

ПАРАЛЕЛЬНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ ПОБУДОВИ РОЗКЛАДУ ЗАНЯТЬ

У роботі описана розробка та реалізація паралельного генетичного алгоритму (ГА) побудови розкладу ВНЗ на основі моделі «фермер-робітники» з елементами евристики для вибору аудиторій та пар під час ініціалізації, застосування нового (додаткового) оператора мутації, додання вдалих підрозв'язків в популяцію під час ініціалізації початкової популяції. У алгоритмі введено специфічний спосіб кодування хромосоми, запропоновано функцію оцінки хромосоми з урахуванням низки обмежень, що накладаються на результуючий розклад-хромосому, а також розроблено спеціальні оператори кросинговеру та мутації. На основі експериментів запропоновано оптимальні значення параметрів ГА.

Вступ

Розв'язання задачі побудови розкладу у загальній постановці є процесом виконання деякої фіксованої системи завдань за допомогою певної множини ресурсів чи обслуговуючих пристроїв [1]. Задача складання розкладу занять формулюється так: «Для заданого набору навчальних аудиторій і заданого набору часових інтервалів (навчальних пар) побудувати такий розподіл навчальних занять для всіх об'єктів (вчителів і навчальних груп, аудиторій тощо), для якого обраний критерій оптимальності є якнайкращим» [2]. Серед обмежень, які впливають на ефективність розкладу, виділимо необхідність передування лекцій над практичними, лабораторними та побажання викладачів щодо часу проведення занять. Зауважимо, що це лише деякі з можливих вимог до вдалого розкладу навчального закладу.

Для розв'язання задачі побудови розкладу (ЗПР) виділяють [3] використання методів: повного перебору, гілок та границь, імітації відпалу, логічного програмування з обмеженнями, розфарбування графу та інші. Враховуючи належність ЗПР до класу NP, прийнятною альтернативою її розв'язання є використання еволюційного підходу – генетичних алгоритмів (ГА) з акцентом на дослідження впливу різноманітних параметрів алгоритму на ефективність його роботи.

Класична схема розробки ГА [4] в контексті ЗПР включає наступні етапи [5, 6].

На першому – розробляється структура та представлення хромосоми, в якій

буде зберігатися розв'язок. Обрана структура має враховувати всі особливості, які висуваються до шуканого розкладу, а також те, що від її вибору напряму залежить реалізація алгоритмів кросинговеру та мутації.

Потім створюється початкова популяція, розмір якої залежить від розмірності задачі. Для організації оптимізаційного процесу необхідно створити напрямок розвитку популяції. Цим напрямком може виступати мінімізація цільової функції (функції пристосованості чи здоров'я). Ця функція має адекватно відрізнити два варіанти розкладу. На основі значення оцінки хромосоми відбувається відбір хромосом у батьківський пул, подальше застосування операторів кросинговеру та мутації. Помістивши початкову популяцію у створене штучне середовище і реалізувавши процеси селекції, кросинговеру і мутації, приходимо до ітераційного алгоритму пошуку оптимального розв'язку.

Ми прагнутиmemo побудувати такий розклад занять, за якого порушуватиметься якнайменша кількість заданих обмежень.

У розпаралелюванні програмної реалізації запропонованого алгоритму використовувався потоковий підхід. Аналіз ефективності алгоритму проведено на основі низки експериментів з урахуванням загальноприйнятих критеріїв оцінювання ГА.

Наукова новизна отриманих результатів полягає у запропонованій спе-

цифіці кодування та введених генетичних операторах і сформульована рекомендаціями щодо використання певних конфігурацій параметрів ГА.

Задача побудови розкладу занять в НаУКМА

Для уточнення постановки і розв'язання ЗПР занять оберемо навчальний план факультету інформатики Національного університету «Києво-Могилянська академія» (НаУКМА) з урахуванням специфіки навчального процесу університету. Тобто, необхідно побудувати оптимальний розклад занять для всіх курсів факультету інформатики (двох бакалаврських та чотирьох магістерських програм) з невеликою кількістю вільних аудиторій (17 навчальних аудиторій) та п'ятиденним робочим тижнем з обмеженням у шість пар щодня. Особливістю університету є велика кількість вибіркового курсів вільного запису. Склад груп певного курсу буде залежати від кількості студентів, які записалися на нього, тобто склад і розмір груп є змінним. До інших обмежень віднесемо забезпечення побажань викладачів, мінімізацію «вікон» у розкладі, врахування потреби спеціалізованих аудиторій (комп'ютерних класів). Також доцільно, щоб лекції та практичні з одного предмету стояли поруч, і бажано, щоб лекційні заняття проводилися зранку.

Перелічені специфіки обмежень визначають природність застосування ГА для розв'язання нашої задачі. Наприклад,

на основі генетичного оператора мутації можна просто і адекватно реалізувати мутацію складу груп.

Кодування розв'язків. Для ГА першим важливим кроком є вибір способу кодування розв'язків, тобто представлення хромосоми. Ми обрали представлення хромосом за допомогою цілочисельних значень генів з мультихромосомним типом кодування – кодування трьома хромосомами. Першою виступає хромосома послідовності кодів груп, другою – хромосома послідовності кодів аудиторій, третьою – послідовності кодів пар (рис. 1). Один набір відповідних значень елементів у всіх трьох списках відобразить один мультиген кінцевого розкладу певного курсу. Відповідно, елемент такої мультихромосоми можна представити трійкою «група», «аудиторія», «пара». Довжина мультихромосоми дорівнює кількості груп, для яких будується розклад.

За допомогою структурного елемента «група» кодується наступна інформація: «код групи» та «номер групи». Такій парі елементів ставиться у відповідність одне цілочислове значення. Сам код групи інкапсулює у собі структурні одиниці «курс», «предмет», «тип заняття», «№ тижнів», «викладач», «тип групи», «розмір групи» та «склад групи».

Елемент «курс» представляє навчальний курс (рік навчання), для якого будується розклад, а «предмет» – відповідний предмет, який вивчається цим курсом. Параметр «тип заняття» відповідає лекційним чи практичним заняттям з предмету.

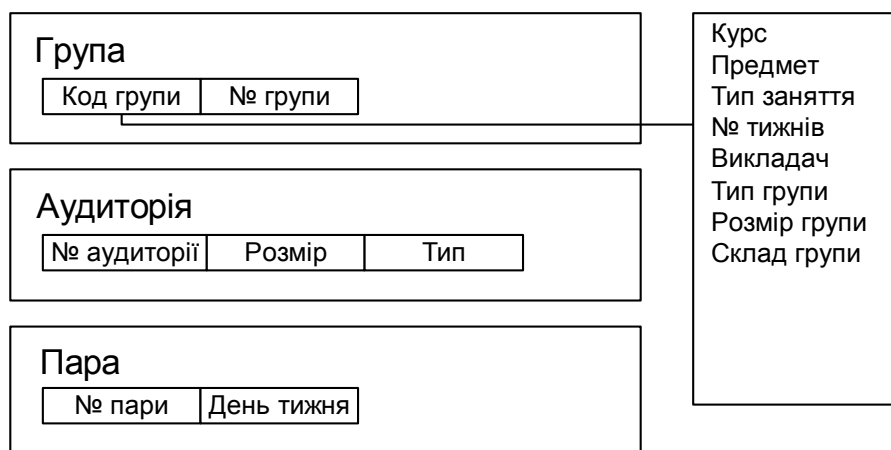


Рис.1. Представлення мультигену мультихромосоми

В залежності від кількості годин, виділених предмету в навчальному плані, він проводиться в певні тижні семестру, за це відповідає структурний елемент «№ тижнів».

Сам структурний елемент «викладач» поділяється на «ПІБ викладача» та «матрицю недоступності викладача», що відображатиме дні та пари, коли викладач не може проводити заняття. Елемент «тип групи» відповідає за те, проста це група чи зведена. Дане поле приймає значення «тип 1», якщо група не належить до зведених груп, або «тип 2», якщо група належить до зведених груп. Для останніх відповідне заняття має проводитися в аудиторії, яка за розміром вміщає всі зведені групи.

Параметр «розмір групи» є допоміжним для реалізації обмеження максимальної кількості студентів в аудиторії та відповідності розміру аудиторії розміру групи.

Параметр «склад групи» потрібний для унеможливлення накладок розкладу для студента. У випадку лекційного заняття з невибіркового предмету склад групи міститиме всіх студентів певного курсу, оскільки невибіркові курси мають вивчати всі студенти. На рис. 2 показано приклад представлення групи.

Елемент «аудиторія» складається з елементів «номер аудиторії», «розмір аудиторії» та «тип аудиторії». «Номер аудиторії» представляє аудиторію з множини можливих аудиторій, поле «розмір

аудиторії» відображає розмір аудиторії, в якій будуть проводитися заняття. Поле «тип аудиторії» визначає тип даної аудиторії, тобто чи вона лекційна, комп'ютерна, чи може використовуватися і для лекційних, і для практичних занять. Представлення аудиторії можна побачити на рис. 3. Відповідно об'єкт аудиторії інкапсулює у собі зазначені структурні елементи.

Структурний елемент «пара» містить у собі наступні параметри: «номер пари» та «день тижня». Відповідно «номер пари» відображає номер пари в розкладі. Структурний елемент «день тижня» показує день проведення пари. На рис. 4 показано представлення пари.

Особина кодується трьома хромосомами, де в першій хромосомі зберігаються гени всіх груп у певному порядку, в другій хромосомі зберігається список аудиторій, відповідних цим групам, а в третій хромосомі закодовані пари, які відповідають певним групам. Така структура кодування була обрана для легкості застосування операторів кросинговеру та мутацій над кожним списком (груп, аудиторій, пар) окремо. Адже використання класичних операторів кросинговеру комбінаторної оптимізації через перестановки генів одного спільного списку могло б суттєво порушити структуру хромосоми, тобто відповідність «група, аудиторія, пара». На рис. 5 показано цю структуру.

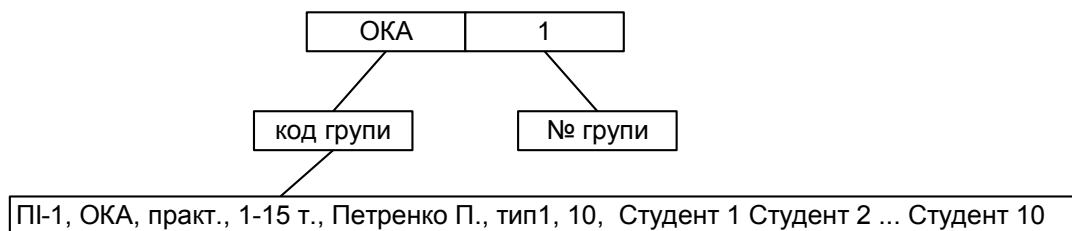


Рис. 2. Представлення групи

1-223	80	лекційна
-------	----	----------

Рис. 3. Представлення аудиторії

ПН	1 пара
----	--------

Рис. 4. Представлення пари (заняття)

Група	Ген 1,1	Ген 1,2	...	Ген 1, <i>n</i>
Аудиторія	Ген 2,1	Ген 2,2	...	Ген 2, <i>n</i>
Пара	Ген 3,1	Ген 3,2	...	Ген 3, <i>n</i>

Рис. 5. Представлення мультихромосоми

Визначення обмежень, що накладаються на розв'язок. Алгоритм потребує встановлення обмежень при побудові та оцінці хромосоми. Такі обмеження дозволять відсіяти завідомо неправильні та неоптимальні можливі розв'язки поставленої задачі. Обмеження можуть бути жорсткими та нежорсткими. Жорсткі обмеження показують недопустимість, а нежорсткі – небажаність розв'язку.

До жорстких обмежень віднесемо наступні: у варіанті розв'язку мають бути представлені всі групи, для яких відбувалася побудова розкладу; викладач не може проводити практичне заняття на одній парі більш ніж в одній групі, допускається проведення лекційних занять з предмету, для якого сформовані зведені групи; аудиторія не може бути зайнята для проведення занять двома різними групами одночасно, окрім випадку проведення лекцій з предмету, для якого сформовані зведені групи; розмір аудиторії має бути більшим чи рівним розміру групи; забезпечення відповідності аудиторії типу заняття; забезпечення уникнення накладок у розкладі для груп з не попарно незалежними списками студентів на певній парі одночасно (визначимо *попарно незалежні* групи як групи, порядковий номер яких різний і склад студентів яких не перетинається). Розв'язок-індивід вважається недопустимим, якщо в задованому ним розкладі порушуються жорсткі обмеження.

До нежорстких обмежень належать наступні: забезпечення побажань викладачів; якнайменше вікон у робочих планах викладачів; мінімізація вікон в індивідуальних планах студентів (зрозуміло, що це

обмеження матиме невелику вартість, адже більше переваг надається викладачам); проведення лекційних занять зранку (бажано до третьої пари включно); забезпечення порядку проведення пар для певного предмету: лекція-практика; лекційні та практичні заняття з одного предмету стоять у розкладі поруч та бажано у тих же аудиторіях.

Беручи до уваги згадані вище обмеження можна побудувати хромосому адекватну вимогам нашої задачі.

Функція здоров'я має враховувати визначені обмеження, і її значення має покращуватися у випадку знаходження кращого варіанта розкладу. Ми обрали наступну функцію здоров'я:

$$f(x) = \sum_{i=1}^n w_i c_i,$$

де $w_i c_i$ – добуток кількості порушень i -того обмеження w_i на вартість i -того обмеження c_i . Покращення у цьому випадку визначатиметься шляхом мінімізації значення коефіцієнта здоров'я, а найкращим розв'язком буде особина зі значенням функції здоров'я, рівним нулю. Для жорстких обмежень встановлюється вища вартість обмеження у порівнянні з нежорсткими, що дає нам можливість отримувати кращі варіанти розв'язків, коригуючи значення вартостей обмежень.

Оператори відбору особин у батьківський пул. Ми використовували пропорційний відбір (а саме відбір за методом рулетки), турнірний відбір та елітарний відбір.

Оператори схрещування. Оператор кросинговеру або схрещування (crossover operator) – це оператор, за допомогою якого реалізується обмін генетичною інформацією між особинами популяції шляхом рекомбінації частин хромосом-батьків для утворення нащадків.

Закодована хромосома груп побудована так, що вона містить коди груп, які не повторюються (у випадку, якщо в розкладі є предмет, пари з якого мають проводитися двічі на тиждень, то для нього формуватимуться дві різні групи, відповідно для кожного заняття з даного предмету). Тому застосування класичних операторів кросинговеру може порушити структуру хромосоми, адже вони не забезпечують виконання обмеження, щоб жодне значення з заданої множини кодованих груп не повторювалося більше одного разу, та не забезпечують присутність всіх значень множини кодованих груп в хромосомі. Тому для кросинговеру хромосом груп застосовувалися наступні оператори схрещування: кросинговер часткового відображення (PMX-кросинговер) та упорядкований кросинговер з однією точкою схрещування (OX-кросинговер). Закодовані хромосоми аудиторії та пар можуть містити повтори значень, так як

множина можливих аудиторій та пар є обмеженою. Відповідно до цього обмеження у роботі для реалізації схрещування хромосом аудиторій та пар були застосовані двоточковий та однорідний кросинговери.

Розглянемо алгоритми виконання цих операторів.

За двоточкового кросинговеру (two-point crossover) випадковим чином генерується дві точки перехрещування, та індивіди обмінюються ділянками між ними. Наступний приклад (рис. 6) демонструє цю роботу.

Нехай у нас є дві хромосоми аудиторій довжини 10, що кодуються двозначними цілими числами (відповідно, ціле число буде представляти код закодованого об'єкта аудиторії, сам об'єкт, як уже зазначалося, в кодуванні складається з номеру, типу та розміру аудиторії), що можуть повторюватися. Відповідне ціле число кодує одну аудиторію. Для схрещування обираються точки 2 та 5.

У випадку однорідного кросинговеру (uniform crossover) кожен з генів хромосоми першого батька має 50 % шансів обмінятися місцями з відповідним йому геном хромосоми другого батька (рис. 7).

Батьки:	11	15	16	13	17	12	11	18	16	15
	17	13	18	12	11	18	15	16	17	13

Нашадки:	11	15	18	12	11	12	11	18	16	15
	17	13	16	13	17	18	15	16	17	13

Рис. 6. Двоточковий кросинговер

Батьки:	11	15	16	13	17	12	11	18	16	15
	17	13	18	12	11	18	15	16	17	13

Нашадки:	11	13	16	12	11	12	11	16	16	13
	17	15	18	13	17	18	15	18	17	15

Рис. 7. Однорідний кросинговер

Роботу кросинговеру часткового відображення, або РМХ-кросинговеру (Partially-Mapped Crossover), можна описати в два кроки [4]. Спочатку випадковим чином обираються дві точки схрещування та відбувається обмін генетичним матеріалом між цими точками. Потім відбувається усунення дублів. Для цього визначається відображення відповідно до значень генів, якими обмінялися хромосоми. Розглянемо приклад роботи РМХ кросинговеру на прикладі 2-х хромосом груп довжини 9, точки схрещування 3 та 7 (рис. 8).

Як видно з прикладу, після обміну генетичним матеріалом хромосоми стають некоректними, тому будуються відображення 101 -> 104; 108 ->105 для першого протонащадка та 104 -> 101; 105 ->108 для другого протонащадка. Після усунення дублів отримуємо допустимі хромосоми.

Розглянемо роботу упорядкованого ОХ-кросинговеру (Order Crossover) з однією точкою схрещування. На початку

його роботи випадковим чином обирається певна точка схрещування. Далі значення генів першого батька до цієї точки копіюються в першого нащадка без змін. А значення генів цього нащадка, що стоять після точки схрещування, заповнюються значеннями генів другого батька, але тими що ще не потрапили в генотип даного нащадка, причому вони записуються в тому порядку, в якому зустрічаються в генотипі другого батька [4]. Відповідно таким же чином формується другий нащадок. Роботу ОХ-кросинговеру відображає наступний приклад, в якому теж відбувається схрещування хромосом груп довжини 9, точка схрещування 4 (рис. 9).

Розглядалися два випадки застосування цих кросинговерів. У першому випадку кросинговер застосовувався з певною ймовірністю лише до одного списку генів особини (тобто груп, аудиторій чи пар). В другому випадку оператори схрещування застосовувалися до усіх списків, тобто і груп, і аудиторій, і пар.

Батьки:

101	102	103	104	105	106	107	108	109
104	105	102	101	108	107	106	109	103

Протонащадки:

101	102	103	101	108	107	106	108	109
104	105	102	104	105	106	107	109	103

Нашадки:

104	102	103	101	108	107	106	105	109
101	108	102	104	105	106	107	109	103

Рис. 8. РМХ-кросинговер

Батьки:

101	102	103	104	105	106	107	108	109
104	105	102	101	108	107	106	109	103

Нашадки:

101	102	103	104	105	108	107	106	109
104	105	102	101	103	106	107	108	109

Рис. 9. ОХ-кросинговер

Оператори мутації. Оператор мутації в генетичних алгоритмах відображає природний процес мутації. Його застосування дозволяє запобігати передчасній збіжності популяції. У роботі розглянуто наступні типи мутацій: мутація обмінами та мутація вставкою. Ці типи мутації можуть бути застосовані і до хромосом груп, і до хромосом аудиторій та пар.

У випадку мутації обмінами (Exchange Mutation) дві випадково обрані позиції хромосоми обмінюються значеннями генів. Кількість обмінів може бути один чи більше і є фіксованою. Розглянемо роботу мутації на прикладі хромосоми довжиною 10, в якому відбувається обмін між 3 і 7 позиціями генів (рис. 10).

При мутації вставкою (Insertion Mutation) вибирається певна позиція в хромосомі і значення гену в цій позиції переноситься у випадкову обрану нову позицію. Розглянемо роботу даної мутації на прикладі, в якому відбувається вставка елемента з сьомої в третю позицію хромосоми (рис. 11).

Оператор мутації, як і оператор кросинговеру, теж розглядався в двох випадках: з застосуванням оператора до всіх трьох списків особини (груп, аудиторій, пар) чи з застосуванням оператора мутації лише до одного із списків.

Умови завершення роботи генетичного алгоритму. Для генетичного алгоритму розв'язання задачі побудови розкладу виділимо три основні умови зупинки:

- знаходження розв'язку зі значенням коефіцієнта пристосованості, рівним нулю:

відповідно цей розв'язок не порушує жодне обмеження;

- досягнення максимальної кількості поколінь, заданих для роботи алгоритму;
- збіжність популяції, тобто середнє значення функції здоров'я не покращується більше ніж на задане число певну задану кількість ітерацій.

При досяганні однієї з вище наведених умов, алгоритм завершуватиметься та виводитиме результат своєї роботи, тобто особину з найкращим рівнем пристосованості (найближчим до нуля).

Застосування евристик для вибору аудиторій та пар під час ініціалізації. При ініціалізації особин спочатку генерується випадкова послідовність груп, далі аудиторій і в кінці пар. У випадку вибору значень аудиторій, можна запропонувати наступні евристики.

По-перше, під час ініціалізації особин використовувати ітеративний метод побудови списку аудиторій. На етапі ініціалізації індивідів спочатку відбувається ініціалізація списку груп. Коли відбувається перехід до ініціалізації списку аудиторій, то на кожній ітерації будемо виконувати перевірку, чи не порушує наступна бажана для додання в список аудиторія обмеження того, що в одній аудиторії не може проводитися одночасно заняття в двох групах (окрім зведених груп). Відповідно, якщо випадковим чином обрана аудиторія порушує це обмеження, то необхідно ще раз генерувати випадкову аудиторію зі списку аудиторій, попередньо виключивши з нього дану, завідомо некоректну аудиторію для цієї позиції у списку.

Батько:

11	14	13	19	22	10	15	18	21	20
----	----	----	----	----	----	----	----	----	----

Нащадок:

11	14	15	19	22	10	13	18	21	20
----	----	----	----	----	----	----	----	----	----

Рис. 10. Мутація обмінами

Батько:

11	14	15	19	22	10	13	18	21	20
----	----	----	----	----	----	----	----	----	----

Нащадок:

11	14	13	15	19	22	10	18	21	20
----	----	----	----	----	----	----	----	----	----

Рис. 11. Мутація вставкою

По-друге, під час ініціалізації також можна забезпечувати вибір аудиторій для лекцій з аудиторій більшого розміру, а для практичних – з аудиторій меншого розміру. Тому варто проводити переініціалізацію аудиторії, поки не буде уникнено обмеження розміру. Така генерація аудиторій на етапі ініціалізації забезпечує виконання жорсткого обмеження розміру аудиторії до розміру групи. Зрозуміло, що схожу схему можна застосовувати і до обрання комп'ютерних чи некомп'ютерних класів для ініціалізації відповідно до предметів груп.

При ініціалізації значень списку пар теж можна застосовувати ітеративну побудову, базуючись на частині списку вже обраних варіантів пар для груп певного курсу. Якщо в множині вже обраних пар для певного курсу обрана пара вже зустрічається для лекційного заняття (повного складу групи), то потрібно генерувати іншу пару для додання до списку, відповідно попередньо виключивши дану зі списку для генерації. У випадку практичних занять треба виконувати перевірку на попарну незалежність груп. Якщо групи є попарно незалежними, то вибір пари для проведення заняття дозволяється для двох чи більше попарно незалежних груп, в іншому випадку відбувається регенерація пари для групи.

Мутація складу груп. Для НаУК-МА необхідно також врахувати наявність у розкладі курсів вибіркового предметів. Відповідно, це може спричинити накладки в розкладі, бо списки різних груп студентів можуть перетинатися, тобто групи не є попарно незалежними.

Щоб покращити значення функції пристосованості хромосоми особини, для якої це обмеження порушується, введемо мутацію складу груп. Ця мутація працюватиме наступним чином: якщо дві групи на певній парі мають різні порядкові номери, але їхні списки студентів не є попарно незалежними, то відбуватиметься обмін списками студентів, що повторюються в обох групах, зі списком студентів інших груп з цього предмету. Ця операція застосовуватиметься лише до однієї з груп, які порушують дане обмеження, з більшою ймові-

рністю до тієї групи, для якої зустрічаються інші можливі групи з даного предмету. У випадку, коли для обох предметів немає груп для заміни, тоді доведеться проводити обмін самих груп у розкладі.

Додавання в початкову популяцію вдалих підрозв'язків. Це однією модифікацією, яка може покращити якість знайдених генетичним алгоритмом розв'язків, є додавання в популяцію існуючих розв'язків. У початкову популяцію додаються існуючі розв'язки розкладу, тобто кілька варіантів розкладу, складених методистом у минулому навчальному році з заміною попереднього на поточний склад груп. Таке додання в початкову популяцію дозволяє визначити напрям розвитку популяції та покращити кінцевий результат.

Паралельна реалізація алгоритму

Розглянемо основні ідеї реалізації запропонованого алгоритму у вигляді програмного застосування.

Застосунок розроблено на мові програмування Java з використанням програмного інструменту maven для управління Java проектами. Графічний інтерфейс забезпечувала бібліотека Swing. Для збереження вхідних даних програми використовувалася СКБД MySQL, а програмний зв'язок з базою даних побудований за допомогою Hibernate. Збереження pdf-файлів з розкладом на диску проводилося на базі бібліотеки itext.

З метою пришвидшення роботи алгоритму було вирішено розробити паралельну версію ГА, що базується на моделі «фермер-робітники» [7]. Застосування створює декілька робочих потоків, кількість яких дорівнює кількості процесорів, встановлених на комп'ютері, на якому виконується програма. Серед процесорів віділяють «фермера» та «робітників». Потoki «робітники» відповідають за породження нащадків, їхнє оцінювання та мутацію. Після оцінювання опрацьовані особини передаються «фермеру» для відбору в нове покоління. На «фермерові» відбувається відбір в нове покоління та застосування оператора кросинговеру. В загальному випадку мутація може вико-

нуватися теж на «фермері», але в даній програмі вона теж розподіляється на «робітників», які після опрацювання повертають особин «фермеру». Коли нове покоління повністю сформоване, воно знову передається «робітникам» на обробку. Такий поділ особин між фермером і робітником обумовлений тим, що функція оцінювання вимагає багато ресурсів для обчислення.

Після кодування, створення початкової популяції та застосування фітнес-функції та функції-валідації на «робітниках» до особин популяції застосовується метод селекції на «фермері», а до обраних селектором індивідів застосовується кросинговер. Для мутації особини порівню розпаралелюються між «робітниками» і до кожної з них з певною ймовірністю застосовується мутатор. Після того як всі «робітники» завершили свою роботу над особинами, сформоване нове покоління знову подається на вхід наступній ітерації циклу для подальшого оцінювання, селекції і т. д., поки дійдемо до останнього покоління. Коли програма опрацювала останнє покоління – вона знаходить найкращу особину популяції відповідно до функції оцінювання, яка є найкращим варіантом розв'язку.

Висновки

У цій роботі описано розроблений нами паралельний генетичний алгоритм побудови розкладу занять ВНЗ на основі моделі «фермер-робітники». Робота алгоритму тестувалася програмним застосунком із графічним інтерфейсом користувача та збереженням даних в СКБД на основі навчального плану занять факультету інформатики НаУКМА на різних конфігураціях та значеннях параметрів.

Досліджувалися елітарний, пропорційний та турнірний методи відбору особин в батьківський пул, РМХ-кросинговер та упорядкований ОХ-кросинговер як методи кросинговеру груп, двоточковий та однорідний кросинговери для породження нащадків списків аудиторій та пар. Розглядалося застосування операторів мутації вставкою та обміном.

У результаті експериментів можна рекомендувати для розв'язання задачі побудови розкладу наступну конфігурацію параметрів:

- застосування турнірного відбору в батьківський пул з невеликим розміром турніру (наприклад, розміром рівним п'яти) для забезпечення різноманітності особин популяції;
- застосування кросинговеру РМХ для схрещування хромосом груп, результати застосування якого у всіх тестах є дещо кращими за результати ОХ-кросинговеру;
- застосування однорідного кросинговеру для породження нащадків хромосом аудиторій та пар;
- застосування операторів кросинговеру одразу до трьох хромосом (груп, аудиторій і пар) з високою ймовірністю – від 0.9 до 1.0;
- застосування оператора мутації обмінами з ймовірністю до 0.05 для спрощених задач та з ймовірністю від 0.1 до 0.2 для побудови розкладу усього факультету;
- застосування розроблених модифікацій алгоритму (а саме евристик при ініціалізації особин, мутації складу груп, додавання існуючих вдалих підрозв'язків), оскільки їх застосування покращує швидкість знаходження розв'язків.

За вищенаведеною конфігурацією значення найкращого середнього відхилення задачі побудови розкладу магістерської програми становило 0.115 %. Значення середнього відхилення задачі побудови розкладу програми бакалаврату є рівним 2.115 %, а для задачі побудови розкладу факультету дане значення становить 4.2 %.

Проведені експерименти підтверджують хороші оцінки практичного застосування запропонованого генетичного алгоритму складання розкладу.

1. *Коффман Э.Г.* Теория расписаний и вычислительные машины. М.: Наука, 1984. – 335 с.
2. *Конькова И.С.* Генетические алгоритмы в решении задачи составления расписания

занятий в вузе // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XII Междунар. научно-техн. конф. – Пенза: ПДЗ, 2012. – С. 26–29.

3. *Buseti F.* Simulated annealing overview, 2009. – Available at: http://www.cs.ubbcluj.ro/~csatol/mestint/pdfs/Buseti_AnnealingIntro.pdf.
4. *Глибовець М.М., Гулаєва Н.М.* Еволюційні алгоритми. – К.: НаУКМА, 2013. – 828 с.
5. *Глибовець Н.Н., Медвідь С.А.* Генетические алгоритмы и их использование для решения задачи составления расписания // Кибернетика и системный анализ. – 2003. – № 1. – С. 95–108.
6. *Бевз С.В., Вои́тко В.В., Бурбело С.М. та інші.* Розробка автоматизованої системи формування розкладу магістратури // Наукові праці ВНТУ. – 2009. – №4. – С. 91–94.
7. *Бидюк П.И., Литвиненко В.И., Токарь А.А.* Параллельные генетические алгоритмы // Системні дослідження та інформаційні технології. – 2002. – № 4. – С. 7–16.

Одержано 18.12.2014

Про авторів:

Глибовець Микола Миколайович,
доктор фізико-математичних наук,
професор, декан факультету
інформатики НаУКМА,

Гулаєва Наталія Михайлівна,
кандидат фізико-математичних наук,
доцент кафедри інформатики НаУКМА,

Пасічник Мар'яна Михайлівна,
магістр факультету інформатики
НаУКМА.

Місце роботи авторів:

Національний університет
«Києво-Могилянська Академія»,
04655, м. Київ,
вул.Сковороди, 2.
Тел.: (044) 463 6985.
Факс: (044) 416 4515.
E-mail: glib@ukma.kiev.ua
E-mail: ngulayeva@yahoo.com
E-mail: mariana.pasichnyk@gmail.com