

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Удосконалений адаптивний алгоритм імітації відпалу»

Виконав: студент 2-ого курсу групи ІСТ-21м
спеціальності 126 – Інформаційні системи та
технології

Боднаренко Б. О.

Керівник: к.т.н., доц. каф. АІТ

Іванов Ю. Ю.

« » _____ 2022 р.

Опонент: _____

« » _____ 2022 р.

Допущено до захисту

в.о. зав. кафедри АІТ

д.т.н., проф. Бісікало О. В.

« » _____ 2022 р.

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

Рівень вищої освіти II-ий (магістерський)

Галузь знань – 12 – Інформаційні технології

Спеціальність – 126 – Інформаційні системи та технології

Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ

в. о. завідувача кафедри АІТ

д.т.н., проф. Бісікало О. В.

« ___ » _____ 2022 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Боднаренко Богдану Олександровичу

1. Тема роботи: Удосконалений адаптивний алгоритм імітації відпалу.
Керівник роботи: к.т.н., доцент каф. АІТ Іванов Ю. Ю.
Затверджені наказом ВНТУ від «14» 09 2022 року № 203.
2. Строк подання роботи студентом: до «до 23» 12 2022 р.
3. Вихідні дані до роботи: розмірність простору пошуку (n); стартова температура T ; температурний режим; закон генерації точки; параметр Больцмана k_b ; максимальна кількість ітерацій N ; набір тестових функцій; середовище програмування MatLab.
4. Зміст текстової частини: вступ; аналіз предметної області; розробка математичної моделі удосконаленого адаптивного алгоритму імітації відпалу; розробка програмного забезпечення та експериментальні дослідження; економічний розділ; список використаних джерел; висновки.
5. Перелік ілюстративного матеріалу: принцип роботи алгоритму імітації відпалу; геометрична інтерпретація переходів; ілюстрація пошуку оптимуму; схема програми; ілюстрація перенагріву; приклад роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Іванов Ю. Ю., к.т.н., доцент каф. АІТ	19.09.2022	07.12.2022
4	Лесько О. Й. к.е.н., проф. каф. ЕПтаВМ		

7. Дата видачі завдання: «19» 09 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	10.10.22	
2	Розробка математичної моделі удосконаленого адаптивного алгоритму імітації відпалу	25.10.22	
3	Розробка програмного забезпечення та експериментальні дослідження	10.11.22	
4	Підготовка економічного розділу	20.11.22	
5	Оформлення пояснювальної записки і графічного матеріалу	30.11.22	
6	Попередній захист роботи	до 07.12.22	
7	Остаточний захист роботи	до 23.12.22	

Студент

_____ (підпис)

Боднаренко Б. О.
(прізвище та ініціали)

Керівник роботи

_____ (підпис)

Іванов Ю. Ю.
(прізвище та ініціали)

АНОТАЦІЯ

УДК 519.85 + 004.8

Боднарєнко Б. О. Удосконалений адаптивний алгоритм імітації відпалу. Магістерська кваліфікаційна робота зі спеціальності 126 – Інформаційні системи та технології, освітньо-професійна програма – Інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2022. 89 с.

На укр. мові. Бібліогр.: 30 назв; рис.: 9; табл.: 16.

У роботі проаналізовано, теоретично обґрунтовано та удосконалено метаевристичний алгоритм імітації відпалу, який інспіровано фізичним процесом відпалу металів. Розроблено алгоритмічне та програмне забезпечення, яке дозволяє виконати експериментальні дослідження у ході розв'язання задачі оптимізації функції.

Ключові слова: оптимізація, функція, екстремум, метаевристика, імітація відпалу, температурний режим.

ABSTRACT

Bodnarenko B. O. Improved adaptive simulation annealing algorithm. Master's thesis in specialty 126 – Information systems and technologies, educational and professional program – Information technologies of data and image analysis. Vinnitsa: VNTU, 2022. 89 p.

In Ukrainian language. Bibliography: 30 titles; fig.: 9; tabl.: 16.

In this work has been analyzed, theoretically substantiated and improved a metaheuristic simulated annealing algorithm, which is inspired by the physical process of metals annealing. The algorithms and software, allows us to perform some experimental researches for solving a function optimization task, have been developed.

Keywords: optimization, function, extremum, metaheuristics, simulation annealing algorithm, temperature mode.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Основні поняття.....	7
1.2 Постановка задачі оптимізації.....	14
1.3 Огляд методів розв’язання задачі оптимізації функції.....	16
1.4 Висновки.....	24
2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ УДОСКОНАЛЕНОГО	
АДАПТИВНОГО АЛГОРИТМУ ІМІТАЦІЇ ВІДПАЛУ	26
2.1 Алгоритм імітації відпалу.....	26
2.2 Функції визначення температури.....	31
2.3 Удосконалення алгоритму.....	36
2.4 Висновки.....	38
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА	
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	39
3.1 Опис роботи програми.....	39
3.2 Результати експериментів.....	44
3.3 Висновки.....	51
4 РОЗДІЛ ЕКОНОМІКИ	52
4.1 Технологічний аудит удосконаленого алгоритму імітації відпалу.....	52
4.2 Розрахунок витрат на проведення досліджень.....	56
4.3 Розрахунок ефекту для потенційного інвестора від можливої.....	
комерціалізації розробки.....	65
4.4 Висновки.....	69
ВИСНОВКИ	70

	3
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
ДОДАТКИ	75
Додаток А (обов'язковий). Технічне завдання.....	76
Додаток Б (обов'язковий). Ілюстративна частина.....	80
Додаток В (обов'язковий). Лістинг головної функції програми.....	84
Додаток Г (обов'язковий). Довідка про можливість впровадження.....	86
Додаток Д (обов'язковий). Протокол перевірки МКР.....	88

ВСТУП

Актуальність. Розвиток різноманітних галузей діяльності людини призводить до ускладнення задач управління та підвищення вимог до ефективності їх розв'язання. Тому необхідно вдосконалювати засоби розв'язання задач оптимізації, управління та оброблення інформації. У ході розробки великих транспортних, виробничих, інформаційних, енергетичних та інших складних технічних систем виникають задачі, які пов'язані з оптимізацією взаємодії об'єктів та їх функціонуванням, що призводить до необхідності вирішення оптимізаційних задач з використанням ефективних методів та сучасних засобів обчислювальної техніки. Зазвичай такі задачі містять дуже великий обсяг даних, що зумовлює високу трудомісткість обчислень у ході пошуку раціональних рішень у багатовимірному просторі альтернатив. Показники якості, як правило, описуються нелінійними залежностями і оцінюються за допомогою складних моделюючих алгоритмів. Застосування класичних методів пошуку екстремуму функцій зі складним рельєфом поверхонь рівня стає малоефективним [1, 2].

Але розв'язання подібних оптимізаційних задач у ході проектування та управління технічними системами необхідне для підвищення технічних та економічних показників ефективності систем від окремих пристроїв до глобальних систем. Тому велика увага приділяється стохастичним методам розв'язання задач оптимізації, які дозволяють знайти рішення "високої якості" за "прийнятний час" роботи програми [2, 3]. Серед таких методів оптимізації окремий клас утворюють метасевристики, які представляють собою гібридний метод розв'язання різноманітних обчислювальних задач.

Стохастичність робить застосування метаевристик нетривіальною задачею, оскільки для кожного алгоритму та різних його реалізацій, а також для кожного класу оптимізаційних задач швидкість роботи, точність, збіжність, вплив умов задачі та евристичних коефіцієнтів вимагають особливого дослідження. Тому важливим питанням є розробка та дослідження метаеврестик, вивчення їх поведінки під час роботи зі складними задачами.

У науковій літературі описано досить багато метаевристичних методів оптимізації. Можна виділити, наприклад, наукові праці Е. Осаба [2], С.В. Грибкова [4], Д. Делахайе [5], І. Суареза [6] та інших вчених. Вони розглядають алгоритм імітації відпалу, інспірований фізичним процесом кристалізації речовини, як один із найбільш ефективних та зручних метаевристичних алгоритмів оптимізації. Тому актуальною задачею є його удосконалення та вивчення особливостей поведінки на практиці.

Мета і задачі дослідження. Метою роботи є підвищення ефективності розв'язання задачі неперервної оптимізації функції від багатьох змінних за допомогою алгоритму імітації відпалу за рахунок використання модифікованих генерації рішення-кандидата та адаптивного температурного режиму.

Для досягнення мети необхідно розв'язати наступні задачі:

- проаналізувати особливості роботи метаевристичних алгоритмів оптимізації;
- удосконалити математичну модель алгоритму імітації відпалу;
- розробити програмні засоби та оцінити ефективність розробки.

Об'єктом дослідження є неперервна оптимізація функцій.

Предметом дослідження є моделі, методи та інструментальні засоби для розв'язання задач неперервної оптимізації функцій.

Методи дослідження. У роботі використано елементи математичної фізики, теорії оптимізації та обчислювального інтелекту під час дослідження роботи алгоритму імітації відпалу у ході розв'язання задачі оптимізації функції. Для аналізу та перевірки достовірності отриманих теоретичних положень застосовано експериментальне дослідження. Оброблення даних виконано за допомогою середовища *MatLab*.

Наукова новизна результатів: запропоновано удосконалений адаптивний алгоритм імітації відпалу, особливістю якого є використання модифікованих генерації рішення-кандидата та адаптивного температурного режиму, що дозволяє підвищити точність розв'язання задачі неперервної оптимізації функції від багатьох змінних.

Практичне значення результатів: розроблене математичне, алгоритмічне та програмне забезпечення дозволяє дослідити вплив параметрів алгоритму імітації відпалу на ефективність розв'язання задач оптимізації.

Апробація результатів та публікації. За результатами роботи опубліковано тези доповіді [7] на науково-практичній конференції з міжнародною участю "Актуальні задачі медичної, біологічної фізики та інформатики" (м. Вінниця, 2022). Основні результати можна використати на практиці, що підтверджено довідкою про можливість впровадження розробки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття

У практичній роботі інженера-дослідника виникають ситуації, коли явища і процеси описуються більше, ніж двома величинами. Не завжди можна розкласти розв'язання таких задач на одновимірні складові. Наприклад, прибуток підприємства залежить не тільки від кількості виробленої продукції, а й від її собівартості, від ринкового попиту і, як наслідок, ціни продажу, від курсу валют і ще від багатьох факторів. Для вивчення такого роду залежностей вводиться поняття функції від багатьох змінних $y = f(x)$, яка представляє правило, що ставить у відповідність кожному значенню аргументу x єдине значення y [8].

Функція є однопараметричною, якщо аргумент x – скаляр, і – багатопараметричною, якщо x – вектор, тобто $x = (x_1, x_2, \dots, x_n)^T$. Функція $f(x)$ називається монотонно зростаючою, якщо для будь-яких x_1 і x_2 , таких що $x_1 < x_2$, виконується умова $f(x_1) \leq f(x_2)$, і монотонно спадаючою, якщо $f(x_1) \geq f(x_2)$ [9].

Точка x^* функції $f(x)$ називається локальним мінімумом (максимумом) функції $f(x)$, якщо для всіх x , віддалених від точки x^* на відстань не більшу деякої малої величини ϵ , виконується умова $f(x^*) \leq f(x)$ ($f(x^*) \geq f(x)$).

Точка x^{**} функції $f(x)$ називається глобальним (абсолютним) мінімумом (максимумом), якщо для будь-якого x виконується умова

$f(x^{**}) \leq f(x)$ ($f(x^{**}) \geq f(x)$). Глобальний мінімум (максимум) є локальним, зворотнє виконується не завжди (рисунок 1.1) [10].

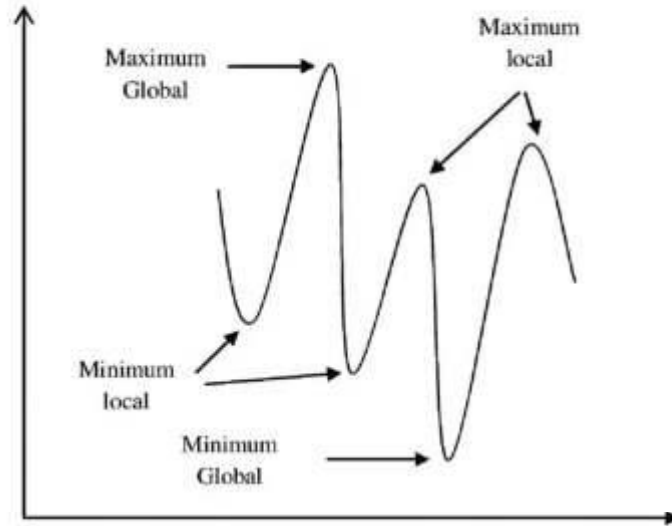


Рисунок 1.1 – Функція з локальним та глобальним оптимумом

Мінімум (максимум) x^* функції $f(x)$ називається сильним, якщо $f(x^*) < f(x)$ ($f(x^*) > f(x)$), і слабким, якщо $f(x^*) \leq f(x)$ ($f(x^*) \geq f(x)$).

Точки мінімуму і максимуму разом називають точками оптимуму (екстремуму). Функція може не мати точок екстремуму ($y = x^3$), мати один локальний мінімум ($y = x^2$) або локальний максимум ($y = -x^2$), мати кілька локальних мінімумів і максимумів ($y = \sin(x)$) тощо.

Функція $f(x)$ називається монотонно зростаючою, якщо для будь-яких x_1 і x_2 , таких що $x_1 < x_2$, виконується умова $f(x_1) \leq f(x_2)$, і монотонно спадаючою, якщо $f(x_1) \geq f(x_2)$.

Функція $f(x)$ називається унімодальною (рисунок 1.2) на інтервалі $[a; b]$, якщо вона монотонна по обидві сторони від єдиної на інтервалі

точки оптимуму (екстремуму) x^* . Якщо функція унімодальна, то локальний оптимум є глобальним [8-10].

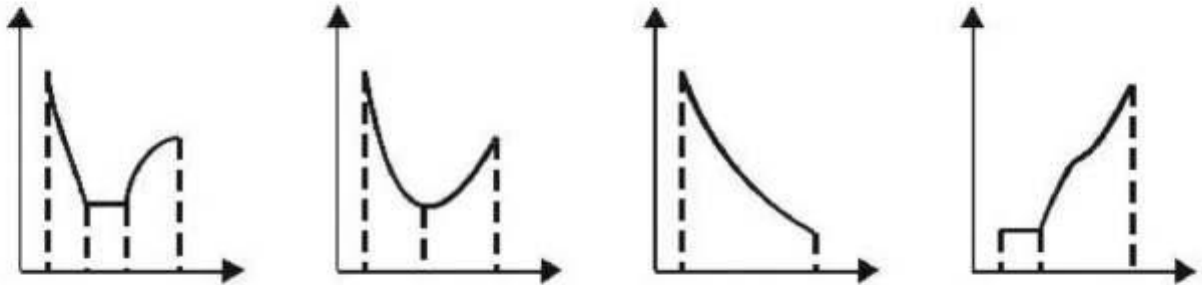


Рисунок 1.2 – Унімодальна функція

Поверхнею рівня функції $f(x)$ називають множину точок, в яких функція приймає однакові значення. Для функцій двох змінних поверхні рівня представляють собою лінії рівня.

Градiєнтом $\nabla f(x)$ неперервної диференційовної функції $f(x)$ в точці x називається вектор-стовпчик, елементами якого є частинні похідні першого порядку, обчислені в цій точці

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix} = \left(\frac{\partial f(x)}{\partial x_1} \quad \frac{\partial f(x)}{\partial x_2} \quad \dots \quad \frac{\partial f(x)}{\partial x_n} \right)^T. \quad (1.1)$$

Градiєнт функції направлений перпендикулярно до дотичної площини, проведеної в точці x , в бік найбільшого зростання функції в точці обчислення градієнта, а антиградiєнт — навпаки [9].

Нормою градієнта $\|\nabla f(x)\|$ функції $f(x)$ в точці x називають величину

$$\|\nabla f(x)\| = \sqrt{\sum_{i=1}^n \left(\frac{\partial f(x)}{\partial x_i} \right)^2}. \quad (1.2)$$

Необхідна умова екстремуму: якщо в точці x^* функція $f(x)$ досягає екстремуму, то її градієнт в цій точці дорівнює нулю. Відповідно точки екстремуму або стаціонарні точки для заданої функції слід шукати серед коренів рівняння $\nabla f(x) = 0$.

Достатні та необхідні умови екстремуму другого порядку визначаються за допомогою матриці Гессе. Матрицею Гессе $H(x)$ неперервної диференційовної в точці x функції $f(x)$ називають матрицю частинних похідних другого порядку, обчислених в цій точці

$$H(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}. \quad (1.3)$$

Матриця Гессе має розмір $n \times n$ і є симетричною відносно головної діагоналі [8-10].

Квадратична форма $\Delta x^T H(x) \Delta x$, а також відповідна матриця Гессе $H(x)$ називається:

- додатньо визначеною ($H(x) > 0$), якщо для будь-якого ненульового Δx виконується строга нерівність $\Delta x^T H(x) \Delta x > 0$;
- додатньо напіввизначеною ($H(x) \geq 0$), якщо для будь-якого ненульового Δx виконується нестрога нерівність $\Delta x^T H(x) \Delta x \geq 0$;
- від'ємно визначеною ($H(x) < 0$), якщо для будь-якого ненульового Δx виконується строга нерівність $\Delta x^T H(x) \Delta x < 0$;
- від'ємно напіввизначеною ($H(x) \leq 0$), якщо для будь-якого ненульового Δx виконується нестрога нерівність $\Delta x^T H(x) \Delta x \leq 0$;
- невизначеною ($H(x) \ngtr 0$), якщо існують такі вектори Δx і $\Delta \bar{x}$, що виконуються нерівності $\Delta x^T H(x) \Delta x < 0$, $\Delta \bar{x}^T H(x) \Delta \bar{x} > 0$;
- нульовою ($H(x) = 0$), якщо для будь-якого Δx виконується рівність $\Delta x^T H(x) \Delta x = 0$.

Операція множення матриць $\Delta x^T H(x) \Delta x$ є досить трудомісткою, тому для визначення характеру квадратичної форми використовують критерій Сільвестра (критерій перевірки достатніх умов екстремуму) [9]:

1) матриця Гессе $H(x)$ додатньо визначена тоді, коли всі її визначники (кутові мінори) додатні

$$\Delta_1 = h_{11}, \Delta_2 = \begin{vmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{vmatrix}, \dots, \Delta_n = \begin{vmatrix} h_{11} & \dots & h_{1n} \\ \dots & \dots & \dots \\ h_{n1} & \dots & h_{nn} \end{vmatrix}; \quad (1.4)$$

2) матриця Гессе $H(x)$ від'ємно визначена тоді, коли всі непарні кутові визначники від'ємні, а парні – додатні (таблиця 1.1).

Таблиця 1.1 — Критерії перевірки достатніх та необхідних умов екстремуму другого порядку при $\nabla f(x^*) = 0$ (перший спосіб)

$H(x^*)$	Умови	Перший спосіб	Тип точки x^*
> 0	Достатні умови	$\Delta_1 > 0, \Delta_2 > 0, \dots, \Delta_n > 0$	Локальний мінімум
< 0	Достатні умови	$\Delta_1 < 0, \Delta_2 > 0, \dots, (-1)^n \Delta_n > 0$	Локальний максимум
≥ 0	Необхідні умови другого порядку	Всі головні мінори визначника матриці Гессе $H(x^*)$ невід'ємні	Може бути локальний мінімум, потрібні дослідження
≤ 0	Необхідні умови другого порядку	Всі головні мінори парного порядку невід'ємні, а непарного недодатні	Може бути локальний максимум, потрібні дослідження
$= 0$	Необхідні умови другого порядку	Матриця Гессе складається з нульових елементів	Потрібні додаткові дослідження
$\begin{matrix} \geq 0 \\ < 0 \end{matrix}$	Необхідні умови другого порядку	Не виконуються умови пунктів 1-5	Немає екстремуму

Також потрібно перевірити необхідні умови екстремуму першого порядку, прирівнявши градієнт до нуля, та другого порядку, що можна записати наступним чином [9]:

1) матриця Гессе $H(x)$ додатньо напіввизначена тоді, коли всі головні мінори її визначника невід'ємні;

2) матриця Гессе $H(x)$ від'ємно напіввизначена, коли всі головні мінори парного порядку її визначника невід'ємні, а непарного – недодатні.

Головними мінорами називаються визначники m -ого порядку ($m \leq n$), які отримують із визначника матриці Гессе $H(x)$ викреслюванням будь-яких $(n-m)$ стрічок та $(n-m)$ стовпців з одними і тими ж номерами.

Інший спосіб перевірки достатніх та необхідних умов екстремуму другого порядку – власні значення λ_i матриці Гессе (таблиця 1.2).

Таблиця 1.2 — Критерії перевірки достатніх та необхідних умов екстремуму другого порядку при $\nabla f(x^*) = 0$ (другий спосіб)

$H(x^*)$	Умови	Другий спосіб	Тип точки x^*
>0	Достатні умови	$\lambda_1 > 0, \dots, \lambda_n > 0$	Локальний мінімум
<0	Достатні умови	$\lambda_1 < 0, \dots, \lambda_n < 0$	Локальний максимум
≥ 0	Необхідні умови другого порядку	$\lambda_1 \geq 0, \dots, \lambda_n \geq 0$	Може бути локальний мінімум, потрібні дослідження
≤ 0	Необхідні умови другого порядку	$\lambda_1 \leq 0, \dots, \lambda_n \leq 0$	Може бути локальний максимум, потрібні дослідження
$=0$	Необхідні умови другого порядку	$\lambda_1 = 0, \dots, \lambda_n = 0$	Потрібні дослідження
$\begin{matrix} \geq 0 \\ < \end{matrix}$	Необхідні умови другого порядку	λ_i мають різні знаки	Немає екстремуму

$$\begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} - \lambda & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} - \lambda & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} - \lambda \end{pmatrix} = 0. \quad (1.5)$$

Можна сформулювати такі правила:

- 1) Матриця Гессе $H(x)$ додатньо визначена тоді, коли власні значення λ_i додатні;
- 2) матриця Гессе $H(x)$ від'ємно визначена тоді, коли власні значення λ_i від'ємні;
- 3) при будь-якій іншій комбінації власних значень λ_i квадратична форма буде невизначеною.

1.2 Постановка задачі оптимізації

Цільова функція $f(x)$ (функція втрат, критерій якості) – це функція, яка дозволяє розрахувати значення критерію оптимальності системи.

Керовані змінні x (проектні параметри) – параметри системи, які впливають на значення цільової функції, в процесі оптимізації можуть бути змінені.

Обмеження (умови, область допустимих рішень) – множина допустимих значень керованих змінних.

Дана цільова функція $f(x) = f(x_1, x_2, \dots, x_n)$, визначена на множині допустимих рішень $D \subseteq R^n$. У найбільш загальному вигляді задача безумовної неперервної оптимізації функції від багатьох змінних формулюється так: потрібно знайти глобальний мінімум функції $f(x)$ на множині допустимих рішень D , тобто таку точку $x^* \in D$, що [8-10]

$$f(x^*) = \min f(x), \quad (1.6)$$

де $x = (x_1, x_2, \dots, x_n)^T$, $D = \{x \mid x_i \in [LB_i, UB_i], i = 1, 2, \dots, n\}$.

У даний час, в зв'язку з доступністю обчислювальної техніки, велика увага приділяється використанню аналітичного та чисельних методів оптимізації, які можна розділити на дві великі групи: методи безумовної та умовної оптимізації. Переважна більшість цих методів призначена для розв'язання задач пошуку мінімуму, тому задача пошуку максимуму функції $f(x)$ зводиться до задачі пошуку мінімуму шляхом заміни знака перед функцією на протилежний:

$$f(x) \rightarrow \max \Rightarrow -f(x) \rightarrow \min. \quad (1.7)$$

Функція $f(x)$ може бути багатоекстремальною, тому шукане рішення в загальному випадку не єдине.

1.3 Огляд методів розв'язання задачі оптимізації функції

При вирішенні конкретної задачі оптимізації дослідник, перш за все, повинен вибрати математичний метод, який призводить би до результатів з найменшими витратами на обчислення або давав можливість отримати найбільший обсяг інформації про рішення. Вибір того чи іншого методу значною мірою визначається постановкою задачі, а також математичною моделлю об'єкта оптимізації. Для вирішення задач оптимізації застосовують переважно такі методи [8-10]:

- класичний математичний аналіз;
- використання невизначених множників Лагранжа;
- динамічне програмування;
- лінійне програмування;
- нелінійне програмування;
- метаевристичні схеми.

Як правило, не можна рекомендувати один метод, який можна використовувати для вирішення всіх без винятку задач, що виникають на практиці. Деякі методи спеціально розроблені або найкраще підходять для розв'язання задач оптимізації із математичними моделями певного виду.

Методи дослідження функцій класичного математичного аналізу є найбільш відомими методами вирішення нескладних задач оптимізації. Звичайною областю використання даних методів є задачі з відомим аналітичним виразом критерію оптимальності, що дозволяє знайти не дуже складний, але також аналітичний вираз для похідних. Отримані прирівнюванням нулю похідних рівняння, що визначають екстремальні рішення оптимальної задачі, вкрай рідко вдається розв'язати аналітичним

шляхом, тому зазвичай застосовують обчислювальні машини. Потрібно розв'язати систему рівнянь, найчастіше нелінійних, використовуючи чисельні методи, аналогічні методам нелінійного програмування [9].

Додаткові труднощі під час вирішення задач оптимізації методами дослідження функцій класичного аналізу виникають тому, що система рівнянь, одержана внаслідок їх застосування, забезпечує лише необхідні умови оптимальності. Тому всі рішення даної системи (а їх може бути багато) мають бути перевірені на достатність. У результаті такої перевірки спочатку відкидають рішення, які не визначають екстремальні значення критерію оптимальності, а потім серед екстремальних рішень, що залишаються, обирають рішення, що задовольняє умовам задачі.

Екстремумом функції $f(x)$ у стаціонарній або сідловій точці $M_0(x^*)$ (в якій градієнт дорівнює нулю) називають таке її значення, яке менше (більше) всіх інших її значень, що набуваються в точках $M(x)$, достатньо близьких до точки $M_0(x^*)$ і відмінних від неї. Точки екстремуму функції можна шукати, використовуючи аналітичний підхід, тобто застосовуючи необхідні і достатні умови екстремуму [8].

Для ідентифікації оптимуму спочатку потрібно перевірити необхідні умови екстремуму першого порядку, прирівнявши градієнт до нуля, тобто

$$\text{grad } f(x) = \nabla f(x) = 0. \quad (1.8)$$

Далі знайти стаціонарні точки в результаті розв'язання n рівнянь (у загальному випадку нелінійних алгебраїчних) з n невідомими. Для наближеного розв'язку можна використовувати методи простої ітерації, Ньютона тощо.

У знайдених точках перевірити достатні умови екстремуму з використанням критерію Сільвестра та, якщо потрібно, необхідні умови екстремуму другого порядку. Можна також застосувати спосіб на основі обчислення власних значень матриці Гессе $H(x)$. Потім обчислити значення функції $f(x)$ в точках екстремуму.

Пошук точок локального екстремуму функції від багатьох змінних за допомогою необхідних і достатніх умов екстремуму називають непрямим методом [9].

Метод множників Лагранжа застосовують для вирішення задач такого ж класу складності, як і при використанні звичайних методів дослідження функцій, але за наявності обмежень типу рівності на незалежні змінні. До вимоги можливості отримання аналітичних виразів похідних від критерію оптимальності при цьому додається аналогічна вимога щодо аналітичного виду рівнянь обмежень [8].

Деяке ускладнення виникає від введення додаткових невизначених множників, унаслідок чого порядок системи рівнянь збільшується на кількість обмежень. Множники Лагранжа можна застосовувати для вирішення задач оптимізації об'єктів на основі рівнянь із частинними похідними та задач динамічної оптимізації. При цьому замість розв'язання системи рівнянь для пошуку оптимуму необхідно інтегрувати систему диференціальних рівнянь.

Динамічне програмування є ефективним методом вирішення задач оптимізації для дискретних багатостадійних процесів, коли критерій оптимальності задається як адитивна функція критеріїв оптимальності з окремих стадій процесу. Фактично метод динамічного програмування є алгоритмом визначення оптимальної стратегії управління всіма стадіями

процесу. У цьому випадку закон управління для кожної стадії знаходять шляхом вирішення часткових задач оптимізації послідовно [10].

Результати зазвичай не можуть бути виражені в аналітичній формі, а записуються у вигляді таблиць. Обмеження на змінні задачі не впливають на загальний алгоритм рішення, а враховуються під час вирішення часткових задач оптимізації кожної стадії процесу. При вирішенні задач методом динамічного програмування, як правило, використовують обчислювальні машини, які мають достатній обсяг пам'яті для зберігання проміжних результатів рішення.

Лінійне програмування представляє собою математичний апарат, розроблений для вирішення задач оптимізації з лінійними виразами для критерію оптимальності та лінійними обмеженнями на область визначення змінних [8]. Такі задачі зазвичай зустрічаються у ході оптимального планування виробництва з обмеженою кількістю ресурсів, при визначенні оптимального плану перевезень (транспортні задачі), задачі розміщення тощо. Тип обмежень (рівності або нерівності), які використовуються, не позначається на можливості застосування зазначеного апарату. Додаткової перевірки оптимальності для одержаних рішень виконувати не потрібно.

Методи нелінійного програмування застосовують для розв'язання задач оптимізації із нелінійними цільовими функціями. На незалежні змінні можуть бути накладені обмеження у вигляді нелінійних виразів, які мають вигляд рівностей або нерівностей. Такі методи іноді називають також прямими методами вирішення задач оптимізації. Крім того, ці методи є основою побудови систем автоматичної оптимізації – оптимізаторів, які безпосередньо застосовуються для управління виробничими процесами [9].

Розв'язання задач мінімізації функцій прямими методами пов'язане з

побудовою послідовності точок, які задовольняють умові

$$f(x^{k+1}) < f(x^k), \quad k = 0, 1, \dots \quad (1.9)$$

при цьому пошук точки мінімуму x^* починається у довільно вибраній точці x^0 , а наступні точки послідовності знаходять за формулою

$$x^{(k+1)} = x^{(k)} + l^{(k)}d^{(k)}, \quad (1.10)$$

де $x^{(k+1)}$ – наступна точка; $x^{(k)}$ – поточна точка; $l^{(k)}$ – довжина кроку; $d^{(k)}$ – напрямок переходу з точки $x^{(k)}$ в точку $x^{(k+1)}$; k – номер ітерації.

Методи рішення задач безумовної оптимізації відрізняються відносно високим рівнем розвитку в порівнянні з іншими методами нелінійного програмування. Багатовимірні методи, що реалізують процедуру пошуку оптимуму на основі обчислення значень функції, із загальних позицій можна розділити на евристичні та теоретичні.

Евристичні методи, як це випливає з назви, реалізують процедуру пошуку за допомогою інтуїтивних геометричних уявлень і забезпечують отримання деяких емпіричних результатів. З іншого боку, теоретичні методи засновані на фундаментальних математичних теоремах і володіють такими операційними властивостями, як збіжність (принаймні при виконанні деяких певних умов) [9].

У залежності від найвищого порядку частинних похідних функції $f(x)$, які використовуються для формування $l^{(k)}$ і $d^{(k)}$, методи мінімізації функцій кількох змінних ділять на три групи [8, 9]:

1) методи нульового порядку, які використовують інформацію тільки про значення функції. Частина цих методів не має строгого теоретичного обґрунтування. Вони побудовані на основі евристичних міркувань. Тому питання збіжності даних методів мало вивчені, а оцінки швидкості збіжності зазвичай відсутні. Разом з тим вони пов'язані з методами першого і другого порядків, що у ряді випадків дозволяє оцінювати ефективність алгоритмів прямого пошуку у випадку застосування до мінімізації деяких класів функцій.

До методів нульового порядку пошуку екстремуму функції багатьох змінних відносяться методи, що не використовують похідні для вибору напрямку спуску (метод Хука-Дживса, Розенброка, метод деформованого многогранника Нелдера-Міда, Пауелла, випадкового пошуку);

2) методи першого порядку, які використовують інформацію про перші похідні функції. До цієї групи відносяться методи градієнтного та покоординатного спуску, Гауса-Зейделя, Флетчера-Рівса, Девідона-Флетчера-Пауелла, метод кубічної інтерполяції тощо;

3) методи другого порядку, які використовують також інформацію про другі похідні функції. Це методи Ньютона, Ньютона-Рафсона, Марквардта тощо.

Моделювання на сучасних комп'ютерах є чи не найпотужнішим засобом дослідження комплексних динамічних систем, природних та технологічних процесів виробництва, економіки тощо. Фактично комп'ютерні моделі використовують для отримання нових знань про об'єкт або для наближеної оцінки поведінки систем, які дуже складні для аналітичного дослідження [2, 7].

В останні роки інтенсивно розвиваються алгоритми комп'ютерної пошукової оптимізації, які називають поведінковими, інтелектуальними, метаевристичними, натхненними (інспірованими) природою, роевими, багатоагентними, популяційними. За їх допомогою успішно вирішуються складні оптимізаційні задачі, наприклад, автоматизованого проектування, синтезу складних хімічних сполук, оптимального управління динамічними системами тощо [7].

Метаевристичні методи забезпечують знаходження квазіоптимального розв'язку. До них відносяться методи розв'язку без використання колективу інтелектуальних програмних агентів та методи з використанням колективу агентів.

Метаевристичні методи комбінують методи пошуку локальних та глобальних розв'язків у абстрактні стратегії евристичної оптимізації задач. Такі методи включають сучасні стратегії пошуку, які інспіровані живою та неживою природою. До них прийнято відносити алгоритм імітації відпалу, пошук із заборонами, імовірнісні жадібні алгоритми, генетичні методи, наслідування поведінки мурашиної колонії, рою бджіл, зграї птахів, вовків, котів, кажанів, колонії мурах, косяку риб, рою часток, бактеріального рою, імунних систем тощо [4].

Ідея цих методів заснована на припущенні, що цільова функція $f(X)$ має багато локальних екстремумів, а аналіз усіх допустимих рішень неможливий, не дивлячись на їх кінцеву кількість. У такій ситуації потрібно зосередити пошук на найбільш перспективних частинах області допустимих рішень. Таким чином, задача зводиться до виявлення таких частин і швидкого їх перегляду. У принципі, цими методами можна знаходити як досить якісні розв'язки, так і досить віддалені від

оптимального. Якість результатів та тривалість обчислень залежать від реалізації окремих елементів методу.

Метаевристичні алгоритми пошукової оптимізації, в порівнянні з класичними, мають незаперечні переваги, перш за все, при вирішенні задач високої розмірності, мультимодальних та погано формалізованих задач. Розглянуті алгоритми програють алгоритму повного перебору при вирішенні задачі оптимізації невисокої розмірності. До недоліків слід віднести також сильну залежність їхньої ефективності від значень вільних параметрів, кількість яких у більшості алгоритмів досить велика [4].

Найбільш універсальним алгоритмом оптимізації є алгоритм імітації відпалу (simulated annealing), який заснований на аналогіях із термодинаміки. Великою перевагою даного алгоритму є можливість уникнути так званих "локальних пасток" у локальних мінімумах функції, яка оптимізується, за рахунок прийняття змін, що тимчасово погіршують результат. Така процедура відображає нагрівання розплаву для запобігання його швидкого охолодження при протіканні відповідного фізичного процесу. Ще однією перевагою є те, що навіть за умов нестачі обчислювальних ресурсів для знаходження глобального мінімуму даний алгоритм, як правило, видає непогане рішення – один із хороших локальних мінімумів [5].

Слід зазначити, що ні один метод або клас методів не відрізняється високою ефективністю при вирішенні оптимізаційних задач різних типів. Зокрема, можливі випадки, коли відбувається переповнювання пам'яті обчислювальної машини; в інших ситуаціях обчислення значень цільової функції вимагає надмірних витрат часу; в деяких завданнях потрібно отримати рішення з дуже високим ступенем точності. У ряді програм або

неможливо, або дуже важко знайти аналітичні вирази для похідних цільової функції.

Тому, якщо передбачається використовувати градієнтні методи, слід застосувати процедуру різницевої апроксимації похідних. У свою чергу це призводить до необхідності експериментального визначення довжини кроків, що дозволяє встановити належну відповідність між помилкою округлення і помилкою апроксимації. Таким чином, інженер змушений пристосовувати застосовуваний метод до конкретних характеристик розв'язуваної задачі.

Навіть хороший метод можна “знищити” поганим програмуванням, невдалим вибором параметрів алгоритму тощо. Крім того, методи можуть поводитися по-різному на етапах процесу мінімізації. Задовільного способу подолання вказаних труднощів не існує. Єдине, що можна зробити в подібній ситуації — це привести дані про результати обчислень в розгорненій формі, що дозволяє порівнювати методи по різних критеріях. Не слід забувати, що пошук рішення завжди залишається мистецтвом, якому можна навчитися лише шляхом проб і помилок, застосовуючи різні методи для вирішення конкретних задач [1].

1.4 Висновки

У даному розділі визначено, що під час проектування або керування різними об'єктами, системами і процесами досить часто виникає необхідність досягнути мети оптимальним способом. Процес оптимізації лежить в основі всієї інженерної діяльності, оскільки класичні функції

інженера полягають у тому, щоб, з одного боку, проектувати нові, ефективніші та менш дорогі технічні системи, а з другого — розробляти методи підвищення якості функціонування існуючих систем.

У ході роботи розглянуто основні поняття теорії оптимізації, представлено загальну постановку задачі оптимізації та наведено аналітичний алгоритм пошуку оптимуму. Ефективність методів оптимізації тісно пов'язана із досягненнями у галузі математики, а саме з розробкою метаевристичних схем, ідею формування яких розглянемо у наступному розділі роботи.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ УДОСКОНАЛЕНОГО АДАПТИВНОГО АЛГОРИТМУ ІМІТАЦІЇ ВІДПАЛУ

2.1 Алгоритм імітації відпалу

Оригінальна назва даного алгоритму виходить із аналогії з фізичним процесом відпалу, який направлений на отримання твердих тіл з низькою енергією стану. У кожного металу є кристалічна решітка, яка описує геометричне положення атомів речовини. Сукупність позицій усіх атомів називається станом системи, кожному стану відповідає певний рівень енергії [11-14].

У фізиці відпалом є процес, в якому тверде тіло спочатку розплавляють шляхом збільшення температури, потім поступово знижують температуру для відновлення твердого стану з низькою енергією (кристалізація речовини). Мета відпалу – привести систему в стан з найменшою енергією. Чим нижчий рівень енергії, тим краща кристалічна решітка, тобто тим менше у неї дефектів і міцніший метал.

У ході відпалу метал спочатку нагрівають до деякої температури T_{max} , що примушує атоми кристалічної решітки покинути свої позиції. Потім починається повільне і контрольоване охолодження. Атоми прагнуть потрапити в стан з меншою енергією, проте, з певною ймовірністю p , вони можуть перейти і в стан з більшою енергією. Значення p зменшується разом з температурою T . Ретельний відпал з рядом температурних рівнів, на яких температура T достатньо довго зберігається з метою досягнення

рівноваги системи, приводить до більш регулярних структур, що відповідають твердим тілам з низькою енергією. Процес завершується, коли температура падає до заданого значення $T_{\min} \rightarrow 0$ [15].

Аналогії між фізичним процесом охолодження і розв'язанням задачі оптимізації наведено в таблиці 2.1.

Таблиця 2.1 – Відповідність понять

Фізичні системи	Задачі оптимізації
Відпал	Моделювання відпалу
Стан, x_i	Рішення, x_i
Енергія, $E(x_i)$	Цільова функція, $f(x_i)$
Охолоджений стан, x^*	Оптимальне рішення, x^*
Поступове охолодження, T	Покрокове покращення, T

Фактично дана метаевристика представляє собою загальний алгоритмічний підхід для розв'язання задач оптимізації. Вона є одним із прикладів застосування локального пошуку та методу Монте-Карло, що дозволяє уникнути "поганих" локальних оптимумів. Моделюючи процес відпалу, шукається така точка або множина точок, на якій досягається оптимум (зазвичай мінімум) деякої функції $f(x)$. У процесі пошуку оптимального рішення з деякою ймовірністю допускається перехід у стан із більш високим значенням цільової функції. Ця властивість дозволяє виходити з локальних оптимумів.

На початку роботи алгоритму ймовірність переходу в стан із більш високим значенням цільової функції повинна бути досить велика, щоб існувала можливість переходу від вибраного наближення до будь-якого

іншого рішення. У процесі роботи алгоритму ймовірність такого переходу поступово зменшується за допомогою схеми охолодження [16]. Алгоритм імітації відпалу схожий на градієнтний спуск, але за рахунок випадковості вибору проміжної точки він буде потрапляти в локальні оптимуми рідше.

На стартовому етапі алгоритму випадковим чином вибирається початковий стан системи x для початкової температури T_{\max} . Наступний стан вибирається відповідно до заданого породжуючого сімейства імовірнісних розподілів $G(x, T)$, яке при фіксованих x і T задає випадковий елемент $\bar{x} = G(x, T)$ зі значеннями в просторі Ω . Це означає, що на кожній ітерації алгоритм стохастично вибирає новий стан термодинамічної системи \bar{x} . Далі необхідно обчислити зміну енергії системи ΔE :

$$\Delta E = E(\bar{x}) - E(x) = f(\bar{x}) - f(x), \quad (2.1)$$

де $E(\bar{x})$ – значення енергії (цільової функції) нового стану (вектора x_i);
 $E(x)$ – значення енергії поточного стану (вектора x_{i-1}).

Якщо $\Delta E \leq 0$, то це означає, що зроблена зміна покращує стан системи, а новий стан замінює старий, тобто приймається з ймовірністю $P = 1$; інакше розраховуємо певне значення ймовірності (відповідно до розподілу Гіббса). Це можна представити процедурою Метрополіса [7, 12]:

$$P = \begin{cases} 1, & \text{якщо } \Delta E = E(x_i) - E(x_{i-1}) \leq 0; \\ \frac{1}{1 + \exp\left(\frac{\Delta E}{k_b \cdot T_{i-1}}\right)} \approx \exp\left(-\frac{\Delta E}{k_b \cdot T_{i-1}}\right), & \text{якщо } \Delta E = E(x_i) - E(x_{i-1}) > 0, \end{cases} \quad (2.2)$$

де k_b – константа Больцмана або константа налаштування (часто приймається рівною одиниці); T_i – температура на $(i-1)$ -ій ітерації алгоритму.

Геометрично ідею даної формули можна представити на рисунку 2.1. Одиничний відрізок (ймовірність) розіб'ємо на дві частини відповідно до значення отриманої імовірності. Якщо число потрапило в ліву частину – здійснюємо перехід до нового стану, в праву – не робимо нічого (залишаємо старий стан).



Рисунок 2.1 – Геометрична інтерпретація переходів

Важливою особливістю алгоритму є те, що процес починається з такої великої температури T , яка б робила ймовірність знаходження системи в деякому стані незалежною від значення цільової функції E в цьому стані. Температура впливає на величину переміщень між послідовно отриманими рішеннями задачі оптимізації.

Високій температурі, з якою починається процес розв'язання, відповідають великі переміщення (алгоритм ще уникає локального мінімуму), а низькій температурі – маленькі переміщення (алгоритм поступово відшукує рівновагу). Температура повинна зменшуватися поступово до мінімального значення $T_{\min} \rightarrow 0$ так, щоб для кожного нового значення T процедура Метрополіса виконала таку кількість ітерацій, яка б гарантувала те, що буде досягнута температурна рівновага.

Описаний процес повторюється певну кількість ітерацій, враховуючи постійну зміну температури за певною схемою і кількість змін станів на одній ітерації алгоритму. На рисунку 2.2 показана типова поведінка даного алгоритму.

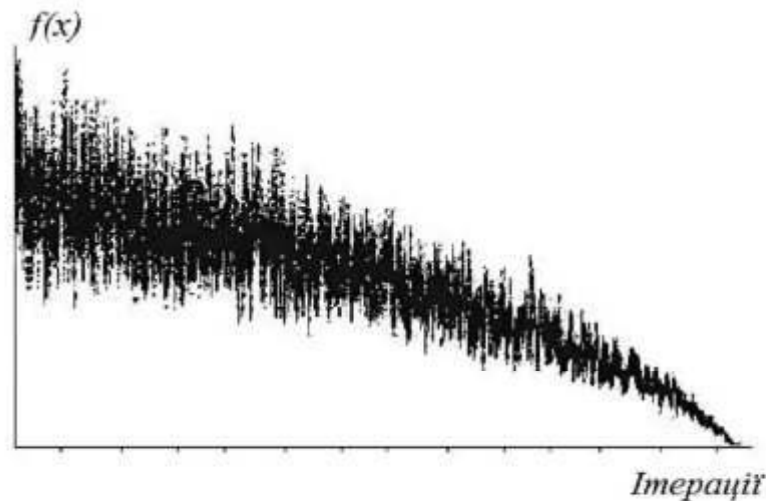


Рисунок 2.2 – Процес пошуку оптимуму за алгоритмом імітації відпалу

На перших ітераціях при високій температурі спостерігається великий розкид значень цільової функції, а процес пошуку нагадує броунівський рух. Зі зниженням температури розкид зменшується, спостерігається явна тенденція до падіння середньої величини цільової функції. При низькій температурі можна побачити стагнацію процесу пошуку і зупинку в одному з локальних оптимумів [16].

Доведено: якщо охолодження відбувається безмежно великий час, процес імітації відпалу виконується безмежно велику кількість ітерацій, то кінцевий стан відповідає оптимуму, тобто значення цільової функції, отримане таким чином, відповідає глобальному мінімуму.

2.2 Функції визначення температури

Слід зазначити, що функція визначення температури (температурний режим) повинна бути монотонно спадною. Спершу розглянемо, відпал Больцмана, в якому зміна температури на i -ій ітерації задається виразом

$$T_i = \frac{T_{\max}}{\ln(1+i)}, \quad i > 0. \quad (2.3)$$

Сімейство розподілів $G(x, T)$ для генерації нового стану вибирається як сімейство нормальних розподілів з математичним очікуванням x і дисперсією T , тобто задається щільністю ймовірності

$$g(\bar{x}; x, T) = \frac{1}{\sqrt{(2\pi T)^n}} \cdot \exp\left(-\frac{\|\bar{x} - x\|^2}{2T}\right), \quad (2.4)$$

де $\|\bar{x} - x\| = \sqrt{\sum_{j=1}^n (\bar{x}'_j - x'_j)^2}$ – евклідова норма.

Для даної схеми доведено, що при великих значеннях T_{\max} та кількості ітерацій i , вибір такого сімейства розподілів гарантує знаходження глобального оптимуму. Основним недоліком такого відпалу є повільне зменшення температури, що може призводити до повільної збіжності алгоритму [15].

Щоб подолати цей недолік Х.Х. Цу та Р.Л. Хартлі [17] запропонували швидкий відпал (відпал Коші) без втрати гарантії знаходження глобального оптимуму. Використано такий режим охолодження

$$T_i = \frac{T_{\max}}{i}. \quad (2.5)$$

Для генерації нового стану використано сімейство розподілів Коші $G(x, T)$ з параметрами зсуву x та масштабу T . Щільність ймовірності можна представити у вигляді

$$g(\bar{x}; x, T) = \frac{T}{\pi \cdot (\|\bar{x} - x\|^2 + T^2)^{\frac{n+1}{2}}}. \quad (2.6)$$

Даний розподіл не дуже зручно моделювати в просторі $n > 1$. Тому застосовують множення n одновимірних розподілів Коші

$$g(\bar{x}; x, T) = \frac{1}{\pi^n} \prod_{j=1}^n \frac{T}{|x_j - \bar{x}_j|^2 + T^2}. \quad (2.7)$$

Знаходження глобального оптимуму гарантується, але при використанні дуже повільного закону зміни температури

$$T_i = \frac{T_{\max}}{i^{1/n}}. \quad (2.8)$$

Недоліки попередніх схем привели до того, що у 1989 році американським вченим Л. Інґбером був розроблений адаптивний алгоритм відпалу [18-20]. У ньому температура для кожної із координат може відрізнятися, тобто T також є вектором розмірності n . Для побудови сімейства розподілів вводиться функція

$$g_T(y) = \prod_{j=1}^n \frac{1}{2 \cdot (\|y_j\| + T_j) \cdot \ln(1 + 1/T_j)} = \prod_{j=1}^n g_{(j;T)}(y_j), \quad y_j \in [-1,1]. \quad (2.9)$$

Значення випадкової величини y_j можна знайти за виразом

$$y_j = \operatorname{sgn}\left(\alpha_j - \frac{1}{2}\right) \cdot T_j \cdot \left((1 + 1/T_j)^{|2\alpha_j - 1|} - 1\right), \quad (2.10)$$

де $\alpha_j \in [0,1]$ – незалежні випадкові величини, які розподілені рівномірно.

Доведено, що закон зміни температури

$$T_j(i) = T_{(j;\max)} \cdot \exp(-c_j \cdot i^{1/n}), \quad c_j > 0 \quad (2.11)$$

дає статистичну гарантію знаходження глобального екстремуму. Для ймовірності прийняття використовується окрема шкала температури, яка змінюється за таким же законом. При реалізації схеми охолодження параметр c_j управляється двома змінними m_j та k_j :

$$c_j = m_j \cdot \exp(-k_j/n). \quad (2.12)$$

Переваги алгоритму:

- експоненціальне спадання температури;
- не потрібно змінювати всі координати одночасно;
- швидке моделювання розподілу незалежно від розмірності задачі.

Недоліки:

- багато евристичних параметрів;
- час налаштування алгоритму для розв'язання певної задачі.

Схему, розроблену К. Яо у 1995 році, можна отримати повторним застосуванням ідеї Інгбера [21]. Для побудови сімейства розподілів використано функцію

$$g_T(y) = \prod_{j=1}^n g_{(j,T)}(y_j) = \prod_{j=1}^n \frac{1}{2 \cdot \left(\|y_j\| + \frac{1}{\ln(1/T_j)} \right) \cdot \ln(1 + \ln(1/T_j))}. \quad (2.13)$$

Тоді при зміні температури за законом

$$T_j(i) = T_{(j,\max)} \cdot \exp(-\exp(c_j \cdot i^{1/n})), \quad c_j > 0 \quad (2.14)$$

досягається статистична гарантія знаходження глобального екстремуму.

Але збільшення швидкості охолодження не означає прискорення розв'язання задачі оптимізації. "Особливість" розподілу призводить до того, що алгоритм генерує велику кількість "довгих" переходів, які відкидаються через низьку ймовірність їхнього прийняття. Отже, незважаючи на те, що даний процес можна повторювати нескінченно, отримуючи такі закони зміни температури як

$$T_j(i) = T_{(j,\max)} \cdot \exp(-\exp(\exp(\dots \exp(c_j \cdot i^{1/n}) \dots))), \quad c_j > 0, \quad (2.15)$$

цінність таких "модифікацій" видається сумнівною. Більш того, можна прийти до тривіального алгоритму випадкового пошуку, яким стає алгоритм імітації відпалу при температурі $T = 0$.

Отже, проблема пошуку схем охолодження, які забезпечують не таке швидке зниження температури, але більшу швидкість збіжності до оптимуму, залишається відкритою.

Далеко не завжди вистачає обчислювальних ресурсів на пошук глобального екстремуму. Схеми "гасіння" не гарантують знаходження глобального оптимуму, але дозволяють отримати "хороше" субоптимальне рішення задачі, а інколи – глобальний екстремум. Основа таких схем полягає в тому, щоб скомбінувати сімейство розподілів однієї з розглянутих схем з більш швидким законом охолодження. Наприклад, можна розглядати нормальний розподіл з відпалу Больцмана, а температуру зменшувати за законом [15]

$$T_{i+1} = \alpha \cdot T_i, \quad (2.16)$$

де $\alpha = 0,7...0,9999$ – коефіцієнт охолодження.

Для схеми надшвидкого гасіння можна використати формули для надшвидкого відпалу з введенням параметра гасіння b , тобто [22]

$$T_j(i) = T_{(j,\max)} \cdot \exp(-c_j \cdot i^{b/n}), \quad c_j > 0. \quad (2.17)$$

Схеми гасіння дуже швидко збігаються, а для певних задач можуть давати досить непогані субоптимальні рішення, але необхідно методом проб і помилок підбирати необхідні параметри, що може потребувати багато часу.

Слід зазначити, що генерацію нового стану для схем Больцмана і Коші можна виконати за формулами

$$x_j = x_{j-1} + \Delta x = x_{j-1} + T_{j-1} \cdot G(x_{j-1}, T_{j-1}), \quad (2.18)$$

$$x_j = x_{j-1} + \Delta x = x_{j-1} + G(x_{j-1}, T_{j-1}), \quad (2.19)$$

а для адаптивного відпалу використати рівняння

$$x_j = x_{j-1} + \Delta x = x_{j-1} + (UB - LB) \cdot y_{j-1}, \quad (2.20)$$

де LB , UB – матриці-стовпці, нижня та верхня межі допустимих значень області пошуку D розмірності n , $UB > LB$; $y_{j-1} \in [-1; 1]$.

2.3 Удосконалення алгоритму

У рівнянні (2.20) інтервал $[LB, UB]$ представляє допустимий діапазон зміни j -ої змінної, а $y_{j-1} \in [-1; 1]$. У випадках, коли він є досить широким, складова Δx матиме велике значення, яке вийде за допустимі межі. Тут можна навести низку недоліків цього процесу:

1. Відповідно буде згенероване випадкове значення Δx , що по суті представляє випадковий пошук.

2. Інший варіант згенерувати багато випадкових y_{j-1} , щоб знайти можливі значення для змінної x_j , але це займе зайвий час.

3. Допустимі значення для y_{j-1} сильно змінюють значення x_j , тому нове значення для x_j дуже сильно відрізняється від попереднього.

Враховуючи причини, зазначені вище, запропоновано простий варіант уникнення даних недоліків із популяційних алгоритмів оптимізації. Отримане значення y_{j-1} слід помножити на коефіцієнт $0 < \alpha \leq 1$, щоб знайти

рішення-кандидат x_j , яке не виходить за межі допустимого діапазону $[LB, UB]$. Запропонований вираз можна навести у такій формі:

$$x_j = x_{j-1} + \Delta x = x_{j-1} + \alpha \cdot y_{j-1} \cdot (UB - LB). \quad (2.21)$$

Значення α залежить від задачі та може бути вибране постійним або змінним залежно від кількості ітерацій алгоритму.

Всі алгоритми імітації відпалу з деякою ймовірністю допускають перехід у стан з більш високим значенням цільової функції у процесі пошуку рішення для того, щоб точка могла залишити окіл локального мінімуму. Але для забезпечення цієї властивості у звичайному методі імітації відпалу потрібно знижувати температуру T дуже повільно, що значно збільшує час обчислень. В адаптивному алгоритмі імітації відпалу цей недолік усувається за рахунок введення нового закону зменшення температури та можливості її зміни по кожній координаті окремо з урахуванням специфіки впливу кожного параметра конкретної задачі оптимізації.

Запропоновано комбінувати схеми охолодження та виконувати періодичний нагрів або перенагрів та повторний відпал у випадку, якщо алгоритм зациквився і не знайшов розв'язок із заданою точністю. Обидві процедури дозволяють виконати більш ретельний аналіз області пошуку, уникнути застрягання у локальному оптимумі [15, 16].

У ході перенагріву необхідно підвищувати температуру у тому випадку, якщо значення функції f не змінюється певну кількість кроків k , а температура опустилась нижче певного мінімального порогу, який вже не дозволяє виконувати "стрибки". Встановити температуру в такому випадку

можна або на стартовий рівень, або дещо нижче, або ж до показника, на якому було знайдене найкраще рішення.

У роботі використано температурний режим з адаптивним відпалом (T_{ad}) та лінійним перенагрівом (T_{lin}). Якщо найкраще рішення з використанням адаптивного відпалу не змінюється впродовж $k = 500 \dots 1500$ кроків і температура не більша 0,25, тоді застосовуємо режим перенагріву з коефіцієнтом $a = 2,75$ (визначено експериментально):

$$T_{i+1} = a \cdot T_i, \quad (2.22)$$

де a – коефіцієнт перенагріву.

Кількість повторних відпалів можна встановити в межах $h = 3 \dots 10$. Він застосовується з тими ж стартовими параметрами, оскільки алгоритм має стохастичну природу.

2.4 Висновки

У даному розділі сформульовано основні принципи роботи та наведено математичну модель удосконаленого алгоритму імітації відпалу для розв'язання оптимізаційних задач. Визначено, що даний алгоритм можна використовувати, коли задача характеризується великою розмірністю та складністю цільової функції. У наступному розділі будуть проведені експериментальні дослідження ефективності роботи даного алгоритму.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Опис роботи програми

У даному підрозділі детально розглянемо кроки алгоритму реалізованого у програмному забезпеченні:

Крок 1. Задати початкові значення глобальних параметрів:

- розмірність простору пошуку n ;
- температура $T : T_{\max} = (T_{(1,0)}, T_{(2,0)}, \dots, T_{(n,0)})$, $T_{\min} \rightarrow 0$;
- вектор параметрів зменшення температури $M = (P_1, P_2, \dots, P_n)^T$ та

$$K = (Q_1, Q_2, \dots, Q_n)^T;$$

– початкове значення t_{\max} та коефіцієнти зменшення m, k глобальної температури t_i ;

- коефіцієнт $0 < \alpha \leq 1$;
- кількість ітерацій локальної оптимізації l ;
- кроків до перенагріву $steps$;
- коефіцієнт нагріву a ;
- температура перенагріву $T_{re-heat}$;
- константа Больцмана $k_b > 0$;
- максимальна кількість ітерацій N ;
- повторних відпалів h .

Крок 2. Задати початковий n -вимірний вектор аргументів функції

$$x = (x_1, \dots, x_n):$$

$$x_j^{(0)} = LB_j + rand_j \cdot (UB_j - LB_j), \quad (3.1)$$

де $rand$ – незалежне дійсне випадкове число, рівномірно розподілене на інтервалі $[0, 1]$; LB, UB – нижня та верхня межі допустимих значень області пошуку D розмірності n , $UB > LB$.

Крок 3. Встановити ітератор алгоритму $i = 1$.

Крок 4. Виконати локальну оптимізацію, повторивши пункти 5-7 l разів.

Крок 5. Згенерувати рішення-кандидат. Для цього виконати такі дії:

а) покласти $j = 1$;

б) обчислити j -ту координату вектора x :

$$y_j^{(i)} = \text{sgn}\left(rand_j - \frac{1}{2}\right) \cdot T_{(j,i)} \cdot \left((1 + 1/T_{(j,i)})^{|2rand_j - 1|} - 1\right), \quad (3.2)$$

$$x_j^{(i)} = x_j^{(i-1)} + \alpha \cdot y_j^{(i)} \cdot (UB_j - LB_j). \quad (3.3)$$

Якщо координата $x_j \notin [LB_j, UB_j]$, то згенерувати за допомогою рівномірного розподілу (3.1) на відріжку $[LB_j, UB_j]$ випадкову величину та вважати x_j рівним її значенню;

в) якщо $j < n$, то покласти $j = j + 1$ і перейти до пункту (б), інакше формування вектора z завершити та перейти далі.

Крок 6. Обчислити значення цільової функції $f(x^{(i-1)})$ та $f(x^{(i)})$, а також знайти приріст енергії $\Delta f = f(x^{(i)}) - f(x^{(i-1)})$.

Крок 7. Знайти нове рішення:

- якщо у точці $x^{(i)}$ значення цільової функції не більше, ніж у точці $x^{(i-1)}$ ($\Delta f \leq 0$), то перейти до точки $x^{(i-1)}$, присвоїти $x^{(i+1)} = x^{(i)}$;
- якщо в точці $x^{(i)}$ значення цільової функції більше, ніж у точці $x^{(i-1)}$ ($\Delta f > 0$), то згенерувати згідно з рівномірним законом розподілу випадкове число $rand$ на проміжку $[0,1]$ і порівняти його з величиною $\exp\left(-\frac{\Delta f}{k_b \cdot t_i}\right)$. Якщо згенероване число є меншим, то в якості наступної точки $x^{(i+1)}$ вибрати $x^{(i)}$, інакше залишити $x^{(i+1)} = x^{(i-1)}$.

Крок 8. Якщо нове обчислене значення цільової функції не більше за старе найкраще рішення, то запам'ятати його, тобто $x^* \approx x, f(x^*) = f(x)$.

Крок 9. Перевірити умову закінчення:

- якщо $i = N$, то присвоїти $x^* \approx x, f(x^*) = f(x)$ та перейти далі;
- якщо $i < N$, то задати $i = i + 1$ та перейти до наступного кроку.

Крок 10. Перевірити умови закінчення всього процесу:

- а) якщо виконано h відпалів, то присвоїти $f(x^*) = \min_{o=1,h}\{f(x_o)\}, x^* = x_o$

та вивести найкращий результат, зупинити роботу алгоритма;

- б) інакше перейти далі.

Крок 11. Якщо глобальний розв'язок $x^* \approx x, f(x^*) = f(x)$ не змінюється впродовж k ітерацій, а температура не більша певного фіксованого малого значення, то виконати перенагрів, піднявши температуру.

Крок 12. Зменшити температуру. Для цього виконати такі дії:

- а) обчислити

$$t_i = t_{\max} \exp(-c \cdot i^{1/n}), \quad (3.4)$$

де

$$c = m \cdot \exp\left(-\frac{k}{n}\right). \quad (3.5)$$

б) покласти $j = 1$;

в) обчислити j -ту координату вектора T :

$$T_{(j,j)} = T_{(j;\max)} \exp(-C_j \cdot i^{1/n}), \quad (3.6)$$

де

$$C_j = M_j \cdot \exp\left(-\frac{K_j}{n}\right). \quad (3.7)$$

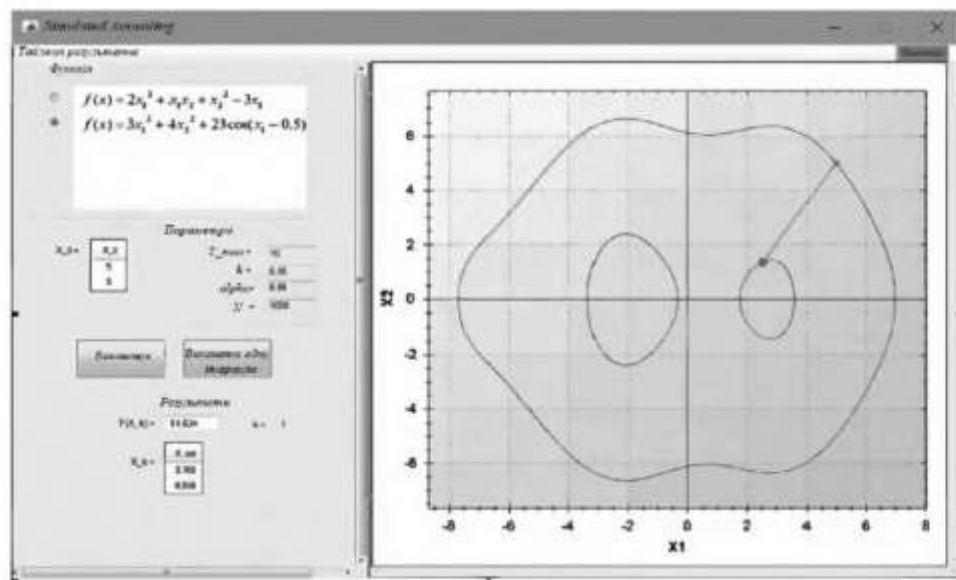
г) якщо $j < n$, то покласти $j = j + 1$ та перейти до пункту (в), інакше перейти до кроку 4.

Рекомендується розпочинати процес пошуку з високих значень параметрів T_{\max} та C (наприклад, $C = 0,9$, перед $C = 0,2$ або $C = 0,1$).

Для досягнення найкращого результату в більшості випадків (особливо при великій кількості змінних) у процесі пошуку рішення необхідно поступово зменшувати значення координат вектора T та

параметра t , так, щоб наприкінці роботи вони були близькі до нуля. Це потрібно враховувати під час підбору параметрів M, K, m, k .

Наведений алгоритм реалізовано у середовищі математичного моделювання MatLab [23]. Робота з програмою починається з вибору необхідних даних на інтерфейсі. Головне вікно програми складається з панелей вибору тестового прикладу та присвоєння значень параметрів алгоритму, початкового вектора аргументів; кнопок управління; панелей виведення результатів та формування графіку; інших опцій (рисунк 3.1).



Рисунк 3.1 – Головне вікно користувача програми

Розроблене програмне забезпечення дозволяє регулювати масштаб графіка простими у використанні способами. Поряд із підписами до осей графіка передбачено можливість визначити координати будь-якої точки на графіку, підвівши до неї курсор “миші”.

Далі вибирається вид розрахунків: кнопка “Виконати” або “Виконати одну ітерацію”. При натисканні на кнопку ”Виконати” програма проводить

розрахунок і виводить кінцеві результати: вектор аргументів x^* і значення $f(x^*)$, а також буде показано стартову і кінцеву точки (для одновимірного випадку – на графіку функції, для двовимірного випадку – на лініях рівня, які проходять через ці точки). Для обчислення по ітераціях слід натиснути на кнопку "Виконати одну ітерацію". У цьому випадку алгоритм зробить один крок і програма виведе проміжні результати. За замовчуванням, якщо процес не перейшов у нову точку, то на графіку з'являється і впродовж декількох секунд "мерехтить" відкинута точка.

3.2 Результати експериментів

Тестування роботи алгоритму з детальним дослідженням параметрів проведемо на тестових функціях [24].

1. Тестова функція $f(x) = 2x_1^2 + x_1x_2 + x_2^2 - 3x_1$. Глобальний мінімум: $x^* \approx (0,8571; -0,4285)^T$, $f(x^*) = -1,2857$. Поверхня функції показана на рисунку 3.2.

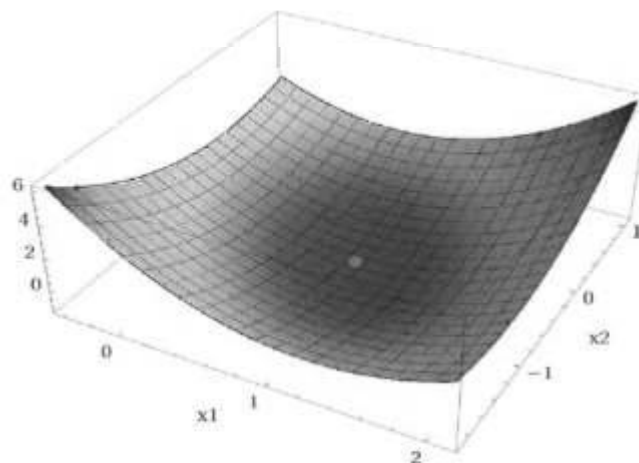


Рисунок 3.2 – Тестова функція № 1

Ілюстрація процесу пошуку показана на рисунку 3.3.

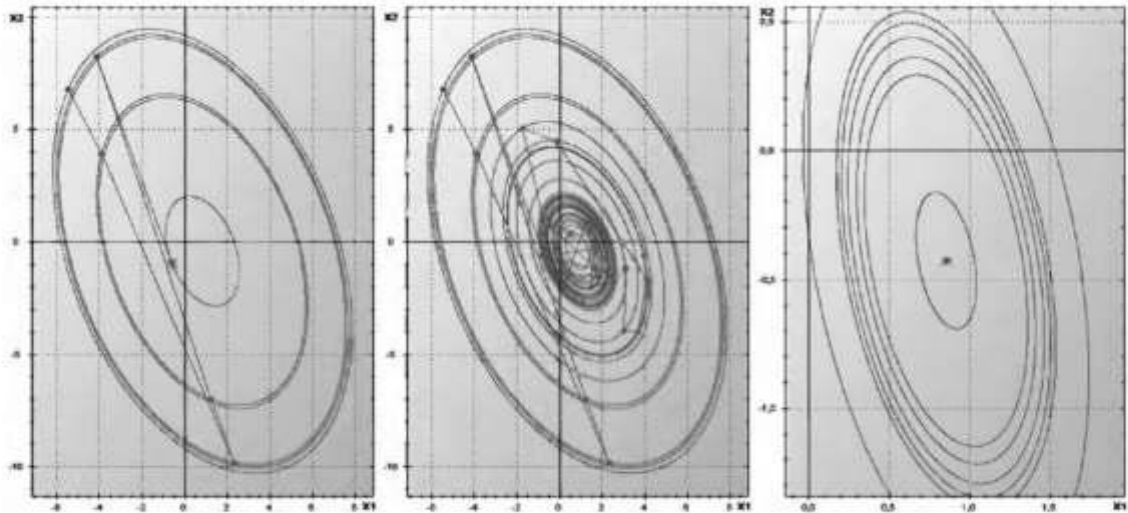


Рисунок 3.3 – Ілюстрація процесу пошуку оптимуму
на 25, 500 та 1000 ітераціях алгоритму

Чисельне рішення наведено в таблиці 3.1, при цьому показник $\alpha = rand$. Результат, отриманий у випадку 1, вимагає уточнення, оскільки задано занадто велике значення T_{max} , однак у випадку 2 це компенсується високою швидкістю зменшення параметра T .

Довжини інтервалів допустимих рішень кожної координати у випадках 4 і 5 істотно відрізняються один від одного. У таких випадках адаптивний метод імітації відпалу дозволяє ефективно проводити пошук розв'язку за рахунок масштабування параметра T і коефіцієнтів його зменшення для кожної координати.

У випадку 4 масштабування не проведено – всі параметри для кожної з координат однакові, а у випадку 5 параметри підібрані окремо, що суттєво вплинуло на покращення результату.

Таблиця 3.1 – Результати обчислень для функції 1

№	Множина допустимих рішень	Параметри алгоритму							Результати	
		N	T_{max}	M^T, K^T	t_{max}	m	k	k_b	x^{*T}	$f(x^*)$
1	$x_1 \in [-20; 20]$ $x_2 \in [-20; 20]$	500	(0,1; 0,1)	(1,1; 1,1), (1; 1)	5	1	2	0,2	(0,933; -0,348)	-1,261
2	$x_1 \in [-10^6; 10^6]$ $x_2 \in [-20; 20]$	1000	(100; 0,1)	(1; 1,1), (1; 1)	5	1	3	0,2	(1,866; -4,724)	14,867
3	$x_1 \in [-10^6; 10^6]$ $x_2 \in [-20; 20]$	1000	(100; 0,1)	(5; 1,1), (1; 1)	5	1	3	0,2	(0,848; -0,43)	-1,285
4	$x_1 \in [1; 2; 10^6]$ $x_2 \in [-20; 20]$	1000	(1; 1)	(1; 1), (1,5; 1,5)	5	1	3	0,2	(1,259; -1,351)	-0,481
5	$x_1 \in [1; 2; 10^6]$ $x_2 \in [-20; 20]$	1000	(10; 1)	(2; 1), (1,5; 1,5)	5	1	3	0,2	(1,003; -0,501)	-1,248

2. Тестова функція $f(x) = 3x_1^2 + 4x_2^2 + 23\cos(x_1 - 0,5)$. Глобальний мінімум $x^* = (-2,0709; 0)^T$, $f(x^*) = -6,4892$. Найближчий до глобального мінімуму локальний мінімум: $x = (2,81640)^T$, $f(x) = 8,1927$. Поверхня функції показана на рисунку 3.4.

Чисельне рішення наведено в таблиці 3.2, при цьому показник $\alpha = rand$. У всіх випадках, крім 5-го, множина допустимих рішень включає точку глобального мінімуму функції, тому результат пошуку прямує до нього. У випадку 5 алгоритм знайшов найближчий локальний оптимум функції.

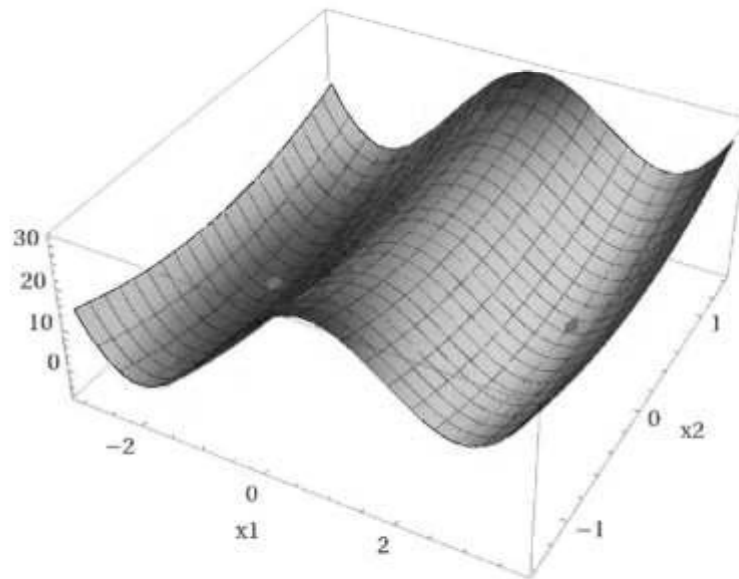


Рисунок 3.4 – Тестова функція № 2

Таблиця 3.2 – Результати обчислень для функції 2

№	Множина допустимих рішень	Параметри алгоритму							Результати	
		N	T_{max}	M^T, K^T	t_{max}	m	k	k_b	x^{*T}	$f(x^*)$
1	$x_1 \in [-4;6]$ $x_2 \in [-10^6;10^6]$	1000	(1; 1)	(1; 1), (1; 1)	5	1	2	0,25	(-1,985; -0,027)	-6,396
2	$x_1 \in [-4;6]$ $x_2 \in [-10^6;10^6]$	1000	(0,1; 5)	(1; 2), (1; 1)	5	1	2	0,3	(-2,084; -0,011)	-6,486
3	$x_1 \in [-10^7;0]$ $x_2 \in [-10^6;10^6]$	1000	(120; 100)	(1,2; 1,2), (0,1; 0,1)	5	1	3	0,2	(-2,103; -0,284)	-6,151
4	$x_1 \in [-10;10]$ $x_2 \in [-10;10]$	1000	(1; 1)	(1,1; 1,1), (1,1; 1,1)	5	1	3	0,2	(-2,065; -0,011)	-6,488
5	$x_1 \in [0;6]$ $x_2 \in [-10^6;10^6]$	1000	(5; 5)	(1,1; 1,1), (0,1; 0,1)	5	1	3	0,2	(2,832; -0,003)	8,195

3. Розглянемо інженерну задачу оцінювання параметрів для частотно-модульованих звукових хвиль (*parameter estimation for frequency-modulated (FM) sound waves*) у сучасних музичних системах. FM-синтез із використанням двох операторів показаний на рисунку 3.5, на якому несучий тон f_c (220 Гц) модулюється за допомогою модулюючого тону f_m (440 Гц) з індексом нечастотної модуляції β [25].

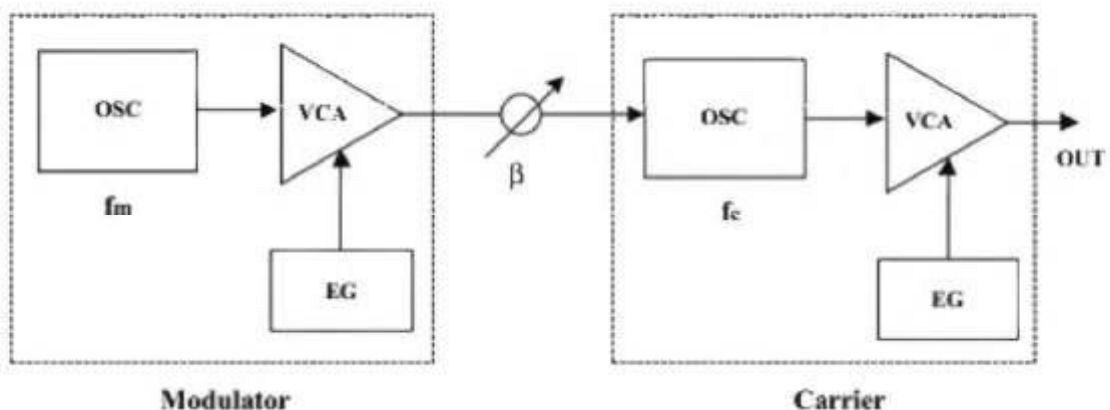


Рисунок 3.5 – Схема для FM-синтезу:

OSC: осцилятор; VCA: підсилювач з керуванням напругою;

EG: генератор огинаючої

Математично задача полягає у генерації певного звуку $y(t)$ на відрізок часу t , схожого на цільовий звук $y_0(t)$, що можна звести до такого вигляду:

$$f = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \rightarrow \min, \quad (3.8)$$

$$y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta))), \quad (3.9)$$

$$y_0(t) = 1,0 \cdot \sin(5,0 \cdot t \cdot \theta - 1,5 \cdot \sin(4,8 \cdot t \cdot \theta + 2,0 \cdot \sin(4,9 \cdot t \cdot \theta))), \quad (3.10)$$

$$\theta = \frac{2 \cdot \pi}{100}, -6,4 \leq \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\} \leq 6,35, f^* = 0. \quad (3.11)$$

Дана задача має величезну кількість глобальних екстремумів.

Порівняння результатів роботи різних алгоритмів показано в таблиці 3.3. Для тестування застосовано такі параметри алгоритму:

- $q = 30$ незалежних запусків;
- розмірність простору пошуку $n = 6$;
- межі пошуку $LB = -6.4$; $UB = 6.35$;
- температура за координатами $T_{\max} = (1.2, 1.2, 1.2, 1.2, 1.2, 1.2)$;
- параметри зменшення $M = (1, 1, 1, 1, 1)^T$ та $K = (1.5, 1.5, 1.5, 1.5, 1.5, 1.5)^T$;
- глобальна температура $t_{\max} = 5$;
- параметри зменшення $m = 1$ та $k = 3$;
- константа Больцмана $k_b = 0,25$;
- кількість ітерацій локальної оптимізації $l = 6$;
- коефіцієнт стиснення $\alpha = rand$;
- кроків до перенагріву $steps = 500$;
- коефіцієнт нагріву $a = 2,75$;
- температура перенагріву $T_{re-heat} = 0,25$;
- кількість ітерацій $N = 6500$;
- якщо 1000 ітерацій найкраще значення не змінюється, тоді

виконати повторний відпал.

У таблиці 3.3 визначено найкраще f_{best} та найгірше f_{worst} знайдені значення, а також середнє значення $mean$ за q запусків програми. Отже, можна зробити висновок, що при правильному виборі параметрів алгоритм імітації відпалу дозволяє досягти високої точності у ході обчислення

глобального мінімуму функції, але у деяких випадках для цього необхідно виконати велику кількість ітерацій та затратити досить багато часу. Особливо чутливим алгоритм є до температурного бар'єру $T_{re-heat}$, кількості кроків $steps$ та коефіцієнта нагріву α , які можна вибрати експериментально.

Таблиця 3.3 – Порівняння результатів роботи метаевристик

Автор	Алгоритм	f_{best}	f_{worst}	$mean$
Рашеді, Саріазді та інші [26]	Алгоритм гравітаційного пошуку	12,600	26,000	16,300
Мааді, Рамезані та інші [27]	Модифікований алгоритм оптимізації на основі поведінки зозуль	0,675	15,00	7,540
Квін, Жанг та інші [28]	Алгоритм рою часток з внутрішньо ройовою стратегією навчання	0,000	24,000	12,800
Кіркпатрік, Желатт та інші [13]	Алгоритм імітації відпалу	19,634	26,000	20,695
Дана робота	Удосконалений адаптивний алгоритм імітації відпалу	$2,490 \cdot 10^{-15}$	14,100	5,260

3.3 Висновки

У даному розділі описано принципи роботи розробленого програмного забезпечення. Програма дозволяє керувати параметрами удосконаленого алгоритму імітації відпалу й аналізувати ефективність його роботи. Визначено, що загалом алгоритм стабільний і збігається до оптимуму, якщо вдало вибрати ключові параметри. Крім того його можна використовувати для розв'язання різноманітних задач оптимізації, як неперервних, так і дискретних, практично при повній відсутності припущень про характер цільової функції.

4 РОЗДІЛ ЕКОНОМІКИ

4.1 Технологічний аудит удосконаленого алгоритму імітації відпалу

Метою проведення комерційного і технологічного аудиту дослідження за темою «Удосконалений адаптивний алгоритм імітації відпалу» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями (таблиці 4.1) [29].

Таблиця 4.1 – Критерії оцінювання рівня комерційного потенціалу розробки та їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти на навчання	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні дорогі та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій > 10-ти рр.	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій > 5-ти рр.	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти рр.	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Для проведення технологічного аудиту залучено 3 незалежних експертів:

– кандидат технічних наук, доцент Кабачій В.В. Викладач дисциплін, які пов'язані з математичним моделюванням та методами оптимізації;

– кандидат технічних наук, доцент Богач І.В. Займається розробкою комп'ютеризованих систем автоматички, інтелектуальних інформаційних технологій;

– кандидат технічних наук, професор Папінов В.М. Займається дослідженням та автоматизованим проектуванням комп'ютерних систем управління.

За результатами оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки, наведених у таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [30]. Згідно проведених досліджень рівень комерційного потенціалу розробки становить 38,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Експерт		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	4	4
2. Ринкові переваги (наявність аналогів)	1	2	1
3. Ринкові переваги (ціна продукту)	2	2	2
4. Ринкові переваги (технічні властивості)	2	2	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	3	2	3
7. Ринкові перспективи (конкуренція)	3	2	3
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	4	4	3
10. Практична здійсненність (необхідність матеріалів)	5	5	5
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	5	5	5
Сума балів	40	39	40
Середньоарифметична сума балів CB_c	39,7		

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень наукового та комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

4.2 Розрахунок витрат на проведення досліджень

При виконанні магістерської кваліфікаційної роботи було зроблено такі витрати [30]:

1. Витрати на основну заробітну плату дослідників розраховуємо у відповідності до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_m \cdot t_i}{T_p} = 14250,00 \cdot 32 / 22 = 20727,27 \text{ грн.}, \quad (4.1)$$

де k – кількість посад дослідників залучених до процесу досліджень; M_m – місячний посадовий оклад конкретного дослідника, грн; t_i – кількість днів роботи конкретного дослідника, дні; T_p – середня кількість робочих днів в місяці, $T_p = 24$ дні.

Проведені розрахунки зведемо до таблиці 4.4.

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн.	Кількість днів роботи	Витрати на заробітну плату, грн.
Науковий керівник НДР	14250,00	647,73	32	20727,27
Інженер-програміст	14150,00	643,18	32	20581,82
Провідний фахівець з математичного моделювання	14000,00	636,36	9	5727,27
Технік 1-ї категорії	7065,00	321,14	16	5138,18
Всього				52174,55

2. Витрати на основну заробітну плату робітників за відповідними найменуваннями робіт НДР розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн./год.; t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.3)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати, прийmemo $M_M = 6700,00$ грн.; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середнє число робочих днів в місяці, приблизно $T_p = 24$ дні; $t_{зм}$ – тривалість зміни, год.

Результати запишемо в таблицю 4.5. Тоді маємо:

$$C_i = 6700,00 \cdot 1,10 \cdot 1,65 / (22 \cdot 8) = 69,09 \text{ грн.}$$

$$Z_{pi} = 69,09 \cdot 7,20 = 497,48 \text{ грн.}$$

3. Додаткову заробітну плату розраховуємо як 10...12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%} = (52174,55 + 5506,77) \cdot 12 / 100\% = 6921,76 \text{ грн.}, \quad (4.4)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати; прийmemo 12%.

Таблиця 4.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год.	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн.	Величина оплати на робітника грн.
Підготовка робочого місця розробника	7,20	2	1,10	69,09	497,48
Інсталяція програмного забезпечення математичного моделювання	5,40	3	1,35	84,80	457,90
Введення програмних блоків на основі адаптивного алгоритму	11,50	4	1,50	94,22	1083,52
Налагодження програмних блоків математичної моделі за алгоритмом імітації відпалу	3,80	5	1,70	106,78	405,77
Формування бази даних дослідження моделі	25,00	3	1,35	84,80	2119,92
Тестування ПЗ	10,00	4	1,50	94,22	942,19
Всього					5506,77

4. Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%} = (52174,55 + 5506,77 + 6921,76) \cdot \frac{22}{100} = 14212,68 \text{ грн.}, \quad (4.5)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату; приймаємо 22%.

5. Витрати на матеріали розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej} = 4,0 \cdot 286,00 \cdot 1,11 - 0 \cdot 0 = 1269,84 \text{ грн.} \quad (4.6)$$

де H_j – норма витрат матеріалу j -го найменування, кг; n – кількість видів матеріалів; C_j – вартість матеріалу j -го найменування, грн./кг; K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$); B_j – маса відходів j -го найменування, кг; C_{ej} – вартість відходів j -го найменування, грн./кг.

Проведені розрахунки зведемо до таблиці 4.6.

6. Витрати на комплектуючі, які використовують при проведенні НДР відсутні.

7. Витрати на спецустаткування, яке додатково закуповується при проведенні НДР відсутні.

8. Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прз}} = \sum_{i=1}^k C_{\text{инпрз}} \cdot C_{\text{прз.д}} \cdot K_i = 9860,00 \cdot 1 \cdot 1,11 = 10944,60 \text{ грн.}, \quad (4.7)$$

де $C_{\text{инпрз}}$ – ціна придбання одиниці програмного засобу даного виду,

грн.; $C_{пра.д}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, $K_i = 1,10 \dots 1,12$; k – кількість найменувань програмних засобів.

Отримані результати зведемо до таблиці 4.7.

Таблиця 4.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн.	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн./кг	Вартість витраченого матеріалу, грн.
Папір офісний А4 500	286,00	4,0	0	0	1269,84
Папір для записів А4 250	156,00	6,0	0	0	1038,96
Органайзер офісний	168,00	3,0	0	0	559,44
Канцелярське приладдя	220,00	4,0	0	0	976,80
Картридж для принтера	1010,00	2,0	0	0	2242,20
Диск оптичний CD-R	23,00	4,0	0	0	102,12
Flesh-пам'ять 64 GB	575,00	1,0	0	0	638,25
Всього					6827,61

9. У спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо,

розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_а} \cdot \frac{t_{вик}}{12} = (8420,00 \cdot 2) / (2 \cdot 12) = 701,67 \text{ грн.}, \quad (4.8)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн.; $t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців; $T_а$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт.	Ціна за одиницю, грн.	Вартість, грн.
Математичне середовище MatLab	1	9860,00	10944,60
Прикладне ПЗ Mathematica	1	7580,00	8413,80
Середовище програмування Gekko	1	6985,00	7753,35
Всього			27111,75

10. Витрати на силову електроенергію розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вт}}{\eta_i} = 0,25 \cdot 240,0 \cdot 6,20 \cdot \frac{0,95}{0,97} = 372,00 \text{ грн.}, \quad (4.9)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт; t_i – тривалість роботи обладнання на етапі дослідження,

год.; C_e – вартість 1 кВт-години електроенергії, грн.; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 6,20$ грн.; K_{enl} – коефіцієнт, що враховує використання потужності, $K_{enl} < 1$; η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
ОС Windows	8420,00	2	2	701,67
Microsoft Office	7350,00	5	2	245,00
Персональний комп'ютер	22600,00	4	2	941,67
Комп'ютерне обладнання	45820,00	5	2	1527,33
Робоче місце розробника ПЗ	11000,00	5	2	366,67
Пристрої передачі даних	8320,00	3	2	462,22
Оргтехніка	10325,00	5	2	344,17
Приміщення лабораторії	294000,00	25	2	1960,00
Всього				6548,72

Таблиця 4.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год.	Сума, грн.
Персональний комп'ютер розробника ПЗ	0,25	240,0	372,00
Комп'ютерне обладнання	0,41	200,0	508,40
Робоче місце розробника	0,16	240,0	238,08
Пристрої передачі даних	0,06	12,0	4,46
Оргтехніка	0,56	22,0	76,38
Всього			1199,33

11. Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%} = (52174,55 + 5506,77) \cdot 25 / 100 = 14420,33 \text{ грн.}, \quad (4.10)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 25\%$.

12. Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%} = (52174,55 + 5506,77) \cdot 40 / 100 = 23072,53 \text{ грн.}, \quad (4.11)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 40\%$.

13. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%} = (52174,55 + 5506,77) \cdot 100 / 100 = 57681,32 \text{ грн.}, \quad (4.12)$$

де H_{ie} – норма нарахування за статтею «Інші витрати», $H_{ie} = 100\%$.

14. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{исв} = (Z_o + Z_p) \cdot \frac{H_{исв}}{100\%} = (52174,55 + 5506,77) \cdot 100 / 100 = 57681,32 \text{ грн.}, \quad (4.13)$$

де $H_{исв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{исв} = 100\%$.

15. Витрати на проведення науково-дослідної роботи розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{вс}} = Z_o + Z_p + Z_{\text{од}} + Z_n + M + K_v + B_{\text{скл}} + B_{\text{исв}} + A_{\text{скл}} + B_e + B_{\text{от}} + B_{\text{от}} + I_e + B_{\text{исв}}, \quad (4.14)$$

$$B_{\text{вс}} = 52174,55 + 5506,77 + 6921,76 + 14212,67659 + 6827,61 + 0,00 + 0,00 + \\ + 27111,75 + 6548,72 + 1199,33 + 14420,33 + 23072,53 + 57681,32 + \\ + 57681,32 = 273358,65 \text{ грн.}$$

16. Загальні витрати на завершення науково-дослідної роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{\text{вс}}}{\eta} = 273358,65 / 0,85 = 321598,42 \text{ грн.}, \quad (4.15)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,85$.

4.3 Розрахунок ефекту для потенційного інвестора від можливої комерціалізації розробки

У ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Результати дослідження проведені передбачають комерціалізацію впродовж 4-ох років реалізації на ринку. У цьому випадку майбутній економічний ефект буде формуватися на основі:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик (1-й рік – +60 споживачів; 2-й рік – +80 споживачів; 3-й – +100 споживачів; 4-й – +65 споживачів);

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 300 осіб;

$Ц_o$ – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 80000,00 грн;

$\pm\Delta Ц_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 6040,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi_i$ для кожного із 4-х років, впродовж яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [30]:

$$\Delta\Pi_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{g}{100}\right), \quad (4.16)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість; у 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda=0,8333$; ρ – коефіцієнт, який враховує рентабельність

інноваційного продукту; прийmemo $\rho = 40\%$; \mathcal{G} – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році $\mathcal{G} = 18\%$.

Збільшення чистого прибутку 1-го року:

$$\begin{aligned}\Delta\Pi_1 &= (6040,00 \cdot 300,00 + 86040,00 \cdot 60) \cdot 0,8333 \cdot 0,4 \cdot (1 - 0,18) = \\ &= 1898710,66 \text{ грн.}\end{aligned}$$

Збільшення чистого прибутку 2-го року:

$$\begin{aligned}\Delta\Pi_2 &= (6040,00 \cdot 300,00 + 86040,00 \cdot 140) \cdot 0,8333 \cdot 0,4 \cdot (1 - 0,18) = \\ &= 3772593,02 \text{ грн.}\end{aligned}$$

Збільшення чистого прибутку 3-го року:

$$\begin{aligned}\Delta\Pi_3 &= (6040,00 \cdot 300,00 + 86040,00 \cdot 240) \cdot 0,8333 \cdot 0,4 \cdot (1 - 0,18) = \\ &= 6114945,98 \text{ грн.}\end{aligned}$$

Збільшення чистого прибутку 4-го року:

$$\begin{aligned}\Delta\Pi_4 &= (6040,00 \cdot 300,00 + 86040,00 \cdot 305) \cdot 0,8333 \cdot 0,4 \cdot (1 - 0,18) = \\ &= 7637475,41 \text{ грн.}\end{aligned}$$

Приведена вартість збільшення всіх чистих прибутків, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (4.17)$$

де $\Delta\Pi_t$ – збільшення чистого прибутку у кожному з років, впродовж яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, впродовж якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки; τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,14$; t – період часу (в роках) від моменту початку впровадження розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Тоді маємо:

$$\begin{aligned} \text{ПП} &= 1898710,66 / (1 + 0,14)^1 + 3772593,02 / (1 + 0,14)^2 + \\ &+ 6114945,98 / (1 + 0,14)^3 + 7637475,41 / (1 + 0,14)^4 = \\ &= 1665535,66 + 2902887,83 + 4127414,36 + 4521998,56 = 13217836,41 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.18)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 3,2$; $3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 321598,42 грн.

Тоді:

$$PV = k_{инв} \cdot 3B = 3,2 \cdot 321598,42 = 1029114,93 \text{ грн.}$$

Абсолютний економічний ефект для потенційного інвестора від можливого впровадження та комерціалізації розробки становитиме:

$$E_{\text{обс}} = \text{ПП} - \text{PV} = 13217836,41 - 1029114,93 = 12188721,48 \text{ грн.} \quad (4.19)$$

Внутрішня економічна дохідність інвестицій, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки ($T_{\text{жс}} = 4$ роки):

$$E_e = \sqrt[T_{\text{жс}}]{1 + \frac{E_{\text{обс}}}{\text{PV}}} - 1 = (1 + 12188721,48 / 1029114,93)^{1/4} = 0,89. \quad (4.20)$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій:

$$\tau_{\text{min}} = d + f, \quad (4.21)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = 0,10$; f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,45.

Тоді:

$$\tau_{\text{min}} = 0,10 + 0,45 = 0,55 < E_e = 0,89,$$

тобто інвестувати в науково-дослідну роботу доцільно.

Період окупності інвестицій, які можуть бути вкладені у впровадження та комерціалізацію науково-технічної розробки:

$$T_{\text{ок}} = \frac{1}{E_e} = 1 / 0,89 = 1,12 \text{ років.} \quad (4.22)$$

Оскільки $T_{\text{ок}} < 3$ -ох років, що свідчить про комерційну привабливість науково-технічної розробки, то це може спонукати потенційного інвестора профінансувати впровадження та виведення її на ринок.

4.4 Висновки

Аналіз комерційного потенціалу розробки показав, що вона за своїми характеристиками випереджає аналоги, що підтверджує її перспективність. Результати економічної частини магістерської кваліфікаційної роботи зведено у таблицю 4.10. Отже, основні техніко-економічні показники удосконаленого алгоритму, визначені у технічному завданні, виконані.

Таблиця 4.10 – Результати економічної частини

Показники	Задані у технічному завданні	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку удосконаленого алгоритму	не більше 350 000 грн.	321 598,42 грн.	Виконано
2. Абсолютний ефект від впровадження	не менше 14 000 грн. за копію	15 235,90 грн. за копію	Виконано
3. Внутрішня дохідність інвестицій	не менше 50 %	55 %	Виконано
4. Термін окупності потенційних інвестицій	до 3-ох років	1,12 років	Виконано

ВИСНОВКИ

Визначено, якщо задачу не можна розв'язати детермінованими методами оптимізації, застосовують гібридні ітеративні схеми – метаевристики. Однією з найбільш ефективних та зручних метаевристик є алгоритм імітації відпалу, інспірований фізичним процесом кристалізації речовини. Теоретична основа та математична модель від глауберівської динаміки до класичного та адаптивного алгоритмів імітації відпалу з різними схемами охолодження розглянута у другому розділі роботи. Запропонована модель з використанням удосконаленого рівняння для вибору рішення-кандидата та адаптивним температурним режимом.

Описано особливості розробленого програмного забезпечення, яке дозволяє керувати параметрами алгоритму імітації відпалу та візуалізувати результати на графіках. Тестування ефективності алгоритму проведено на низці задач. Проведено порівняння результатів роботи з іншими метаевристиками. Визначено, що при вдалому виборі параметрів, алгоритм збігається до оптимуму, не потребуючи громіздких обчислень, але не може гарантувати знаходження глобального екстремуму. Отже, запропонований алгоритм можна використовувати для розв'язання задач оптимізації, завдяки простоті, гнучкості та ефективності.

Також у роботі проведено технологічний аудит запропонованого алгоритму. Аналіз комерційного потенціалу програмного забезпечення показав, що його розробка є перспективною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Henderson H. Encyclopedia of Computer Science and Technology. New-York, 2003. 593 p.
2. Osaba E., Molina D., Yang X.-S. and others. Bio-Inspired Computation: Where We Stand and What's Next. *Swarm and Evolutionary Computation*. 2019. Vol. 48. P. 220-250.
3. Almufti S.M. Historical Survey on Metaheuristics Algorithms. *International Journal of Scientific World*. 2019. Vol. 7 (1). P. 1-12.
4. Грибков С.В. Інформаційні технології прийняття рішень на основі модифікованих евристичних і еволюційних методів та алгоритмів для харчових підприємств: дис. ... д-ра техн. наук: 05.13.06. Київ, 2021. 395 с.
5. Delahaye D., Chaimatanan S., Mongeau M. Simulated Annealing: From Basics to Applications. *International Series in Operations Research & Management Science*. 2019. 43 p. URL: <https://hal-enac.archives-ouvertes.fr/hal-01887543> (дата звернення: 22.08.2022).
6. Suarez J., Millan C., Millan E. Improved Modified Simulated Annealing Algorithm for Global Optimization. *Contemporary Engineering Sciences*. 2018. Vol. 11 (96). P. 4789-4795.
7. Боднаренко Б.О., Іванов Ю.Ю., Яровий І.К. та інші. Застосування стохастичних алгоритмів оптимізації з декомпозицією простору пошуку *Актуальні задачі медичної, біологічної фізики та інформатики: матер. науково-практичної конференції з міжнародною участю*. Вінниця: ВНМУ ім. М.І. Пирогова, 2022. С. 72-73.
8. Бартіш М.Я., Дудзяний І.М. Дослідження операцій: підручник. Львів, 2007. 168 с.

9. Жалдак М.І., Триус Ю.В. Основи теорії і методів оптимізації. Черкаси: Брама-Україна, 2005. 608 с.
10. Гуляницький Л.Ф., Мулеса О.Ю. Прикладні методи комбінаторної оптимізації. К.: Видавничо-поліграфічний центр “Київський університет”, 2016. 142 с.
11. Азенкотт Р. Процедура “отпуска”. *Труды семинара Н. Бурбаки*. 1990. С. 235-251.
12. Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*. 1953. № 21. P. 1087-1092. URL: <https://bayes.wustl.edu/Manual/EquationOfState.pdf> (дата звернення 10.09.2022).
13. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by Simulated Annealing. *Science*. 1983. P. 671-680. URL: <http://www2.stat.duke.edu/~scs/Courses/Stat376/Papers/TemperAnneal/KirkpatrickAnnealScience1983.pdf> (дата звернення 10.09.2022).
14. Cerny V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory Applications*. 1985. № 45. P. 41-51.
15. Franzin A., Stutzle T. Revisiting Simulated Annealing: A Component-Based Analysis. Technical Report Series. TR/IRIDIA/2018-010. 2018. 35 p. URL: <https://iridia.ulb.ac.be/IridiaTrSeries/rev/IridiaTr2018-010r001.pdf> (дата звернення 25.08.2022).
16. Peprah A.K., Appiah S.K., Amponsah S.K. An Optimal Cooling Schedule Using a Simulated Annealing Based Approach. *Applied Mathematics*. 2017. № 8. P. 1195-1210.

17. Szu H.H., Hartley R.L. Fast Simulated Annealing. *Physical Letters*. 1987. №122. P. 157-162. URL: https://www.researchgate.net/publication/234014515_Fast_simulated_annealing (дата звернення 10.09.2022).
18. Ingber L. Very Fast Simulated Re-annealing. *Mathematical and Computer Modelling*. 1989. № 112. P. 967-973. URL: https://www.researchgate.net/publication/2836339_Very_Fast_Simulated_Re-Annealing (дата звернення 25.08.2022).
19. Ingber L. Adaptive Simulated Annealing (ASA): Lessons Learned. *Journal of Control and Cybernetics*. 1996. № 25. P. 33-54. URL: <https://arxiv.org/pdf/cs/0001018.pdf> (дата звернення: 25.08.2022).
20. Ingber L. Simulated Annealing: Practice Versus Theory. *Mathematical and Computer Modelling*. 1993. Vol. 18 (11). P. 29-57.
21. Yao X. A New Simulated Annealing Algorithm. *International Journal of Computer Mathematics*. 1995. Vol. 56. P. 161-168.
22. Ingber L., Rosen B. Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison. *Mathematical and Computer Modeling*. 1992. Vol. 16(11). P. 87-100.
23. MATLAB Programming Fundamentals. URL: https://ayouty.com/4AI/MATLAB/MATLAB_PROG.pdf. (дата звернення 28.08.2022).
24. Кісельова О.М., Шевельова А.Є. Чисельні методи оптимізації. Дніпро: Видавництво ДНУ, 2008. 208 с.
25. Das S., Suganthan P.N. Problem Definitions and Evaluation Criteria for CEC Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Technical Report, 2010. 42 p.

26. Rashedi E., Nezamabadi-pour H., Saryazdi S. GSA: A Gravitational Search Algorithm. *Information Science*. 2009. Vol. 179 (13). P. 2232-2248.
27. Maadi M., Javidnia M., Ramezani R. Modified Cuckoo Optimization Algorithm to Solve Precedence Constrained Sequencing Problem. *Applications of Artificial Intelligence*. 2018. Vol. 48 (6). P. 1407-1422.
28. Qin Q., Cheng S., Zhang Q. and others. Particle Swarm Optimization With Interswarm Interactive Learning Strategy. *IEEE Transactions in Cybernetics*. 2016. Vol. 46(10). P. 2238-2251.
29. Кавецький В.В., Козловський В.О., Причепя І.В. Економічне обґрунтування інноваційних рішень. Вінниця: ВНТУ, 2016. 113 с.
30. Козловський В.О., Лесько О.Й., Кавецький В.В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця: ВНТУ, 2021. 42 с.

ДОДАТКИ

Додаток А
(обов'язковий)
Технічне завдання

ЗАТВЕРДЖУЮ
в. о. завідувача кафедри АІТ
д.т.н., проф. Бісікало О. В.
«__» _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«Удосконалений адаптивний алгоритм імітації відпалу»
08-31.МКР.001.02.000 ТЗ

Керівник роботи:
к.т.н., доц. каф. АІТ
Іванов Ю. Ю.
«__» _____ 2022 р.

Виконавець:
ст. гр. ІСТ-21м
Боднаренко Б. О.
«__» _____ 2022 р.

1. Назва та галузь застосування

Робота спрямована на дослідження удосконаленого адаптивного алгоритму імітації відпалу, який можна використовувати для розв'язання різноманітних задач практично при повній відсутності припущень про характер цільової функції.

2. Підстави для розробки

Розробку системи здійснювати на підставі наказу по університету №203 від 14.09.2022 та завдання до магістерської кваліфікаційної роботи, складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій».

3. Мета та призначення розробки

Метою роботи є підвищення ефективності розв'язання задачі неперервної оптимізації функції від багатьох змінних з використанням алгоритму імітації відпалу за рахунок використання модифікованої генерації рішення-кандидата та адаптивного температурного режиму.

4. Джерела розробки

1. Ingber L. Adaptive Simulated Annealing (ASA). *Journal of Control and Cybernetics*. 1996. № 25. P. 33-54.

2. Praph A.K., Appiah S.K., Amponsah S.K. An Optimal Cooling Schedule Using a Simulated Annealing Based Approach. *Applied Mathematics*. 2017, № 8. P. 1195-1210.

3. Suarez J., Millan C., Millan E. Improved Modified Simulated Annealing Algorithm for Global Optimization. *Contemporary Engineering Sciences*. 2018, Vol. 11 (96). P. 4789-4795.

4. Delahaye D., Chaimatanan S., Mongeau M. Simulated Annealing: From Basics to Applications. *International Series in Operations Research & Management Science*. 2019. 43 p.

5. Показники призначення

Розроблено програмне забезпечення, яке дозволяє провести експериментальні дослідження алгоритму імітації відпалу у ході розв'язання задач оптимізації функцій.

Основні технічні вимоги та мінімальні системні вимоги до програми: операційна система *Windows 7, 8 або 10*; підключення клавіатури та миші; кількість ядер комп'ютера – не менше 4 (*Core i5 або Ryzen 5*); оперативна пам'ять – не менше 8 ГБ; відеопам'ять – 512 Mb; жорсткий диск – 500 ГБ і більше; встановлене програмне забезпечення *MatLab*.

Вхідні дані: розмірність простору пошуку (n); стартова температура T ; температурний режим; закон генерації точки; параметр Больцмана k_b ; максимальна кількість ітерацій N ; набір тестових функцій.

Результати роботи програми: значення екстремуму x^* функції y .

6. Економічні показники

До економічних показників входять:

- витрати на розробку – не більше 350 тис. грн.;
- вартість копії продукту – не менше 14 тис. грн.;
- мінімальна дохідність – 50 %;
- термін окупності – не менше 3-ох років;
- інші економічні переваги у порівнянні з аналогами.

7. Стадії розробки

1. Розділ 1 «Аналіз предметної області» має бути виконаний до 10.10.22.

2. Розділ 2 «Розробка математичної моделі удосконаленого адаптивного алгоритму імітації відпалу» має бути виконаний до 25.10.22.

3. Розділ 3 «Розробка програмного забезпечення та експериментальні дослідження» має бути виконаний до 10.11.22.

4. Економічний розділ має бути виконаний до 20.11.22.

8. Порядок контролю та приймання

1. Рубіжний контроль провести до 07.12.22.
2. Попередній захист магістерської кваліфікаційної роботи провести до 07.12.22.
3. Захист магістерської кваліфікаційної роботи провести до 23.12.22.

Додаток Б
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

УДОСКОНАЛЕНИЙ АДАПТИВНИЙ АЛГОРИТМ ІМІТАЦІЇ ВІДПАЛУ

В.о. зав. кафедри АПТ _____ д-р техн. наук, професор каф. АПТ
Бісікало О. В.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Керівник роботи _____ канд. техн. наук, доцент каф. АПТ
Іванов Ю. Ю.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Тех. контроль _____ канд. техн. наук, доцент каф. АПТ
Іванов Ю. Ю.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Нормоконтроль _____ канд. техн. наук, доцент каф. АПТ
Іванов Ю. Ю.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Опонент _____

(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Студент гр. ІСТ-21м _____ Боднарєнко Б. О.
(підпис) (ініціали та прізвище)

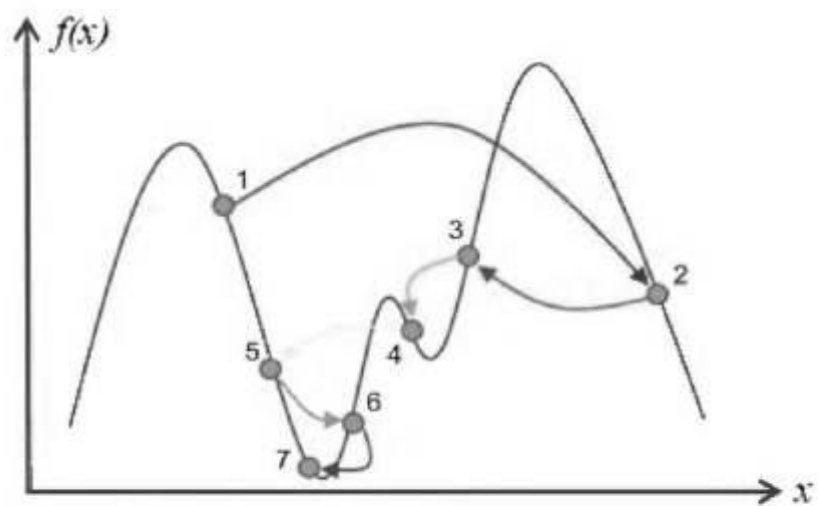


Рисунок Б.1 – Принцип роботи алгоритму імітації відпалу



Рисунок Б.2 – Геометрична інтерпретація переходів

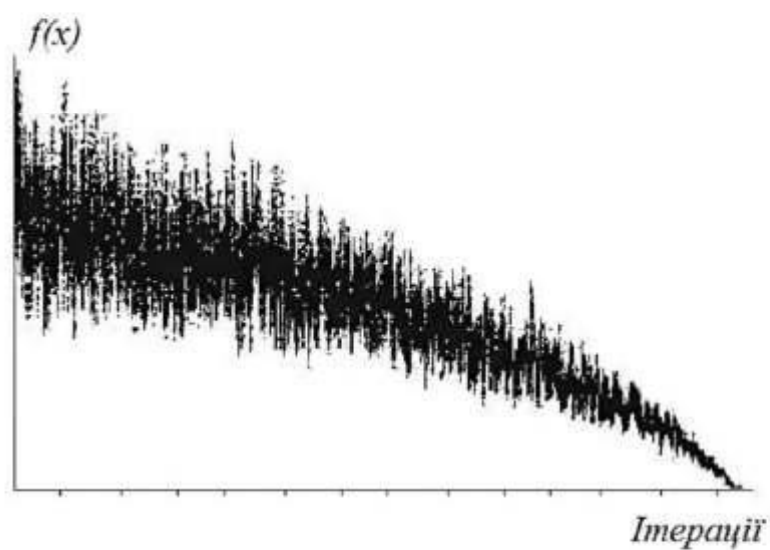


Рисунок Б.3 – Ілюстрація пошуку оптимуму



Рисунок Б.4 – Схема програми

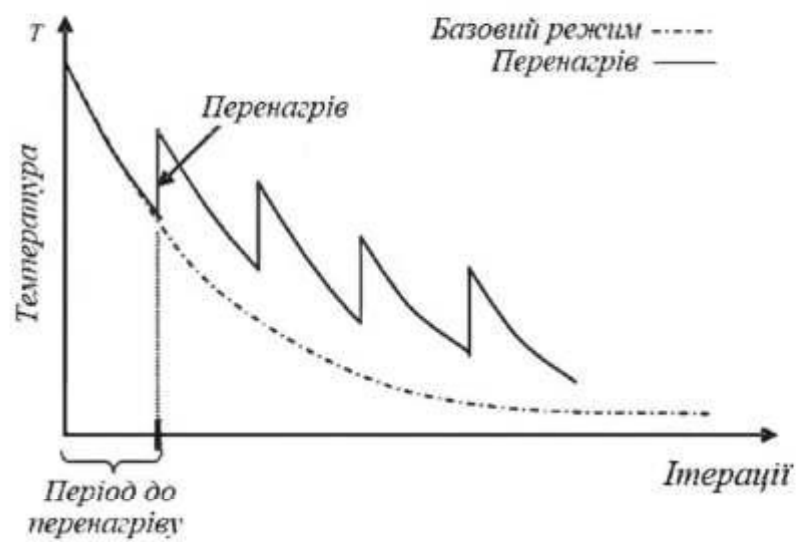


Рисунок Б.5 – Ілюстрація перенагріву

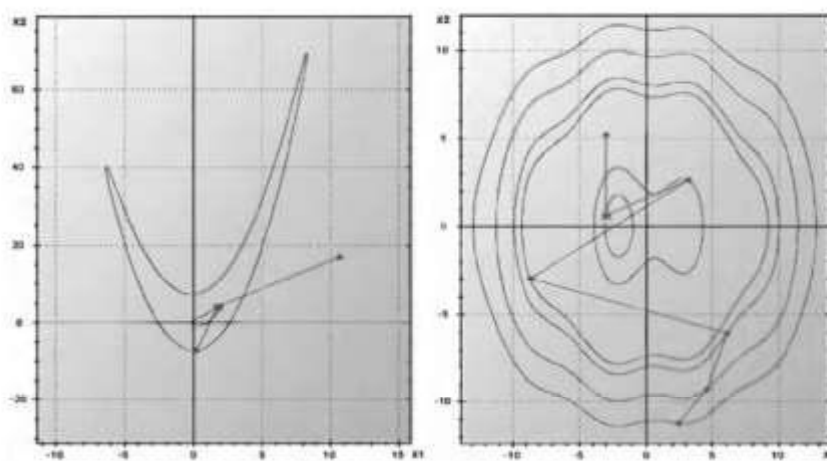


Рисунок Б.6 – Приклад роботи програми

Додаток В

(обов'язковий)

Лістинг головної функції програми

```
%%%%%%%%%%  
function [best_solution, solution_point, mas_best] = SA(func)  
  
% func = 'x.^2+y^2';  
n = 2;  
current_point = [0, 0];  
low = [-500, -500];  
up = [500, 500];  
Tmax = [10, 10];  
tmax = 5;  
M = [1, 1];  
K = [3, 3];  
m = 1;  
k = 3;  
alpha = rand;  
limit = 10;  
k = 800;  
a = 2.75;  
TR = 0.25;  
stop = 1;  
exit = 10000;  
  
f = vectorize(inline(func));  
[x, y] = meshgrid(low(1):0.3:up(1), low(2):0.3:up(2));  
z = f(x, y);  
solution_point = current_point;  
current_solution = our_function(current_point, value);  
mas_current(stop) = current_solution;  
best_solution = current_solution;  
T = Tmax;  
  
rng('shuffle');  
  
while (stop <= exit)  
  
    for test_points = 1:limit  
        for i = 1:length(current_point)  
            a = rand;  
            dx(i) = sign(a - 0.5) * T * ((1 + 1/T)^abs(2*a - 1) - 1);  
        end  
    end  
end
```



```

test_point = current_point + dx;
test_point = (test_point < low).*low +...
              (low <= test_point).*(test_point <= up).*test_point...
              + (up < test_point).*up;
test_solution = our_function(test_point, value);
delta_errors = test_solution - current_solution;

if (delta_errors <= 0) || (rand < exp(-delta_errors/T))
    current_point = test_point;
    current_solution = test_solution;
end
end

mas_current(stop+1) = current_solution;
mas_temperature(stop) = T;
if current_solution <= best_solution
    solution_point = current_point;
    best_solution = current_solution;
end

mas_best(stop) = best_solution;
T = alpha * T;
fprintf('\n %.1d. Best = %.6f, New = %.6f, Temp = %.6f,
Process work = %.3f %%%', stop, best_solution, current_solution, T,
100*(stop/(hard*exit)));
stop = stop + 1;
end
%%

```

Додаток Г
(обов'язковий)

Довідка про можливість впровадження

Додаток Д
(обов'язковий)
Протокол перевірки МКР

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Удосконалений адаптивний алгоритм імітації відпалу.

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації

Показники звіту подібності Plagiat.pl (StrikePlagiarism)

Оригінальність 93,8% Схожість 6,2%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень

Особа, відповідальна за перевірку _____
(підпис)

Маслій Р. В.
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Plagiat.pl (StrikePlagiarism) щодо роботи.

Автор роботи _____
(підпис)

Боднарєнко Б. О.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

Іванов Ю. Ю.
(прізвище, ініціали)