УДК 519.854.2

М.А. ПАХОМОВ, НТУ «ХПІ», Харків, Україна
Н.А. МАРЧЕНКО, канд. техн. наук, доц., НТУ «ХПІ», Харків, Україна

Розробка математичного та програмного забезпечення для автоматизації складання розкладу

Завдання складання розкладів є одним з найбільш поширених завдань, що вирішуються кожною людиною (свідомо чи ні) практично щодня. В загальній постановці воно являє собою процес розподілу деякого кінцевого набору подій у часі в умовах ресурсних та інших обмежень. Таким чином, проста людина, плануючи робочий день, і диспетчер, складаючи розклад занять у ВНЗ або графік робіт на підприємстві, вирішують завдання складання розкладу. Але якщо в першому випадку завдання може вирішуватися інтуїтивно на основі життєвого досвіду, то в другому воно може виявитися непосильно складним навіть для групи фахівців. Тому з розвитком обчислювальних технологій ведуться розробки автоматизованих систем складання розкладу. В окремих випадках вдалося розробити алгоритми, здатні знайти розв’язок за обмежений час. В той же час більшість реальних завдань складання розкладу відносяться до класу NP-повних [1]. Це робить розробку алгоритму, здатного вирішити їх за допустимий час, дійсно складним завданням, навіть якщо відповідне завдання можна поставити як однокритеріальне. Ситуація істотно ускладнюється тим, що більшість реальних завдань складання розкладів багатокритеріальні.

Головною метою роботи є розробка математичного та програмного забезпечення для автоматизації складання розкладу.

Точним методам розв’язування NP-важких завдань приділена чимала увага в роботах з теорії розкладів. Найбільшого поширення набули методи скороченого перебору, звані методами гілок і меж [2]. Для скорочення перебору обчислюються нижні оцінки цільової функції (в разі її мінімізації) і використовуються комбінаторні властивості завдань. Також для розв’язання задач теорії розкладів широко застосовується метод динамічного програмування [3].

Генетичні алгоритми – це стохастичні евристичні оптимізаційні методи, основна ідея яких була взята з теорії еволюційного розвитку видів. Першим кроком при розробці математичної моделі, заснованої на генетичному алгоритмі, є розробка структури хромосоми, в якій буде зберігатися розв’язок. У даному випадку такою «хромосомою» є розклад. Обрана структура повинна враховувати всі особливості та обмеження, що пред’являються до розв’язку, а також те, що від її вибору безпосередньо залежать реалізації алгоритмів кросинговеру і мутації. Вибір «хромосоми» впливає не тільки на швидкість, а й на збіжність алгоритму взагалі.

Одним з найбільш зручних уявлень розв’язку розглянутої задачі є тривимірна матриця, по осях i, j, k якої відкладаються відповідно групи,

навчальні години і аудиторії. Елементом матриці є запит на проведення заняття з i-ю групою даним викладачем з даної дисципліни в j-у годину в k-й аудиторії. Після цього кожна зв'язка дисципліна-група-викладач розглядається як цілісність з умовою, що алгоритм знає, які дисципліни в якій групі який викладач проводить і який тип аудиторій для цього підходить. Така структура хромосоми зручна тим, що вже на етапі завдання початкових даних можна виключити свідомо невдалі розв'язки, заблокувавши відповідні комірки.

На наступному кроці алгоритму створюється початкова популяція, розмір якої залежить від розмірності задачі і становить зазвичай кілька сотень розв'язків. Для організації оптимізаційного процесу необхідно створити спрямовуючу силу розвитку популяції. В якості такої сили виступає вимога мінімізації цільової функції або, в термінах генетичних алгоритмів, фітнес-функції.

Таким чином, помістивши початкову популяцію в створене штучне середовище і реалізувавши процеси селекції, кросинговеру і мутації, можна отримати ітераційний алгоритм пошуку оптимального розв'язку, на кожній ітерації якого виконуються наступні дії.

1. Кожна особина популяції оцінюється за допомогою фітнес-функції.

2. Кращі розв'язки (зазвичай близько 5%) копіюються в нову популяцію без зміни. Такий принцип запобігає втраті кращих розв'язків і забезпечує підвищену збіжність алгоритму.

3. На основі пропорційного відбору з поточної популяції вибираються два розв'язки, які піддаються рекомбінації. Для цього хромосоми батьків обмінюються відповідними ділянками.

4. Отриманий в попередньому пункті розклад може виявитися некоректним, наприклад, може не відповідати навчальному плану. В цьому випадку можна повторювати операцію рекомбінації до тих пір, поки не буде отримано коректний розклад.

5. Якщо нова популяція сформована, то стара видаляється, після чого переходимо до етапу 1. В іншому випадку переходимо до етапу 3.

Розглянутий алгоритм є не тільки стійким до локальних мінімумів, а й завдяки внутрішньому паралелізму, вираженому в роботі не з окремими розв'язками, а з цілими класами розв'язків, забезпечує відносно швидкий пошук оптимального рішення.

Список літератури:

1. *Кормен Т.* Алгоритмы: построение и анализ, 2-е издание/ *Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн* // Пер. с англ. – М. : Вильямс. – 2011. – 1296 с.

2. *Сигал И. Х.* Введение в прикладное дискретное программирование: теория и вычислительные алгоритмы./ *И. Х. Сигал, А. П. Иванова //* М.: Физматлит. – 2002. – 240 с.

3. *Беллман Р.* Динамическое программирование/ *Р. Беллман //* М.: ИЛ. – 1960. – 400 с.