок оптимізації з'єднань за допомогою класичних методів розрахунку (транспортна задача), та виведено умова балансу. В подальшому аналізі за допомогою розподільного методу було досягнуто зменшення вартості пересилання. Отримані результати свідчать про наявну можливість пониження витрат, але досить незначне. Отже, з цього виходить необхідність вдосконалення процесу пошуку оптимального розв'язку шляхом застосування генетичних алгоритмів.

3 ГЕНЕТИЧНИЙ АЛГОРИТМ ТРАНСПОРТНОЇ ЗАДАЧІ

3.1 Загальне поняття генетичного алгоритму

Генетичним алгоритмом називають евристичний алгоритм пошуку, що в свою чергу застосовується для розв'язання задач оптимізації та моделювання шляхом послідовного підбирання, комбінування та варіації параметрів з застосуванням механізмів: що схожі на еволюцію в галузі біології. Генетичний алгоритм є різновидом еволюційних числень, але головною відмінною рисою названого алгоритму є акцентування на використанні оператора що має назву «схрещування».

В свою чергу робить операцію рекомбінації розв'язків кандидатів, роль якого аналогічна ролі схрещування в живій природі.

При описі генетичних алгоритмів використовуються терміни, що були запозичені з розділу біології, а саме з генетики [3]. Наприклад, мова йде про популяцію осіб, а як базові поняття застосовуються ген, хромосома, генотип, фенотип, аллель. Також використовуються відповідні цим термінам слова з технічного лексикона, наприклад, ланцюг, двійкова послідовність, структура.

Генетичні алгоритми отримали досить широку застосування, а саме:

- задачі для розрахунку найкоротшого шляху,
- створення розкладу,
- задачі на пошук точок мінімуму та максимуму,
- добір вхідних даних,
- навчання штучної нейромережі та інше.

Популяція - це кінцева множина особин.

Особини, що входять у популяцію, у генетичних алгоритмах представляються хромосомами з закодованим у них множинами параметрів задачі, тобто розв'язань, які також називаються точками в просторі пошуку.

При описі генетичних алгоритмів використовуються терміни, запозичені з генетики, а саме:

- популяцією називають кінцеву множину осіб,
- особи – ті, що входять в популяцію, у генетичних алгоритмах представляють у вигляді хромосом з закодованими у їх складі множинами параметрів задач, тобто розв’язків що мають назву точок в просторі пошуку,
- хромосомами називають упорядковані послідовності генів,
- геном називають елемент геному (зокрема хромосому),
- генотипом (друга назва структура) називають набір хромосом особи,
- що вивчається в конкретному випадку,
- фенотипом називають набір значень, що відповідають даному генотипу, тобто декодована структура (інша назва множина параметрів задачі),
- аллеллю називають значення, конкретного гена, що також обумовлюють як значення властивості чи варіант властивості,
- поколінням називають чергову популяцію в генетичному алгоритмі, причому до знову створеної популяції застосовуються такий термін як «нове покоління».

Генетичне спадкування моделюється в наступному вигляді:

- хромосомою називають вектор (друга назва послідовність) з нулів та одиниць, кожний біт (інша назва позиція) називають геном,
- кросовером називають операцію при якій хромосоми обмінюються своїми частинами,
- мутацією називають випадкові зміни одного чи декількох позицій всередині хромосоми,
- індивідум = генетичний код – набір хромосом = варіанти розв’язання,
- геном називають відрізок ланцюга ДНК що відповідає за визначену властивість.

При розв’язанні задач застосовуються відомі природні способи оптимізації. В ході розв’язку розглядається кожен варіант, що має назву інвестування (набір

значень змінних) а прибутковість обраного варіанта – як пристосованість цього варіанта розв’язку. Таким чином в процесі еволюції пристосованість буде зростати, отже будуть з’являтися більш кращі та дохідні варіанти розв’язку.

Зупинивши еволюцію в потрібний момент, можна одержати найбільш вдале розв’язання задачі.

Основним механізмом еволюції є природний відбір. Його сутність полягає в принципі, при якому більш пристосовані особи мають набагато більші можливості для виживання, з чого виходить що вони приносять більше нащадків ніж ті особи що погано пристосувалися.

В такому випадку, завдяки передачі генетичної інформації (інша назва – генетичне спадкування) від батьків нащадки успадковують основні їх якості.

Таким чином можна зробити з цього деякі висновки :

— після зміни декількох (від одиниць, десятків до сотень) поколінь середня пристосованість досліджуваного виду зростає,

— нащадки сильних досліджуваних індивідів будуть також добре пристосованими а їх кількість в загальній масі досліджуваного середовища буде поступово зростати,

— локус або позиція вказує місце розміщення даного гена в хромосомі (ланцюжку). Множина позицій генів - це локи.

Дуже важливим поняттям у о генетичних алгоритмах вважається функція пристосованості, інакше називана функцією оцінки. Вона представляє міру пристосованості даної особи в популяції. Ця функція відіграє найважливішу роль, оскільки дозволяє оцінити ступінь пристосованості конкретних особин у популяції і обрати з них найбільш пристосовані (тобто найбільші значення функції, що мають пристосованості) відповідно до еволюційного принципу виживання «найсильніших» (найкраще пристосувалися). Функція пристосованості також одержала свою назву безпосередньо з генетики. Вона о впливає на функціонування генетичних алгоритмів і повинна мати точне і коректне означення. У задачах оптимізації функція пристосованості, як правило,

оптимізується (точніше кажучи, максимізується) і називається цільовою функцією. У задачах мінімізації цільова функція перетворюється, і проблема зводиться до максимізації. У теорії керування функція пристосованості може приймати вид функції похибки, а в теорії ігор - вартісної функції. На кожній ітерації генетичного алгоритму пристосованість кожної особи даної популяції оцінюється за допомогою функції пристосованості, і на цій основі створюється наступна популяція особей, що складають множину потенційних рішень проблеми, наприклад, задачі оптимізації.

3.2 Особливості генетичних алгоритмів

Генетичні алгоритми є достатньо новітнім, однак не єдиним зі способів розв'язку задач з оптимізації. Відомо два основних шляхи розв'язку задач – переборний та локально – градієнтним. Названі методи мають свої переваги та недоліки, через що необхідно ретельно обмірковувати, в якому випадку обирати конкретний метод розрахунку.

Отже, розглянемо недоліки та переваги класичних методів перебору розв'язку та генетичних алгоритмів на прикладі класичної задачі комівояжера.

Сутність названої задачі полягає в знаходженні найбільш короткого замкненого шляху обходу декількох точок, заданих певними координатами. Наприклад для 20 -30 точок оптимального шляху є досить складною задачею, що спонукає пошук та розвиток більш ефективних методів. Так, кожен варіант розв'язку (для 35) це числовий ряд, в якому на J місці стоїть номер J зв'язку. Таким чином в цій задачі нараховується 35 параметрів, однак не всі параметри припустимі.

На перший погляд можливим варіантом розв'язку є повний перебір всіх можливих варіантів обходу. Цей метод називають переборний.

Названий метод є найбільш простим та тривіальним у складанні програми для нього. Щоб знайти оптимальне розв'язання (точки мінімуму чи максимуму функції) необхідно послідовно обчислювати значення цільової функції, причому у всіх можливих точках та не забуваючи максимальне та мінімальне значення з них.

Недоліком названого методу є занадто велика обчислювальна вартість – адже у задачі комівояжера необхідно розрахувати довжину, що становить до 10^{35} можливих варіантів шляхів, що при певних обставинах відсутня можливість обчислення.

Інший спосіб застосовується на методі градієнтного спуску. В даному випадку обирають певні випадкові значення параметрів, після чого значення поступово змінюють для досягнення якомога більшої швидкості зростання цільової функції

У випадку досягання локального максимуму такий алгоритм зупиняється тому для знаходження оптимуму необхідно прикласти додаткових зусиль. Також важливо пам'ятати, що градієнтні методи мають велику швидкість обробки, однак не гарантують оптимальності знайдених розв'язків. Названий метод найбільш зручний у випадку «унімодальних задач», в яких цільова функція має лише один максимум (його також називають глобальним), його зображено на рис. 3.2.

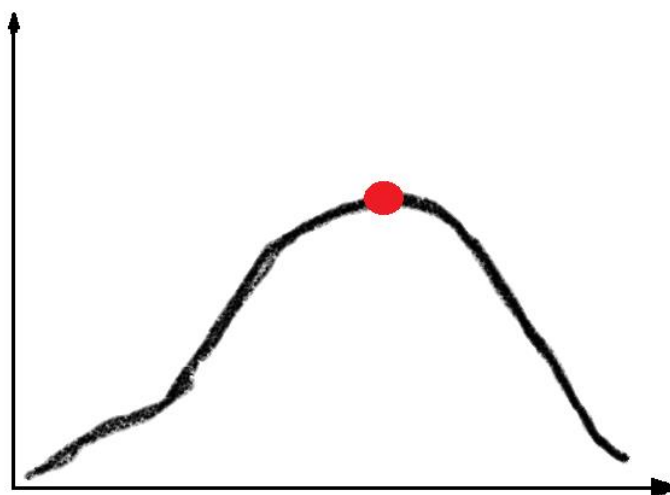


Рисунок 3.2 - Графік унімодальної задачі

Однак при комбінуванні переборного та градієнтного методу з'являється можливість отримання хоча б наближеного розв'язку, причому точність буде зростати при збільшенні часу за який проводиться розрахунок. Генетичний алгоритм є саме таким, у якому комбінуються перераховані методи. Механізми мутації та схрещування в певному місці роблять реалізацію переборної частини методу, а в свою чергу добір найкращих розв'язань – градієнтний спуск.

На рис. 3.3 можна побачити, що показана комбінація дозволяє забезпечити стійку та добру ефективність генетичного пошуку для будь-яких відомих типів задач.

Отже, на деякій множині задається складна функція, що має у складі багато змінних, то в такому випадку генетичний алгоритм є програмою, що за досить короткий час знаходить точку в якій значення функції достатньо близьке до максимально можливого. Вибираючи прийнятий час, за який буде проводитися розрахунок, ми одержимо одне з можливих кращих розв'язань, що можливо отримати за відведений час.

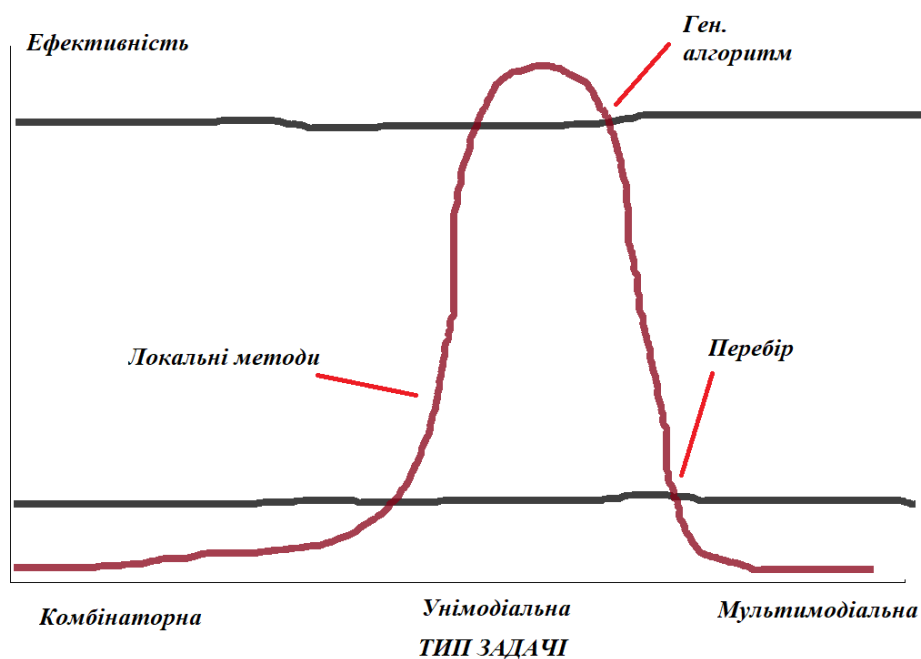


Рисунок 3.3 – Типи транспортної задачі.

Опис алгоритму розв'язку є приблизно наступним – при розв'язанні задачі вона кодується таким чином, щоб її розв'язки стало можливим представити у формі вектора (хромосоми). Далі випадковим чином створюють певну кількість початкових векторів (тобто початкову популяцію для розрахунку).

Створені вектори оцінюють з застосуванням функції пристосованості, як результат, кожному створеному вектору присвоюється обчислене значення пристосованості, що в свою чергу дає поняття про імовірність виживання модельованого організму, який представлений даним вектором.

Після виконання цих дій застосовуючи отримані значення пристосованості, обирають селекції (вектори) що можна допустити до подальшого схрещування, до отриманих векторів тепер можна застосувати генетичні оператори (найчастіше механізми схрещування, тобто мутації). Особи наступного покоління, що утворилася оцінюються, після чого проводиться селекція. Таким чином проходить моделювання процесу еволюції, що в свою чергу буде проходити доти, поки не буде застосований критерій зупинки алгоритму. При цьому може змінитися декілька поколінь. В свою чергу критерієм зупинки вважається:

- вичерпання числа поколінь, що застосовані для еволюції,
- вичерпання часу що було відведено на еволюцію,
- отримання відповідного розв'язку.

Таким чином, можна виділити такі етапи для генетичного алгоритму:

- створення початкової популяції,
- обчислювання функції пристосованості для популяції,
- селекція з поточної популяції індивідів,
- розрахунок пристосованості для всіх індивідів, що є в задачі,
- формування нового покоління,
- якщо виконуються умови задачі, то розрахунок припиняється, у випадку невідповідності умов цикл обчислень починається спочатку.

Початкова популяція перед першим кроком створюється в будь – якому випадковому вигляді. Навіть у випадку її повної неконкурентоспроможності генетичний алгоритм все одно здатен швидко перевести її в життєздатний стан. Таким чином на початку обчислень не має необхідності створювати добре пристосованих індивідів. Достатньо буде відповідність формату особин популяції, та вже на них з'являється можливість розраховувати функцію пристосованості. Підсумками цього кроку є популяція «Н» що складається з «N» особин.

Після створення початкової популяції настає процес відбору. На цьому етапі необхідно вибрати визначену певну її частку, що «залишиться в живих». Імовірність виживання особи h повинна залежати від значення функції пристосованості. Причому частка що вважається «вижившою» є параметром генетичного алгоритму, котра заздалегідь задається. За підсумками відбору з «N» особин популяції «Н» має залишитися «S» особин, що пізніше ввійдуть до підсумкової популяції «Н'». Інші що не ввійшли за умовами задачі «гинуть».

Розмноження в генетичних алгоритмах як правило статеве - для створення нащадка необхідно декілька батьків, найчастіше два. Розмноження в різних випадках може визначатися по різному, та зазвичай залежить від представлених даних. Найголовніша вимога до процесу розмноження – нащадки повинні успадкувати риси обох батьків, «змішавши» будь-яким раціональним способом. До мутацій відносять теж саме, що і розмноження, так є деяка частина мутантів m , що входить до параметру, та на кроці мутацій необхідно обрати m індивідів, після чого змінити їх у відповідності до визначених операцій з мутації.

Класичний генетичний алгоритм (також відомий під назвою елементарного або простого) складається з таких кроків:

- ініціалізація чи вибір вихідної популяції хромосоми,
- оцінювання пристосованості хромосоми у популяції,
- перевірка умов зупинки генетичного алгоритму,
- селекція хромосом,

- застосування генетичних операторів,
- сформування нової більш кращої популяції,
- відбір найкращої за умовами задачі хромосоми.

3.3 Робота з генетичним алгоритмом в середовищі MATLAB

В програмний пакет MATLAB входить вагомий функціонал, що дозволяє його застосувати для задач оптимізації [4]. Таким пакетом є Genetic Algorithm Tool.

Робота з Genetic Algorithm Tool виглядає наступним чином:

- створюємо новий скрипт, та вказуємо в ньому назву функції (може бути будь – яка, наприклад «simple_fitness»), у подальшому вказане буде також і назвою файлу, що присвоїться автоматично,
- далі у скрипті вказується функція, що буде оброблятися,
- наступним кроком вказують обмеження, початкові точки, кількість змінних (nvars),

Якщо скрипт складено вірно, то його можна запустити через команду «RUN», або іншим способом, а саме :

- у вкладці «APP» обираємо «Optimization». Відкриється нове вікно. Його можна умовно поділити на дві частини – ліву та праву,
- зліва знаходиться панель запуску та обирання типу обробки. В цьому вікні обираємо тип розв’язку (Solver) – «ga – Genetic Algorithm»,
- після цього вказуємо в рядку «Fitness function» назву функції (в нашому випадку це виглядає так – «@simple_fitness»),
- в рядку «numbers of variables» вказуємо кількість змінних,
- за необхідності можна ввести обмеження функції.

Так, у віконці «Constraints» - «Linear inequalities» вказують обмеження лінійної нерівності наступного формату:

$$A \cdot x \leq b \quad (3.1)$$

В свою чергу, в віконці «Linear equalities» цієї панелі задаються лінійні обмеження рівності:

$$A \cdot x = b \quad (3.2)$$

де b – вектор

A – деяка матриця

В рядку «Bounds» у векторному вигляді задають нижню та верхню границю обмеження змінних, а в «Nonlinear constraint function» за необхідності задають деяку довільну нелінійну функцію обмежень. У випадку, якщо завдання не потребує обмежень, всі поля на панелі «Constraints» залишають незаповненими.

Далі, нижче розглянутого поля знаходяться кнопки керування, а саме запуску, тимчасової паузи та остаточної зупинки обробки генетичного алгоритму. Поряд виводиться значення кінцевої точки роботи генетичного алгоритму.

Правіше знаходиться панель налаштування «Options». Вона дозволяє виводити різні графіки, що відображають інформацію про роботу генетичного алгоритму. На основі цієї інформації можна змінювати налаштування алгоритму з метою підвищення ефективності його роботи.

До параметрів, що настраюються у вкладці «Options» відносяться:

— Stopping criteria - критерії зупинки генетичного алгоритму, що визначають дані, котрі призводять до завершення алгоритму. В цьому розділі також наявні такі підпункти як : максимальна кількість ітерацій визначає найбільшу кількість ітерацій, які виконується алгоритмом; максимальна кількість оцінок функції, що вказує найбільш можливу чисельність оцінок цільових

функцій та функцій обмежень, що виконує алгоритм. В свою чергу функціональний допуск визначає дозвіл завершення для значення цільової функції, а допуск нелінійного обмеження виявляє припущення для максимального порушення нелінійного обмеження,

- Поріг необмеженості припиняє ітерації, якщо цільова функція опускається нижче нього,

- Function value check – даний рядок служить для перевірки значення функції. Вона повертає надану користувачем цільову функцію або функція нелінійних обмежень, та видає певний результат в залежності від даних, після чого виводиться Inf, NaN або комплексні значення,

- Validate user-supplied derivatives - Перевіряє надані користувачем похідні, та виконується оцінка похідних у цільових і нелінійних функціях обмежень. Під час виконання в командному рядку відображається попереджувальне повідомлення, якщо обчислена та надана похідні розходяться в початковій точці, після чого обробка призупиняється,

- Approximated derivatives - (апроксимовані похідні) застосовують для виведення кінцевих різниць для наближення похідних,

- Hessian – визначення типу апроксимації або точної функції Гессе. Має наступні типи апроксимації – BFGS ,LBFGS ,кінцева різниця на градієнтах.

Algorithm Settings – має декілька підпунктів, такі як:

- Typical X values – відповідає за встановлення типових значення змінної x , щоб визначити типову величину змінних. Це використовується для масштабування, наприклад, кінцевих різниць,

- Constraints to honor at each iteration слугує для встановлення обмеження, котре потрібно дотримувати на кожній ітерації, для задоволення межі на всіх ітераціях. У випадку відсутності необхідності обробки цього пункту встановлюють значення «NONE»,

- Initial trust region radius або початковий радіус довірчої області, слугує для введення радіусу довірчої області для обробки першої ітерації,

- Initial barrier parameter встановлює бар'єрний параметр μ для першої ітерації,
- Inner iteration stopping criteria – або проєктована спряжена ітерація, функція що визначає такі параметри - Maximum iterations або максимальна кількість ітерацій, визначає кількість спрогнозованих ітерацій спряженого градієнта; Relative tolerance - це відносний допуск завершення на спроєктованій ітерації спряженого градієнта; Absolute tolerance— це абсолютний допуск на завершення на спроєктованій ітерації спряженого градієнта,
- Plot function –вказує вихідні дані для графіка оптимізації на кожній ітерації.

В наявності такі ділянки:

- Current point— зображення точки на поточній ітерації,
- Function count — алгоритм підрахунку кількості оцінок функції на кожній ітерації,
- Function value— створює найкраще значення функції на кожній ітерації,
- Max constraint— створює графік максимального значення порушення обмеження на кожній ітерації,
- Current step— створює графік розміру кроку алгоритму на кожній ітерації,
- First order optimality – (або оптимальність першого порядку) — створює графік порушення умов оптимальності для розв'язувача на кожній ітерації,
- Custom function— вводиться дескриптор функції у формі @plotfun, де plotfun.m — файл функції з тим самим синтаксисом, що й функція виведення,
- Output Function - Вихідна функція викликається на кожній ітерації розв'язувача. Для її виклику необхідно ввести у формі @outputfun, де outputfun.m файл функції з описаним тут синтаксисом.

У вкладці налаштування популяцій користувач має можливість вибрати тип математичних об'єктів, до якого відноситимуться особини всіх популяцій

(подвійний вектор, бітовий рядок або користувальницький тип). мутації та схрещування особин. Так, наприклад, при виборі як форма подання. особин бітового рядка для оператора схрещування не можна використовувати гібридну функцію або нелінійну функцію, що обмежує.

Також вкладка популяції дозволяє налаштовувати розмір популяції (з кількох осіб їй буде складатися кожне покоління) і яким чином буде створюватися початкове покоління (Uniform – якщо відсутні обмеження, що накладаються, в іншому випадку – Feasible population). Крім того, в даній вкладці є можливість задати вручну початкове покоління (використовуючи пункт Initial population) або його частину, початковий рейтинг особин (пункт Initial scores), а також ввести обмежувальний числовий діапазон, якому повинні належати особини початкової популяції (Initial range).

У вкладці масштабування (Fitness Scaling) користувач має можливість вказати функцію масштабування, яка конвертує значення, що досягаються оптимізованою функцією, у значення, що лежать в межах, допустимих для оператора відбору. При виборі як функцію масштабування параметра Rank масштабування буде призводитися до рейтингу, тобто особам присвоюється рейтинговий номер (для кращої особини – одиниця, для наступної – двійка, і так далі). Пропорційне масштабування (Proportional) задає можливість пропорційно заданому числовому ряду для особин. При виборі опції Top найбільше рейтингове значення присвоюється відразу декільком найвидатнішим особам (їх число вказується як параметр). Нарешті, при виборі масштабування типу Shift linear можна вказати максимальну ймовірність найкращої особини.

Вкладка Selection дозволяє вибрати оператор відбору батьківських особин на основі даних із функції масштабування [5]. Як доступні для вибору варіанти оператора відбору пропонуються такі:

- Tournament – випадково обирається вказана кількість особин, серед них на конкурсній основі обираються найкращі,
- Roulette – імітується рулетка, в якій розмір кожного сегмента

встановлюється відповідно до його ймовірності;

- Uniform – батьки вибираються випадковим чином згідно з заданим розподілом та з урахуванням кількості батьківських особин та їх ймовірностей,

- Stochastic uniform – будується лінія, в якій кожному з батьків ставиться у відповідність її частина певного розміру (залежно від ймовірності батька), потім алгоритм пробігає під лінії кроками однакової довжини і вибирає батьків залежно від того, на яку частину лінії потрапив крок,

- Вкладка Reproduction уточнює, яким чином відбувається створення нових особин. Пункт Elite count дозволяє вказати кількість особин, які гарантовано перейдуть у наступне покоління. Пункт Crossover fraction вказує на частку особин, які створюються шляхом схрещування. Решта частка створюється шляхом мутації.

У вкладці оператора мутації вибирається тип оператора. Доступні такі варіанти:

- Gaussian – додає невелике випадкове число (згідно з розподілом Гауса) до всіх компонентів кожного вектора-особи,

- Uniform - вибираються випадковим чином компоненти векторів і замість них записуються випадкові числа з допустимого діапазону,

- Adaptive feasible - генерує набір напрямків залежно від останніх найбільш вдалих і невдалих поколінь і з урахуванням обмежень, що накладаються, просувається вздовж усіх напрямків на різну довжину,

- Custom - дозволяє задати власну функцію,

- Вкладка Crossover дозволяє вибрати тип оператора схрещування (однокрапкове, двокрапкове, евристичне, арифметичне або розсіяне (Scattered), при якому генерується випадковий двійковий вектор відповідності батьків). Також є можливість завдання довільної (custom) функції схрещування,

- У вкладці Migration можна налаштовувати правила, згідно з якими особи переміщатимуться між підпопуляціями в межах однієї популяції. Підпопуляції створюються, якщо розміром популяції зазначений вектор, а чи не

натуральне значення. У цій вкладці можна вказати напрямок міграції (forward – у наступну підпопуляцію, both – у попередню та наступну), частку мігруючих особин та частоту міграції (скільки поколінь проходить між міграціями). Якщо створення підпопуляцій не потрібне, цю вкладку слід залишати без змін.

Вкладення спеціальних опцій алгоритму дозволяє налаштовувати параметри розв'язання системи нелінійних обмежень, що накладаються на алгоритм. Значення параметра Initial penalty визначає початкове числове значення критики алгоритму, Penalty factor використовується як множник цього значення у випадках, коли розробника не влаштовує точність оптимізації або при виході за межі, визначені у вкладці обмежень. Як правило, ці опції детально налаштовуються для вирішення задач високої складності.

Вкладка Hybrid function дозволяє встановити ще одну функцію мінімізації, яка буде використовуватися після закінчення роботи алгоритму. Як можливі гібридні функції доступні наступні вбудовані в середу MATLAB функції:

- none (не використовувати гібридну функцію),
- fminsearch (пошук мінімального із значень),
- patternsearch (пошук за зразком),
- fminunc (для необмеженого алгоритму),
- fmincon (для алгоритму із заданими обмеженнями).

У вкладці критерію зупинки (Stopping criteria) вказуються ситуації, у яких алгоритм робить зупинку. При цьому налаштовуються такі параметри:

- Generations - максимальна кількість поколінь, після перевищення якого відбудеться зупинка,
- Time limit - ліміт часу на роботу алгоритму,
- Fitness limit – якщо значення, що оптимізується, менше або дорівнює даному ліміту, то алгоритм зупиниться,
- Stall generations – кількість поколінь, що мало відрізняються, після яких алгоритм зупиниться,
- Stall time limit – те саме, що й попередній параметр, але застосовний до

часу роботи алгоритму,

- Function tolerance та Nonlinear constraint tolerance – мінімальні значення змін оптимізованої та обмежуючої функцій відповідно, за яких алгоритм продовжить роботу.

Важливою частиною є вкладка Plot Functions, яка дозволяє вибирати різну інформацію, яка виводиться по ходу роботи алгоритму і показує як коректність його роботи, так і конкретні результати, що досягаються алгоритмом. Найбільш важливими параметрами, що використовуються для відображення, є:

- Plot interval – кількість поколінь, після якого відбувається чергове оновлення графіків,

- Best fitness – виведення найкращого значення оптимізованої функції для кожного покоління,

- Best individual – висновок найкращого представника покоління за найкращого оптимізаційного результату у кожному з поколінь,

- Distance - виведення інтервалу між значеннями особин у поколінні,

- Expectation – виводить низку ймовірностей та відповідні їм особини покоління,

- Genealogy - Висновок генеалогічного дерева особин,

- Range – висновок найменшого, найбільшого та середнього значень оптимізованої функції для кожного покоління,

- Score diversity – показ гістограму рейтингу у кожному поколінні,

- Scores - виведення рейтингу кожної особи в поколінні,

- Selection - показ гістограми батьків,

- Stopping - виведення інформації про стан всіх параметрів, що впливають на критерії зупинки,

- Custom – відображення на графіку певної функції.

Вкладка виведення результатів у вигляді нової функції (Output function) дозволяє увімкнути виведення історії роботи алгоритму в окремому вікні із заданим інтервалом поколінь (прапор History to new window та поле Interval

відповідно), а також дозволяє задати та вивести довільну вихідну функцію, що задається в полі Custom функція.

Вкладка User function evaluation описує, в якому порядку відбувається обчислення значень оптимізується і обмежує функції (окремо, паралельно в одному виклику або одночасно).

Вкладка Display to command window дозволяє налаштовувати інформацію, яка відображається в основному командному вікні MATLAB під час роботи алгоритму. Можливі наступні значення: Off - немає виведення в командне вікно, Iterative - виведення інформації про кожну ітерацію працюючого алгоритму, Diagnose - виведення інформації про кожну ітерацію та додаткові відомості про можливі помилки та змінені ключові параметри алгоритму, Final - виводиться тільки причина зупинки та кінцеве значення .

3.4 Висновки за третім розділом

В даному розділі розглянуто поняття генетичних алгоритмів, та його близькість до галузі біології, і основні поняття.

Показано основні методи, а саме – переборний, градієнтний спуск.

Також відведено вивчення інтерфейсу MATLAB для застосування його в обробці генетичного алгоритму транспортної задачі.

4 РОЗВ'ЯЗАННЯ ЗАДАЧІ ОПТИМІЗАЦІЇ ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ У СЕРЕДОВИЩІ MATLAB

4.1 Оптимізація результатів розрахунку класичного методу за допомогою MATLAB

Виконаємо вдосконалення розрахунків цільової функції мінімізації за допомогою генетичного алгоритму в середовищі MATLAB. Однак спочатку створимо систему розрахунку на базі класичного методу. Для цього створюємо новий скрипт. В самому початку потрібно ввести назву оброблюваної функції, що після збереження стане назвою `m` – файлу. Наприклад:

```
function [x,fval]=OptomizationNumberone
```

В скрипті мають бути такі вихідні дані, представлені в лістингу 4.1:

- `X, fval` – результати оптимізації та значення цільової функції,
- `@optimFunc` – цей розділ вказує на обробку цільової функції,
- `[30, 0, 20 ...]` – початкова точка з якої алгоритм шукає оптимальне розв'язання. В нашому випадку початковим буде 35 значень, що були отримані завдяки класичному методу. Введені значення відповідають $x(1) = 30$, $x(2) = 0$ і так далі,

- `[ ]` – пусті дужки вказують на відсутність лінійних нерівностей,
- `[25.709, 5.002, ...]` – позначення лінійних рівностей, кожне число позначає своє значення C_{ij} , а вираз в свою чергу $C_{11}*x(1) + \dots$,

- `fmincon` – тип оброблюваного алгоритму, даний шукає мінімальне значення багатовимірної функції,

- `[0, 0 , 0 ...]` – встановлення нижнього порогу на змінні, в нашому випадку $x(1) \geq 0$, ... $x(35) \geq 0$. Також можна встановити обмеження на змінні зверху, але у подальшій роботі вони не потрібні.

Лістинг 4.1 – Скрипт простої функції мінімізації

```

function [x,fval]=OptomizationNumberone
[x,fval] = fmincon(@EasiestObjective,[30, 0, 20, 0, 0, 0, 0, 0, 0, 0,
70, 0, 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, 150, 0, 0, 0, 50, 150, 0, 0, 0,
60, 0, 70, 0, 170, 0],[[],[]],[25.709, 5.002, 10, 5.681, 6.451, 5.016,
7.369, 13.803, 3.074, 5, 3.426, 3.824, 3.105, 4.3, 10.06, 2.469, 4,
2.718, 3, 2.491, 3.337, 8.423, 2.202, 4, 2.406, 2.638, 2.221, 2.913,
7.485, 2.050, 3, 2.229, 2.43, 2.069, 2.671 ],3030,[0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0],[[]],@nonlcon);
end

function Obj=EasiestObjective(x)
Obj=25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+
7.369*x(7)+13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.
105*x(13)+4.3*x(14)+10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(
19)+2.491*x(20)+3.337*x(21)+8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(
25)+2.638*x(26)+2.221*x(27)+2.913*x(28)+7.485*x(29)+2.050*x(30)+3*x(
31)+2.229*x(32)+2.43*x(33)+2.066*x(34)+2.671*x(35);
end

function [x,ceq]=nonlcon(x)
25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+7.36
9*x(7)<=50;
13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.105*x(13)+4
.3*x(14)<=100;
10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(19)+2.491*x(20)+3.33
7*x(21)<=150;
8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(25)+2.638*x(26)+2.221*x(27)+
2.913*x(28)<=200;
7.485*x(29)+2.050*x(30)+3*x(31)+2.229*x(32)+2.43*x(33)+2.066*x(34)+2
.671*x(35)<=300;

25.709*x(1)+13.803*x(8)+10.06*x(15)+8.423*x(22)+7.485*x(29)<=30;

```


В даній таблиці представлені отримані результати змінних x та їх сума за рядками А та стовпчиками В.

Таблиця 4.1 – Результати розрахунку з $C = 3153$

x1	x2	x3	x4	x5	x6	x7	A1
45,2917	7,5714	19,8015	8,5606	9,6825	7,5918	11,0200	173,46
x8	x9	x10	x11	x12	x13	x14	A2
20,3940	4,7623	25,5314	5,2752	5,8551	4,8075	14,0997	80,71
x15	x16	x17	x18	x19	x20	x21	A3
14,9407	3,8809	6,1115	4,2437	4,6545	3,9129	43,9321	106,61
x22	x23	x24	x25	x26	x27	x28	A4
12,5556	3,4918	6,1115	16,5461	42,9137	3,5195	4,5278	89,66
x29	x30	x31	x32	x33	x34	x35	A5
11,1890	18,6304	4,6545	21,4942	3,8240	47,2906	4,1752	111,25
B1	B2	B3	B4	B5	B6	B7	C
91,8454	38,3368	62,1779	56,1198	66,3298	83,4345	77,7548	3153

Бачимо, що результати обчислень не відповідають умовам балансу. Тому спробуємо зменшити введені значення цільової функції що було задано в початку з 3153 до 3060 у.о.в./у.о.п. (це значення нижче за результат обробки розподільного методу). Результати представлені в таблиці 4.2.

Також додатково зробимо розрахунок з введенням $C=3050$. Результати представлені в таблиці 4.3. Подальші зниження призводять до розбалансу та отримання некоректних результатів, тому далі роботу з простою функцією мінімізації можна завершити.

Таблиця 4.2 – Результати розрахунку з $C = 3060$

x1	x2	x3	x4	x5	x6	x7	A1
27,8824	0,6904	19,1814	0,6371	0,5767	0,6893	0,5047	49,6573
x8	x9	x10	x11	x12	x13	x14	A2
0,0000	0,8416	69,2398	0,8140	0,7828	0,8391	29,5618	102,0791
x15	x16	x17	x18	x19	x20	x21	A3
0,2936	0,8890	0,7689	0,8695	0,8474	0,8873	148,8362	153,3919
x22	x23	x24	x25	x26	x27	x28	A4
0,4220	0,9100	0,7689	49,5768	148,8911	0,9085	0,8542	202,3315
x29	x30	x31	x32	x33	x34	x35	A5
0,4956	59,53800	0,8474	69,4572	0,8921	168,8022	0,8732	300,9057
B1	B2	B3	B4	B5	B6	B7	C
29,0936	62,869	90,8064	121,3546	151,9901	172,1264	180,6301	3060

Таблиця 4.3 – Результати розрахунку з $C = 3050$

x1	x2	x3	x4	x5	x6	x7	A1
27,8855	0,6904	19,1835	0,6372	0,5768	0,6893	0,5047	50,1674
x8	x9	x10	x11	x12	x13	x14	A2
0,0000	0,8417	69,2474	0,8141	0,7828	0,8392	29,5650	102,0902
x15	x16	x17	x18	x19	x20	x21	A3
0,2936	0,8891	0,7690	0,8696	0,8475	0,8874	148,8525	153,4087
x22	x23	x24	x25	x26	x27	x28	A4
0,4221	0,9101	0,7690	49,5822	148,9073	0,9086	0,8543	202,3536
x29	x30	x31	x32	x33	x34	x35	A5
0,4956	59,5445	0,8475	69,4648	0,8922	168,8206	0,8733	300,9385
B1	B2	B3	B4	B5	B6	B7	
29,0968	62,8758	90,8164	121,3639	152	172,1451	180,6498	

Тепер після уточнення, переходимо до створення скрипту генетичного алгоритму. Його синтаксис схожий з попереднім, але наявні деякі відмінності.

Роздивимось його ближче:

— `function [x,fval] = OptimizationNumberTwo` – функція, що для вхідних аргументів повертає значення `fval` тобто значення функції пристосованості в точці `x`,

— далі вводиться назва функції (а саме генетичний алгоритм – `ga`) та наступні вихідні дані: кількість змінних (`nvars`), лінійні рівності та нерівності, нижня та верхня гранична точки обмежень та функція вводу виразів для створення умови обмежень для балансу (`nonlcon`).

В іншому дані такі самі, як і у попередньому випадку, що і бачимо в лістингу 4.2 та рис.4.2.

Лістинг 4.2 – Скрипт функції генетичного алгоритму

```
function [x,fval] = OptimizationNumberTwo
[x, fval] = ga(@optimFunc2, 35, [],[],[],[],[27.8855 0.6904 19.1835
0.6372 0.5768 0.6893 0.5047 0.0000 0.8417 69.2474 0.8141 0.7828
0.8392 29.5650 0.2936 0.8891 0.7690 0.8696 0.8475 0.8874 148.8525
0.4221 0.9101 0.7690 49.5822 148.9073 0.9086 0.8543 0.4956
59.5445 0.8475 69.4648 0.8922 168.8206 0.8733],[], @nonlcon );
end
function y = optimFunc2 (x)
y =
25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+7.36
9*x(7)+13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.105*
x(13)+4.3*x(14)+10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(19)+
2.491*x(20)+3.337*x(21)+8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(25)+
2.638*x(26)+2.221*x(27)+2.913*x(28)+7.485*x(29)+2.050*x(30)+3*x(31)+
2.229*x(32)+2.43*x(33)+2.066*x(34)+2.671*x(35);

function [x, ceq] = nonlcon (x)
```

```

25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+7.36
9*x(7)<=50;
13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.105*x(13)+4
.3*x(14)<=100;
10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(19)+2.491*x(20)+3.33
7*x(21)<=150;
8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(25)+2.638*x(26)+2.221*x(27)+
2.913*x(28)<=200;
7.485*x(29)+2.050*x(30)+3*x(31)+2.229*x(32)+2.43*x(33)+2.066*x(34)+2
.671*x(35)<=300;

```

```

25.709*x(1)+13.803*x(8)+10.06*x(15)+8.423*x(22)+7.485*x(29)<=30;
5.002*x(2)+3.074*x(9)+2.469*x(16)+2.202*x(23)+2.050*x(30)<=60;
10*x(3)+5*x(10)+4*x(17)+4*x(24)+3*x(31)<=90;
5.681*x(4)+3.426*x(11)+2.718*x(18)+2.406*x(25)+2.229*x(32)<=120;
6.451*x(5)+3.824*x(12)+3*x(19)+2.638*x(26)+2.43*x(33)<=150;
5.016*x(6)+3.105*x(13)+2.491*x(20)+2.221*x(27)+2.066*x(34)<=170;
7.369*x(7)+4.3*x(14)+3.337*x(21)+2.913*x(28)+2.671*x(35)<=180;
ceq = 0;

```

```
end
```

```
end
```

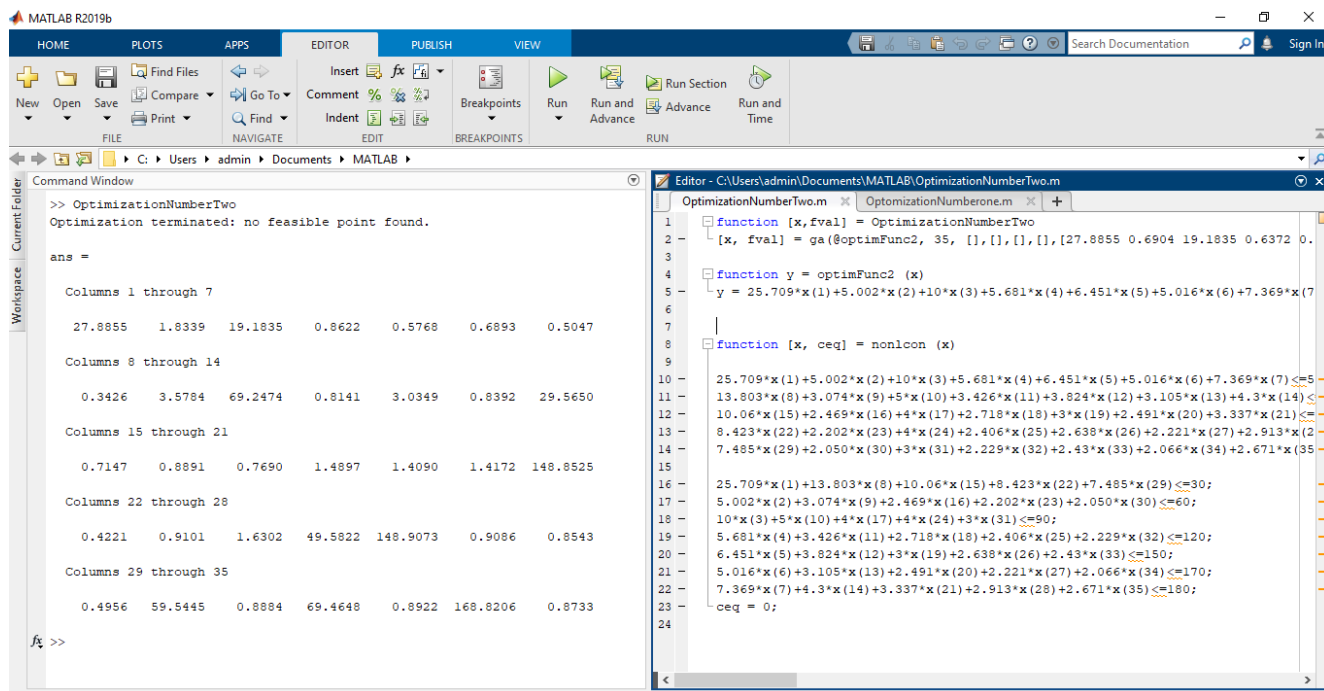


Рисунок 4.2 – Результати обробки скрипта генетичного алгоритма

Таблиця 4.3 – Результат першого досліді

x1	x2	x3	x4	x5	x6	x7	A1
27,8855	1,8339	19,1835	0,8622	0,5768	0,6893	0,5047	51,5359
x8	x9	x10	x11	x12	x13	x14	A2
0,3426	3,5784	69,2474	0,8141	3,0349	0,8392	29,5650	107,4216
x15	x16	x17	x18	x19	x20	x21	A3
0,7147	0,8891	0,7690	1,4897	1,4090	1,4172	148,8525	155,5412
x22	x23	x24	x25	x26	x27	x28	A4
0,4221	0,9101	1,6302	49,5822	148,9073	0,9086	0,8543	203,2148
x29	x30	x31	x32	x33	x34	x35	A5
0,4956	59,5445	0,8884	69,4648	0,8922	168,8206	0,8733	300,9794
B1	B2	B3	B4	B5	B6	B7	C
29,8505	66,756	91,7185	122,213	154,8202	172,6749	180,6498	3125

Отримавши перші результати оптимізації, бачимо що вартість пересилання знизилася у порівнянні з класичним методом. Але має місце великі розбіжності по сумі пересиланих умовних одиниць потужності від джерел до споживачів по

рядкам та стовпчикам, що не співпадає з реальними значеннями. Тому потрібно виконати повторну обробку генетичного алгоритму.

Результати другої спроби представлені в таблиці 4.4.

Таблиця 4.4 – Результати другої спроби оптимізації

x1	x2	x3	x4	x5	x6	x7	A1
27,8855	0,6904	19,1835	0,6372	1,5561	0,6893	0,5047	51,1467
x8	x9	x10	x11	x12	x13	x14	A2
0	0,8417	69,2474	0,8141	0,7828	1,1636	29,5650	102,4146
x15	x16	x17	x18	x19	x20	x21	A3
0,2936	1,7819	1,3308	1,9060	0,8475	2,2702	148,8525	153,5613
x22	x23	x24	x25	x26	x27	x28	A4
0,4221	0,9101	0,7690	49,5822	148,9073	0,9086	0,8543	202,3536
x29	x30	x31	x32	x33	x34	x35	A5
0,4956	59,5445	1,3313	69,4648	0,8922	168,8206	0,8733	301,4223
B1	B2	B3	B4	B5	B6	B7	C
29,0968	63,7686	91,862	122,4043	152,9859	173,8523	180,6498	3104

Бачимо що вартість пересилання знизилася. В свою чергу значення потужності джерел та споживачів наблизилися до реальних. Тому доцільно провести додатковий дослід з оптимізації.

Запускаємо повторну (третю) обробку. Вхідні дані залишаємо такі самі, що і у попередніх двох дослідях. Результати представлені в таблиці 4.5.

Таблиця 4.5 – Третій дослід оптимізації

x1	x2	x3	x4	x5	x6	x7	A1
27,8855	0,6904	19,1835	0,6372	0,5768	0,6893	0,5047	50,1674
x8	x9	x10	x11	x12	x13	x14	A2
0,0695	0,8417	69,2474	0,8141	0,7828	2,3322	29,5650	103,6527
x15	x16	x17	x18	x19	x20	x21	A3
0,3291	0,8891	0,7690	0,8696	0,8876	1,5107	148,8525	153,8201
x22	x23	x24	x25	x26	x27	x28	A4
0,4221	0,9101	1,1460	49,5822	148,9073	0,9086	0,8543	202,7506
x29	x30	x31	x32	x33	x34	x35	A5
0,4956	59,5445	0,8475	69,4648	0,8922	168,8206	0,8733	300,9385
B1	B2	B3	B4	B5	B6	B7	C
29,2018	62,8758	91,1934	121,3679	152,0467	174,2614	180,6498	3093,659

З отриманих результатів можна зробити висновок щодо підвищення ефективності оптимізації транспортної задачі. Виходячи з цього висновок щодо вдалості оптимізації.

Фактично, маємо варіант схеми в випадку задіяння всіх можливих з'єднань.

4.2 Висновки за четвертим розділом

В четвертому розділі було проведено розрахунок та оптимізація за допомогою генетичного алгоритму в середовищі MATLAB, на основі результатів розрахунку класичним методом. При використанні декількох спроб у підсумку, маємо зниження вартості пересилання у порівнянні з класичним та розподільним методами розрахунку. Однак через наявність похибки обчислення генетичного алгоритму має місце розбіжності суми потужностей споживачів та джерел.

ВИСНОВКИ

Оптимізація структури електричної розподільної мережі є важливою задачею при наявності великої кількості промислових підприємств. Найбільш поширеним методом розрахунку оптимального з'єднання є застосування транспортної задачі.

В першому розділі було приведено поняття оптимізації структури з'єднань електричних мереж та приведено основні методи оптимізації, а саме: метод покрокової оптимізації, динамічного програмування, нелінійного програмування, метод табу – пошуку та еволюційний метод на базі генетичних алгоритмів. Також розглянуто поняття транспортної задачі та хід її розв'язання класичним методом.

У другому розділі було створена початкова схема з'єднань та виконано розрахунок оптимізації кількості цих зв'язків за допомогою класичного методу. Також створено умови балансу за яких розв'язок та отримані результати можна вважати правильними. Отримане значення цільової функції мінімізації додатково було перевірено за допомогою розподільного методу, що призначений вдосконалити результати обчислень. В підсумку була отримана нова, більш досконала схема з'єднань та зниження вартості пересилання у порівнянні з класичним методом.

В третьому розділі був описаний порядок роботи з генетичним алгоритмом в середовищі програми MATLAB, та розглянуто основні функції при обробці.

В четвертому розділі було створено скрипт генетичного алгоритму, а також проведено розрахунок та оптимізація за його допомогою в середовищі MATLAB, на основі результатів розрахунку класичним методом. При використанні декількох спроб у підсумку, маємо зниження вартості пересилання у порівнянні з класичним та розподільним методами розрахунку. Однак через наявність похибки обчислення генетичного алгоритму має місце розбіжності суми потужностей споживачів та джерел.

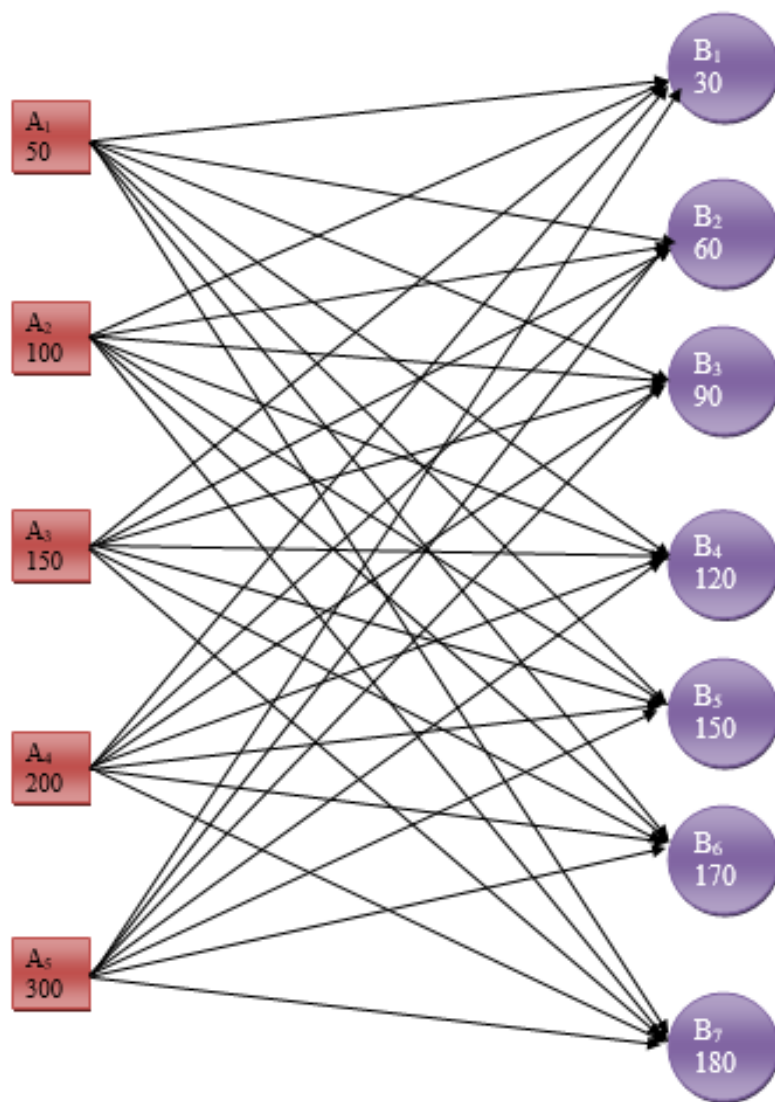
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Моделі оптимального розвитку енергосистем: Методи оптимізації розвитку електричних мереж енергосистем. Навчальний посібник для студентів спеціальності 141 «Електроенергетика. електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського; уклад.: В. А. Баженов. –(1 файл: 462 кбайт). – Київ : КПІ ім. Ігоря Сікорського. 2022. – 38 с.
2. Оптимізаційні задачі в енергетиці сільського господарства: Навч. посібник / Г.Б. Іноземцев. В.В. Козирський; За ред. Г.Б. Іноземцева. – К.: Видавничий центр НУБіП України. 2014 – 172 с.
3. Іноземцев Г.Б. Математичне моделювання та оптимізація систем електроспоживання у сільському господарстві / Г.Б. Іноземцев. В.В. Козирський; за ред.. Г.Б. Іноземцева. – К.: АграрМедіаГруп. – 2010. – 139 с.
4. Кононюк А.Ю. Нейронні мережі і генетичні алгоритми - К.: «Корнійчук» 2008 – 446 с
5. Лазарєв Ю.Ф. Довідник з Matlab / Електронний навчальний посібник з курсового і дипломного проектування – К.:НТУУ «КПІ» . 2013 – 132 с
6. M. Mohammadi. S. Hosseinian. and G. Gharehpetian. “Ga-based optimal sizing of microgrid and dg units under pool and hybrid electricity markets.” International Journal of Electrical Power and Energy Systems. vol. 35. no. 1. pp. 83 – 92. 2012.
7. Каплун В.В. Кравченко О.П.. Василенко В.В.. Макаревич С.С. Каплун Р.В. Аналіз методів оптимізації мікроенергетичних систем (MICROGRID) на основі джерел розподіленої генерації – К.: Київський національний університет технологій та дизайну, Національний університет біоресурсів та природокористування України

8. Кулик В.В., Гриник В.А. Порівняльна характеристика методів оптимізації електроенергетичних систем – В.: ВНТУ
9. Bajpai. Pratibha & Kumar. M.. (2010). Genetic Algorithm – an
10. Approach to Solve Global Optimization Problems. Indian Journal of Computer Science and Engineering. 1. 199-206.
11. L. Arya. L. Titare. and D. Kothari. “Improved particle swarm optimization applied to reactive power reserve maximization.” International Journal of Electrical Power and Energy Systems. vol. 32. no. 5. pp. 368 – 374. 2010.
12. A. K. Basu. S. Chowdhury. S.P. Chowdhury. “Operational Management of CHP-Based Microgrid”. Proceedings of International Conference on Power System Technology (POWERCON- 2010). 24-28 October. 2010. pp. 1 – 5.
13. Omar A. Imran MatLab Department of Chemical Eng. Diyala university / College of Engineering. 2019.
14. P. Valli. C. Antony Jeyasehar . Genetic algorithm based equipment selection method for construction project using MATLAB tool / International journal of optimization in civil engineering . 2012.
15. Hsieh. P.-H. Decision support for unrelated parallel machine scheduling with discrete controllable processing times / P.-H. Hsieh. S.-J. Yang. D.-L. Yang // Applied Soft Computing. – 2015.
16. Basctin A. A Decision Support system for optimal equipment selection in open pitmining: analytical Hierarchy process. Report of Mining Engineering Department. 2003.
17. Goldenberg M. and Shapira A. Soft Considerations in Equipment for Building Construction Project. J Constr Eng Manage. Proc. ASCE. 2007
18. Qijun Deng.Xing Gao. Hong Zhou .Wenshan Hu. “System modeling and optimization of microgrid using Genetic Algorithm”.Proceedings of 2nd International Conference on Intelligent Control and Information Processing (ICICIP). 2011 .

19. H. M. Khodr. J. F. Gomez. L. Barnique. J. H. Vivas. P. Paiva. J. M. Yusta. and A. J. Urdaneta. "A linear programming methodology for the optimization of electric power-generation schemes." *Power Systems. IEEE Transactions on.* vol. 17. 2002.
20. Michalewicz, *Genetic Algorithms i Data Structures Evolution Programs*, Springer-Verlag, 3rd, revised and extended edition, 1999.
21. J.-H. Kim and H. Myung, "Evolutionary Programming Techniques for Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 121 X. Yao, *Evolutionary Computation: Theory and Applications*, World Scientific, Singapore, 1999.

ДОДАТОК А – ГРАФІЧНА ЧАСТИНА



Первинна схема з'єднань

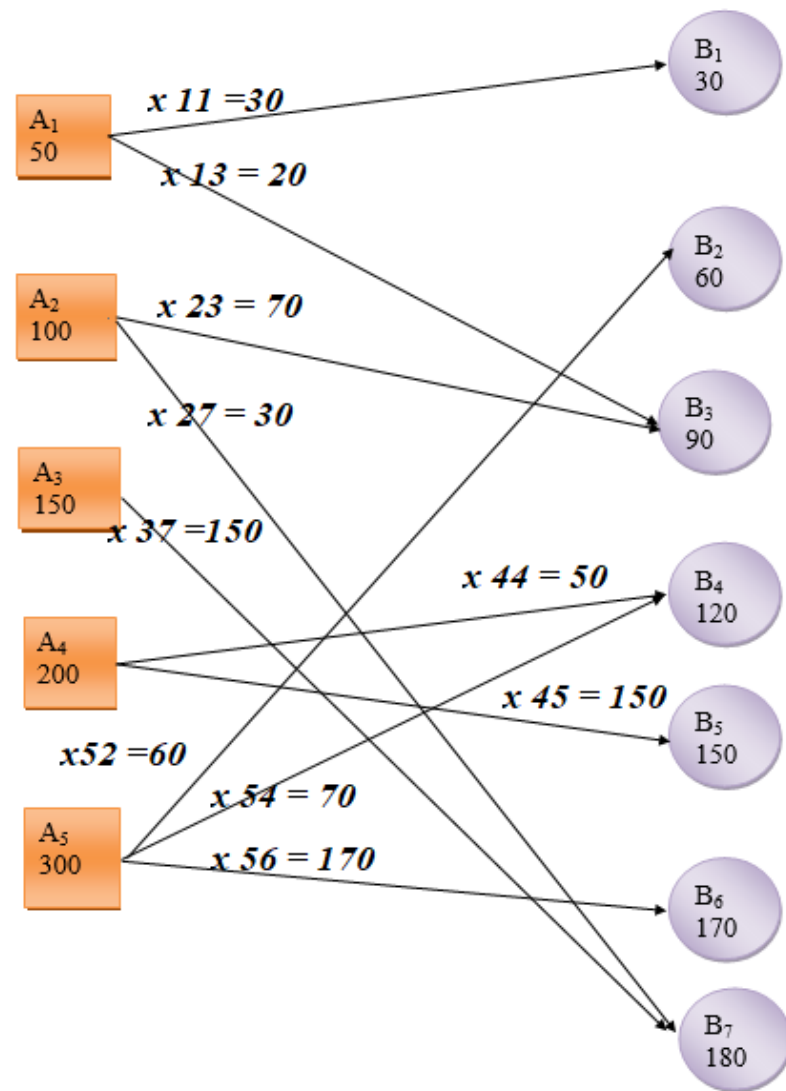


Схема з'єднань мережі після розрахунку класичним методом,

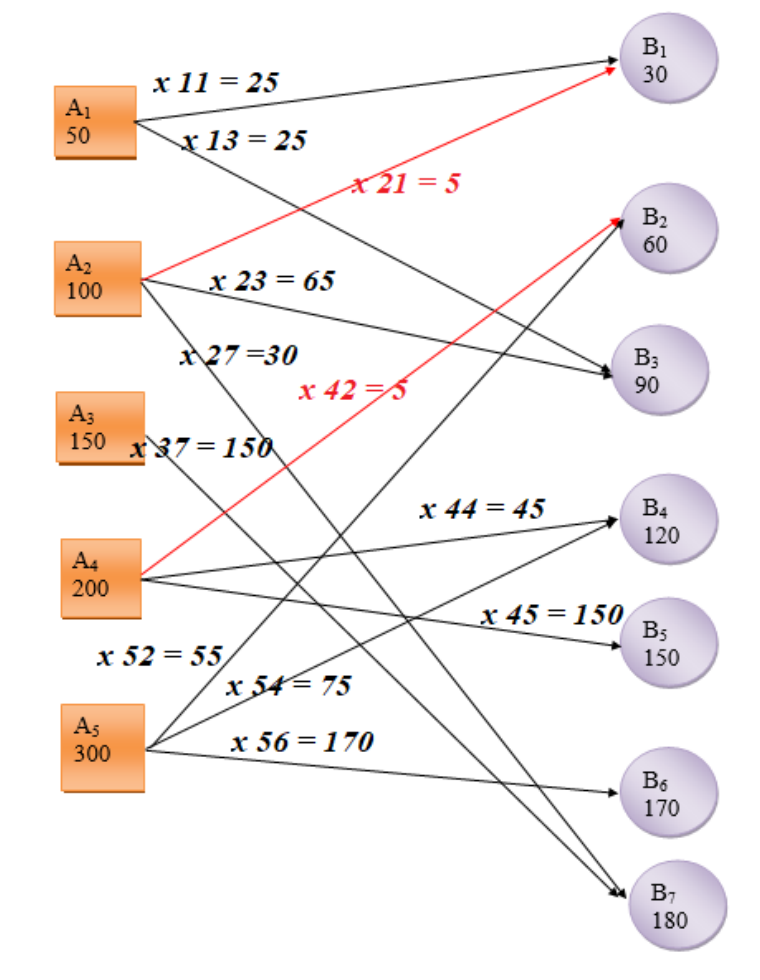


Схема з'єднань з урахуванням результатів розрахунку розподільним методом

ДОДАТОК Б – Скрипт роботи генетичного алгоритму

```

function [x,fval] = OptimizationNumberTwo
[x, fval] = ga(@optimFunc2, 35, [],[],[],[],[27.8855 0.6904 19.1835
0.6372 0.5768 0.6893 0.5047 0.0000 0.8417 69.2474 0.8141 0.7828
0.8392 29.5650 0.2936 0.8891 0.7690 0.8696 0.8475 0.8874 148.8525
0.4221 0.9101 0.7690 49.5822 148.9073 0.9086 0.8543 0.4956
59.5445 0.8475 69.4648 0.8922 168.8206 0.8733],[], @nonlcon );
end
function y = optimFunc2 (x)
y =
25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+7.36
9*x(7)+13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.105*
x(13)+4.3*x(14)+10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(19)+
2.491*x(20)+3.337*x(21)+8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(25)+
2.638*x(26)+2.221*x(27)+2.913*x(28)+7.485*x(29)+2.050*x(30)+3*x(31)+
2.229*x(32)+2.43*x(33)+2.066*x(34)+2.671*x(35);
end
function [x, ceq] = nonlcon (x)

25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+7.36
9*x(7)<=50;
13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.105*x(13)+4
.3*x(14)<=100;
10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(19)+2.491*x(20)+3.33
7*x(21)<=150;
8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(25)+2.638*x(26)+2.221*x(27)+
2.913*x(28)<=200;
7.485*x(29)+2.050*x(30)+3*x(31)+2.229*x(32)+2.43*x(33)+2.066*x(34)+2
.671*x(35)<=300;

25.709*x(1)+13.803*x(8)+10.06*x(15)+8.423*x(22)+7.485*x(29)<=30;
5.002*x(2)+3.074*x(9)+2.469*x(16)+2.202*x(23)+2.050*x(30)<=60;

```

```
10*x(3)+5*x(10)+4*x(17)+4*x(24)+3*x(31)<=90;  
5.681*x(4)+3.426*x(11)+2.718*x(18)+2.406*x(25)+2.229*x(32)<=120;  
6.451*x(5)+3.824*x(12)+3*x(19)+2.638*x(26)+2.43*x(33)<=150;  
5.016*x(6)+3.105*x(13)+2.491*x(20)+2.221*x(27)+2.066*x(34)<=170;  
7.369*x(7)+4.3*x(14)+3.337*x(21)+2.913*x(28)+2.671*x(35)<=180;  
ceq = 0;  
end  
end
```

ДОДАТОК В – Скрипт простої функції мінімізації

```

function [x,fval]=OptomizationNumberone
[x,fval] = fmincon(@EasiestObjective,[30, 0, 20, 0, 0, 0, 0, 0, 0, 0,
70, 0, 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, 150, 0, 0, 0, 50, 150, 0, 0, 0,
60, 0, 70, 0, 170, 0],[],[],[25.709, 5.002, 10, 5.681, 6.451, 5.016,
7.369, 13.803, 3.074, 5, 3.426, 3.824, 3.105, 4.3, 10.06, 2.469, 4,
2.718, 3, 2.491, 3.337, 8.423, 2.202, 4, 2.406, 2.638, 2.221, 2.913,
7.485, 2.050, 3, 2.229, 2.43, 2.069, 2.671 ],3030,[0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0],[],@nonlcon);

function Obj=EasiestObjective(x)
Obj=25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+
7.369*x(7)+13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.
105*x(13)+4.3*x(14)+10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(
19)+2.491*x(20)+3.337*x(21)+8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(
25)+2.638*x(26)+2.221*x(27)+2.913*x(28)+7.485*x(29)+2.050*x(30)+3*x(
31)+2.229*x(32)+2.43*x(33)+2.066*x(34)+2.671*x(35);

end

function [x,ceq]=nonlcon(x)
25.709*x(1)+5.002*x(2)+10*x(3)+5.681*x(4)+6.451*x(5)+5.016*x(6)+7.36
9*x(7)<=50;
13.803*x(8)+3.074*x(9)+5*x(10)+3.426*x(11)+3.824*x(12)+3.105*x(13)+4
.3*x(14)<=100;
10.06*x(15)+2.469*x(16)+4*x(17)+2.718*x(18)+3*x(19)+2.491*x(20)+3.33
7*x(21)<=150;
8.423*x(22)+2.202*x(23)+4*x(24)+2.406*x(25)+2.638*x(26)+2.221*x(27)+
2.913*x(28)<=200;
7.485*x(29)+2.050*x(30)+3*x(31)+2.229*x(32)+2.43*x(33)+2.066*x(34)+2
.671*x(35)<=300;

25.709*x(1)+13.803*x(8)+10.06*x(15)+8.423*x(22)+7.485*x(29)<=30;

```

```
5.002*x(2)+3.074*x(9)+2.469*x(16)+2.202*x(23)+2.050*x(30)<=60;  
10*x(3)+5*x(10)+4*x(17)+4*x(24)+3*x(31)<=90;  
5.681*x(4)+3.426*x(11)+2.718*x(18)+2.406*x(25)+2.229*x(32)<=120;  
6.451*x(5)+3.824*x(12)+3*x(19)+2.638*x(26)+2.43*x(33)<=150;  
5.016*x(6)+3.105*x(13)+2.491*x(20)+2.221*x(27)+2.066*x(34)<=170;  
7.369*x(7)+4.3*x(14)+3.337*x(21)+2.913*x(28)+2.671*x(35)<=180;
```

```
ceq=0;
```

```
end
```

```
end
```

ДОДАТОК Г – Перелік зауважень нормоконтролера до дипломної роботи

Позначення документа	Документ	Умовне позначення	Зміст зауваження

Дата _
