

Кемарський Микита Олександрович, здобувач магістерського освітнього рівня

КПІ ім. Ігоря Сікорського, Україна

Головченко Максим Миколайович, старший викладач

КПІ ім. Ігоря Сікорського, Україна

Науковий керівник: Павлов Олександр Анатолійович, доктор технічних наук, професор, професор кафедри інформатики та програмної інженерії

КПІ ім. Ігоря Сікорського, Україна

МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ СКЛАДАННЯ РОЗКЛАДІВ ЗАНЯТЬ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

Анотація. Було спроектовано та розроблено кросплатформну програмну бібліотеку, що ефективно реалізує модифікований генетичний алгоритм для розв'язання нової постановки задачі складання розкладів занять у ВНЗ.

КЛЮЧОВІ СЛОВА: розклад навчальних занять ВНЗ, академічний розклад, крос-платформна програмна бібліотека, генетичний алгоритм, алгоритми оптимізації.

Abstract. We have designed and developed a cross-platform software library that effectively implements a modified genetic algorithm for solving a new formulation of the problem of creating university class schedules.

KEY WORDS: university class schedule, academic timetable, cross-platform software library, genetic algorithm, optimization algorithms.

Вступ. Правильно складений розклад є ключовим елементом успішної навчальної програми. Для ефективного та якісного процесу навчання необхідно забезпечити оптимальне використання часу та ресурсів. Для студентів і викладачів важливо мати розклад, який враховує їхні індивідуальні потреби та дозволяє виконувати всі навчальні завдання вчасно та ефективно. Крім того, складання розкладу навчальних занять може бути витратним та складним процесом, що потребує значних зусиль від адміністративного персоналу ВНЗ. Використання готових програмних засобів може значно полегшити процес складання розкладу, покращити його якість та знизити навантаження на персонал ВНЗ.

Постановка задачі. Розробити кросплатформну програмну бібліотеку, що ефективно реалізує модифікований генетичний алгоритм для розв'язання нової постановки задачі складання розкладів занять у ВНЗ.

В якості вхідних даних при складанні розкладу навчальних занять виступають:

1. Множина $D = \{d_1, d_2, \dots, d_{N_d}\}$ дисциплін, де N_d – кількість дисциплін;
2. Множина $G = \{g_1, g_2, \dots, g_{N_g}\}$ груп, де N_g – кількість груп;
3. Множина $P = \{p_1, p_2, \dots, p_{N_p}\}$ викладачів, де N_p – кількість викладачів;
4. Множина $A = \{a_1, a_2, \dots, a_{N_a}\}$ аудиторій, де N_a – кількість аудиторій;
5. Множина $T = \{t_1, t_2, \dots, t_{N_t}\}$ комірок розкладу, де N_t – кількість комірок розкладу.

Комірка розкладу t_i містить номер заняття, день тижня та номер тижня в циклі.

Опишемо сутності формально [3]. Формально ми можемо описати дисципліни D наступним чином:

$$D = \{d_j\},$$

$$d_j = \begin{pmatrix} d_{maxGroupsPerClass}^j \\ d_{lecturersRequiredPerClass}^j \\ d_{online}^j, d_{classesPerWeek}^j \\ d_{roomTypes}^j \\ d_{faculty}^j, d_{facultyDepartment}^j \end{pmatrix},$$

$$j = \overline{1; N_d}$$

Формально ми можемо описати групи

$$G = \{g_l\},$$

$$g_l = \begin{pmatrix} g_{normativeDisciplines}^l, g_{availability}^l \\ g_{maxClassesPerDay}^l, g_{faculty}^l \\ g_{facultyDepartment}^l \end{pmatrix},$$

$$l = \overline{1; N_g}$$

Формально ми можемо описати викладачів P наступним чином:

$$P = \{p_m\},$$

$$p_m = (p_{availability}^m, p_{disciplines}^m),$$

$$m = \overline{1; N_p}$$

Формально ми можемо описати аудиторії A наступним чином (для задання характеристик аудиторії):

$$A = \{a_n\},$$

$$a_n = \begin{pmatrix} a_{type}^n, a_{building}^n, a_{capacityGroups}^n, a_{faculty}^n \\ a_{facultyDepartment}^n, a_{availability}^n \end{pmatrix},$$

$$n = \overline{1; N_a}$$

Формально ми можемо описати комірки розкладу T наступним чином:

$$T = \{t_k\}, t_k = (t_{week}^k, t_{day}^k, t_{classNumber}^k),$$

$$k = \overline{1; N_t}$$

Також ми можемо формалізувати теоретичну множину Z з об'єктів, що

об'єднують дисципліни, групи, та викладачів:

$Z = \{z_1, z_2, \dots, z_{N_z}\}$, де N_z – кількість елементів множини Z .

$$z_i = \begin{pmatrix} z_P^i, z_d^i, z_G^i, z_{classesPerCycle}^i \\ z_{roomTypes}^i, z_{online}^i, z_{availability}^i \\ z_{faculty}^i, z_{facultyDepartment}^i \end{pmatrix},$$

$$i = \overline{1; N_z},$$

де z_P^i – множина викладачів (іноді потребується > 1 викладача для проведення заняття, $z_P^i \subseteq P$),

z_d^i – дисципліна,
 z_G^i

– множина груп (множина, що складається з однієї групи або груп потоку, $z_G^i \subseteq G$),

$z_{classesPerCycle}^i$

$= N * z_{d_{classesPerWeek}}^i$

– кількість занять на цикл з N тижнів,

$z_{roomTypes}^i = z_{d_{roomTypes}}^i$

– допустимі типи аудиторій

для проведення заняття,

z_{online}^i

$= z_{d_{online}}^i$

– флаг онлайн заняття (якщо $z_{online}^i = 1$, то $z_{roomTypes}^i = \emptyset$),

$z_{availability}^i = \cap_i (z_{Gi} \cap z_{Pi})$

– множина комірок розкладу,

доступних для проведення занять

з дисципліни z_d^i , (перетин доступності викладачів z_P^i та груп z_G^i),

$z_{availability}^i \subseteq T$,

$z_{faculty}^i$

$= z_{d_{faculty}}^i$

– факультет, до якого прив'язано дисципліну,
 $z_{facultyDepartment}^i$
 $= z_{a_{facultyDepartment}}^i$ – кафедра,
до якої прив'язано дисципліну.

При складанні розкладу будемо оперувати трьома множинами об'єктів:

$$Z = \{z_i\}, i = \overline{1; N_z}$$

$$A = \{a_n\}, n = \overline{1; N_a}$$

$$T = \{t_k\}, k = \overline{1; N_k}$$

Таким чином, задачу складання розкладу навчальних занять декомпонуємо [2] на дві підзадачі:

1. Розподілення груп, дисциплін та викладачів у множину Z ;

2. Розподілення елементів множини Z по аудиторіях та комірках розкладу;

Розклад навчальних занять, який необхідно отримати в якості результату, можна представити у вигляді двох двовимірних векторів:

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_{N_z})$$

$$\tau = (\tau_1, \tau_2, \dots, \tau_i, \dots, \tau_{N_z}),$$

де $\alpha_i \in A$,

A - множина всіх аудиторій,

α_i - аудиторія, яку назначено для проведення занять Z_i ,

$$\tau_i = \left\{ \tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_{z_{classesPerCycle}}} \right\}, \tau_{i_j}$$

$\in T$ -вектор з комірками розкладу, які назначено для проведення занять Z_i , $i = \overline{1; N_z}$.

Було складено математичну модель нової постановки задачі складання розкладів занять у ВНЗ та модифіковано [1] генетичний алгоритм для розв'язання даної постановки задачі.

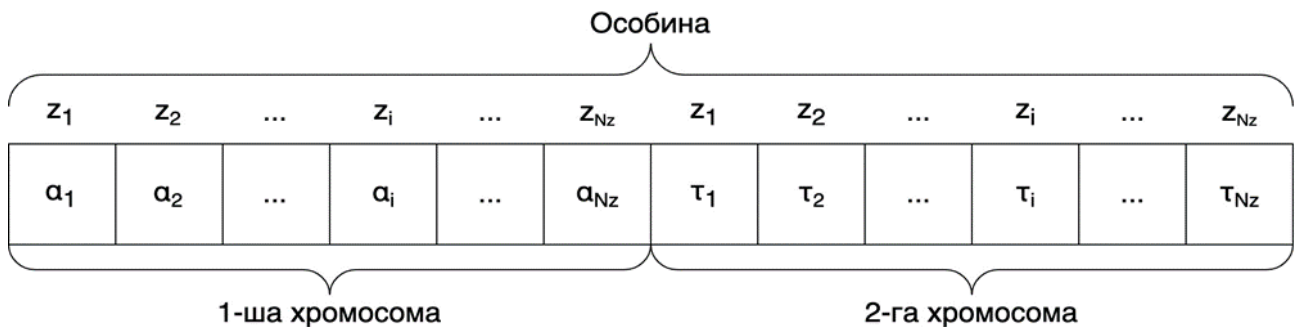


Рис. 1 – Особина генетичного алгоритму з двох хромосом

В модифікованому генетичному алгоритмі кожна особина є одним з можливих рішень задачі, тобто варіантом розкладу. Відповідно до запропонованого представлення розкладу занять у вигляді двох векторів, пропонується розглядати особину, що складається з двох хромосом.

Кожна хромосома, в свою чергу, складається з генів, що позначаються цілими числами $1, 2, \dots, i, \dots, N_z$, причому порядковий номер гену відповідає порядковому номеру елемента множини Z , тобто i -й ген в першій та другій хромосомах характеризує елемент Z_i з множини Z .

Інформаційним наповненням першої хромосоми є аудиторії, що назначено для проведення занять; другої хромосоми – комірки розкладу, що назначено для проведення занять. Таким чином, в першій хромосомі значенням i -го гену є номер аудиторії, що назначена для проведення занять Z_i ; в другій хромосомі значенням i -го гену є вектор з комірками розкладу, які назначено для проведення занять Z_i .

Пошук оптимального розкладу з використанням модифікованого генетичного алгоритму буде відбуватись в декілька етапів. Розглянемо кожен з них.

Створення початкової популяції. Початкова популяція задається кількома професіоналами в складанні розкладів.

Селекція особин. На цьому етапі відбувається відбір найбільш пристосованих особин (варіантів розкладу), тобто тих, які мають найкращі значення фітнес-функції порівняно з іншими особинами. У даній роботі пропонується використовувати метод, відомий як "елітний відбір" або "елітна стратегія", який при вирішенні цього завдання полягає в наступному: з попереднього покоління вибираються задана кількість особин (варіантів розкладів), які мають найменше значення фітнес-функції - критерію, що відображає якість розкладу. Таким чином виявлені "елітні" особини без будь-яких змін переходять до наступного покоління. Решта "вільних місць" у новому поколінні заповнюється особинами, отриманими в результаті схрещування (кросинговера) та мутації особин. Відсотки особин від решти "вільних місць" у новому поколінні, що отримуються в результаті схрещування (кросинговера) та мутації задаються на рівні вхідних даних.

Схрещування (кросинговер). Схрещування особин відбувається за наступною схемою: серед найбільш пристосованих "елітних" особин випадковим чином обираються дві особини. За основу для схрещеної особини береться перша особина. Далі по черзі для кожної i -ї позиції виконується спроба замінити аудиторію α_i першої обраної особини на аудиторію α_i другої обраної особини. Якщо в наявності є достатня кількість доступних комірок розкладу для проведення занять z_i в новій аудиторії, то випадковим чином обираються комірки розкладу з доступних та виставляються в якості значення τ_i . Якщо в наявності немає достатньої кількості доступних комірок розкладу, то заміна аудиторії не відбувається.

За запропонованою схемою схрещування двох особин не відбувається

зміна загальної кількості генів, які відповідають елементам z_i множини Z . Також не змінюється порядок генів всередині кожної хромосоми. Враховуючи той факт, що при виборі аудиторії та комірок розкладу враховувались обмеження задачі оптимізації, після процедури схрещування не вимагається перевірка особин на "правильність" вибору аудиторії та комірок розкладу для проведення занять. Це дозволяє уникнути зайвих перевірок на коректність особин, отриманих у результаті процедури схрещування.

Мутація. Мутація особин відбувається за наступною схемою: по черзі для кожної i -ї позиції з заданою вірогідністю виконується спроба виконання мутації гену α_i , що відповідає за аудиторію проведення занять. Відповідно до вимог задачі оптимізації, з множини аудиторій, що підходять для проведення занять z_i випадковим чином обирається аудиторія a . Якщо в наявності є достатня кількість доступних комірок розкладу для проведення занять z_i в новій аудиторії, то випадковим чином обираються комірки розкладу з доступних та виставляються в якості значення τ_i . Якщо в наявності немає достатньої кількості доступних комірок розкладу, то заміна аудиторії не відбувається.

Перевірка умови зупинки алгоритму. В якості умови зупинки алгоритму пропонується задати фіксовану кількість ітерацій генетичного алгоритму на рівні вхідних даних. Тоді алгоритм буде зупинено автоматично після виконання заданої кількості ітерацій. В якості результату роботи алгоритму виступатиме найкраща особина (варіант розкладу) з популяції.

Розробка кросплатформної програмної бібліотеки. Було розроблено кросплатформну програмну бібліотеку мовою JavaScript з використанням платформи Node.js, що надає інтерфейс для складання розкладів занять у ВНЗ. Зовнішній інтерфейс бібліотеки включає в

себе інтерфейс Scheduler, що відповідає безпосередньо за складання розкладу занять. Розроблена бібліотека також включає в себе

реалізацію GeneticAlgorithmScheduler інтерфейсу Scheduler, що використовується для складання розкладів занять генетичним алгоритмом.

Висновки. Було розроблено кросплатформну програмну бібліотеку, що ефективно реалізовує модифікований генетичний алгоритм для розв'язання нової постановки задачі складання розкладів занять у ВНЗ. Розроблена програмна бібліотека здатна покращити ефективність та якість навчання, покращити результати студентів, знизити навантаження на викладачів та адміністративний персонал ВНЗ. Бібліотека може бути інтегрована в інформаційні системи ВНЗ.

Список інформаційних джерел

1. Леонова М.В. Моделювання задач складання розкладу занять у ВНЗ: огляд та різні підходи до розв'язування / М.В. Леонова // Вісник Запорізького національного університету. Фізико-математичні науки. - 2013. - № 1. - С. 52-59. - Режим доступу: http://nbuv.gov.ua/UJRN/Vznu_mat_2013_1_8.
2. Заневич О., Кухарський В. Про розв'язування задачі складання розкладу занять, використовуючи генетичний алгоритм // Вісник Львівського університету. Серія прикладна математика та інформатика. – 2019. – Вип. 27. – С. 133- 146.
3. Димова, Г. О. (2022). РОЗРОБКА МОДЕЛІ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ МЕТОДОМ ЕВОЛЮЦІЙНОГО ПОШУКУ. Таврійський науковий вісник. Серія: Технічні науки, (2), 3-9. <https://doi.org/10.32851/tnv-tech.2022.2.1>