

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

**на тему «Інформаційна технологія формування розкладу
навчальних занять»**

Виконав: студент 2 курсу,
групи КН-18 м
спеціальності 122 «Комп'ютерні науки»
Давидюк Р. О.

Керівник: к.т.н., доц. Месюра В.І.

Рецензент: к. т. н., доц. Войтко В. В.

Вінниця
2019

ЗАТВЕРДЖУЮ
Завідувач кафедри КН _____
д.т.н., проф.. Яровий А.А.

_____ (підпис)
“ _____ ” _____ 2019 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра наук зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.027.18.000.ПЗ

Магістранта групи ІКН-17м Давидюка Романа Олександровича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія формування розкладу навчальних занять»

Вхідні дані: 1) кількість слотів часу – не менше 320, 2) кількість лекційних аудиторій – не менше 3, 3) кількість лабораторій – не менше 7, 4) кількість дисциплін – не менше 21, 5) кількість викладачів – не менше 15, 6) кількість академ. груп – не менше 12.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Граф-схема алгоритму роботи програмного забезпечення формування розкладу навчальних занять, структура інформаційної технології формування розкладу навчальних занять, діаграма класів програмного забезпечення, початкові вікна програмного забезпечення, робочі вікна програмного забезпечення формування розкладу навчальних занять.

2. Текстова (пояснювальна записка): Вступ, аналіз предметної області формування розкладу навчальних занять, розробка інформаційної технології формування розкладу навчальних занять, програмна реалізація інформаційної технології формування розкладу навчальних занять, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня розвитку інформаційних технологій формування розкладу навчальних занять. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Побудова математичних моделей формування розкладу навчальних занять			Математичні моделі, розділ 2
3	Практичне застосування та оцінка ефективності розроблених моделей			розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, проф. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ В.І.Месюра
ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц., доц. каф. ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ М. В. Бальзан
ініціали та прізвище

Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____ канд. техн. наук, доц., доц. кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)
 _____ В. В. Войтко
ініціали та прізвище

Завдання видав науковий керівник _____ канд. техн. наук, проф. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 _____ В.І.Месюра
ініціали та прізвище
 “ ____ ” _____ 20__ р.

Завдання отримав магістрант _____ Р.О. Давидюк
(підпис) ініціали та прізвище
 “ ____ ” _____ 20__ р.

АНОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для формування розкладу навчальних занять за допомогою гібридного метаевристичного алгоритму. Були розглянуті основні методи гібридних метаевристичних алгоритмів, визначені їх переваги і недоліки, та на основі цього був зроблений вибір на користь гібридного метаевристичного алгоритму. Було досліджено структуру, математичну модель та алгоритм роботи гібридного метаевристичного алгоритму, який слугував базисом для розробки алгоритму роботи програмного забезпечення. Був побудований програмний засіб формування розкладу навчальних занять на мові програмування C#, який характеризується підвищеною в 3,8 рази швидкістю формування розкладу навчальних занять.

ABSTRACT

This master's degree work is devoted to the development of software for scheduling training sessions using a hybrid meta-heuristic algorithm. The basic methods of hybrid meta-heuristic algorithms were considered, their advantages and disadvantages were determined, and on the basis of this a choice was made in favor of the hybrid meta-heuristic algorithm. The structure, mathematical model and algorithm of the hybrid meta-heuristic algorithm, which served as the basis for the development of the software algorithm, were investigated. A software tool for the formation of training schedule in the C # programming language was built, which is characterized by a 3.8-fold increase in the speed of formation of the training schedule.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ФОРМУВАННЯ РОЗКЛАДУ НАЧАЛЬНИХ ЗАНЯТЬ.....	12
1.1 Формалізація задачі формування розкладу навчальних занять	12
1.2 Аналітичний огляд існуючих методів формування розкладу навчальних занять	17
1.3 Аналіз аналогів до програмного забезпечення формування розкладу навчальних занять	21
1.4 Постановка задачі формування розкладу навчальних занять	23
1.5 Висновок	26
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ РОЗКЛАДУ НАЧАЛЬНИХ ЗАНЯТЬ.....	27
2.1 Адаптація генетичного алгоритму до формування розкладу начальних занять	27
2.2 Модифікація алгоритму імітації відпалу	36
2.3 Розробка гібридного алгоритму формування розкладу начальних занять на основі генетичного алгоритму та методу імітації відпалу.....	40
2.4 Структура інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму	44
2.5 Висновок	46
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ РОЗКЛАДУ НАЧАЛЬНИХ ЗАНЯТЬ.....	48
3.1 Обґрунтування вибору мови програмування	48
3.2 Розробка структури програмного забезпечення формування розкладу начальних занять на основі гібридного алгоритму.....	50
3.3 Розробка програмного забезпечення формування розкладу начальних занять на основі гібридного алгоритму	53
3.4 Аналіз результатів тестування програми.....	56
3.5 Висновок	64
4 ЕКОНОМІЧНА ЧАСТИНА.....	66
4.1 Оцінювання комерційного потенціалу розробки.....	66
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.	67

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.	71
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності..	72
4.5 Висновок	75
ВИСНОВКИ.....	77
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
Додаток А Інструкція користувача.....	83
Додаток Б Лістинг програми.....	85
Додаток В Графічна частина.....	96

ВСТУП

Актуальність. Проблемі автоматизації формування навчальних розкладів, як акції оптимального управління, присвячено величезну кількість наукових робіт, що різняться між собою у залежності від вимог освітянських структур. Формально дана тематика відноситься до класу задач нелінійного цілочисельного програмування великої розмірності. Її цільова функція, як правило, передбачає мінімізацію сумарних штрафів за порушення базових, але необов'язкових властивостей розкладу, а як обмеження формалізуються вимоги, які мають бути виконані обов'язково. Точний розв'язок такої задачі є практично неможливим за реальний час у зв'язку з її великою розмірністю (десятки тисяч змінних), тому на практиці використовують наближені пошукові методи.

Задача складання розкладу навчальних занять вищого навчального закладу (University Course Timetabling Problem - UCTP) є типовою проблемою усіх університети світу. Існує велика кількість постановок цієї задачі, що мають різну ступінь математичної формалізації, запропоновані різні методи і алгоритми вирішення UCTP і розроблені численні інформаційні системи, у тому числі і комерційні. Актуальність пошуку ефективних методів і алгоритмів вирішення UCTP підтверджується фактом проведення з 2002 р міжнародних змагань з цієї проблематики (International Timetabling Competition).

У серпні 2020 року у Відні, Австрія, відбудеться 14-а міжнародна конференція з практики і теорії автоматичного формування навчальних розкладів (14th International Conference on the Practice and Theory of Automated Timetabling).

Можна констатувати, що на сьогоднішній день відсутні загальноновизнані і незаперечні моделі, методи і алгоритми в цій галузі прикладної математики та задача продовження досліджень в напрямку побудови оптимальних розкладів, як і раніше, стоїть на порядку денному.

Постійно розвивається ринок автоматизованих систем складання розкладів занять. Верхні рядки світових рейтингів займають також такі зарубіжні розробки, як Lantiv Scheduling Studio 7 (Ізраїль) і OROLOGIO 13.x (Греція), «ІС: Автоматизоване складання розкладу. Університет» (Росія).

Все це свідчить про актуальність проведення теоретичних і прикладних досліджень за тематикою UCTP, а також побудови ефективних комп'ютерних реалізацій розроблених методів і алгоритмів.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення швидкодії формування розкладу навчальних занять за рахунок застосування гібридних метаевристичних алгоритмів.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз проблеми формування розкладу навчальних занять;
- розглянути існуючі методи вирішення задачі формування розкладу навчальних занять та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- запропонувати модель формування розкладу навчальних занять, на основі моделі розробити алгоритм формування розкладу навчальних занять,
- спроектувати структуру інформаційної технології для формування розкладу навчальних занять,
- здійснити програмну реалізацію інформаційної технології формування розкладу навчальних занять,
- провести тестування та аналіз результатів роботи програмних засобів формування розкладу навчальних занять.

Об’єкт дослідження – процес комп’ютеризованого формування розкладу навчальних занять з використанням гібридних метаевристичних алгоритмів.

Предмет дослідження – методи та програмні засоби формування розкладу навчальних занять з використанням гібридних метаевристичних алгоритмів та їх швидкодія.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, дослідження операцій, систем штучного інтелекту, теорії генетичних алгоритмів для реалізації інформаційної технології формування розкладу навчальних занять, методи математичної статистики для розробки процесу формування розкладу навчальних занять та обрахунків результатів експериментів із програмним засобом, об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія формування розкладу навчальних занять, яка відрізняється використанням гібридного метаевристичного методу на основі генетичного алгоритму та методу імітації відпалу, що дозволило підвищити швидкодію формування розкладу навчальних занять.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення формування розкладу формування розкладу навчальних занять.

Запропонована інформаційна технологія сприяє підвищенню якості процесу формування розкладу навчальних занять, зокрема:

– розроблено алгоритм роботи програмного забезпечення формування розкладу навчальних занять на основі гібридного метаевристичного алгоритму;

– розроблено програмні засоби для формування розкладу навчальних занять на основі гібридного метаевристичного алгоритму.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації

інформаційної технології формування розкладу навчальних занять. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів та методів підвищення швидкодії [1].

Апробація результатів роботи. Результати досліджень апробовані на науково-технічній конференції «Молодь в науці: дослідження, проблеми, перспективи-2020», Вінниця, 2019 року.

Публікації. За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ФОРМУВАННЯ РОЗКЛАДУ НАЧАЛЬНИХ ЗАНЯТЬ

1.1 Формалізація задачі формування розкладу навчальних занять

Розглянемо навчальний заклад, у якому виділяються такі групи об'єктів:

- a) множина G груп, що навчаються;
- b) множина A аудиторій;
- c) множина D дисциплін;
- d) множина V викладачів;
- e) множина T навчальних уроків (часових інтервалів проведення занять).

Якщо у зазначеній групі G проводяться заняття в аудиторії A з дисципліни D , викладачем V , під час навчальної пари T , то функція набуває значення рівне 1, в протилежному випадку - 0.

Нехай необхідно визначити

$$\begin{aligned} \alpha &= (\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_{N_{\text{блоків}}}), \\ t &= (t_1, t_2, \dots, t_i, \dots, t_{N_{\text{блоків}}}), \\ v &= (v_1, v_2, \dots, v_i, \dots, v_{N_{\text{блоків}}}), \end{aligned} \quad (1.1)$$

де $\alpha_i \in A$ - код аудиторії, призначений блоку занять $z_i \in Z$; $t_i \in T$ - код навчальної пари, призначений першого заняття з блоку занять $z_i \in Z$. $v_i \in V$ - код викладача, призначений блоку занять $z_i \in Z$.

До розкладу пред'являється множина вимог і обмежень [2, 3]. Умовно, весь ряд обмежень розбивається на обов'язкові та бажані обмеження. До обов'язкових належать:

- відсутність одночасного проведення різних занять в одній аудиторії;
- у заняттях студентів не мають з'являтися «вікна», хоча для викладачів вони допускаються;
- обладнання аудиторій має відповідати типу занять, які в них проводяться;

- місткість аудиторій має бути достатньою для груп, які у ній займаються, при цьому в одній аудиторії можуть водночас проводитися заняття для кількох груп;
- обмеження на обсяг щоденних занять;
- обов'язкове проведення всіх занять, запланованих робочим навчальним планом.

До бажаних (неосновних) вимог віднесемо:

- проведення лекцій на початку дня, а практичних та лабораторних робіт – наприкінці;
- мінімізація переходів між аудиторіями або корпусами;
- побажання викладацького складу;
- надання викладачам одного вільного від занять дня на тиждень для проведення наукових досліджень;
- рівномірність навантаження студентів протягом усього семестру, а також конкретного навчального дня.

Формалізуємо обмеження, що накладаються на розклад [4-5].

Обмеження (1.2) гарантує відсутність накладок для аудиторій. Для кожної впорядкованої двійки елементів: аудиторія - урок, в аудиторії існує або єдиний блок занять з множини Z , що означає проведення заняття цього блоку в цій аудиторії під час даної пари, або відсутність блоку заняття, що вказує на те, що аудиторія вільна.

$$\forall (a_i, t_j) : a_i \in A, t_j \in T$$

$$(\exists! z_k : (a_i = a_k) \wedge (z_k \in Z^{t_j})) \vee (\neg \exists z_k : (a_i = a_k) \wedge (z_k \in Z^{t_j})), \quad (1.2)$$

де Z^{t_j} - множина блоків занять, що проводяться під час уроку t_k .

Обмеження (1.3) гарантує відсутність накладок для викладачів: існує або єдиний блок занять, які веде даний викладач під час заданого уроку, або цього блоку не існує взагалі.

$$\begin{aligned} & \forall (v_i, t_j) : v_i \in V, t_j \in T \\ & (\exists! z_k : (v_i = v_k) \wedge (z_k \in Z^{t_j})) \vee (\neg \exists z_k : (v_i = v_k) \wedge (z_k \in Z^{t_j})), \end{aligned} \quad (1.3)$$

де Z^{t_j} - множина блоків занять, що проводяться під час уроку t_k .

Обмеження (1.4) забезпечує відсутність накладок для навчальних груп, тобто для кожної пари елементів: група - урок, сума компонент z_i^e вектора z_j блоків з множини $Z^{g_j} \cap Z^{t_k}$ не перевищує одиниці.

$$\forall (g_n, t_j) : g_n \in G, t_k \in T \sum z_i^e \leq 1, i \in Z^{g_n} \cap Z^{t_j}, \quad (1.4)$$

де Z^{g_n} - множина блоків занять, на яких присутня група g , а Z^{t_j} – множина блоків занять, що проводяться під час уроку t_k .

Під час конкретної пари група знаходиться на одному занятті, або проводиться заняття тільки у однієї з підгруп, або у обох, або заняття відсутні взагалі.

$$\forall z_i \in Z, a_i \in A^{z_i^a} \quad (1.5)$$

Вираз (1.5) описує відповідність типу аудиторії заняттю, що проводиться, тобто для кожного блоку занять $z_i, z_i \in Z$ аудиторія вибирається з допустимої підмножини аудиторій, код цієї підмножини зберігає компонента z_i^a .

$$\forall (b_\tau, g_n) : b_\tau \in B, g_n \in G \sum z_i^e \leq N_{max}, i \in I_{g_n}^{b_\tau}, \quad (1.6)$$

де $B = \{b_1, b_2, \dots, b_N\}$ - множина навчальних днів. Кожен елемент описаної множини, виділяється так:

$$b_\tau = \{t_j \in T : t_j^d = b_\tau\}.$$

Таким чином виглядає обмеження, накладене на кількість навчальних уроків, що проводяться протягом одного навчального дня, що означає, що для кожної пари елементів: група і день, кількість проведених уроків не перевищує максимально допустимого – N_{max} .

$$\forall (b_\tau, g_n): b_\tau \in B, g_n \in G \left(\sum_{i \in I_{g_n}^{b_\tau}} z_i^e = t_max_number_{g_n}^{b_\tau} - t_min_number_{g_n}^{b_\tau} + 1 \right) \wedge$$

$$\wedge (\forall t: t_min_l_{g_n}^{b_\tau} \leq t \leq t_max_l_{g_n}^{b_\tau} \sum_{z_i \in Z^{g_n} \wedge t_j^p = t} z_i^e = 1) \quad (1.7)$$

де $t_max_number_{g_n}^{b_\tau} - t_min_number_{g_n}^{b_\tau} + 1$ - відсутність вікон у груп;

$t_max_number_{g_n}^{b_\tau}$ - максимальний номер пари протягом дня b_t для групи g_n

$t_min_number_{g_n}^{b_\tau}$ - мінімальний номер пари протягом дня b_t , для групи g_n ,

$I_{g_n}^{b_\tau} = \{i: (z_i \in Z^{g_n}) \wedge (t_j^d = b_\tau)\}$ - множина номерів блоків занять, що проводяться для груп g_n під час дня b_t .

Таким чином записується обмеження відсутності вікон для навчальних груп, тобто для кожної пари елементів: день - група, кількість пар, що проводяться в цій групі у поточний день, має дорівнювати цій величині.

Потрібно знайти такий варіант вибору векторів α , t , ρ , який задовольняє обмеженням (1.2) - (1.7), а також мінімізує значення критерію втрати якості K . Критерій якості ґрунтується на бажаних вимогах і має такий вигляд;

$$K = \varphi(\alpha, t, v) = \sum_{i=1}^N c_i w_i(\alpha, t, v)$$

де c_i - значення штрафного коефіцієнта за невиконання i -ої вимоги, а w_i – оцінка ступеня невиконання i -ої бажаної вимоги.

До найбільш значущих бажаних вимог відносяться:

1. Побажання викладацького складу

Для формулювання даної вимоги розглядаються дві матриці: матрицею заборон і матриця зайнятості:

$$M_{\text{заборон}} = \begin{cases} 1, \text{ заборона проведення заняття під час } k \text{ – го уроку} \\ 0, \text{ відсутність заборони} \end{cases}$$

Друга матриця - формується таким чином:

$$M_{\text{зайнятості}} = \begin{cases} 1, \text{ проведення заняття для викладача } i \text{ під час уроку } k \\ 0, \text{ відсутність занять} \end{cases}$$

$$\sum_{p_i \in P} \sum_{t_j \in T} (M_{\text{заборон}} \wedge M_{\text{зайнятості}}) \rightarrow \min$$

Вимога врахування побажань.

2. Мінімізація кількості вікон у викладачів.

$$\sum \sum (t_{\max_{p_i}^{b_\tau}} - t_{\min_{p_i}^{b_\tau}}) - N_{\text{блоков}} \rightarrow \min,$$

де $t_{\max_{p_i}^{b_\tau}}$ - максимальний номер уроку в день b_τ у викладача p_j ,

$t_{\min_{p_i}^{b_\tau}}$ - мінімальний номер уроку в день b_τ у викладача p_j .

3. Вимога рівномірності занять.

$$D_n^{\text{сер}} = \frac{1}{N_{\text{днів}}} \sum_{b_\tau} (M_y^{\text{сер}} - |I_{g_n}^{b_\tau}|)^2$$

Якщо середнє відхилення кількості занять для групи g_n складає:

$$M_n^{\text{сер}} = \frac{1}{N_{\text{днів}}} \sum_{b_{\tau} \in B} |I_{g_n}^{b_{\tau}}|,$$

то вимога набуває такого вигляду:

$$\sum_{n=1, N_{\text{груп}}} D_n^{\text{сер}} \rightarrow \min$$

Цільова функція будується на основі мінімізації штрафних показників. Кожне порушення обмеження або бажаної вимоги збільшує значення цільової функції відповідно до коефіцієнта значущості вимоги. В результаті цільова функція в загальному вигляді описується формулою:

$$F_{\text{ц}} = \sum_{i=1, N_{\text{кільк.обмежень}}} \text{obm}_i * k_{\text{obm}_i} + K$$

На підставі побудованої математичної моделі (1.1) – (1.7) розкладу здійснюється реалізація еволюційного пошуку оптимального розкладу з використанням генетичного алгоритму.

1.2 Аналітичний огляд існуючих методів формування розкладу навчальних занять

Проаналізуємо основні методи [6], що зустрічаються на практиці при вирішенні задачі формування розкладу в освітніх установах:

- класичні методи (теорія графів, цілочисельне лінійне програмування);
- метаевристичні (генетичні алгоритми, метод відпалу, метод мурашиних колоній);
- багато-агентні системи;
- метод рішення по прецедентах.

- методи повного перебору,
- градієнтні методи,
- методи гілок і меж.

Різноманітність методів і підходів при вирішенні задачі свідчить про відсутність універсального методу автоматичного формування розкладу навчального закладу, тому необхідно здійснити аналіз методів для визначення найбільш оптимального підходу, виявити переваги і недоліки методів, а так само ступень їх універсальності і обмеження, яким вони задовольняють.

Специфічною характеристикою класичних методів є найбільш високий ступінь формалізації (математична точність) як самої задачі формування розкладу занять, так і алгоритмів її реалізації («жорсткі» алгоритми). Дані методи дозволяють:

- а) дослідити методи рішення задачі формування навчальних розкладів;
- б) зрозуміти вплив цікавих для нас факторів на час і чіткість виконання задачі формування розкладів навчальних занять

Метод розмальовування графа заснований на застосуванні математичного апарата теорії графів і полягає у формуванні розкладу таким чином. Кожна вершина подає розрахований навчальним планом урок, накладки виникають в тому випадку, якщо між якимось двома вершинами з'являється суперечність. Наприклад, якщо два уроки проводяться в одному класі, то вони з'єднуються ребрами, що рівносильне забороні одночасного проведення цих уроків. Тоді проблему формування розкладів можна виразити як проблему зменшення кількості кольорів потрібних для розмальовування графа, де кожен колір відповідає одному єдиному періоду розкладу. Наближені методи вирішення задачі про формування розкладу формулюються як підзадачі у гібридних методах.

Використання зазначеного способу при реалізації дійсних проблем формування розкладу для великомасштабних освітніх систем навчання не

продуктивно, але комбінація розглянутого підходу з іншими методами може виявитися практичною.

Другим методом є метод цілочисельного лінійного програмування, який застосовується для вирішенні задачі формування розкладу за допомогою рішення симплекс-методу для знаходження цільової функції.

Евристичні методи базуються на використанні різноманітних видів евристик, або евристичних алгоритмів, при підготовки яких використовуються «сліпі» припущення, не підтвержені відповідним математичним аргументом. Формування деяких правил (евристик) допускає певне прискореного пошуку «найкращого» розкладу, але застосування таких алгоритмів у більшості випадків дає гарантію всього лише виявлення приблизного рішення задачі (наближеного локального екстремуму). В даному випадку виникає питання оцінки наближеності знайденого локального екстремуму до глобального екстремуму. У ряді робіт розв'язування задач великої розмірності намагаються реалізувати шляхом порівняння розкладів, отриманих евристичними методами, і розкладів, отриманих методами перебору [7].

У зв'язку із зазначеним недоліком, евристичні алгоритми залишаються надійним інструментом пошуку «кращого» у певному сенсі рішення в тих випадках, коли знаходження оптимального найкращого рішення є практично важким або неможливим.

Розглянемо основні метаевристичні методи вирішення задачі формування навчального розкладу. Генетичні алгоритми - методи вирішення великорозмірних задач цілочисельного програмування, що інтенсивно розвиваються у даний час. Істотні відмінності і переваги генетичних алгоритмів у порівнянні з класичними методами виражається у такому:

- генетичний алгоритм працює з операторами, у яких представлений список параметрів, що залежать від аргументів цільової функції;
- в процесі пошуку використовується деяка кількість точок пошукового інструменту (запускаються паралельні процеси), а не перемикається від точки до точки, як це виконується в традиційних методах. Тобто генетичний алгоритм

оперує одразу з цілою сукупністю припустимих рішень;

- генетичний алгоритм в процесі роботи не використовує допоміжної інформації, що значно збільшує швидкість його реалізації;

- генетичний алгоритм використовує для перемикання від одних точок до інших як імовірнісні, так і детерміновані підходи до створення нових точок пошуку [8].

У методі рою частинок агентами є частинки, які у кожному мить часу мають в просторі параметрів задачі деякий стан і швидкість. Правила, за якими частинки змінюють своє положення і швидкість, визначаються на основі обчислення цільової функції частки. В основі канонічного методу рою частинок лежить такий принцип: на кожній ітерації для визначення наступного положення частинки враховується інформація про найкращу «сусідню» частку та інформація про дану частку на тому етапі, коли їй відповідало оптимальне значення цільової функції. Так само існує модифікація канонічної моделі, яка враховує значення цільової функції всіх часток рою, в деяких моделях частки групуються у кілька роїв і т. д.

Багатоагентна система - це система, що самоорганізується, що складається з декількох взаємодіючих між собою інтелектуальних агентів. Виділяють дві основні ролі агентів:

- а) агенти - організатори (агенти-вчителя);
- б) агенти - учасники (агенти-класи, агенти-аудиторії, агенти).

Агенти взаємодіють між собою, намагаючись знайти прийнятний час і місце події (заняття). Зауважимо, що багатоагентні системи не знайшли жодного допустимого розкладу на тестах змагань ІТС, яке проводилося в рамках конференції ПАТАТ (Practice and Theory of Automated Timetabling) [9].

У методі рішення по прецедентах, прецеденти подаються у вигляді списку особливості-значення, база прецедентів містить по дві кращих евристики з визначеного набору евристик. Недоліки методу: низька гнучкість в

обмеженнях, низька якість розкладу при невеликій кількості прецедентів, погана масштабованість. Метод не враховує специфічні вимоги.

У результаті аналізу існуючих методів складання розкладу, можна зробити висновок про відсутність у теперішній час повністю автоматизованих систем складання розкладів у вищих навчальних закладах через велику трудомісткість побудови точних математичних моделей та велику кількість вхідних варійованого даних і правил опису обмежень при складанні розкладу, які важко формалізуються.

1.3 Аналіз аналогів до програмного забезпечення формування розкладу навчальних занять

Кількість публікацій в періодичних виданнях [2, 3] і працях конференцій [4] з даної тематики обчислюється тисячами. Для вирішення УСТР пропонують використовувати: класичні методи (теорія графів, целочисленне лінійне програмування); метаевристическіе (еволюційні і генетичні алгоритми, метод імітації відпалу - simulated annealing, табу-пошук - tabu search, метод мурашиних колоній - ant colony optimization, максміна мурашина система - max-min ant system, оптимізація роєм часток - particle swarm optimization); мультиагентні системи; метод рішення по прецедентах.

Останнім часом все більше уваги дослідники приділяють ідеї гібридизації (спільного використання) різних методів і алгоритмів при побудові загальної схеми рішення УСТР. Огляд публікацій, найбільш підходящими "кандидатами" для гібридизації є евристичні алгоритми послідовної (покрокової) локальної оптимізації, що базується на ідеї "жадібних" алгоритмів, і спеціалізовані версії генетичного алгоритму, що використовуються як для пошуку локального оптимуму, так і для керованого перебору варіантів (послідовностей) множини сутностей (навчальних одиниць), що розміщуються в сітці розкладу.

Постійно розвивається ринок автоматизованих систем складання розкладів занять. Верхні рядки світових рейтингів займають також такі

зарубіжні розробки, як Lantiv Scheduling Studio 7 (Ізраїль) і OROLOGIO 13.x (Греція), «ІС: Автоматизоване складання розкладу. Університет» (Росія).

Вид вікна програми Lantiv Scheduling Studio 7 (Ізраїль) показано на рис. 1.1. Її недоліки полягають у тому, що не завжди отримується максимально оптимізований розклад, невисока швидкодія та висока вартість.

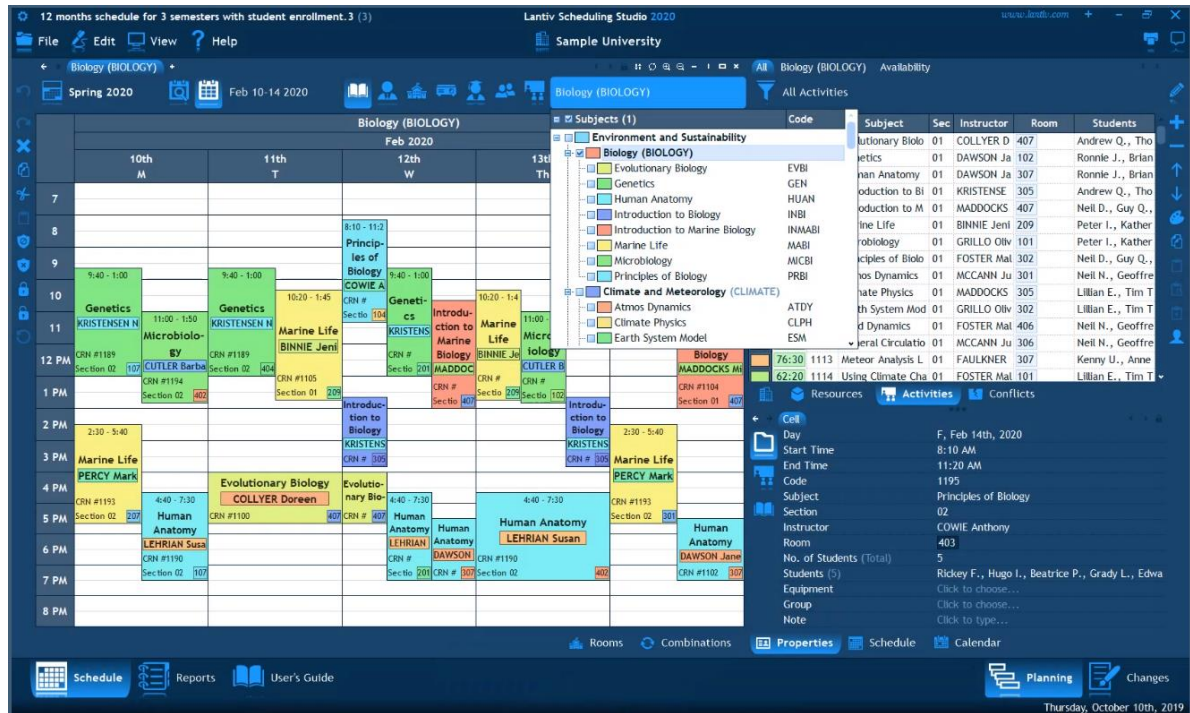


Рисунок 1.1 – Вид вікна програми Lantiv Scheduling Studio 7 (Ізраїль)

Вид вікна програми «ІС: Автоматизоване складання розкладу. Університет» (Росія) показано на рис. 1.2.

Її недоліки полягають у тому, що не завжди отримується максимально оптимізований розклад з точки зору виконання більшості м'яких обмежень, невисока швидкодія та висока вартість. Крім того, використання продуктів фірм країни-агресора заборонено в Україні.

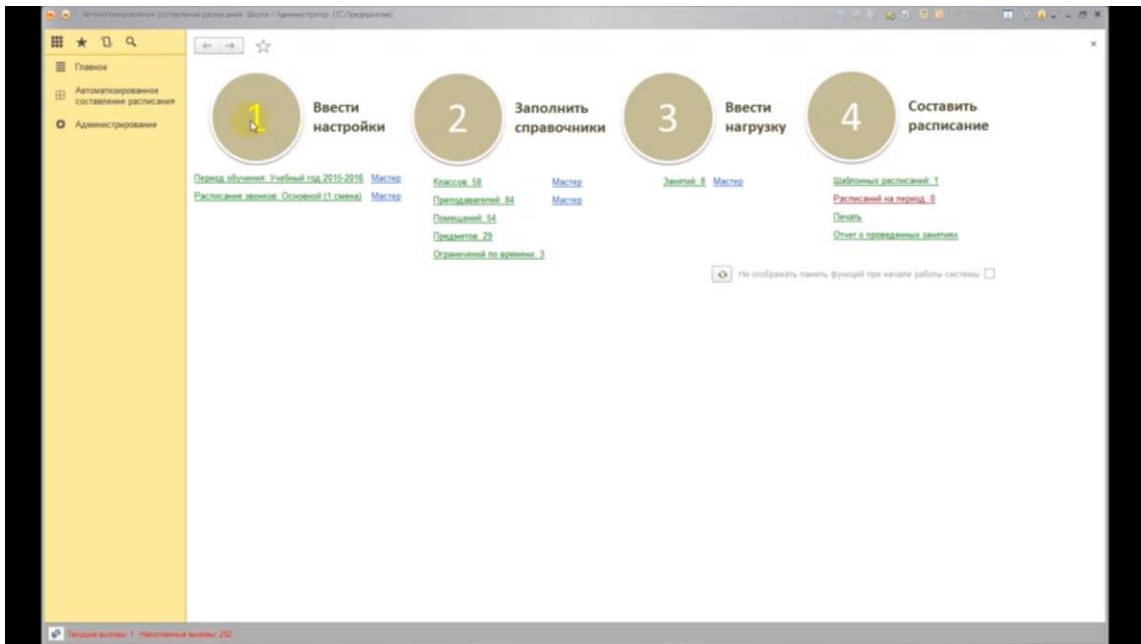


Рисунок 1.2 – Вид вікна програми «1С: Автоматизоване складання розкладу. Університет» (Росія)

Все вищезазначене свідчить про актуальність проведення теоретичних і прикладних досліджень з тематики методів та програмних засобів формування розкладу навчальних занять.

1.4 Постановка задачі формування розкладу навчальних занять

Розклад занять є однією з найважливіших складових навчального процесу будь-якого навчального закладу. Він подається документом, що регламентує трудовий ритм і впливає на творчу віддачу викладацького складу та розглядається як фактор оптимізації навчального процесу.

У розкладі занять містяться інформація про конкретне заняття викладача, котрий його проводить, групу (підгрупу, потік), для якого проводять заняття, та місце проведення (аудиторії).

Розклади зазвичай поділяють на:

- тижневі. В таких розкладах створюється типовий тиждень який дублюється на кожен з тижнів семестру;

- семестрові, що формуються на увесь семестр в цілому.

Значущими проблемними питаннями складання розкладу занять вищого навчального закладу (ВНЗ) є:

- відсутність єдиного джерела вхідної інформації і, як наслідок, необхідність ретельної підготовки, структуризації, збору та обробки великого обсягу вхідної інформації з різних структурних підрозділів ВНЗ, таких як навчальний відділ, факультети, кафедри, відділ кадрів, диспетчерська;

- складність чіткої формалізації й ідентифікації ряду вхідних параметрів і обмежень, ступінь і якість обліку яких повністю залежить від досвіду, кваліфікації і професійної інтуїції працівника диспетчерського відділу;

- суперечливість інтересів основних учасників навчального процесу студентів і викладачів, необхідність урахування наявного аудиторного фонду ВНЗ (тобто цільового призначення аудиторій), і, як наслідок, складність математичної формалізації єдиних вимог до оптимальності рішення задачі складання розкладу.

На даний час указаний аспект процесу планування навчального процесу практично не автоматизований, що призводить до значних втрат часу, численних помилок, низької адаптивності до поточних змін тощо.

Задача складання розкладу навчальних занять за сутністю є задачею складання оптимального розкладу при наявності певної множини обмежень.

Вхідними даними можуть бути: тривалість навчальних занять та їх кількість по днях тижня, тривалість перерв між заняттями, кількість предметів та викладачів, що їх викладають. Можливі обмеження визначатимуться такими аспектами навчального процесу, як необхідність чіткого дотримання порядку проходження дисциплін (тобто, наприклад, практичні заняття повинні проводитися після теоретичного вивчення матеріалу), потреба врахувати міждисциплінарний взаємозв'язок (наприклад, вивчення деяких тем однієї дисципліни ґрунтується на знаннях, отриманих при вивченні попередніх тем іншого предмету). Крім того, в процесі складання розкладу можуть бути враховані інші обмеження: необхідність проведення деяких занять у

спеціалізованих класах, лабораторіях тощо; можливість одночасного проведення заняття з декількома навчальними групами у відповідній аудиторії і т.п. Критерієм оптимальності отриманого розкладу може, наприклад, являтися мінімізація простоїв у навчальному процесі.

Рішення подібних задач теорії розкладів являє відому складність. За змістом ці задачі відносяться до класу комбінаторних [10,11,12], для яких суттєве значення має розмірність. Як правило, розмірність задач формування оптимальних розкладів настільки велика, що розв'язати їх простим перебором варіантів не є можливим. Часто задачі складання розкладів зводяться до задач цілочисельного лінійного програмування [13,14] (в тому числі багатоіндексного), для вирішення яких використовуються широко відомі методи відсікання або гілок і меж [15,16]. Традиційними методами дослідження операцій [17] для задач планування є комбінаторні процедури, імітаційне моделювання [18], нейромережеві методи й евристичні підходи.

Основними задачами роботи є:

1. Формалізація задачі формування розкладу занять з урахуванням вимог і побажань, які відображають реальні умови планування і організації навчальної діяльності вузів. При формалізації і побудові моделі використовувати не тільки загальноприйнятту математичну (теоретико-множинну) нотацію, але і нотації, що використовуються в комп'ютерних науках.

2. Обґрунтування поділу жорстких обмежень, що встановлюють фізичну реалізованість розкладу, на статичні, які можна врахувати до складання розкладу і динамічні, які повинні перевірятися в процесі складання розкладу.

3. Розробити алгоритми, програмні засоби і структури даних для перевірки жорстких обмежень, що знижують обчислювальні витрати на реалізацію основних циклів пошуку оптимального розкладу за рахунок максимально можливої апріорної обробки вихідних даних.

4. Обґрунтувати інтегральний критерій якості розкладу, складовими якого є кількісні оцінки міри порушення м'яких обмежень, «зважені» нормувальними множниками. Реалізувати у вигляді завершеного програмного

засобу алгоритм оцінки якості розкладу, що допускає його використання на різних етапах вирішення оптимізаційної задачі складання розкладу.

5. Сформулювати задачу послідовної локальної оптимізації розкладу; розробити і програмно реалізувати алгоритм її вирішення, заснований на гібридизації генетичного алгоритму та алгоритмів відпалу.

6. Розробити гібридні алгоритми і здійснити їх програмну реалізацію для перевірки ефективності запропонованих рішень.

7. Зробити висновки щодо ефективності реалізованих гібридних алгоритмів та намітити шляхи подальших досліджень в області автоматичного формування навчального розкладу вищого навчального закладу.

1.5 Висновок

У даному розділі магістерської роботи було виявлено високу актуальність задачі автоматичного формування розкладу навчальних занять, як узагальненої проблеми оптимального управління. Було здійснено формалізацію постановки задачі, що дозволило формалізувати групи об'єктів розкладу і основні обмеження, яким він задовольняє. Було здійснено розподіл обмежень на обов'язкові (жорсткі), які визначають коректність розкладу та додаткові (м'які), які визначають його зручність, і кількість задоволення яких може бути використана, як міра якості розкладу. Був проведений аналіз існуючих методів формування розкладу навчальних занять, який засвідчив, що у зв'язку зі значною кількістю обмежень, жоден з них на сьогодні не може претендувати на домінування і гарантувати формування оптимального розкладу. Більше того, деякі дослідники стверджують неможливість створення такого розкладу в повністю автоматичному режимі. Також було обгрунтовано доцільність розробки і дослідження гібридного алгоритму формування навчального розкладу на базі двох (або більше) окремих алгоритмів різної природи, які при сумісному використанні компенсували б недоліки один одного.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ РОЗКЛАДУ НАЧАЛЬНИХ ЗАНЯТЬ

2.1 Адаптація генетичного алгоритму до формування розкладу начальних занять

Генетичний алгоритм є методом оптимізації, який базується на теорії еволюції Дарвіна та сучасних знаннях про генетику. Як і у природі, генетичні алгоритми включають такі поняття, як хромосоми (можливі розв'язки задачі), гени (змінні або властивість розв'язку), спарування (отримання нової хромосоми-рішення з двох старих), схрещування (обмін хромосом своїми частинами), мутація (випадкова зміна гена у хромосомі) та еволюція (створення умов для розвитку кращих хромосом-рішень, і видалення з розгляду неприпустимих та поганих рішень).

Загальну схему генетичного алгоритму (ГА) наведено на рис. 2.1.

Проблема розкладів університетів складається з набору курсів, які мають бути заплановані на набір часових інтервалів (T) та набір номерів аудиторій (r), і може бути подана у вигляді трьохвимірного кубу (рис.2.2).

При цьому мають бути задоволені певні обмеження, які поділяються на жорсткі та м'які.

У залежності від структури та особливостей навчального процесу перелік обмежень може сильно різнитися для різних навчальних закладів, але існує загальновизнаний список широко використовуваних жорстких обмежень, на якому можна порівнювати ефективність різних алгоритмів розв'язання задачі [6]:

Н1: учні можуть одночасно мати лише одну лекцію.

Н2: лектор може одночасно виголосити лише одну лекцію.

Н3: кімнату можна використовувати лише для однієї лекції за один раз.

Н4: лектори повинні бути доступними в той час, коли вони надаються.

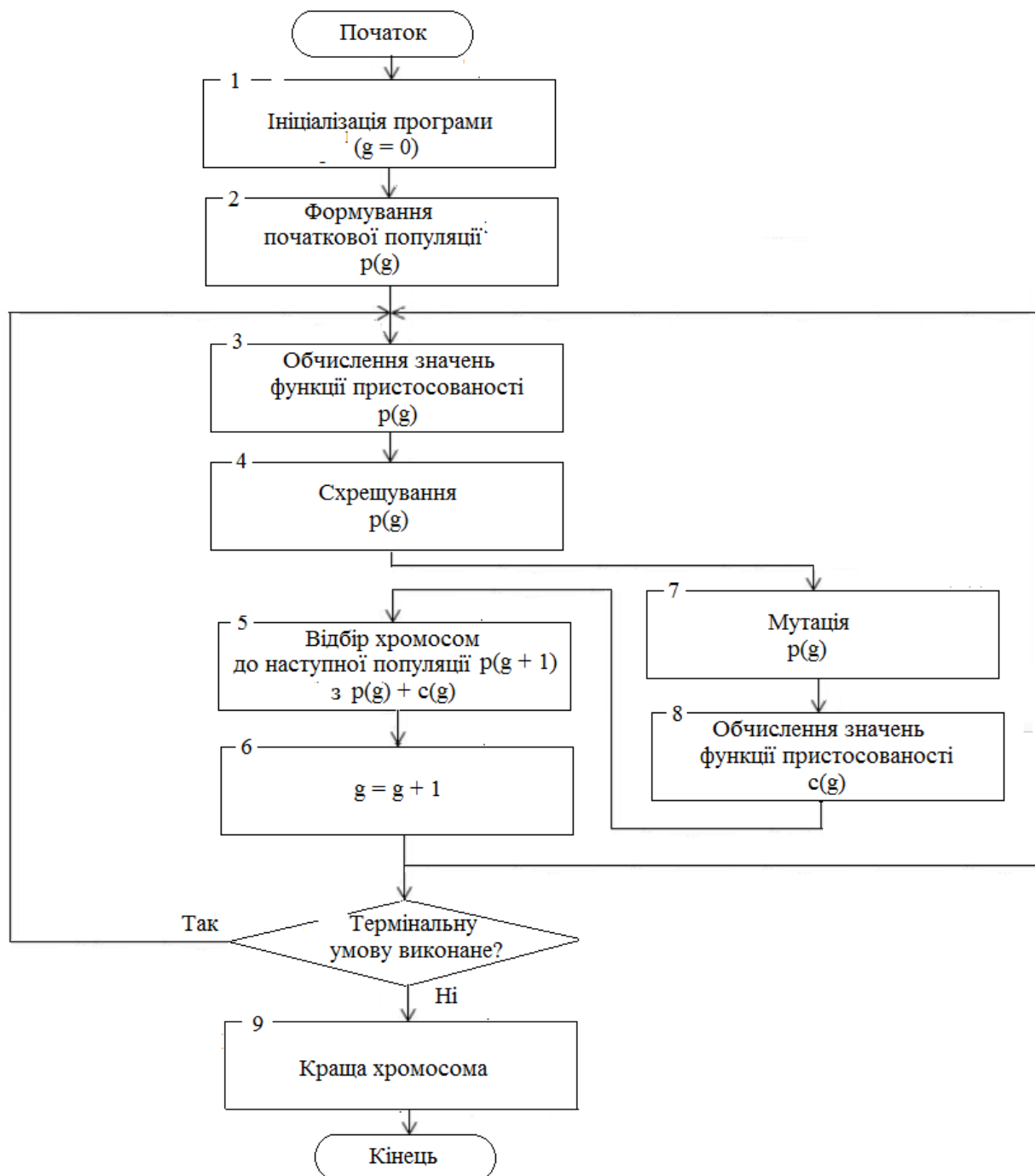


Рисунок 2.1 – Загальна схема генетичного алгоритму

Н5: вимоги щодо спеціальних приміщень (наприклад, лабораторій) враховуються.

Н6: ємність аудиторій повинна відповідати кількості студентів.

Н7: два або більше додаткових блоків курсу не можуть бути заплановані в один і той же день.

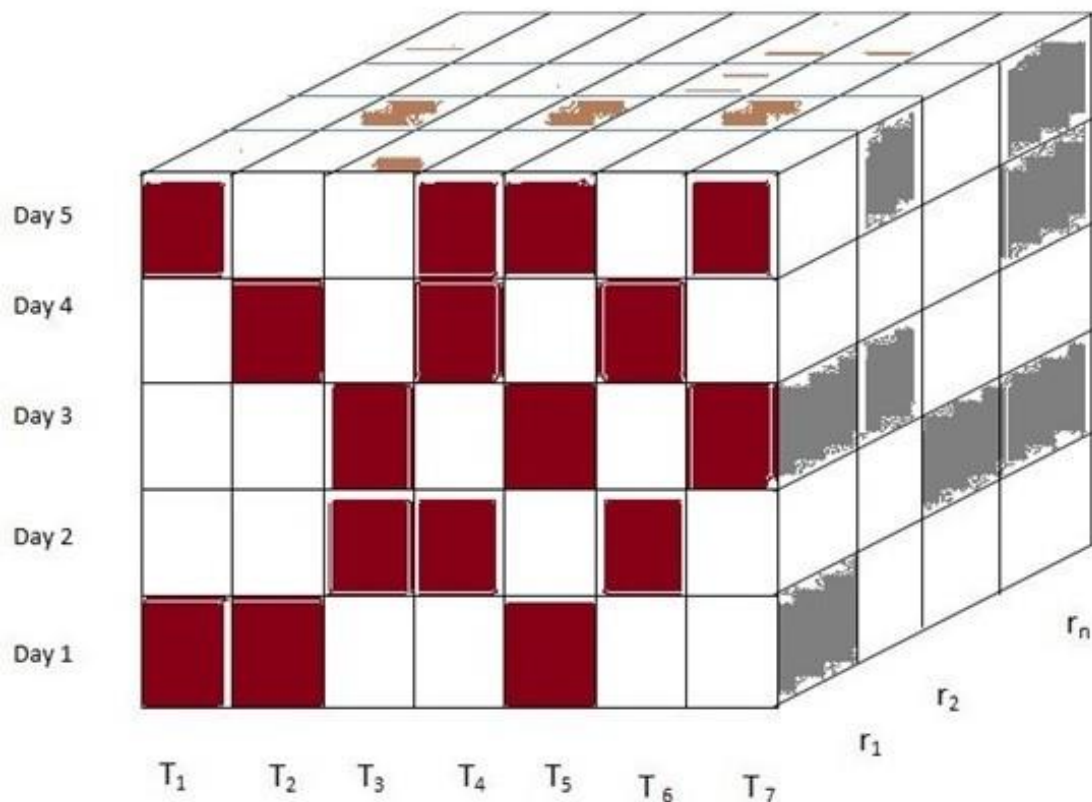


Рисунок 2.2 – Візуалізація сховища даних про навчальний розклад

М'які обмеження також змінюються перевагами академічних установ. Але існує перелік м'яких обмежень, яким має задовольняти розклад високої якості [19]:

S1: викладачі повинні мати принаймні один день для академічних досліджень.

S2: розклад для аудиторій має бути максимально компактним.

S3: кількість вільних періодів у графіках студентів має бути мінімальною.

S4: Перерва на обід має бути запланована з 12:00 до 15:00.

S5: Студенти повинні мати послідовні лекції в тій самій будівлі.

Жорсткі обмеження суворі, їх порушення є неприпустимим. Розклад може бути використаний навчальним закладом, для якого було створено, лише за умов беззаперечного виконання усіх важких обмежень. Такий розклад називається "здійсненням". При цьому задоволення всіх м'яких обмежень не є необхідним для розкладу, але саме кількість задоволених м'яких обмежень

визначає якість розкладу. Отож, основною метою розроблюваного генетичного алгоритму є побудова здійсненого розкладу з високою якістю.

Зауважимо, що на сьогодні відсутні програмні засоби, які б задовольняли усім індивідуальним перевагам.

Першим етапом генетичного алгоритму є ініціалізація (в.1, рис.2.1), яка повинна створити випадкову популяцію хромосом, що покриє весь простір пошуку рішень. Різноманітність початкової популяції надає алгоритму можливість шукати у всьому просторі можливих рішень і не застрягти у локальних оптимумах. Будемо створювати первинну популяцію зі 100 хромосом, і утворювати їх випадковим чином.

Для створення популяції хромосом генетичного алгоритму необхідно визначити форму їх подання (в.2, рис.2.1). Кожна хромосома повинна зберігати всю інформацію, необхідну для побудови усього розкладу. При цьому, подання хромосоми має бути зручним з точки зору побудови ефективних генетичних операцій, таких як схрещування, мутація та розрахунку фізичної придатності відповідного розв'язку.

Здійснимо пряме кодування хромосом, коли кожна хромосома являє собою можливий розв'язок. Будемо вважати. Що кожного дня в університеті може відбуватись 9 уроків, з 08.15 до 14.15 і з 14.45. до 17.45 При п'яти робочих днях отримаємо за тиждень $5 \times 9 = 45$ часових слотів (рис. 2.3).

	Понеділок	Вівторок	Середа	Четвер	П'ятниця
8.15-9.15	1	10	19	28	37
9.15-10.15	2	11	20	29	38
10.15-11.15	3	12	21	30	39
11.15-12.15	4	13	22	31	40
12.15-13.15	5	14	23	32	41
13.15-14.15	6	15	24	33	42
14.45-15.45	7	16	25	34	43
15.45-16.45	8	17	26	35	44
16.45-17.45	9	18	27	36	45

Потрібно побудувати щотижневий розклад для кожної кімнати. Для цього створимо масив RoomsTimeTable, надавши 45 тайм-слотів кожній аудиторії (рис. 2.3). Перші 45 елементів масиву відповідають першій аудиторії; наступні 45 елементів, від 46 до 90, відповідають другій аудиторії і так далі.

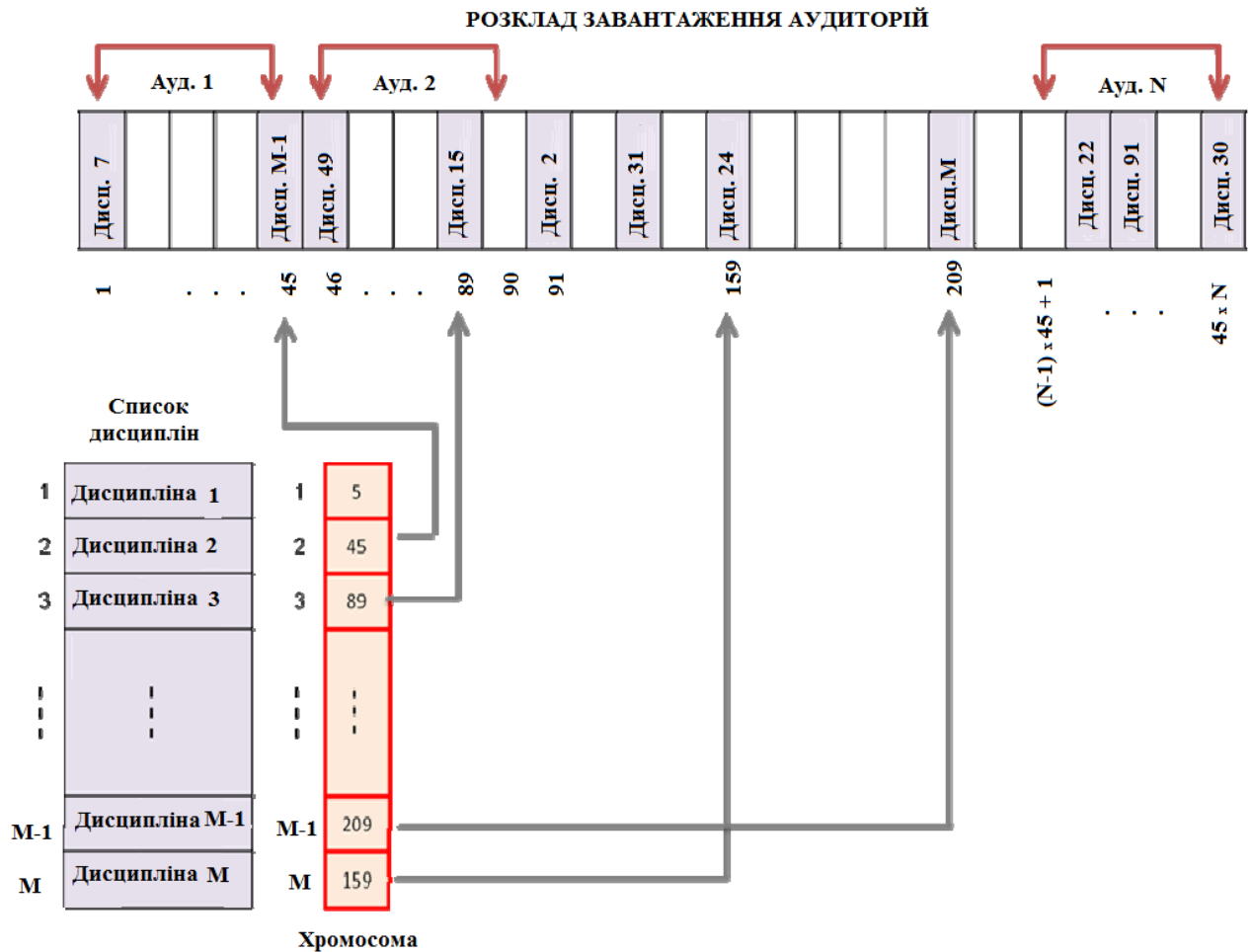


Рисунок 2.3 – Подання хромосоми

Таким чином, масив RoomsTimeTable матиме розмір $45 \times$ кількість аудиторій. Створимо також масив Course List, елементами якого будуть всі дисципліни, що викладаються в одному семестрі. Крім того, елементи цього масиву також зберігають інформацію про лектора та тривалість заняття. Таким чином хромосома подається одновимірним масивом $m \times 1$, де m - кількість курсів у списку курсів. Тому кожен елемент хромосоми має відношення з тим самим нумерованим елементом списку дисциплін. Кожен ген в хромосомі

(кожен елемент масиву), наприклад, `chrn [i]`, має значення індексу елемента розкладу кімнати, де планувався і-й курс.

Наприклад, на рисунку 2.3 другий елемент (ген) хромосоми містить значення 45, оскільки дисципліну М-1 було занесено до 45-го елемента у таблиці розкладу завантаження аудиторій. Це означає, що дисципліна М-1 надається у п'ятницю, починається о 16:30 в аудиторії №1. Деякі заняття тривають дві-три години, тому кожен хромосомний елемент означає час початку кожного заняття.

Інформацію про те, хто читає дисципліну і скільки часу займе заняття вбудуємо в елемент списку курсів.

Для вимірювання якості поданого хромосомою рішення необхідно визначити функцію пристосованості (*fitness-function*). Чим більший кількість м'яких обмежень задовольняє хромосома, тим більшим буде значення функції пристосованості. Отже, оцінювання хромосом і вибір обмежень є наступними важливими кроками генетичного алгоритму.

Частіше за все значення функції пристосованості для кожної хромосоми обчислюється зі значення протилежної функції незадоволених обмежень. Наприклад, часто використовуються функції пристосованості зі значеннями:

$$\frac{1}{1+x}, \frac{1}{1+x^2} \text{ або } \frac{1}{1+\sqrt{x}}$$

Скористаємось таким визначенням функцію пристосованості:

$$\text{fitness}(x) = 1 - \frac{\sum \text{кількість порушених м'яких обмежень}}{\sum \text{кількість м'яких обмежень}}$$

Таким чином, кожна хромосома отримає значення функції пристосованості (в.3, рис.2.1) з інтервалу $[0,1]$, і у разі задоволення усіх обмежень її значенням пристосованості буде 1.

Існує три групи метаевристичних алгоритмів обробки обмежень [20]:

- алгоритми одноетапної оптимізації: задоволення як жорстких, так і м'яких обмежень;
- алгоритми двоетапної оптимізації: задоволення м'яких обмежень здійснюється після побудови «здійсненого» розкладу;
- алгоритми з пом'якшенням покарання: порушення важких обмежень від початку забороняються, пом'якшуючи деякі інші особливості проблеми. Потім намагаються спробувати задовольнити м'які обмеження, а також усунути допущені пом'якшення.

Побудуємо двоступеневий алгоритм оптимізації. На відміну від класичних генетичних алгоритмів, не буде допускати порушення жорстких обмежень на жодній ступені еволюції хромосом. Отже, у кожному варіанті розкладу всі хромосоми будуть можливим рішенням, після чого будемо намагатися звести до мінімуму порушення м'яких обмежень. Тому функція пристосованості буде розглядати лише задоволення м'яких обмежень.

Після оцінки хромосом, деякі з них треба вибрати для створення наступного покоління. Здійснимо вибір у два етапи. На першому етапі здійснимо оцінку пристосованості хромосом і відберемо для розмноження 20 найкращих хромосом. На другому етапі додатково відберемо для розмноження 40 випадкових хромосом. Такий підхід має забезпечити гарний генетичний матеріал за рахунок перших 20 хромосом і різноманітність пошуку, дозволивши розмноження менш пристосованим хромосомам.

Для змішування генетичного матеріалу двох батьків при створенні нащадків використаємо двохточкове схрещування (в.3, рис.2.1) з імовірністю схрещування p_c , яке реалізуємо таким чином (рис. 2.4):

1. Випадково виберемо точки діапазону обміну генами.
2. Здійснимо обмін генами між двома хромосомами.

3. У разі отримання внаслідок схрещування хромосоми з дублікатами генів, розташуємо отриманий дублікат у випадковим чином вибраний вільний і здійснений часовий інтервал.

7	2	5	9	6	8	3	1	4	→	7	2	5	2	9	1	3	1	4
3	7	5	2	9	1	4	6	8		3	7	5	9	6	8	4	6	8

а)

7	2	5	2	9	1	3	1	4	→	7	6	5	2	9	1	3	8	4
3	7	5	9	6	8	4	6	8		3	7	5	9	6	8	4	1	2

б)

Рисунок 2.4 – Генетична операція схрещування з усуненням дублікатів генів:

а) виконання двохточкового схрещування; б) відбудова здійсненої хромосоми

Зауважимо на важливість усунення дублікатів. У змінній частині хромосом один або більше генів можуть зустрічатися два рази. Це означає, що дві різні дисципліни почнуться одночасно, в одній кімнаті. Для подолання цього конфлікту і знаходиться вільний часовий інтервал, де розміщується одна з суперечливих дисциплін і отримується коректно оновлена популяція.

Схрещування хромосом може призвести до потомства з кращою пристосованістю, ніж їхні батьки, але воно не завжди забезпечує можливість дослідження деяких частин пошукового простору. Наприклад, якщо один або кілька генів батьків ідентичні, перетинання цих генів не створить нового гену. У екстремальному випадку, коли вся популяція буде мати один або кілька ідентичних генів, ці гени ніколи не змінювалися би при розвитку .

Подолання цієї проблеми здійснюють за допомогою мутації дочірніх хромосом (в.5, рис. 2.1) шляхом додавання різноманітності до нащадків і, таким чином, забезпечення можливості вивчення більшої частини пошукового простору. Мутація є вторинним оператором пошуку в генетичних алгоритмах, виконуваним з малою ймовірністю p_m , на відміну від, наприклад, еволюційних стратегій, де мутація є основним оператором пошуку.

Виконуватимемо мутацію так:

1. Згенеруємо випадкове число від 0 до 1.
2. Якщо отримане число є меншим за імовірність мутації, виберемо випадковим чином ген і переспрямуємо його на доступний часовий інтервал. Тут доступний часовий інтервал означає часовий слот, який є вільним у розкладах викладачів, студентських груп і аудиторій водночас.
3. Якщо отримане випадкове число є більшим за імовірність мутації, то зберігаємо хромосому без мутації.

Зауважимо, що як схрещування, так і мутацію будемо застосовувати лише за умов задоволення жорстких обмежень, тобто лише до здійснених хромосом.

Для збереження найкращих хромосом у кожній наступній популяції використаємо просту стратегію заміни (в.7, рис. 2.1). Будемо копіювати кращі 20 хромосом кожної попередньої популяції до наступної популяції. Далі, з 80 хромосом попередньої популяції, що залишилися, виділимо випадковим чином 60 хромосом і замінимо їх нащадками (20 з цих 60 нащадків утворимо з кращих 20 хромосом початкової популяції, а решту 40 потомків - з випадково відібраних 40 хромосом). Ця техніка заміни гарантує, що найкращі хромосоми кожної популяції будуть, як мінімум, дорівнювати найкращим хромосомам попередньої популяції.

Генетичний алгоритм ітераційно породжує нові популяції доти, доки не буде виконана термінальна умова (в.9, рис.2.1). Забезпечимо можливість використання однієї з трьох умов зупинки роботи алгоритму (в.9, рис.2.1):

- хоча б одна з хромосом отримала задане значення пристосованості (1);
- значення пристосованості найкращої хромосоми не змінювалось задану максимальну кількість ітерацій;
- виконана задана максимальна кількість ітерацій (популяцій).

У разі виконання термінальної умови генетичний алгоритм повертає хромосому з найкращим значенням пристосованості (в.10, рис.2.1).

2.2 Модифікація алгоритму імітації відпалу

Метод імітації відпалу є одним з методів локальної оптимізації і особливо гарно працює на невеликих просторах пошуку. Алгоритми локального пошуку використовують функцію сусідства та функцію пристосованості для пошуку локального оптимуму. Щоб уникнути локального оптимуму та знайти рішення, найближче до глобального оптимуму, метод імітації відпалу використовує параметр керування температурою та графік охолодження [21].

Концепція методу імітації відпалу натхненна процесом відпалу кристалічних матеріалів. Кристалічну решітку можна уявити як систему частинок, а її енергетичний стан - сукупністю станів частинок. Частинки переходять з одного енергетичного стану раптом довільним чином, але ймовірність переходів залежить від температури системи. Ймовірність переходу з високоенергетичного стану в низько енергетичний велика при будь-якій температурі, також існує відмінна від нуля ймовірність переходу в стан з більш високим значенням енергії. Ця ймовірність тим вище, чим менше різниця між станами і чим вище температура системи.

Якщо в ролі фізичної системи уявити завдання оптимізації, в якості енергії системи - значення цільової функції $f(x)$, а в ролі частинок – керуючі змінні x , то можна вирішувати задачу оптимізації функції $f(x)$, використовуючи механізми та закони, які визначають процес затвердіння. Необхідно задати закон, за яким буде змінюватися температура системи, закон, за яким буде випадковим чином змінюватися значення координат в просторі рішення, і правило визначення переходу частки в точку з новими координатами. Крім того, потрібно обрати початкову і кінцеву температуру (t_0 і t_f).

При завершенні алгоритму збережене в пам'яті найкраще рішення і буде результатом роботи алгоритму.

Метод імітації відпалу є методом Монте-Карло і для цього потрібно початкове рішення. Таке початкове рішення може бути сформовано випадковим чином або за допомогою рішень, генерованих іншим алгоритмом.

Траєкторія методу імітації відпалу заснована на метаевристиці одного рішення. Це означає, що він працює з одним рішенням, яке модифікується і покращується, доки алгоритм не буде зупинено. Інші алгоритми на основі траєкторії, які завжди шукають найкраще рішення, частіше за все потраплять у місцевий оптимум. Перевага методу імітації відпалу полягає в тому, що пристосованість прийнятого рішення сусіда може як зменшуватися, так і зростати для запобігання попадання у місцевий оптимум.

SA використовує температуру як контрольний параметр, який зменшується і збільшується для імітації охолодження та нагрівання. Охолодження відбувається повільно з виконанням послідовності ітерацій. Може бути реалізоване і нагрівання, яке швидко підвищує температуру за певний час або за певні ітерації. Нагрівання та охолодження визначаються графіком охолодження, який визначає зміну температури з часом, і від якого значною мірою залежить ефективність алгоритму [22].

Графік охолодження визначається такими параметрами:

- початкова температура;
- функція зниження температури;
- кінцева температура / умова зупинки.

Графіки охолодження можуть бути визначені як статичні або динамічні. Графік статичного охолодження не може змінювати параметри під час виконання, а динамічний графік охолодження може адаптувати параметри під час виконання алгоритму.

Рішення з кращим значенням функції пристосованості ніж має поточне рішення, завжди приймаються. Щоб вирішити, чи слід прийняти рішення з гіршим значенням функції пристосованості ніж у поточного розв'язку, використовується функція ймовірності разом з випадковим числом від 0 до 1. Функція ймовірності залежить від поточної температури (T) та значення функції пристосованості (C) і призначена для створення більшої ймовірності для великих T і невеликих відмінностей у пристосованості двох рішень ($C1$ і $C2$).

Функція ймовірності ґрунтується на фізиці відпалу та порівнюється з випадково вибраним значенням від 0 до 1. Рішення приймається, якщо ймовірність більша, ніж випадкове значення

$$P(\Delta C) = e^{\frac{-\Delta C}{T}}$$

Пристосованість розраховується за допомогою функції пристосованості, щоб вирішити, чи є якість рішення кращим чи гіршим, ніж поточне рішення. ΔC – змінення пристосованості між поточним та тестовим рішеннями:

$$\Delta C = C_{\text{поточне}} - C_{\text{тестове}}$$

Воно завжди буде позитивним, оскільки краще рішення, яке зробило б значення ОС негативним, завжди буде прийнято і, таким чином, функція ймовірності навіть не буде використана.

Успіх методу імітації відпалу завдячує відносно невеликому часу виконання, високій якості рішення, простоті в реалізації, застосовності та гнучкості. Проте, навіть якщо реалізація є простою, а гнучкість високою, метод не завжди є тривіальним, і часто зовсім не просто створити ефективний алгоритм вирішення заданої проблеми. Ефективність СА часто дуже сильно залежить від якості функції сусідства та графіка охолодження.

Основними елементами імітації відпалу є:

1. Обмежена множина S .
2. Дійсна цінова функція J , визначена на множині S . Позначимо $S^* \subset S$ як множину глобального мінімуму функції J .
3. Для кожного $i \in S$, множину $S(i) \subset S - \{i\}$ назвемо множиною сусідніх вузлів i .
4. Для кожного i , сукупність додатніх коефіцієнтів q_{ij} , $j \in S(i)$ таких, що $\sum_{j \in S(i)} q_{ij} = 1$. Припустимо, що $j \in S(i)$ тоді і лише тоді, коли $i \in S(j)$.

5. Незростаюча функція $T: N \rightarrow (0, \infty)$, що називається графіком замерзання, а $T(t)$ – температурою в момент t .

6. Початкові умови $x(0) \in S$.

Алгоритм базується на дискретному по часу неоднорідному ланцюзі Маркова $x(t)$, розвиток якого наступний:

1. Якщо поточний стан $x(t)$ еквівалентний i , обираємо сусідній вузол j для i випадковим чином, ймовірність вибору окремого $j \in S(i)$ складає q_{ij} .

2. Наступний стан $x(t + 1)$ визначається наступним чином:

2.1. Якщо $J(j) \leq J(i)$, тоді $x(t + 1) = j$.

2.2. Якщо $J(j) \geq J(i)$, тоді $x(t + 1) = j$ з ймовірністю $\exp[-(J(j) - J(i))/T(t)]$, інакше $x(t + 1) = i$.

Формально, $P[x(t + 1) = j | x(t) = i] = q_{ij} * \exp[-\frac{1}{T(t)} \max \theta, J(j) - J(i)]$, якщо $j \neq i, j \in S(i)$.

Якщо $j \neq i, j \notin S(i)$, $P[x(t + 1) = j | x(t) = i] = 0$.

Обґрунтувати описаний метод можна розглянувши неоднорідний ланцюг Маркова $x_T(t)$, для якого температура $T(t)$ утримується на постійному рівні T . Припустимо, що ланцюг Маркова незвідний і аперіодичний, що $q_{ij} = q_{ji}$ для всіх i, j . Тоді $x_T(t)$ – оборотний ланцюг Маркова з його інваріантною ймовірністю розподілу

$$\pi_T(i) = \frac{1}{Z_T} \exp\left[-\frac{J(i)}{T}\right], i \in S,$$

де Z_T – нормалізуюча константа. При $T \rightarrow 0$, ймовірносний розподіл π_T сконцентрується на множині S^* глобального мінімуму.

Умови збіжності алгоритму сформульовані Хаєком такою теоремою:

Стан i комунікує з S^* з мірою h , якщо існує шлях в S , що починається в i і закінчується у деякому елементі множини S^* де h визначається з умови, що

максимальним значенням функції J на шляху від i до елемента множини S^* було $J(i) + h$. Нехай d^* – це найменше значення міри комунікації кожного i з S^* . Тоді, алгоритм імітації відпалу збігається тоді і лише тоді, коли $\lim_{n \rightarrow 0} T(t) = 0$ і $\sum_{t=1}^{\infty} \exp\left[-\frac{d^*}{T(t)}\right] = \infty$.

Загальна послідовність операцій, виконуваних згідно методу імітації відпалу наведено на рис. 2.5.

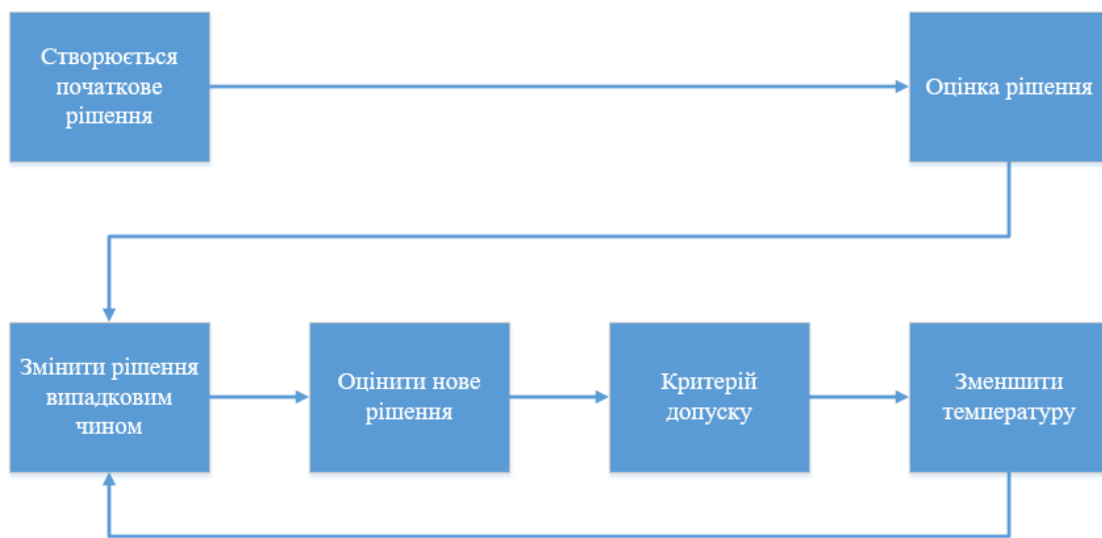


Рисунок 2.5 – Загальна послідовність операцій, виконуваних згідно методу імітації відпалу

Схема алгоритму імітації відпалу для задачі формування розкладу занять навчального закладу наведено на рис. 2.6.

2.3 Розробка гібридного алгоритму формування розкладу начальних занять на основі генетичного алгоритму та методу імітації відпалу

Для застосування методу імітації відпалу до розв'язання задачі формування розкладу навчальних занять, подамо частинки подіями. Події

зв'яжемо з часовими інтервалами для запуску алгоритму. Енергію визначимо функцією пристосованості.

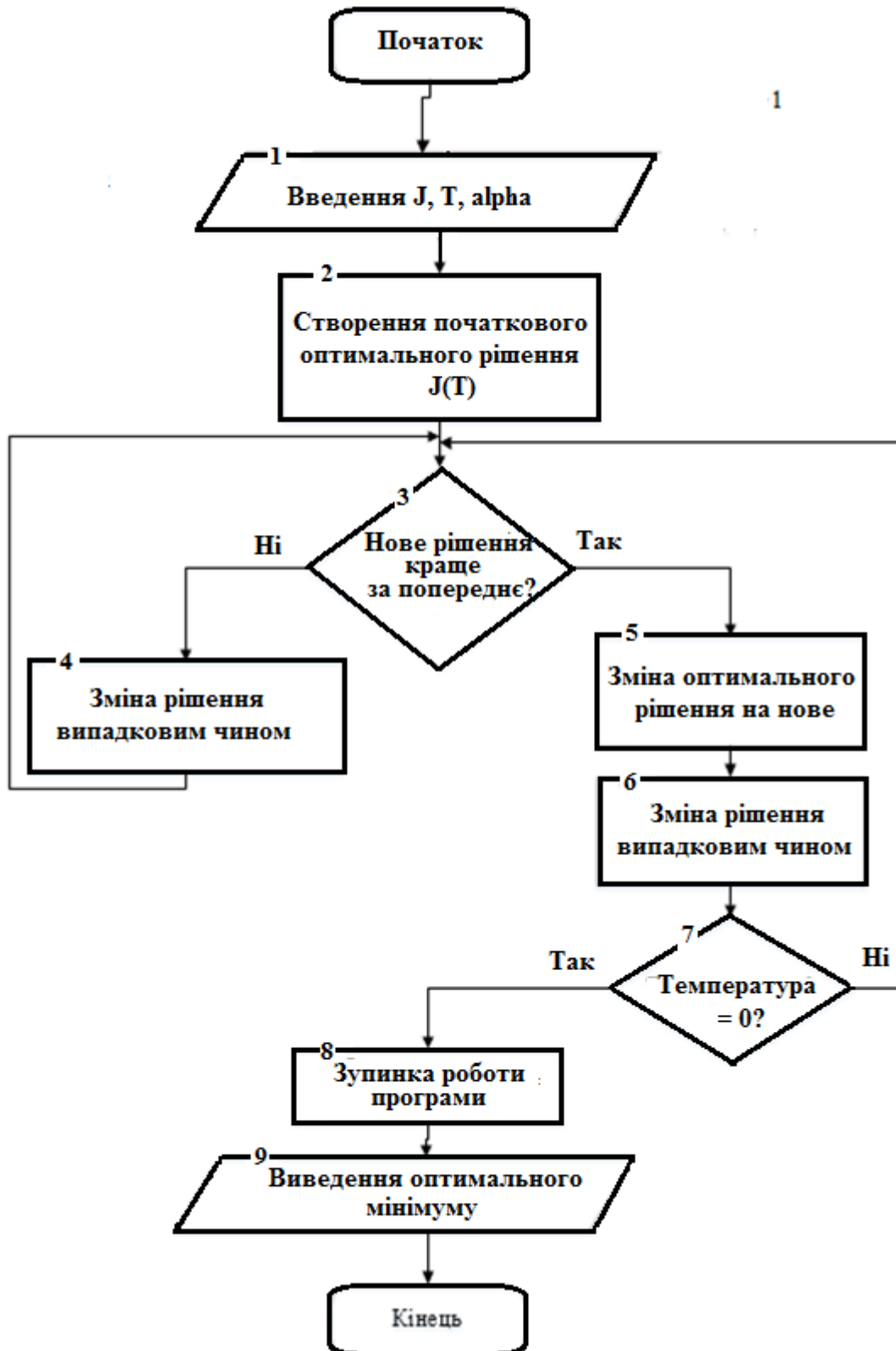


Рисунок 2.6 – Схема алгоритму імітації відпалу для задачі формування розкладу навчальних занять

У [23] наводиться порівняння різних функцій пошуку сусідів для задачі формування розкладу навчальних занять. Досліджувані функції засновувались на обміні, простому пошуку і комбінації обміну з простим пошуком. Простий пошук випадковим чином вибирає одну подію і призначає випадковий часовий інтервал як час початку для події. Обмін випадковим чином вибирає дві дії та обмінює час їх початку. Простий пошук та обмін випадково призначають нові часові інтервали для двох випадкових операцій. Найкращим результатом виявилось поєднання усіх цих трьох стратегій пошуку.

У [24] наведено набір даних для Міжнародного конкурсу навчальних розкладів 2017 року (ІТС-2017), який використовуються багатьма авторами для оцінки алгоритмів формування навчальних розкладів. Написано великі резюме, які порівнюють алгоритми використання цих наборів даних іншими наборами. Набори даних ґрунтуються на алгоритмі, який дозволяє створювати дані різного розміру з відомим ідеальним рішенням .

Сучасні методи формування навчальних розкладів часто включають гібридизацію метаевристичних алгоритмів одного рішення і популяційних метаевристичних алгоритмів [25]. Існує багато різних способів створення гібридних метаевристичних алгоритмів. Деякими прикладами є низькорівневий командний гібридний алгоритм та високорівневий естафетний гібридний алгоритм. Низькорівневі командні гібридні алгоритми вбудовують метаевристику в популяційні метаевристичні алгоритми. Наприклад, у генетичні алгоритми, де генетичні оператори схрещування і мутації можуть бути розширені або замінені на алгоритми локального пошуку. Високорівневі естафетні гібридні алгоритми високого рівня виконують різні алгоритми послідовно. Наприклад, спочатку може бути виконаний жадібний евристичний алгоритм, а знайдене ним рішення буде використано як вхідне для

еволюційного алгоритму після чого алгоритм локального пошуку покращить рішення, створене еволюційним алгоритмом.

У [23] наведені дослідження гібридних алгоритмів формування розкладів навчальних закладів на базі штучної бджолоїної колонії та великого потоку. Вони здійснили значне удосконалення попередніх алгоритмів бджолиних колоній та дослідили отримані алгоритми формування навчальних розкладів на основі пошуку табу та великого потоку на тестових наборах даних ІТС-2007 .

У [25] також досліджено алгоритм формування начального розкладу на основі пошуку табу та великого потоку. Алгоритм показав високі показники пристосованості, але вимагав значного обчислювального часу на середніх тестових випадках. У [24] представлено гібридний алгоритм, що складається з генетичного алгоритму, локального алгоритму пошуку та функції відновлення, яка змінює нездійсненні розклади на прийнятні. Результати порівнянні з найкращими результатами набору даних ІТС-2007.

Гібридні алгоритми, що складаються з GA та SA, були успішно впроваджені для вирішення проблем, відмінних від UCTP. Обидва алгоритми мають різні сильні та слабкі сторони: GA може шукати великий простір пошуку та знайти розумне рішення, але часто не може запропонувати оптимальне рішення в розумний час. SA постійно працює над єдиним рішенням і не шукає достатньо місця для пошуку, щоб отримати огляд усіх перспективних рішень. Коли вони використовуються разом, вони можуть давати багатообіцяючі результати [26]. У [27] також пропонують використовувати GA для пошуку високопродуктивних областей у великому просторі пошуку та алгоритму локального пошуку, щоб знайти оптимальне рішення.

У [26] представлено гібрид GA-SA для вирішення проблеми подорожуючих продавців. Алгоритм використовував чергування між GA і SA, коли алгоритми застрягали. Алгоритм використовував невеликий розмір популяції та багатоступеневий кросовер, що створює багато дітей у порівнянні з розміром населення. Алгоритм давав хороші результати на еталонних наборах даних.

Інший гібрид GA і SA був впроваджений для планування декількох проектів з кількома проблемами обмеження ресурсів, запропонованих компанією [28]. Ця реалізація використовувала зниження частоти мутацій і дозволила трохи погіршити потомство на основі температури та імітації ймовірності відпалу. Алгоритм забезпечив хороші результати протягом розумного часу порівняно з іншими алгоритмами.

2.4 Структура інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму

Розроблений метод формування розкладу начальних занять є високорівневим естафетним гібридним методом і виконує генетичний алгоритм і метод імітації відпалу послідовно, приймаючи від генетичного алгоритму найкращу хромосому як початкове рішення для алгоритму імітації відпалу. Метод вимагає визначення умов зупинки (або точок переривання) при переході від генетичного алгоритму до імітації відпалу.

Досліджено кілька різних варіантів:

- Точка переривання після певного часу роботи генетичного алгоритму.
- Точка переривання генетичного алгоритму за відсутності зміни значення пристосованості за останні X ітерацій.
- Переривання при досягненні генетичним алгоритмом певного значення пристосованості.

Розроблений метод намагатиметься досягти кращої продуктивності ніж окремі генетичний алгоритм або імітація відпалу, завдяки використанню відповідних сильних сторін кожного алгоритму

Структура інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму, детально описана у попередніх підпунктах, зображена на рис. 2.7.

Для роботи інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму спочатку потрібно мати інформацію про

навчальний заклад, яка складається з таких групи об'єктів: а) множина G груп, що навчаються; б) множина A аудиторій; с) множина D дисциплін; д) множина V викладачів; е) множина T навчальних уроків (часових інтервалів проведення занять).



Рисунок 2.7 - Структура інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму

Робота інформаційної технології починається з запуску генетичного алгоритму для пошуку високопродуктивних областей у великому просторі пошуку, після чого використовується алгоритм локального пошуку імітації відпалу для оптимізації знайденого розв'язку.

Таким чином, розроблена структура інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму може бути використана для подальшої розробки програмних засобів.

2.5 Висновок

В даному розділі магістерської кваліфікаційної роботи було обґрунтовано вибір генетичного алгоритму для глобального пошуку квазіоптимального рішення задачі автоматичного формування навчального розкладу, оскільки він є найбільш продуктивним серед інших еволюційних алгоритмів. Запропоновано подання хромосоми, як формалізованого подання можливого розв'язку задачі, зручне з точки зору виконання генетичних операторів, а також перевірки хромосоми на здійсненність. Запропоновано використовувати в якості значення функції пристосованості хромосоми відсоток задоволених рішенням м'яких обмежень, що спрощує процедуру формування нових популяцій у розвитку еволюційного процесу. Обґрунтовано вибір двохетапної оптимізації розв'язку задачі формування навчального розкладу з заборонаю порушення жорстких обмежень на будь-якому етапі еволюції хромосом, що забезпечить подальше спрощення обчислення функції пристосованості за рахунок розгляду нею лише м'яких обмежень. Запропоновано підхід до відбору хромосом для утворення нових поколінь, який дозволяє зберегти найкращий генетичний матеріал і, разом з тим, забезпечити різноманітність пошуку за рахунок репродукції менш пристосованих хромосом. Запропоновані модифікації генетичних операторів схрещування і мутації, які забезпечують задоволення жорстких обмежень при утворенні кожної нової хромосоми. Обґрунтовано доцільність використання методу імітації відпалу для здійснення локального пошуку рішення у визначеній генетичним алгоритмом області простору пошуку, оскільки даний метод дозволяє здійснювати локальний пошук навіть у напрямку погіршення значень функції пристосованості. Запропоновано високорівневий естафетний гібридний алгоритм формування розкладу навчальних занять з узгодженою

роботою генетичного алгоритму і методу імітації відпалу. Розроблено структуру інформаційної технології формування розкладу начальних занять на основі гібридного алгоритму.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ РОЗКЛАДУ НАЧАЛЬНИХ ЗАНЯТЬ

3.1 Обґрунтування вибору мови програмування

Для програмної реалізації інформаційної технології формування розкладу начальних занять оберемо C#, який є об'єктно-орієнтованою мовою програмування. Він був розроблений в 2001 році, інженерами під керівництвом Андерса Хейлсберг в компанії Microsoft. На даний час існує 4 версії мови C#.

До числа принципово важливих рішень, які реалізовані корпорацією Microsoft в мові програмування C#, слід віднести наступні [29]:

- компонентно-орієнтований підхід до програмування (який характерний і для ідеології Microsoft .NET в цілому);
- властивості як засіб інкапсуляції даних (характерно також в цілому для ООП);
- обробка подій (маються розширення, в тому числі в частині обробки винятків, зокрема, оператор try);
- уніфікована система типізації (відповідає ідеології Microsoft .NET в цілому);
- делегати (delegate - розвиток покажчика на функцію в мовах C і C ++);
- індексатори (indexer - оператори індексу для звернення до елементів класу-контейнера);
- перевантажені оператори (розвиток ООП);
- оператор foreach (обробка всіх елементів класів-колекцій, аналог Visual Basic);
- механізми boxing і unboxing для перетворення типів; - атрибути (засіб оперування метаданими в СОМ-моделі);
- прямокутні масиви (набір елементів з доступом за номером індексу і однаковою кількістю стовпців і рядків).

Додаткові переваги C# [30]:

- створювався паралельно з каркасом Framework .Net і в повній мірі враховує всі його можливості - як FCL, так і CLR;

- є повністю об'єктно-орієнтованою мовою, де навіть типи, вбудовані в мову, представлені класами;

- є потужною об'єктною мовою з можливостями успадкування та універсалізації;

- є спадкоємцем мов C/C ++, зберігаючи кращі риси цих популярних мов програмування. Загальний з цими мовами синтаксис, знайомі оператори мови полегшують перехід програмістів від C++ до C#;

Зберігши основні риси свого великого батька, мова стала простіше і надійніше. Простота і надійність, головним чином, пов'язані з тим, що на C# хоча і допускаються, але не заохочуються такі небезпечні властивості C++ як покажчики, адресація, розіменування, адресна арифметика;

Завдяки каркасу Framework .Net, який став надбудовою над операційною системою, програмісти C# отримують ті ж переваги роботи з віртуальною машиною, що і програмісти Java. Ефективність коду навіть підвищується, оскільки виконавче середовище CLR є компілятор проміжної мови, в той час як віртуальна Java-машина є інтерпретатором байт-коду;

Потужна бібліотека каркаса підтримує зручність побудови різних типів додатків на C#, дозволяючи легко будувати Web-служби, інші види компонентів, досить просто зберігати і отримувати інформацію з бази даних і інших сховищ даних;

Реалізація, що поєднує побудова надійного і ефективного коду, є важливим фактором, що сприяє успіху C#.

Виділення і об'єднання кращих ідей сучасних мов програмування робить мову C# не просто сумою їх достоїнств, а мовою програмування нового покоління.

3.2 Розробка структури програмного забезпечення формування розкладу навчальних занять на основі гібридного алгоритму

Здійснимо реалізацію гібридного алгоритму створення оптимізованого розкладу навчальних занять на основі функції пристосованості, загальної для складових алгоритмів: генетичного алгоритму і алгоритму імітацію відпалу.

Розклад складається з подій, призначених для певної аудиторії та часового інтервалу. Усі події вважаються однаковою тривалості та вписуються у один часовий інтервал, і всі дні складаються з однієї і тієї ж кількості часових інтервалів.

Подія визначається дисципліною, курсом, викладачем, групою студентів та діяльністю. Діяльність може бути різних типів: лекції, лабораторії роботи або практичні заняття. Дисципліна складається з певної кількості лекцій, лабораторних робіт і практичних занять. Усі види діяльності мають різні вимоги до приміщення. Наприклад, до обладнання або розміру кімнати. У кімнаті може розміститися максимальна кількість студентів, а також обладнання або засоби, які відповідають вимогам певної діяльності.

Студентська група має визначену кількість студентів, і список дисциплін, які має відвідати група. Практичні заняття і лабораторні роботи часто проводяться в приміщеннях, менших за лекційні аудиторії. Це вимагає такого розподілу події на підподії, щоб кожен студент групи мав змогу її відвідати.

Визначимо структуру даних розкладу як список аудиторій, у кожній з яких є список часових інтервалів для призначення подій. Для певної аудиторії та часового інтервалу можна призначити лише одну подію. Алгоритми мають перевіряти, що усі події завжди заплановані.

Якість розкладу визначається функцією пристосованості. Функція пристосованості розраховує кількість порушень м'яких обмежень з ваговим значенням для різних обмежень. Значення функцією пристосованості використовується як вимірювання якості розкладу, сформованого алгоритмами. Практичне рішення - це рішення, яке не порушує жорстких обмежень.

Алгоритми намагаються максимально збільшити пристосованість. Оцінка пристосованості 0 означає ідеальне рішення.

Реалізація розробленого генетичного алгоритму має забезпечити відбір, двохточкове скрещування та мутацію нащадків з певною ймовірністю. Алгоритм дозволяє існування лише здійснюваних хромосом, які кодують усі заплановані події на різні часові інтервали у різних аудиторіях. Алгоритм відбору будується таким чином, щоб 20 найкращих хромосом попередньої популяції переходили до нової популяції. Хромосома складається з розкладу, і ген складається з події, призначеної для аудиторії та часового інтервалу.

Функціональну схему програмної реалізації генетичного алгоритму наведено на рис. 3.1.

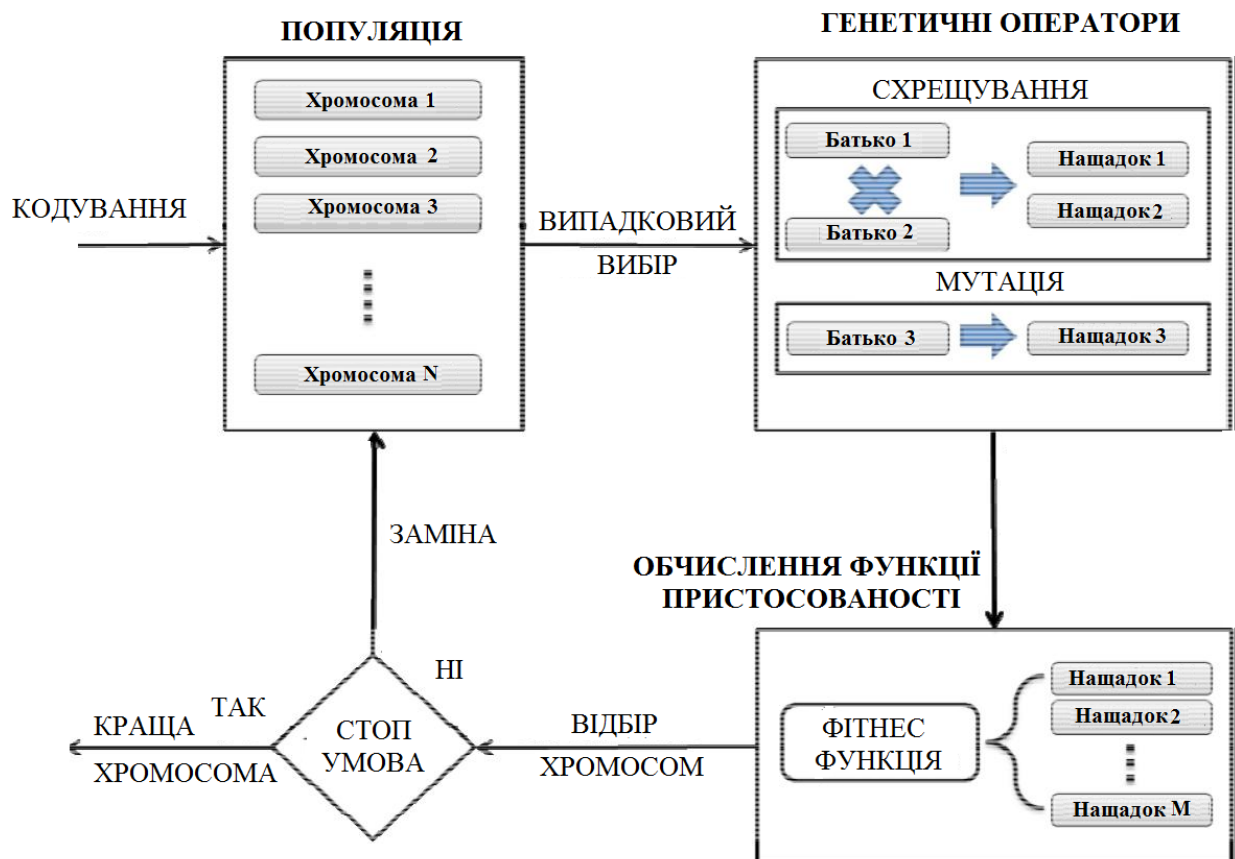


Рисунок 3.1 – Функціональна схема реалізації генетичного алгоритму

Структурну схему програмної реалізації генетичного алгоритму наведено на рисунку 3.2

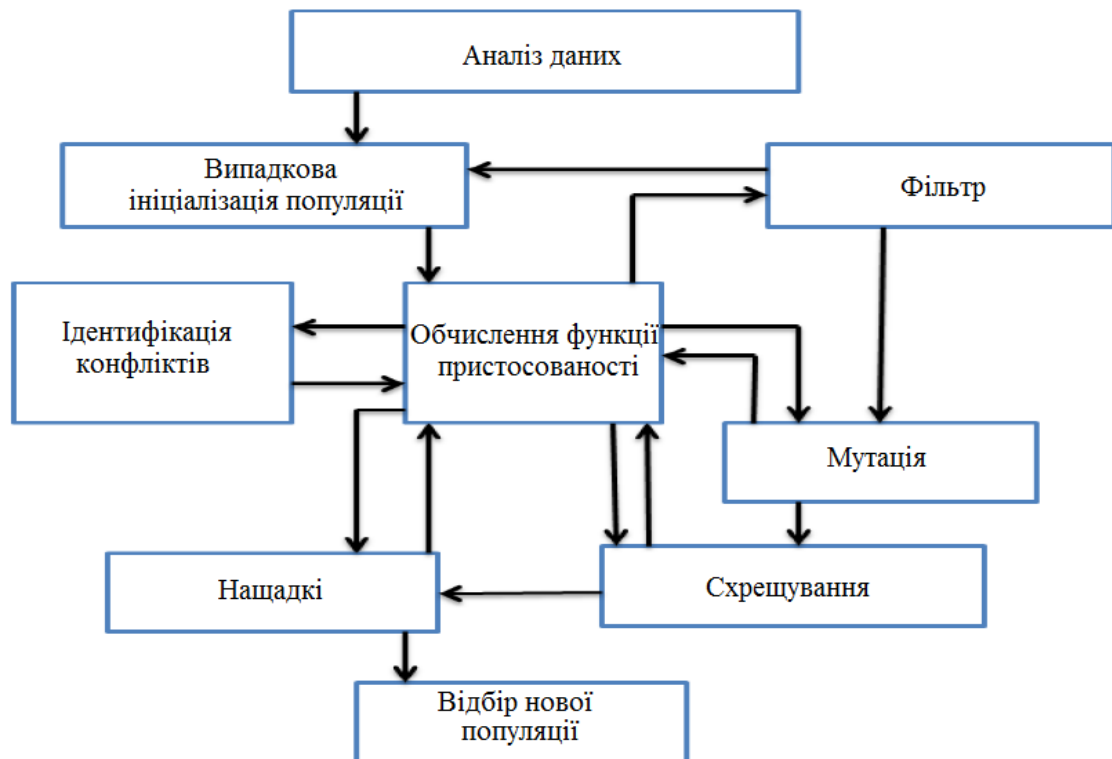


Рисунок 3.2 – Структурна схема програмної реалізації генетичного алгоритму

Реалізація методу імітацій відпалу використовує ті самі структури даних для зберігання розкладу як і генетичний алгоритм статичний графік охолодження. Графік охолодження описується рівнянням

$$T_k = T_0 * e^{\mu * k}$$

де T_k – поточна температура, T_0 - початкова температура, μ - константа, k - кількість ітерацій.

Негативне значення μ гарантує, що температура з часом зменшиться, а ймовірність прийняття гірших рішень зменшується, оскільки метод імітації відпалу наближається до кінця пошуку.

3.3 Розробка програмного забезпечення формування розкладу навчальних занять на основі гібридного алгоритму

Програмне забезпечення формування розкладу навчальних занять розробляється на основі об'єктно-орієнтованого програмування. Діаграму класів генетичного алгоритму наведено на рис. 3.3.

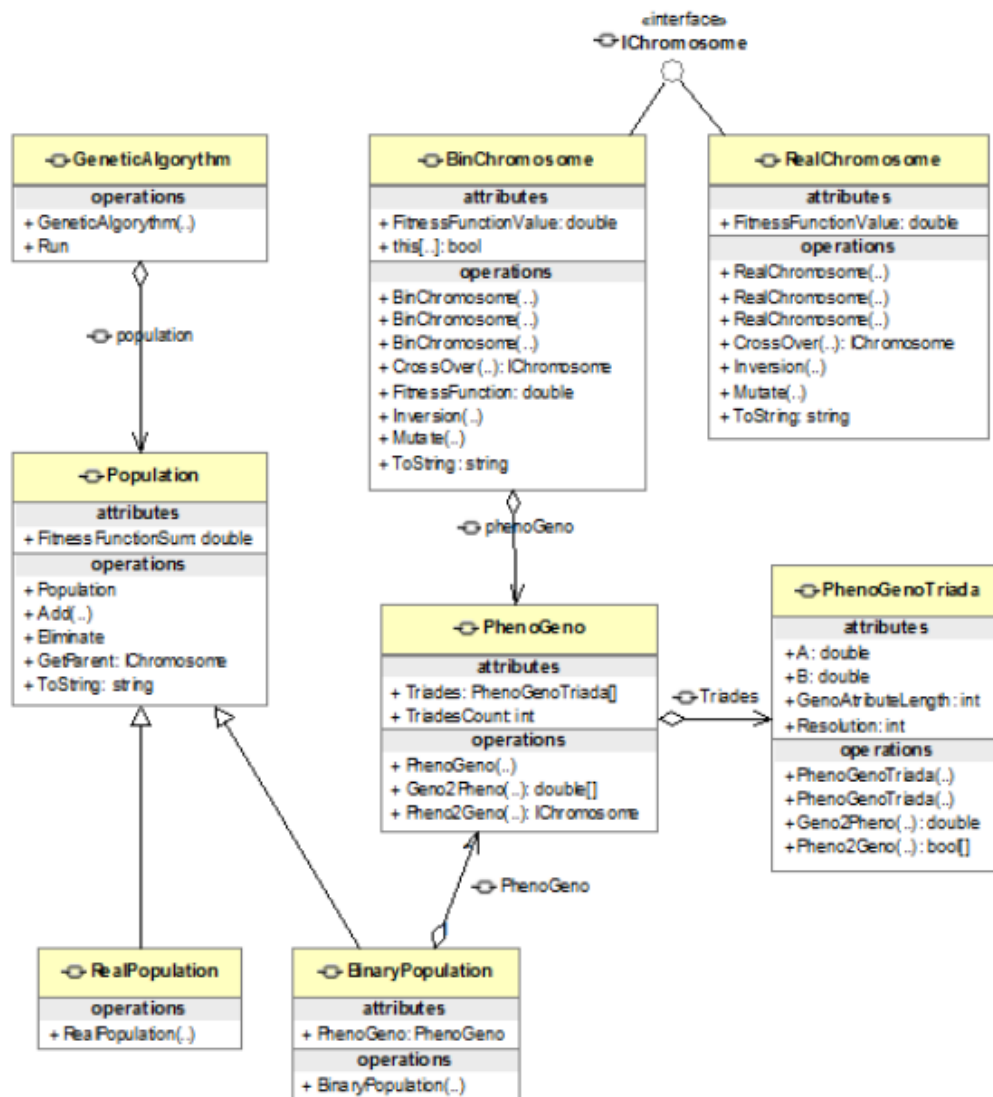


Рисунок 3.3 – Діаграма класів генетичного алгоритму

Первинним типом цієї діаграми є інтерфейс IChromosome, що описує вимоги, що накладаються на програмну реалізацію хромосоми будь-якого типу:

```
public interface IChromosome
```

```

    {FitnessFunctionValue {get; set;} CrossOver (IChromosome chr); Mutate
(double p); // p - ймовірність мутаціїInversion (double p); // p - ймовірність
інверсії
}

```

Клас `BinChromosome` (реалізує інтерфейс `IChromosome`) реалізує «класичну» хромосому, яка використовує бінарне кодування фенотипу.

Закритими змінними цього класу є:

```

protected double fitnessFunctionValue; bool &] alleles; PhenoGeno phenoGeno;
double & phenoAttributes;

```

Змінна `alleles` представляє бінарний масив - генотип хромосоми. Для формування цього масиву в якості першоджерела використовується масив `phenoAttributes` з параметрами фенотипу. Для кодування і декодування хромосома містить спеціальний об'єкт `phenoGeno`.

Це об'єкт класу `PhenoGeno`, який містить в собі масив об'єктів `PhenoGenoTriada`. Кожен такий об'єкт містить діапазон і точність кодування (змінні `a`, `b`, і `resolution`). Використовуючи ці дані і вхідний параметр-фенотип `X` метод `Pheno2Geno` здійснює його перетворення у фрагмент генотипу – бінарний масив. Метод `Geno2Pheno` здійснює зворотне перетворення.

Клас `Geno2Pheno` реалізує стандартні генетичні оператори кросовера, мутації і інверсії.

Клас `BinChromosome` реалізує інтерфейс `IChromosome` для підтримки хромосом з речовим кодуванням. Змінні цього класу:

```

private string id; // ідентифікатор хромосоми
double fitnessFunctionValue;
double [] phenoAttributes; double from; double to; double alpha; // параметр BLX-
кросовера.

```

Для обслуговування ГА зручно визначити службовий клас `Population`, який реалізує корисні функції популяції в цілому. Тут важливо відзначити 2 факти: клас містить множину хромосом в відсортованому в порядку зростання функції пристосованості списку `species`; клас підтримує змінну `fitnessSum`, яка зберігає сумарне значення функцій пристосованості хромосом.

Відповідно до цього метод Add додавання особини в популяції зберігає відсортоване стан списку species і актуальне значення змінної fitnessSum.

Діаграму класів методу імітації відпалу подано на рисунку 3.4.

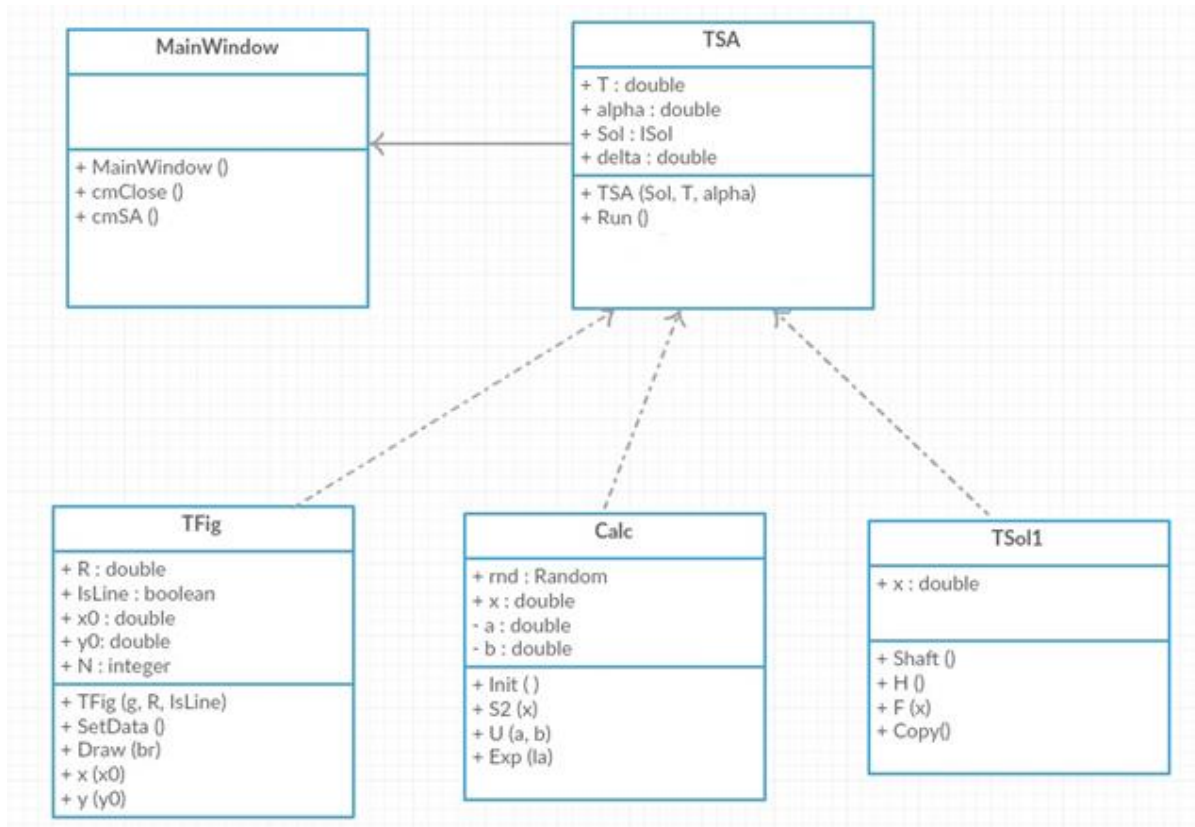


Рисунок 3.4 – Діаграма класів програмної реалізації методу імітації відпалу

Клас «TSA» ініціалізує усі початкові дані, а також звертається до класів «TFig», «Calc», «TSol1» для отримання необхідних даних для безпосереднього обчислення і залучення спеціалізованим математичних операцій.

Клас «TFig» реалізує відображення цільової функції на вказаному проміжку.

Клас «Calc» відповідає за усі математичні операції які залучаються для обчислень методом імітації відпалу.

Клас «TSol1» реалізує виконання алгоритму імітації відпалу на основі введених даних.

3.4 Аналіз результатів тестування програми

Розроблений алгоритм є високорівневим естафетним гібридним алгоритмом і виконує генетичний алгоритм і метод імітації відпалу послідовно, приймаючи від генетичного алгоритму найкращу хромосому як початкове рішення для алгоритму імітації відпалу. Алгоритм вимагає визначення умов зупинки (або точок переривання) при переході від генетичного алгоритму до імітації відпалу.

Досліджено кілька різних варіантів:

- Точка переривання після певного часу роботи генетичного алгоритму.
- Точка переривання генетичного алгоритму за відсутності зміни значення пристосованості за останні X ітерацій.
- Переривання при досягненні генетичним алгоритмом певного значення пристосованості.

Алгоритм намагатиметься досягти кращої продуктивності ніж окремі генетичний алгоритм або імітація відпалу, завдяки використанню відповідних сильних сторін кожного алгоритму

Робота програми починається з запуску генетичного алгоритму для пошуку високопродуктивних областей у великому просторі пошуку, після чого використовується алгоритм локального пошуку імітації відпалу для оптимізації знайденого розв'язку.

Використовуваний набір даних є тим самим, який використовується у [29]. Його характеристики наведені у таблиці 3.1

Розклад складається з п'яти днів і чотирьох часових інтервалів на кожен день. Розмір наборів даних визначається сумою подій та щільністю даних.

Таблиця 3.1 – Характеристики тестового набору даних

Події	Слоти часу	Лекційні аудиторії	Лабораторії	Дисципліни	Викладачі	Академ. групи	Щільність
159	320	3	7	21	15	12	0,5

Щільність може бути визначена рівнянням:

$$SCH = \frac{\text{події}}{\text{аудиторії} \times \text{часові слоти}}$$

Під час тестування випробувані різні параметри, щоб знайти розумні конфігурації параметрів для використання в порівнянні алгоритмів. Випробування проводились на великому наборі даних з обмеженнями за часом та оцінками пристосованості, як умов зупинки. Межу для оцінки пристосованості було встановлено на рівні 50, а часовий ліміт складав до 120 секунд. Очікувані розумні параметри конфігурації для алгоритмів повинні були досягти мети цільової пристосованості до встановленого ліміту. Всі комбінації виконувалися дев'ять разів, щоб врахувати різноманітні результати, пов'язані з імовірнісною евристикою. Завдяки кількості комбінацій параметрів та часу, необхідного для виконання кожної комбінації, більша кількість тестів була практично не можливою.

Параметрами протестованими для генетичного алгоритму були ймовірність мутації, розмір популяції та розмір вибірки, які наведені в таблиці 3.2.

Таблиця 3.2 – Значення протестованих параметрів генетичного алгоритму

Ймовірність мутації, %	4	6	8	10
Розмір популяції	50	100	150	200
Розмір вибірки відносно розміру популяції	0,25	0,5	0,75	1

Були перевірені усі можливі комбінації параметрів. Розмір вибірки був розрахований у відсотках від розміру популяції.

Тести параметрів імітації відпалу були засновані на початкових та кінцевих ймовірностях, наведених у таблиці 3.3, які визначають, коли прийняти гірші рішення.

Параметри для відпалу були розраховані шляхом тестування різних варіантів початкової (P_{start}) та кінцевої (P_{end}) вірогідності методу для вибору гіршого рішення. Значення T_0 та μ розраховувалися за допомогою рівнянь:

$$T_0 = \frac{\Delta fitness}{\ln(P_{start})}$$

$$\mu = \frac{\ln\left(\frac{\Delta fitness}{\ln(P_{end}) * T_0}\right)}{k_{max}}$$

Середнє значення зниження пристосованості спостерігається під час тестування на рівні - 4,8 для м'яких обмежень, а алгоритм повинен був складати не більше 20 000 (k_{max}) ітерацій.

Таблиця 3.3 – Значення протестованих параметрів імітації відпалу

Ймовірність старту, %	0,1	1	5	10
Ймовірність фінішу, %	0,01	0,1	1	5

Були протестовані усі комбінації початкових та кінцевих ймовірностей, де кінцева ймовірність була нижчою, ніж початкова. Параметр повторення було встановлено 10, щоб зменшити час виконання тестів, і тому, що зміна швидкості температури була протестована з різними інтервалами між початковою та кінцевою вірогідністю.

Методи точки переривання (обмеження за часом, цільова пристосованість та обмеження за кількістю незмінних результатів) перевірені окремо з обмеженням часу у 200 секунд.

Ідея тестування полягала у порівняння окремо генетичного алгоритму, імітації відпалу та розробленого гібридного алгоритму в однакових умовах на

однакових тестах з однаковою функцією пристосованості, з однаковим процесом ініціалізації всіх алгоритмів. Ініціалізація здійснюється шляхом присвоєння кожній події випадкового вільного часу. Різниця між алгоритмами ініціалізації полягає в тому, що генетичний і гібридний алгоритми повинні ініціалізувати кожен розклад популяції, тоді як метод відпалу має ініціалізувати лише один розклад як вхідне рішення. Ініціалізація є частиною алгоритмів і, таким чином, буде включена у час виконання.

Щоб зберегти схожі умови для роботи алгоритмів, функції сусідства реалізовані таким чином, щоб задовольняти однаковим жорстким обмеженням. Вони не дозволяють робити подвійне замовлення і переконуються, що всі події є здійсненими.

Вже перше тестування показало, що продуктивність алгоритмів сильно залежить від обраних для них значень параметрів. Тому були перевірені різні конфігурації параметрів, щоб переконатись, що всі алгоритми не будуть працювати настільки погано, наскільки це можливо з погано обраними параметрами. Конфігурації параметрів були оптимізовані під набір даних, який використовувався для порівняння кінцевого алгоритму.

Умови зупинки формування навчального розкладу були визначені значенням 0,5 для пристосованості або тривалістю виконання у 120 секунд, оскільки перші дослідження показали що такого часу вистачає на більшості тестів для досягнення функцією пристосованості значення 0,5.

На рисунках 3.5 -3.7 наведені графіки, що показують точки переходу в процесі пошуку розв'язку від генетичного алгоритму до імітації відпалу і загальний час, витрачений на пошук гібридним алгоритмом.

У найкращому результаті на рис.3.5 умову зупинки досягнуто гібридним алгоритмом за 4 покоління. Причина великого відхилення у 1000 поколінь полягає у імовірнісному характері обох складових гібридного алгоритму, внаслідок чого він не завжди знаходить задовільне вирішення за 200 секунд.

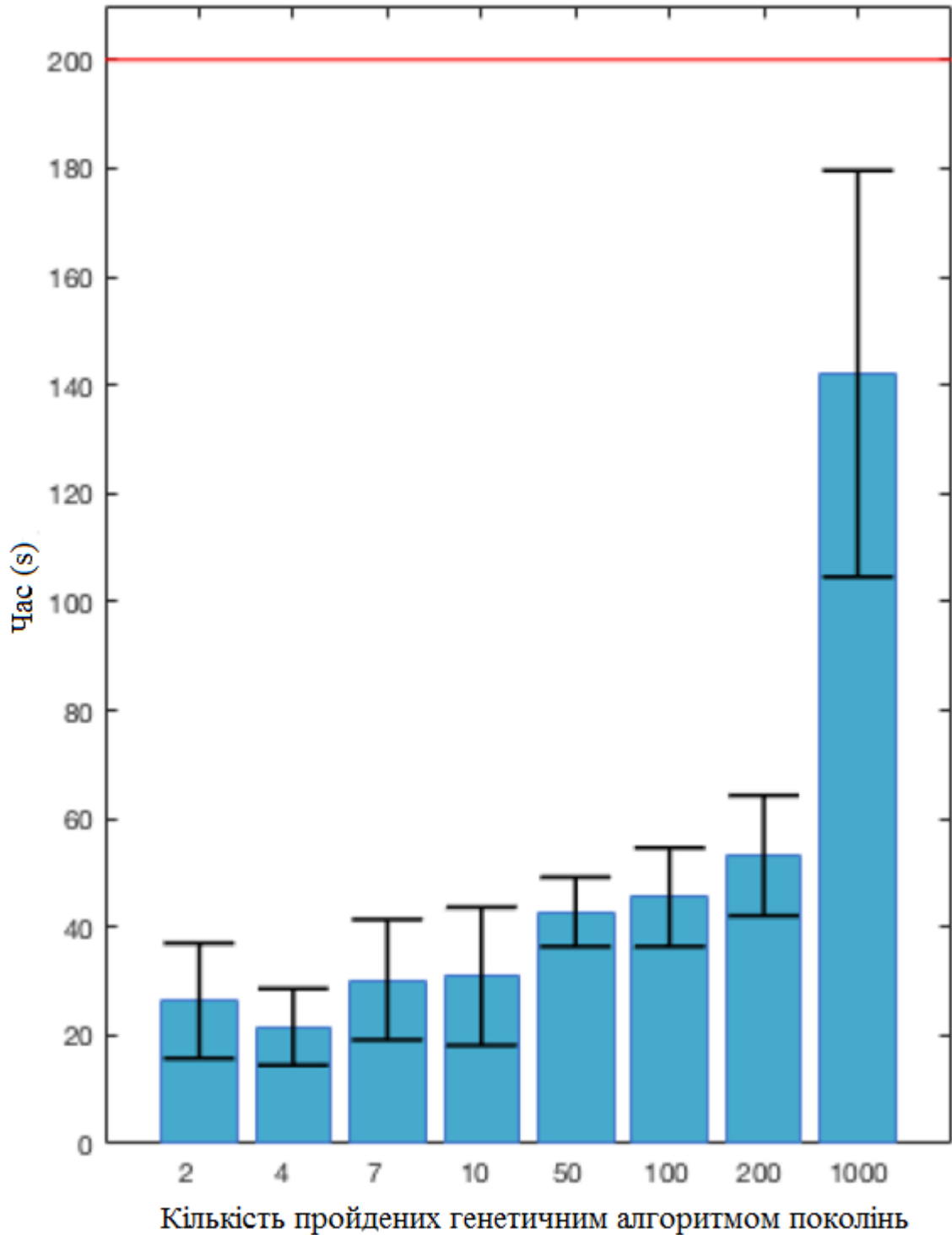


Рисунок 3.5 – Точки переходу від генетичного алгоритму до імітацію відпалу після зміни кількох поколінь хромосом

Рис. 3.6. показує час досягнення гібридним алгоритмом умови зупинки при передачі найкращого рішення від генетичного алгоритму до імітації відпалу, через час, вказаний на осі X.

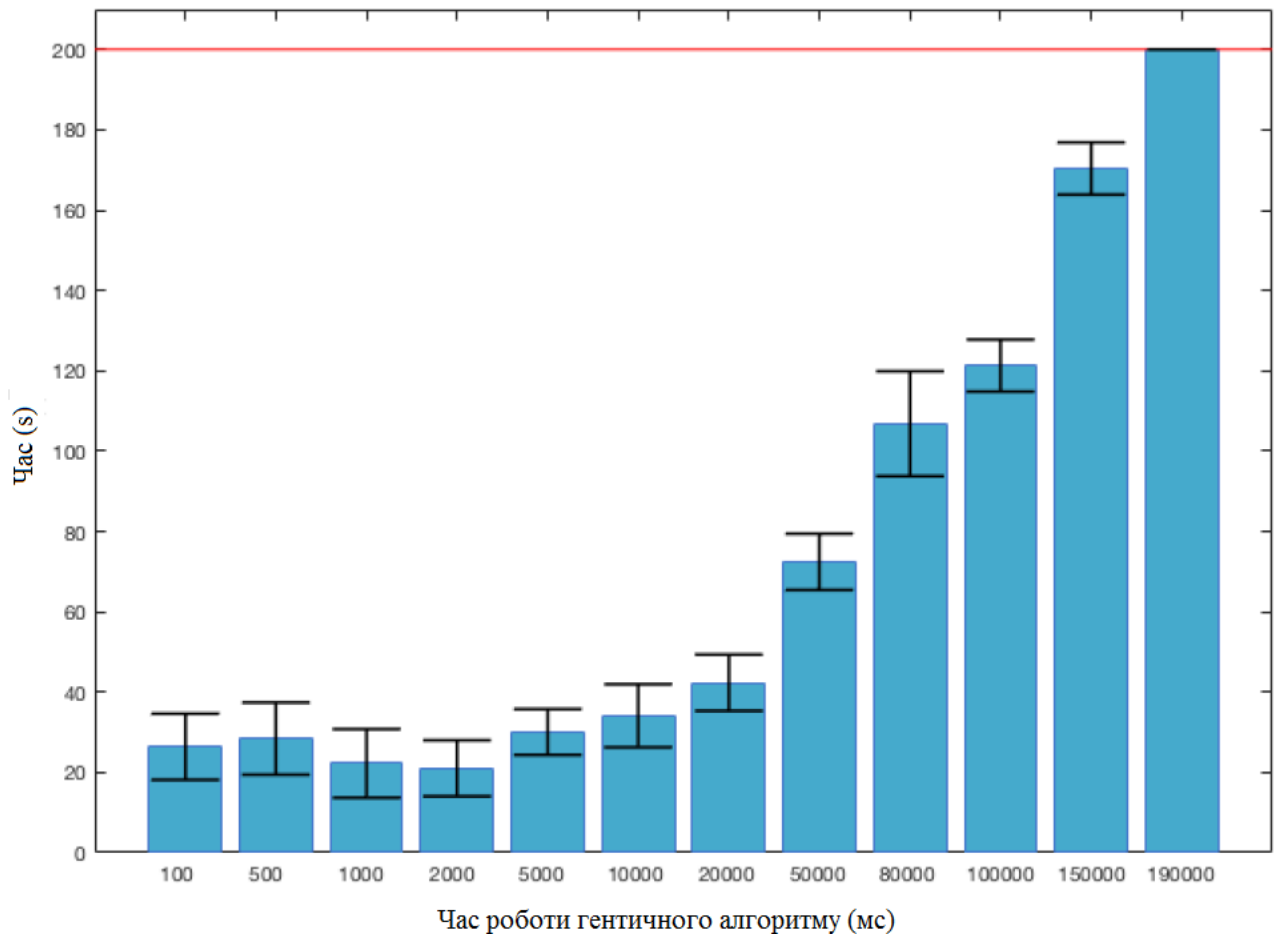


Рисунок 3.6 – Точки переходу до імітації відпалу після певного часу роботи генетичного алгоритму

Рис. 3.7 демонструє час досягнення гібридним алгоритмом умови зупинки при передачі від генетичного алгоритму імітації відпалу рішення з певним значенням пристосованості.

Судячи по рисункам 3.5 -3.7, здається, що гібридний алгоритм має тим кращу продуктивність, чим менше часу займає робота генетичного алгоритму.

Найкраща точка зупинки з кожного тесту дала приблизно такий самий час виконання, як наведено у таблиці 3.4. Для тесту алгоритмів порівняння було обрано час виконання генетичного алгоритму 2 секунди.

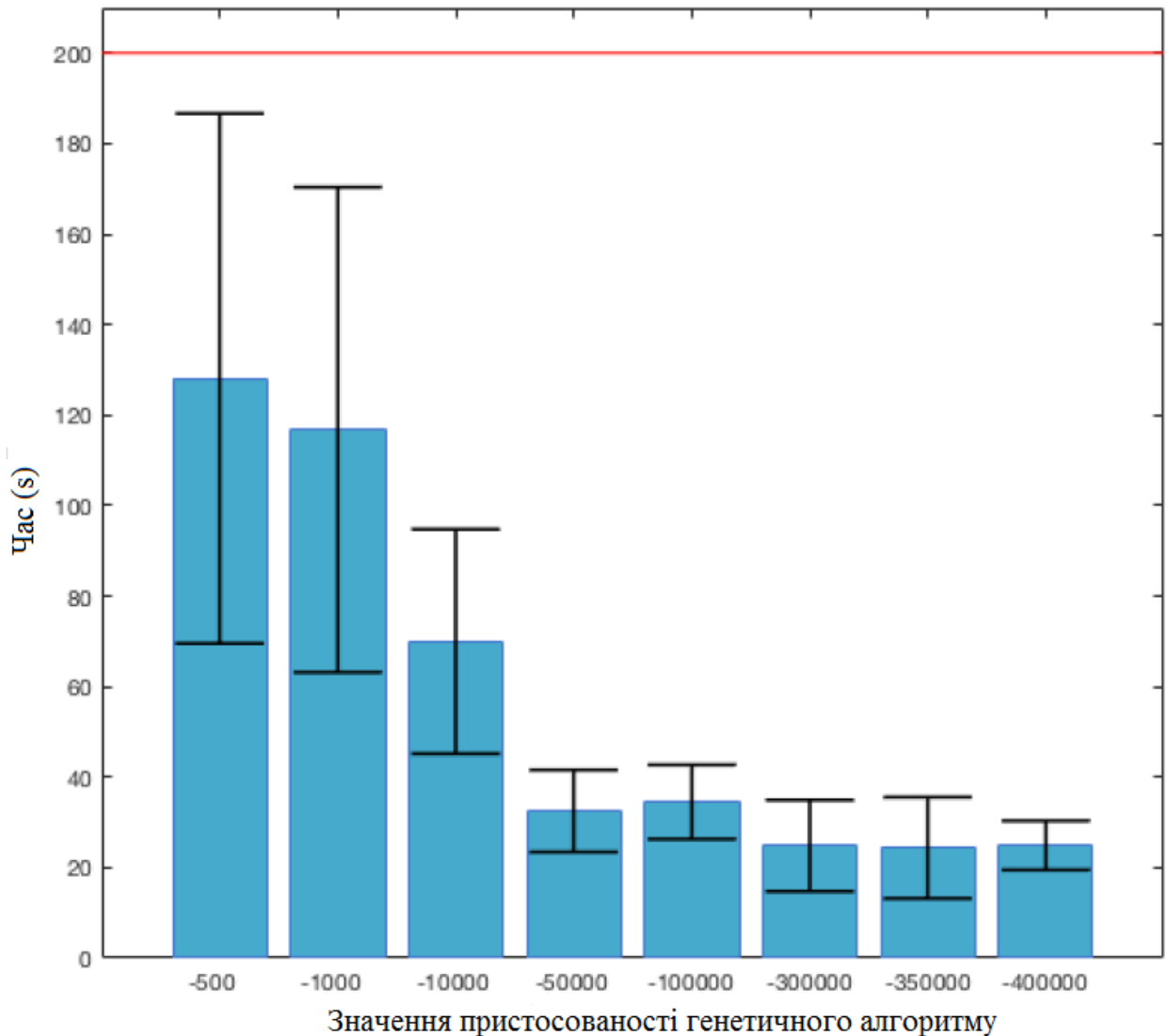


Рисунок 3.7 – Точки переходу до імітації відпалу після досягнення генетичним алгоритмом значення пристосованості у 5%

Таблиця 3.4 – Час виконання гібридного алгоритму

Тест	Параметр	Середній час
Перехід по часу	2000 мс	21015
Перехід по поколінням	4 покоління	21575
Перехід по пристосованості	-350000	24378

Як видно з рис. 3.8. найкращі результати у всіх випадках показав гібридний алгоритм (ГА-ІВ), метод імітації відпалу (ІВ) посідає друге місце, а генетичний алгоритм (ГА) у всіх випадках був останнім. Зауважимо, що вісь Y

на рис. 3.8 є логарифмічною. Графік показує чітку різницю між запропонованим гібридним алгоритмом та двома іншими алгоритмами. Різниця між імітацією відпалу і гібридним алгоритмом становить 2 секунди для всіх рівнів пристосованості. Лише генетичний алгоритм кілька разів не встиг досягти заданого рівня пристосованості за 200 сек.

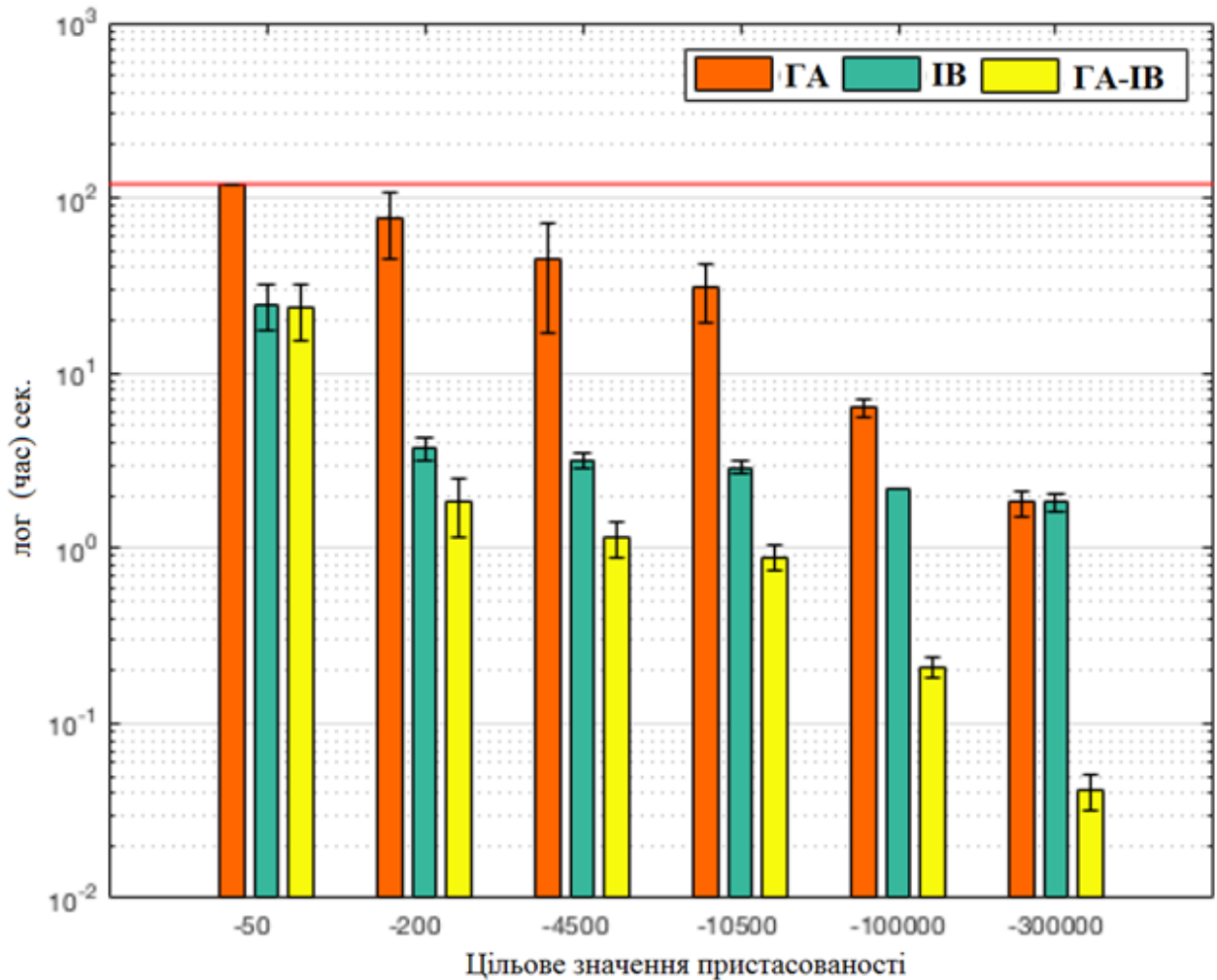


Рисунок 3.8 – Середній час з 50 запусків програми для досягнення певної пристосованості кожним з алгоритмів

Для кількісного аналізу підвищення швидкодії гібридного алгоритму дані з рис. 3.8 занесено у табл. 3.5, де також показано коефіцієнт підвищення швидкодії гібридного алгоритму відносно кращого з двох інших, тобто методу імітації відпалу. Цей коефіцієнт знаходився як відношення часу пошуку

оптимізованого розкладу гібридним алгоритмом до часу пошуку оптимізованого розкладу алгоритмом імітації відпалу.

Таблиця 3.5 – Аналіз швидкодії алгоритмів формування розкладу занять

Алгоритм	Час пошуку оптимізованого розкладу при значенні пристосованості, с					
	-50	-200	-4500	-10500	-100000	-300000
Генетичний	200	80	44	30	6,5	1,9
Імітації відпалу (ІВ)	24	3,9	3,2	2,9	2,1	1,9
Гібридний	23	1,9	1,1	0,9	0,2	0,04
Коефіцієнт збільшення швидкодії запропонованого алгоритму відносно ІВ	1,04	2,05	2,91	3,22	10,05	4,75

Середнє значення коефіцієнту збільшення швидкодії запропонованого алгоритму відносно ІВ буде становити 3,8, тобто запропонований гібридний алгоритм працює в середньому в 3,8 рази швидше алгоритму імітації відпалу. Таким чином, мета роботи досягнута - швидкодія формування розкладу навчальних занять збільшена, в середньому, в 3,8 рази.

3.5 Висновок

В ході практичної реалізації інформаційної технології розглянуто та обґрунтовано переваги та недоліки об'єктно-орієнтованої мови програмування C#, розроблено структурну і функціональну схему програмного забезпечення формування розкладу навчальних занять, які забезпечують узгоджену роботу генетичного алгоритму та імітації відпалу. Визначено основні об'єкти і методи, розроблено діаграми класів генетичного алгоритму і методу імітації відпалу,

створене відповідне програмне забезпечення. Розроблено методику тестування програмного забезпечення, яке з метою проведення досліджень ефективності розробленої програми, передбачає окремі тестування гібридного і генетичного алгоритмів і алгоритму імітації відпалу на тих самих тестах, для забезпечення можливості порівняння отриманих результатів. Аналіз отриманих результатів тестування засвідчив збільшення швидкодії формування розкладу навчальних занять, в середньому, в 3,8 рази.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки [18]. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Колесницький О.К. та Перевозніков С.І.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Колесницький О.К.	2. Перевозніков С.І.
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	3
5	3	4
6	4	4
7	3	3
8	4	4
9	4	3
10	4	3
11	3	4
12	3	4
Сума балів	СБ ₁ = 43	СБ ₂ = 43
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 43$	

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Для розробки нового програмного продукту необхідні такі витрати [18].

Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M - місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 21$ день;

t - число днів роботи розробника, $t = 45$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	5500	261,90	10	2619
Інженер-програміст	4000	190,47	45	8571,15
Всього:				11190,15

Розрахуємо додаткову заробітну плату:

$$Z_{\text{дод}} = 0,1 \cdot 11190,15 = 1119,01 (\text{грн.})$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5...40% від суми їхньої основної та додаткової заробітної плати:

$$H_{\text{зп}} = (Z_o + Z_p) \cdot \frac{\beta}{100}, \quad (4.2)$$

$$H_{\text{зп}} = (11190,15 + 1119,01) \cdot \frac{36,3}{100} = 4468,22 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання (T=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	9000	25	2	375
Всього:				375

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n H_i \cdot C_i \cdot K_i, \quad (4.4)$$

де n – кількість комплектуючих;

N_i – кількість комплектуючих i -го виду;

C_i – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 4.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	180	1	180
Пачка паперу	уп.	130	1	130
Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				352

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi} ; \quad (4.5)$$

де V – вартість 1кВт-години електроенергії ($V=1,7$ грн/кВт);

P – установлена потужність комп'ютера ($P=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=180$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,8$).

$$V_e = 1,7 \cdot 0,6 \cdot 180 \cdot 0,8 = 146,88 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати I_v можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{\text{ін}} = 1 * (11190,15 + 1119,01) = 12309,16 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_0 + Z_d + H_{\text{зп}} + A + K + V_e + I_B$$

$$V = 11190,15 + 1119,01 + 4468,22 + 375 + 352 + 146,88 + 12309,16 = 29960,42 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $V_{\text{заг}}$ за формулою:

$$V_{\text{заг}} = \frac{V_{\text{ін}}}{\alpha} \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$V_{\text{заг}} = \frac{29960,42}{1} = 29960,42$$

Прогнозування загальних витрат ZB на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ZB = \frac{V_{\text{заг}}}{\beta} \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ZB = \frac{29960,42}{0,9} = 33289,35 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}}\Delta N)_i \quad (4.9)$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 20 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 20 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 250 користувачів, протягом другого року – на 200 користувачів, протягом

третього року – 150 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 600 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 300 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 20 \cdot 600 + (300 + 20) \cdot 250 = 92000 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = 20 \cdot 600 + (300 + 20) \cdot (250 + 200) = 156000 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 20 \cdot 600 + (300 + 20) \cdot (250 + 200 + 150) = 204000 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.

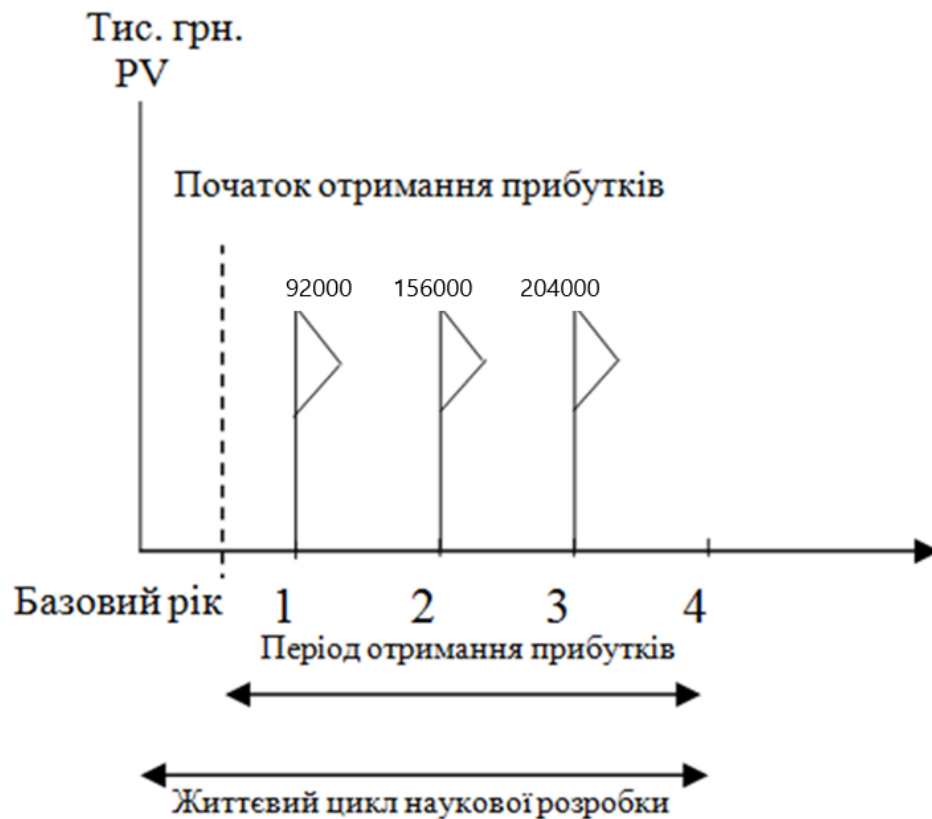


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{33289,35}{(1+0,1)^0} + \frac{92000}{(1+0,1)^2} + \frac{156000}{(1+0,1)^3} + \frac{204000}{(1+0,1)^4} = 365863,9 \text{ (грн.)}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{\text{абс}} = 365863,9 - 33289,35 = 332574,55 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ за формулою:

$$E_{\text{в}} = \sqrt[t]{1 + \frac{E_{\text{абс}}}{\text{PV}}} - 1 \quad (4.12)$$

де $E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $\text{PV} = \text{ЗВ}$, грн;

$T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_{\text{в}} = \sqrt[3]{1 + \frac{332574,55}{33289,35}} - 1 = 1,22 \text{ або } 122 \%$$

Далі, розраховану величина $E_{\text{в}}$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 122\% > \tau_{\min} = 0,3 = 30\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B}$$

$$T_{ок} = \frac{1}{1,22} = 0,81 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновок

В даному розділі було проведено економічне обґрунтування доцільності розробки програми для формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів. Незалежними експертами було здійснено оцінювання комерційного потенціалу розробки, за результатами якого було визначено, що нова розробка має високий рівень комерційного потенціалу, оскільки середньоарифметична сума балів становить 43. Також було виконано прогнозування витрат на виконання розробки, де розраховано основну заробітну плату кожного із розробників, додаткову заробітну плату всіх розробників, нарахування на заробітну плату, амортизацію обладнання, комп'ютерів та приміщень, витрати на допоміжні матеріали, витрати на силову

електроенергію тощо. Загальна сума витрат на виконання означених робіт склала 33289,35 грн. Виконано розрахунок ефективності вкладених інвестицій та періоду їх окупності. Визначено, що абсолютна ефективність вкладених інвестицій становить 365863,9 грн, і це свідчить про те, що вкладання коштів на виконання та впровадження результатів НДДКР є доцільним. Було розраховано відносну (щорічну) ефективність вкладених в наукову розробку інвестицій – 122 %, її величина більша за мінімальну (бар'єрну) ставку дисконтування, отже інвестор буде зацікавлений у фінансуванні даної наукової розробки. Проведено розрахунок терміну окупності - 0,81 року (10 місяців). Це означає, що вже починаючи з 11 місяця розробка буде приносити прибуток. В загальному можна зробити висновок, що фінансування розробки програми для формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів є економічно доцільним проектом.

ВИСНОВКИ

В результаті виконання магістерської кваліфікаційної роботи було розв'язано задачу розробки інтелектуальної інформаційної технології та програмного забезпечення формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів.

У першому розділі магістерської роботи було виявлено високу актуальність задачі автоматичного формування розкладу навчальних занять, як узагальненої проблеми оптимального управління. Було здійснено формалізацію постановки задачі, що дозволило формалізувати групи об'єктів розкладу і основні обмеження, яким він задовольняє. Було здійснено розподіл обмежень на обов'язкові (жорсткі), які визначають коректність розкладу та додаткові (м'які), які визначають його зручність, і кількість задоволення яких може бути використана, як міра якості розкладу. Був проведений аналіз існуючих методів формування розкладу навчальних занять, який засвідчив, що у зв'язку зі значною кількістю обмежень, жоден з них на сьогодні не може претендувати на домінування і гарантувати формування оптимального розкладу. Більше того, деякі дослідники стверджують неможливість створення такого розкладу в повністю автоматичному режимі. Також було обґрунтовано доцільність розробки і дослідження гібридного алгоритму формування навчального розкладу на базі двох (або більше) окремих алгоритмів різної природи, які при сумісному використанні компенсували б недоліки один одного.

У другому розділі магістерської кваліфікаційної роботи було обґрунтовано вибір генетичного алгоритму для глобального пошуку квазіоптимального рішення задачі автоматичного формування навчального розкладу, оскільки він є найбільш продуктивним серед інших еволюційних алгоритмів. Запропоновано подання хромосоми, як формалізованого подання можливого розв'язку задачі, зручне з точки зору виконання генетичних операторів, а також перевірки хромосоми на здійсненність. Запропоновано використовувати в якості значення функції пристосованості хромосоми

відсоток задоволених рішенням м'яких обмежень, що спрощує процедуру формування нових популяцій у розвитку еволюційного процесу. Обґрунтовано вибір двохетапної оптимізації розв'язку задачі формування навчального розкладу з заборорою порушення жорстких обмежень на будь-якому етапі еволюції хромосом, що забезпечить подальше спрощення обчислення функції пристосованості за рахунок розгляду нею лише м'яких обмежень. Запропоновано підхід до відбору хромосом для утворення нових поколінь, який дозволяє зберегти найкращий генетичний матеріал і, разом з тим, забезпечити різноманітність пошуку за рахунок репродукції менш пристосованих хромосом. Запропоновані модифікації генетичних операторів схрещування і мутації, які забезпечують задоволення жорстких обмежень при утворенні кожної нової хромосоми. Обґрунтовано доцільність використання методу імітації відпалу для здійснення локального пошуку рішення у визначеній генетичним алгоритмом області простору пошуку, оскільки даний метод дозволяє здійснювати локальний пошук навіть у напрямку погіршення значень функції пристосованості. Запропоновано високорівневий естафетний гібридний алгоритм формування розкладу навчальних занять з узгодженою роботою генетичного алгоритму і методу імітації відпалу. Розроблено структуру інформаційної технології формування розкладу навчальних занять на основі гібридного алгоритму.

У третьому розділі в ході практичної реалізації інформаційної технології розглянуто та обґрунтовано переваги та недоліки об'єктно-орієнтованої мови програмування C#, розроблено структурну і функціональну схему програмного забезпечення формування розкладу навчальних занять, які забезпечують узгоджену роботу генетичного алгоритму та імітації відпалу. Визначено основні об'єкти і методи, розроблено діаграми класів генетичного алгоритму і методу імітації відпалу, створене відповідне програмне забезпечення. Розроблено методику тестування програмного забезпечення, яке з метою проведення досліджень ефективності розробленої програми, передбачає окремі тестування гібридного і генетичного алгоритмів і алгоритму імітації відпалу на

тих самих тестах, для забезпечення можливості порівняння отриманих результатів. Аналіз отриманих результатів тестування засвідчив збільшення швидкодії формування розкладу навчальних занять, в середньому, в 3,8 рази.

У четвертому розділі було проведено економічне обґрунтування доцільності розробки програми для формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів. Незалежними експертами було здійснено оцінювання комерційного потенціалу розробки, за результатами якого було визначено, що нова розробка має високий рівень комерційного потенціалу, оскільки середньоарифметична сума балів становить 43. Також було виконано прогнозування витрат на виконання розробки, де розраховано основну заробітну плату кожного із розробників, додаткову заробітну плату всіх розробників, нарахування на заробітну плату, амортизацію обладнання, комп'ютерів та приміщень, витрати на допоміжні матеріали, витрати на силову електроенергію тощо. Загальна сума витрат на виконання означених робіт склала 33289,35 грн. Виконано розрахунок ефективності вкладених інвестицій та періоду їх окупності. Визначено, що абсолютна ефективність вкладених інвестицій становить 365863,9 грн, і це свідчить про те, що вкладання коштів на виконання та впровадження результатів НДДКР є доцільним. Було розраховано відносну (щорічну) ефективність вкладених в наукову розробку інвестицій – 122 %, її величина більша за мінімальну (бар'єрну) ставку дисконтування, отже інвестор буде зацікавлений у фінансуванні даної наукової розробки. Проведено розрахунок терміну окупності - 0,81 року (10 місяців). Це означає, що вже починаючи з 11 місяця розробка буде приносити прибуток. В загальному можна зробити висновок, що фінансування розробки програми для формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів є економічно доцільним проектом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Давидюк Р. О., Месюра В. І., Денисов І. К. Інформаційна технологія формування розкладу навчальних занять на основі гібридних метаевристичних алгоритмів // Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи-2020», Вінниця, 2019. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2020/paper/view/8511>. Дата звернення: Грудень 2019.
2. Алексеев С.В. Аналіз методів розв'язку задачі складання розкладу занять / Алексеев С.В., В.О. Мартовицький // Системи обробки інформації – 2014, випуск 2. - с. 84-88. - ISSN 1681-7710.
3. Wren A. Scheduling, Timetabling and Rostering - A special Relationship? / A. Wren. - Lecture Notes in Computer Science. - pp 46-75.
4. Post G. The Third International Timetabling Competition / Gerhard Post, Luca Di Gaspero, Jeffrey H. Kingston, Barry McCollum, Andrea Schaerf//Annals of Operations Research. – Vol.239, No.1. – 2016. – pp. 69-75. – ISSN: 0254-5330 (Print) 1572-9338 (Online)
5. 12th International Conference on the Practice and Theory of Automated Timetabling [Електронний ресурс]. - Режим доступу: <http://www.patatconference.org/patat2018/> (дата звернення 23.11.2019). — Назва з екрану.
6. Каширина И. Л. Моделирование и алгоритмизация задачи составления расписания учебных занятий / И. Л. Каширина, А. Л. Ухин // Вестник ВГУ, Серия: Системный анализ и информационные технологии. – 2015, № 2. - с.66-71
7. H. Babaei. A survey of approaches for university course timetabling problem / Karimpour, J., Hadidi, A.// Computers and Industrial Engineering . - Vol.37, 2015. - pp. 43 - 59.
8. Штовба С.Д. Муравьиные алгоритмы. Exponenta Pro / С.Д. Штовба // Математика в приложениях. – 2003. – № 4. – С. 70-75.

9. Гладков Л.А. Генетические алгоритмы /Л.А. Гладков, В.В. Курейчик, В.М. Курейчик ; под ред.В.М. Курейчика. – 2-е изд., испр. и доп. – М.:ФИЗМАТЛИТ, 2008. – 320 с.
- 10.Бабкина Т.С. Задача составления расписаний: решение на основе мультиагентного подхода / Т.С. Бабкина // Бизнес-информатика. - №1, 2008.
- 11.Андерсон Джеймс А. Дискретная математика и комбинаторика / Джеймс А. Андерсон. – М., 2004. – 960 с.
- 12.Графы и их применение. Комбинаторные алгоритмы для программистов Костюкова Н.И. Интернет- университет информационных технологий. – ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007. – 312 с.
- 13.Рыбников К.А. Введение в комбинаторный анализ / К.А. Рыбников. – М.: Изд-во МГУ, 1985. – 312 с.
- 14.Акулич И.Л. Математическое программирование в примерах и задачах / И.Л. Акулич. – СПб., М.:
- 15.«Лань», 2011. – 185 с.
- 16.Асанов М.О. Дискретная математика: графы матроиды, алгоритмы / М.О. Асанов, В.А. Баранский, В.В. Расин. – Ижевск: НИЦ "РХД", 2001. – 288 с.
- 17.Кафаров В.В. Гибкие автоматизированные производственные системы / В.В. Кафаров. – М.,1990. – 320 с.
- 18.Вентцель Е.С. Исследование операций: задачи, принципы, методология / Е.С. Вентцель. – М.: Наука, Главная редакция физико-математической литературы, 1980.
- 19.Строгалев В.П. Имитационное моделирование / В.П. Строгалев, И.О. Толкачева. – М.: МГТУ им. Баумана, 2008. – С. 697-737.
- 20.Калашников А.В. Параллельный алгоритм имитации отжига для построения многопроцессорных расписаний / А.В. Калашников, В.А. Костенко // Известия РАН. Теория и системы управления. – 2008. – № 3. – С. 101-110.
- 21.Pongcharoen P. Stochastic Optimisation Timetabling Tool for University course scheduling/ Promtet W., Yendaree, P., Hicks,C.// International Journal of Production Economics. – Vol.12, 2014. – hh 19-26/

22. Lewis R. A survey of metaheuristic-based techniques for university timetabling problems / Lewis R. – Springer – Verlag, 2007. pp. 167-190.

23. Костенко В.А. Исследование различных модификаций алгоритмов имитации отжига для решения задачи построения многопроцессорных расписаний / В.А. Костенко, А.В. Калашников // Тр. седьмой междунар.

24. научн. конфер. "Дискретные модели в теории управляющих систем". – М.: МАКС Пресс, 2006.

25. E. Ayca Solving the Course Scheduling Problem Using Simulated Annealing / Ayca, E., Ayav, T // Proceedings of the Advance Computing Conference. – Vol.14, 2009. – pp.462-466.

26. Aarts E. Simulated Annealing. In Burke, E., Kendall, G. (eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques / Aarts E., Kort J., Michiels W. - Springer, 2005. - pp. 187-210

27. Bogdanov D. A Comparative Evaluation of Metaheuristic approaches to the problem of curriculum-Based course timetabling / D. Bogdanov. - Kandidatuppsats, Skolan för teknikvetenskap. Stockholm: KTH. - 2016. – p.136

28. Abramson D. Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms / D. Abramson // Management Science. – Vol.37, 2011. - pp. 98-113.

29. Elhaddad Y. A New Hybrid Genetic and Simulated Annealing Algorithm to Solve the Traveling Salesman Problem / Elhaddad Y., Sallabi O. // Proceeding of the World Congress on Engineering 2010. Vol 7, 2010. – pp. 11-14.

30. Talbi E.-G. A Taxonomy of Hybrid Metaheuristics / E.-G. Talbi // Journal of Heuristics. – Vol.6, 2012. - pp.541-564.

31. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.