

В.А. Турчина, Д.О. Танасієнко

Дніпровський національний університет імені Олеся Гончара

ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДО ЗАДАЧІ СКЛАДАННЯ НАВЧАЛЬНОГО РОЗКЛАДУ

Розглянуто один з відомих підходів, який можна використовувати при складанні навчального розкладу. Запропонована схема генетичного алгоритму, результатом роботи якої є отримання наближеного розв'язку задачі складання розкладу з необхідністю подальшого його покращення іншими евристичними методами.

Ключові слова: генетичний алгоритм, навчальний розклад, евристичні методи.

Рассмотрен один из известных подходов, который можно использовать при составлении учебного расписания. Предложена схема генетического алгоритма, результатом работы которой является получение приближенного решения задачи составления расписания с необходимостью дальнейшего его улучшения другими эвристическими методами.

Ключевые слова: генетический алгоритм, учебное расписание, эвристические методы.

One of the main tasks in organizing the educational process in higher education is the drawing up of a schedule of classes. It reflects the weekly student and faculty load. At the same time, when compiling, there are a number of necessary conditions and a number of desirable. The paper considers seven required and four desirable conditions. In this paper, one of the well-known approaches that can be used in drawing up a curriculum is considered. The proposed scheme of the genetic algorithm, the result of which is to obtain an approximate solution to the problem of scheduling with the need to further improve it by other heuristic methods.

To solve the problem, an island model of the genetic algorithm was selected and its advantages were considered. In the paper, the author's own structure of the individual, which includes chromosomes in the form of educational groups and genes as a lesson at a certain time, is presented and justified. The author presents his own implementations of the genetic algorithms. During the work, many variants of operators were tested, but they were rejected due to their inefficiency. The biggest problem was to maintain the consistency of information encoded in chromosomes. Also, two post-steps were added: to try to reduce the number of teacher conflict conflicts and to normalize the schedule - to remove windows from the schedule. The fitness function is calculated according to the following principles: if some desired or desired property is present in the individual, then a certain number is deducted from the individual's assessment, if there is a negative property, then a certain number is added to the assessment. Each criterion has its weight, so the size of the fine or rewards may be different. In this work, fines were charged for non-fulfillment of mandatory conditions, and rewards for fulfilling the desired.

Keywords: genetic algorithm, educational schedule, heuristic methods.

Вступ. Математиці завжди було притаманне прагнення розробляти ефективні методи розв'язання якомога ширших класів задач. Багаторічний досвід розвитку теорії дискретних і комбінаторних задач та практика їх

розв'язання показали, що ці дві сторони – загальність метода і його ефективність – знаходяться у відомому антагонізмі. Разом з тим, дуже важливо знати, і особливо це стосується задач дискретної оптимізації, що відносяться до NP-складних [1], чи можна в принципі сподіватися на створення досить загальних і ефективних методів або треба свідомо йти шляхом розбиття задач на все більш вузькі класи і, користуючись їх специфікою, розробляти для них ефективні (хороші) алгоритми.

У випадках, коли точні методи мають експоненційну складність, доцільно застосовувати наближені алгоритми. Одним із підходів, що досить ефективно зарекомендував себе для отримання наближених розв'язків, є використання генетичних алгоритмів. Саме цей підхід і розглядається в даній роботі.

Постановка задачі. Однією з основних задач при організації навчального процесу у ЗВО є складання розкладу занять. Він відображає тижневе аудиторне навантаження студентів та викладачів. При цьому, при складанні розкладу занять є ряд необхідних умов та ряд бажаних. До необхідних умов, зокрема, відносяться наступні:

- обсяг аудиторного навантаження на студента визначається навчальним планом підготовки спеціалістів та відповідним робочим навчальним планом, який стосується конкретного року. Зазвичай тижневе аудиторне навантаження змінюється в діапазоні 18-22 год;

- в один день не можна планувати більше 8 год;
- не повинно бути “вікон”, тобто, не повинно бути перерв, більших за 20 хвилин;
- деякі заняття повинні проводитись в спеціалізованих класах;
- один викладач не може проводити одночасно заняття в двох аудиторіях;
- викладання деяких дисциплін планується одночасно для різних спеціальностей;
- заняття повинні починатись і закінчуватись у відведений часовий проміжок.

Розклад, що задовольняє наведеним вимогам, будемо вважати допустимим. Крім обов'язкових, бажаним є виконання і наступних вимог:

- не планувати на викладача в один день обсяг занять більше встановленої норми;
- не планувати на студентів в один день обсяг лекційних занять більше встановленої норми;
- не планувати на один день заняття, що проводяться в різних корпусах;
- наявність дня самостійної роботи.

Розклад, який задовольняє необхідним та бажаним умовам, будемо вважати допустимим.

В даній роботі не враховуються наступні фактори:

- доступність аудиторій та їх тип (лекційна, практична, лабораторія);
- неповне навантаження (заняття за предметом закінчуються (починаються) раніше, ніж закінчується (починається) семестр).

Вибір генетичного алгоритму. Для розв'язання поставленої задачі було обрано острівну модель генетичного алгоритму. Цей вибір мотивовано наступними його перевагами:

- **Асинхронність.** Кожен острів може незалежно від інших виконувати розрахунки. Враховуючи багатопоточність сучасних CPU (також, можна застосовувати GPU), цей факт дає велику перевагу, бо за одну одиницю часу виконується більше роботи, чим інші непаралельні ГА. Це дає більшу вірогідність, що результат є найкращим (або близький до нього).

- **Різноманітність алгоритмів кожного острова.** Кожний острів може мати певну «спеціалізацію» – спеціально підібрані оператори генетичного алгоритму, які виконують певну задачу (широкий пошук по всьому простору «добрих» індивідів або покращення існуючого генотипу, прямування до локального (а можливо і глобального) екстремуму).

- **Міграції між островами.** Цей процес дозволяє популяціям з різних островів отримувати генотипи, яких раніше вони не мали або не могли згенерувати, використовуючи наявні оператори відбору.

Тут використовуються такі відомі поняття:

- острів – елемент алгоритму, який вміщує в себе популяцію та оператори ГА для роботи над нею;

- епоха – час (або крок алгоритму), за який з популяцією на острові відбуваються зміни (з'являються нові та вмирають індивіди);

- GPU (англ. *Graphics Proccesing Unit, GPU*) — окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний рендеринг;

- CPU (англ. *Centralprocessingunit, CPU*) — функціональна частина комп'ютера, що призначена для інтерпретації команд.

Опис основних елементів ГА. Розглянемо модель по спадному принципу. Поняття індивід (рис. 1) ототожнимо з розкладом на тиждень. Такий варіант обрано, так як в навчальних закладах розклади в переважній більшості складаються на тиждень.

Наступним елементом деталізації є хромосоми. Кожна хромосома відповідає розкладу однієї групи на тиждень. Тобто, кожен індивід складається з n хромосом (груп). Наступна структура хромосоми (рис. 1) виявилась найбільш вдалим варіантом. Перед нею випробувано декілька варіантів, які були відкинуті у зв'язку з тим, що для них надто важко, або взагалі неможливо, побудувати генетичні оператори, які б не вироджували інформаційну структуру індивідуума.

Кожна хромосома складається з генів. Кожен ген представляє собою

пару $\{Task, Timepoint\}$. Причому, якщо $Task$ (рис. 2) – є завданням на 2 академічні години, то $Timepoint$ може приймати лише парні значення. Також це означає, що завдання присутнє як і за чисельником, так і за знаменником.

Вхідні дані представляють наряди на розклад, які вже мають певну структуру. А саме, в ньому вказується який викладач (викладачі), в якій групі, скільки академічних годин буде вести конкретний предмет (практичні чи лекційні заняття).

Опис деяких використовуваних операторів генетичного алгоритму.

Оператор відбору батьків. Тут використовувалися звичайні оператори [2]:

- панміксія;
- елітарний;
- рулеточний відбір.

Оператор кроссоверу. Тут задача була складнішою, адже вигляд індивіда є нетиповим для ГА. В ході роботи випробувано велику кількість варіантів оператора, але вони були відкинуті у зв'язку зі своєю неефективністю.

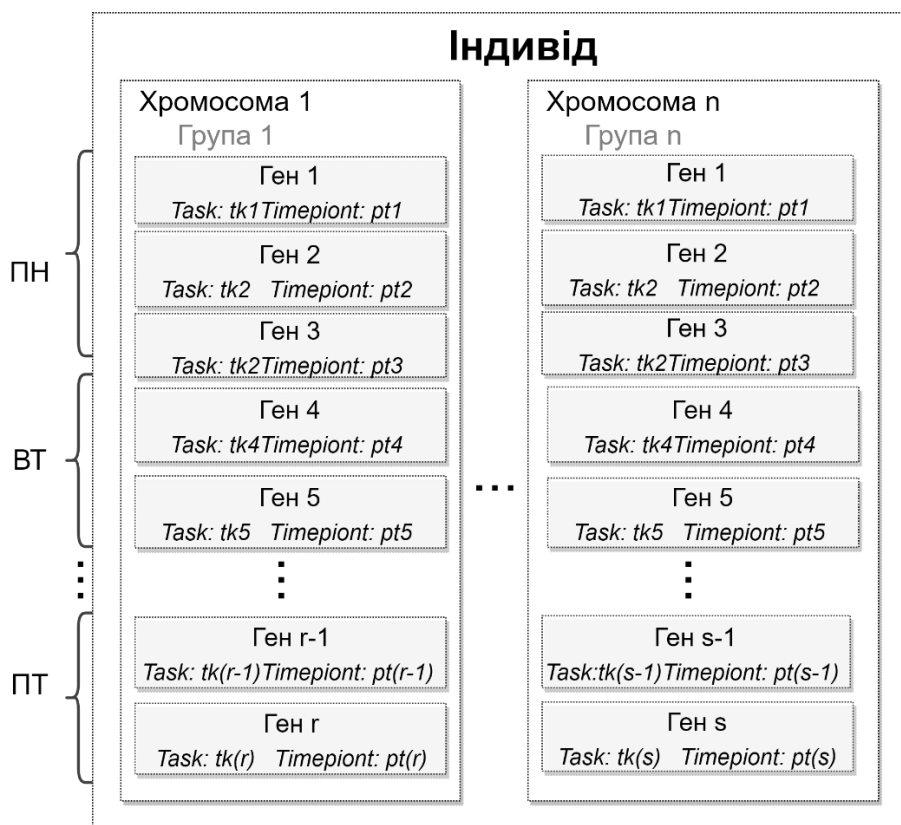


Рис. 1. Схема індивіда та хромосом

Оператор кроссоверу. Тут задача була складнішою, адже вигляд індивіда є нетиповим для ГА. В ході роботи випробувано велику кількість варіантів оператора, але вони були відкинуті у зв'язку зі своєю неефективністю. Найбільшою проблемою було збереження несуперечливості інформації закодованої в хромосомах. Наприклад, якщо використовувати простий обмін генами – однокротовий кроссовер, то з великою ймовірністю на один і той же час назначалося два завдання.

Task	
+	Teacher from teachers
+	Subject from subjects
+	Group from groups
+	Duration from {FULL, HALF}
+	Type from {LECTURE, PRACTICE}
+	empty(): bool

Рис. 2. Схема Task

Тому у роботі використовувався наступний варіант кроссоверу: якщо на один і той же час припадало дві задачі, то час віддавався гену першого батька, у всіх інших випадках – гену другого батька. На спрощеній схемі це виглядає так:

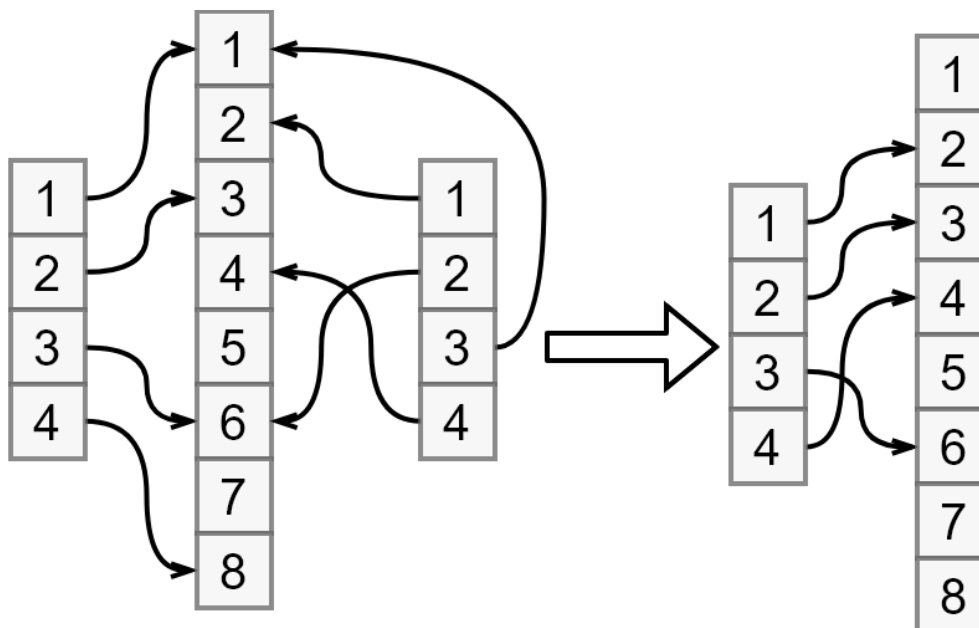


Рис. 3. Схема роботи кроссоверу

При мутації для кожної хромосоми індивіда змінюються деякі значення генів. Зміна може бути наступна: або обмін часом з іншим геном, або перенести на будь-який інший доступний час. Це відбувається так: вибирається випадковий час, якщо цей час зайнятий іншим геном, то обмінюємо місцями час. Якщо ні – то час встановлюємо відповідно отриманому випадковому значенню.

Також було додано два пост-кроки:

- 1) спробувати зменшити кількість конфліктів зайнятості викладача;
- 2) нормалізувати розклад, тобто видалити «вікна» з розкладу.

Пост-кроки були введені у зв'язку з тим, що дія оператора кроссоверу та мутації майже одразу створювала індивідів, з великою функцією фітнесу –

індивіди були далекі від допустимої множини. Але загальний генетичний алгоритм, не передбачає такі кроки. Тому довкола оператора мутації було створено wgarper, який маскується під оператора кроссоверу та вже всередині себе викликає потрібного оператора і робить необхідні додаткові дії.

Для зменшення кількості конфліктів зайнятості викладача знаходяться всі гени, в яких одні і ті ж викладачі, призначені на один і той самий час, та випадковим чином розподіляються по розкладу. Хоч це і не гарантує, що конфлікти знову не будуть створені, але такий метод швидкий і не суперечить принципам ГА.

Для нормалізації розкладу необхідно в межах кожного дня змінити значення часу в генах так, щоб не було «вікон». Це робиться наступним чином: знаходяться всі гени для певного дня, сортуються в порядку слідування та час кожного гена здвигається в напрямку першого.

Функція фітнесу [3] розраховується за наступними принципами: якщо деяка необхідна або бажана властивість присутня у індивіда, то від оцінки індивіда віднімається певне число, якщо ж присутня негативна властивість, то певне число додається до оцінки. Кожен критерій має свою вагу, тому розмір штрафу або винагороди можуть бути різними. В даній роботі штрафи нараховувалися за невиконання обов'язкових умов, а винагороди за виконання бажаних. Як можна бачити, це задача мінімізації цільової функції.

Готовий індивід має таку структуру:

- 24 хромосоми, тобто на факультеті навчаються 24 групи;
- в середньому, кількість генів дорівнювала 10, так як гени генеруються випадковим чином, але так, щоб навантаження на групу, якій відповідає хромосома, була в межах від 18 до 22 годин.

Програмна реалізація відповідає принципам закладеним в [4].

Висновки. Запропонований алгоритм дозволяє знаходити відносно непогані наближені розв'язки. Для подальшого покращення результатів потрібно використовувати класичні алгоритми комбінаторної оптимізації. Для повного завершення програмного комплексу необхідно додати модуль класичної оптимізації, утворивши конвеєр. Також можна зауважити, що розроблений ГА швидко збігається, тому не слід задавати велику кількість епох.

Бібліографічні посилання

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи [Текст] / Гэри М., Джонсон Д. – М.: Мир, 1982. — 416 с.
2. **Панченко Т.В.** Генетические алгоритмы: Учебно-методическое пособие / под ред. Ю.Ю. Тарасевича [Текст] / Панченко Т.В. – Астрахань: АГУ, 2007. –87 с.
3. **Батищев Д.И.** Генетические алгоритмы решения экстремальных задач Нижегородский госуниверситет. [Текст] /Батищев Д.И. – Нижний Новгород: 1995.с. – 62с.
4. **Koza J. R.** Genetic Programming [Текст] / Koza J. R. – Cambridge: The MIT Press, 1998.— 609 с. — ISBN 0-262-11170-5

Надійшла до редколегії 19.09.2018.