УДК 004.023:004.421.2

С.В. Алексєєв, В.О. Мартовицький

Харківський університет Повітряних Сил імені Івана Кожедуба, Харків

АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗКУ ЗАДАЧІ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ

У статті наведено результати аналізу основних методів розв'язку задачі складання розкладу: метод імітації відпалу, метод гілок та меж, генетичні алгоритми, мурашиний алгоритм. Показано, що для розв'язку задач складання розкладу навчальних занять доцільно використати модифікації генетичного алгоритму, які за рахунок можливості виконання заданої кількості ітерацій дозволять отримати наближене до оптимального або допустиме рішення.

Ключові слова: *розклад занять, метод імітації відпалу, метод гілок та меж, генетичний алгоритми, мурашиний алгоритм.*

Вступ

Однією з найважливіших складових навчального процесу будь-якого навчального закладу є розклад занять – документ, що регламентує трудовий ритм і впливає на творчу віддачу викладацького складу та розглядається як фактор оптимізації навчального процесу.

В ньому 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 із 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 курсантів (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 деякої множини обмежень.

Вхідними даними можуть являтися: тривалість навчальних занять та їх кількість по днях тижня, тривалість переривів 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 проведення одночасного проведення заняття з декількома навчальними групами в відповідній аудиторії і т.п. Критерієм оптимальності отриманого розкладу може, наприклад, являтися мінімізація простоїв в навчальному процесі.

Рішення подібних задач теорії розкладів представляє відому трудність. За змістом ці задачі відносяться до класу комбінаторних [1 – 3], для яких суттєве значення має розмірність. Як правило, розмірність задач складання оптимальних розкладів настільки велика, що розв'язати їх простим перебором варіантів не представляється можливим. Часто задачі складання розкладів зводяться до задач цілочисельного лінійного програмування [4, 5] (в тому числі багатоіндексного), для вирішення яких використовуються широко відомі методи відсікання або гілок і меж [6]. Традиційними методами дослідження операцій [7] для задач планування є комбінаторні процедури, імітаційне моделювання [8], мережеві методи й евристичні підходи.

Метою статті є обґрунтування вибору найбільш доцільного методу розв'язку задачі складання розкладів занять ВНЗ в цілях її подальшої автоматизації.

Основна частина

Проведемо аналіз основних відомих методі розв'язку задач складання розкладів.

Метод імітації відпалу.

Алгоритми імітації відпалу [9] – загальний алгоритмічний метод розв'язання задачі глобальної оптимізації; є одним із прикладів методів Монте-Карло. У процесі пошуку оптимального рішення з деякою ймовірністю допускається перехід у стан із більш високим значенням цільової функції. Ця властивість дозволяє їм виходити з локальних оптимумів. На початку роботи алгоритму ймовірність переходу в стан із більш високим значенням цільової функції повинна бути досить велика, щоб була можливість переходу від обраного початкового наближення до будь-якого іншого розв'язання. У процесі роботи алгоритму ймовірність переходу поступово зменшується.

Алгоритм ґрунтується на імітації фізичного процесу, який відбувається при кристалізації речовини (при поступовому зниженні температури), в тому числі при відпалі металів.

За допомогою моделювання такого процесу шукається така точка або безліч точок, на якій досягається мінімум деякої числової функції $F(\overline{x})$, де $\overline{x} = (x_1, ..., x_m) \in X$. Вводиться послідовність точок $\overline{x_0}, \overline{x_1}, ..., \overline{x_n}$ простору X. Алгоритм послідовно знаходить наступну точку по попередній, починаючи з точки $\overline{x_0}$, яка є початковим наближенням. Алгоритм зупиняється після досягнення точки $\overline{x_n}$.

Точка $\overline{x_{i+1}}$ за алгоритмом розраховується на основі поточної точки $\overline{x_i}$ наступним чином. До точки $\overline{x_i}$ застосовується оператор А, який випадковим чином модифікує відповідну точку, в результаті чого виходить нова точка $\overline{x_i}$. Точка $\overline{x^*}$ стає точкою $\overline{x_{i+1}}$ з імовірністю $P(\overline{x^*}, \overline{x_{i+1}})$, яка обчислюється відповідно до розподілу Гіббса:

$$P(\overline{x^*} \to \overline{x_{i+1}}) =$$
$$= \begin{cases} e^{-\frac{F(\overline{x^*}) - F(\overline{x_i})}{Q_i}}, & \text{якщо } F(\overline{x^*}) - F(\overline{x_i}) \geq 0 \sqrt{} \\ 1, & \text{якщо} F(\overline{x^*}) - F(\overline{x_i}) < 0, \end{cases} \quad (1)$$

де $Q_i > 0_i$ – елементи довільної, збіжної до нуля позитивної послідовності, яка задає аналог падаючої температури в кристалі. Швидкість убування і закон убування, як правило, задають окремо відповідно до умов задачі, що розв'язується.

Для багатьох NP-важких задач найкращі розв'язки були отримані алгоритмами імітації відпалу [9]. Однак їхнім недоліком є висока обчислювальна складність. Це зумовлено тим, що для отримання хорошого розв'язання потрібно досить повільне зниження ймовірності переходу в стан із більш високим значенням цільової функції, яке призводить до великої кількості ітерацій алгоритму.

У [10] показано, що алгоритми імітації відпалу можуть застосовуватися для побудови розкладів виконання прикладних програм на всіх стадіях проектування обчислювальних систем реального часу. Однак для їх практичного впровадження актуальною задачею залишається зменшення обчислювальної складності алгоритмів.

Метод гілок та меж [11] є загальним алгоритмічним методом вирішення різноманітних оптимізаційних задач. Він широко застосовується для таких NP-повних задач, як задача комівояжера [12] та задача о ранці [13]. Даний метод є варіацією повного перебору, при якому відкидаються підмножини допустимих рішень, що не містять оптимальних значень цільової функції.

В методі гілок та меж використовуються дві процедури: розгалуження та знаходження оцінок (меж).

За допомогою процедури розгалуження множина допустимих рішень на кожному етапі розділяється на підмножини меншої розмірності. Ці множини стають вузлами дерева пошуку.

Процедура знаходження оцінок (меж) визначає верхні та нижні межі оптимального значення на підмножинах допустимих рішень. Якщо для знайденого вузла дерева пошуку верхня межа співпадає з нижньою, то це значення цільової функції є оптимальним і досягається на відповідній підмножині допустимих рішень.

Формальний опис методу для розв'язання задач на мінімум [14] може бути дано в такому вигляді:

$$f(x) \to \min_{x \in D}, \quad (2)$$

де D – множина допустимих рішень; $d \in D$.

Функцію b(d), яка співвідносить множині d розбиття його на підмножини $d_1,...,d_N$, N>1, називають розгалуженням.

Функція H(d) є нижньою межею для d, якщо

а) $H(d) \le \min_{x \in d} f(x)$;

б) на одноелементній множині {x} вірна рівність H({x}) = f(x).

Алгоритм, що реалізує метод гілок та меж, складається з послідовності однотипних кроків. На кожному кроці відомий рекорд x^0 і підмножини t_1, $t_2,...,T_L$ не розглянутих розв'язків. На початку роботи алгоритму L= 1, t_1 =D, x^0 – довільний елемент множини D, або порожня множина.

На кожному кроці алгоритм починає роботу з перевірки елементів розбиття. Нехай перевіряється множина t_j. Множина t_j відсікається в одному з двох випадків, що послідовно перевіряються:

а) якщо $H(t_j) \ge f(x^0)$;

б) якщо $H(t_j) < f(x^0)$ і знайдено такий елемент $y \in_j t_j$, що $f(y_j) = \min_{x \in t_j} f(x) = H(t_j)$.

У другому випадку відбувається зміна рекорду $x^0 = y_j$.

Нехай $t_1,t_2,...,T_M$ $(M \le L)$ –невідсічені множини.

Якщо M = 0, алгоритм закінчує роботу, і за розв'язок задачі приймається рекорд x^0. При $M \ge 1$ серед множин $t_1,..., t_M$ обирається множина для нового розгалуження. Нехай такою є множина t_1. Тоді здійснюється розгалуження $b(t_1) = (d_1, ..., d_N)$, в результаті якого отримуємо список множин $d_1, ..., d_N$, $t_2, ...,T_M$. Ці множини нумеруються числами від 1 до L, і починається новий крок алгоритму.

При розв'язку конкретної задачі слід визначити способи побудови нижньої і верхньої оцінок, метод розгалуження, а також правило вибору перспективної множини для розбиття.

При визначенні «перспективного» елемента розбиття в основному застосовують одночасне (багатостороннє) й одностороннє розгалуження. При одночасному розгалуженні функція b може бути застосована добудь-якого елементу розбиття. Часто в якості та кого елемента вибирається підмножина t_k з мінімальною нижньою межею

$$f(y_j) = \min_{x \in t_j} f(x) = H(t_j). \quad (3)$$

При односторонньому розгалуженні номер підмножини, що підлягає розбиттю, відомий заздалегідь.

Генетичні алгоритми [15 – 17] дозволяють розв'язувати широке коло складних задач багатокритеріальної оптимізації шляхом випадкового підбору, комбінування та зміни параметрів моделювання способами, що подібні до біологічної еволюції (наслідування, мутація, відбір). Застосовують ці алгоритми й для розв'язку задач на графах. Генетичні алгоритми оперують сукупністю особин (популяцією), які кодують один із можливих розв'язків задачі.

Цим генетичний алгоритм відрізняється від більшості інших алгоритмів оптимізації, які оперують лише з одним розв'язком, який покращує його. Принцип роботи генетичного алгоритму полягає у наступному.

За допомогою функції пристосованості серед всіх особин популяції виділяють:

− найбільш пристосовані, які отримують можливість схрещуватися і давати потомство;

− найгірші, які видаляються з популяції і не дають потомства.

Таким чином, пристосованість нового покоління в середньому вище попереднього.

Існує досить багато модифікацій генетичних алгоритмів. Але всім їм властива універсальність. Тобто від задачі, яку необхідно розв'язати, залежить визначення функції пристосованості й спосіб кодування рішень. Решта же етапів генетичних алгоритмів виконується однаково для будь-яких задач.

Кожен крок алгоритму складається з трьох стадій:

1. Генерація проміжної популяції шляхом відбору поточного покоління.

2. Схрещування особин проміжної популяції шляхом кросовера, у результаті чого формується нове покоління.

3. Мутація нового покоління.

Проміжна популяція формується з особин, які отримали право розмножуватися. Найбільш пристосовані особини можуть бути записані туди кілька разів, найменш пристосовані з великою ймовірністю туди взагалі не потраплять.

Існує кілька способів реалізації даного відбору:

– stochastic sampling. Особини розташовуються на колесі рулетки так, що розмір сектора кожної особини пропорційний її пристосованості. N раз запускаючи рулетку, вибирається необхідна кількість особин для запису в проміжну популяцію;

– remainde rstochastic sampling. Для кожної особини обчислюється відношення її пристосованості до середньої пристосованості популяції. Ціла частина цього відношення вказує, скільки разів потрібно записати особину в проміжну популяцію, а дробова показує її ймовірність потрапити туди ще раз.

Після відбору особини проміжної популяції випадковим чином розбиваються на пари, потім із деякою ймовірністю схрещуються, в результаті чого виходять два нащадка, які записуються до нового покоління, або не схрещуються, тоді в нове покоління записується сама пара.

У класичному генетичному алгоритмі застосовується одноточковий оператор кросовера: для батьківських «генотипів» випадковим чином вибирається точка розділу, нащадки формуються шляхом обміну відсіченими частинами:

$$\begin{array}{lcl} 011010.01010011101 & \rightarrow & 101100.01010011101 \\ 101100.10011101011 & & 011010.10011101011. \end{array}$$

До отриманого в результаті відбору і схрещування нового покоління застосовується оператор мутації. При цьому кожен біт «генотипу» кожної особини популяції з деякою, досить малою ймовірністю, інвертується.

Такий процес еволюції може тривати до безкінечності. Критерієм зупинення може служити задана кількість поколінь або сходження популяції.

Сходження – стан популяції, коли всі рядки популяції знаходяться в області деякого екстремуму і майже однакові. Тобто кросовер практично ніяк не змінює популяції, а особини, що мутували, схильні вимирати, оскільки менш пристосовані. Таким чином, сходження популяції означає, що знайдено близький до оптимального розв’язок.

За розв'язок задачі приймають найбільш пристосовану особина останнього покоління.

Мурашиний алгоритм [18] (ant colony optimization, ACO) – один із ефективних поліноміальних алгоритмів для знаходження наближених рішень задачі комівояжера [19], а також аналогічних завдань пошуку маршрутів на графах. Суть підходу полягає в використанні моделі поведінки мурах, що шукають шляхи від колонії до джерела живлення.

Перший варіант алгоритму, запропонований доктором наук Марко Доріго [20] у 1992 році, отримав подальший розвиток у [21] та більш пізніх роботах.

В основі класичного алгоритму лежить поведінка мурашиної колонії – маркування більш вдалих шляхів великою кількістю ферменту. Робота починається з розміщення мурах у вершинах графа, потім здійснюється рух мурах. Напрямок руху визначається ймовірнісним методом.

Будь-який мурашиний алгоритм, незалежно від модифікацій, складається з наступних етапів, які виконуються циклічно до виконання умов виходу:

1. Створення мурах.

Вибір початкових вершин, куди розміщуються мурахи, і спосіб їх розміщення залежить від обмежень за умовами задачі. Мурахи можуть поміщатися в одну вершину, в різні вершини з повтореннями, або без повторень.

На цьому ж етапі задається первинний рівень феромону. Він, зазвичай, ініціалізується невеликим позитивним числом для того, щоб на початковому кроці ймовірності переходу в наступну вершину не були нульовими.

2. Пошук розв’язку.

Імовірність переходу з вершини i в вершину j визначається за формулою

$$P_{i,j,k}(t) = \frac{\left[\tau_{ij}(x)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum\limits_{j \in J_{j,k}} \left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}, \quad (4)$$

де $\tau_{ij}(t)$ – рівень феромону;

η_{ij} – евристична відстань;

α, β – константні параметри.

Необхідний компроміс між цими величинами знаходиться експериментально.

3. Оновлення феромону.

Рівень феромону оновлюється відповідно до наведеної формули

$$P_{i,j,k}(t) = \frac{\left[\tau_{ij}(x)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum\limits_{j \in J_{j,k}} \left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}, \quad (5)$$

де ρ – інтенсивність випаровування феромону;

$L_k(t)$ – ціна поточного розв’язку для k-ї мурахи;

Q – параметр порядку ціни оптимального рішення, тобто $Q/L_k(t)$ – феромон, відкладався k-ю мурахою на ребрі (i, j).

4. Додаткові дії (за необхідності).

Зазвичай на цьому етапі використовується алгоритм локального пошуку, проте він може також застосовуватися й після пошуку всіх розв’язків задачі.

Висновки

Таким чином, враховуючи велику розмірність задачі складання розкладу навчальних занять та значну кількість обмежень, що необхідно врахувати, маємо наступне:

– методи імітації відпалу характеризуються великою обчислювальною складністю, тому їх застосування буде досить проблематичним;

– метод гілок і меж потребує визначення процедур розгалуження й знаходження оцінок верхніх і нижніх меж оптимального значення на підмножинах допустимих рішень, що при його ітеративності й наявності можливо протиречивих обмежень також призведе до великої складності застосовуваних алгоритмів;

– генетичні алгоритми вимагають кодування можливого рішення у вигляді «генома» й визначення цільової функції, за допомогою якої необхідно оцінювати оптимальність рішення. Розмірність задачі напряму збільшить довжину «генома», але можливість виконати задану кількість ітерацій може дозволити отримати наближене до оптимального або допустиме непротиречиве рішення;

– адаптація мурашиного алгоритму до розв’язку задачі складання розкладу потребує його значної модифікації й, вочевидь, буде мати значну обчислювальну складність.

Відповідно, для розв'язку задач складання розкладу занять доцільно використати генетичні алгоритми.

Тому напрямком подальших досліджень є визначення та формалізація наявних обмежень щодо проведення навчальних занять, модифікація генетичного алгоритму, проведення експерименту та оцінка ефективності складання розкладів занять з метою подальшої автоматизації цієї задачі.

Список літератури

1. Андерсон Джеймс А. Дискретная математика и комбинаторика / Джеймс А. Андерсон. – М., 2004. – 960 с.
2. Графы и их применение. Комбинаторные алгоритмы для программистов Костюкова Н.И. Интернет-университет информационных технологий. – ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007. – 312 с.
3. Рыбников К.А. Введение в комбинаторный анализ / К.А. Рыбников. – М.: Изд-во МГУ, 1985. – 312 с.
4. Акулич И.Л. Математическое программирование в примерах и задачах / И.Л. Акулич. – СПб., М.: «Лань», 2011. – 185 с.
5. Асанов М.О. Дискретная математика: графы матроиды, алгоритмы / М.О. Асанов, В.А. Баранский, В.В. Расин. – Ижевск: НИЦ "РХД", 2001. – 288 с.
6. Кафаров В.В. Гибкие автоматизированные производственные системы / В.В. Кафаров. – М.,1990. – 320 с.
7. Вентцель Е.С. Исследование операций: задачи, принципы, методология / Е.С. Вентцель. – М.: Наука, Главная редакция физико-математической литературы, 1980.
8. Строгалев В.П. Имитационное моделирование / В.П. Строгалев, И.О. Толкачева. – М.: МГТУ им. Баумана, 2008. – С. 697-737.
9. 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.
10. Костенко В.А. Исследование различных модификаций алгоритмов имитации отжига для решения задачи построения многопроцессорных расписаний / В.А. Костенко, А.В. Калашников // Тр. седьмой междунар. научн. конфер. "Дискретные модели в теории управляющих систем". – М.: МАКС Пресс, 2006.
11. Land A.H. An automatic method of solving discrete programming problems / A.H. Land, A.G. Doig // Econometrica. – 1960. – V. 28. – P. 497-520.
12. Алгоритм для решения задачи коммивояжера / Д.Ж. Литтл, К. Мурти, Д. Суини, К. Кэрел // Экономика и математические методы. – 1965. – Т. 1, вып. 1. – С. 90-107.
13. Бурков В.Н. Элементы теории графов / В.Н. Бурков, Д.А. Новиков. – 2001. – С. 28.
14. Гончаров Е.Н. Исследование операций. Примеры и задачи: учебное пособие / Е.Н. Гончаров, А.И. Ерзин, В.В. Залюбовский. – Новосибирск: Новосибирский государственный университет, 2005. – 78 с.
15. Deb K. Understanding Interactions Among Genetic Algorithm Parameters / K. Deb, S. Agrawal. – 1998.
16. 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, 2006. – 320 с.
17. Еремеев А.В. Генетический алгоритм для задачи о покрытии / А.В. Еремеев // Дискрет. анализ и исслед. операций. Сер. 2. – 2000. – Т. 7, № 1. – С. 47-60.
18. Штовба С.Д. Муравьиные алгоритмы. Exponenta Pro / С.Д. Штовба // Математика в приложениях. – 2003. – № 4. – С. 70-75.
19. Литтл Д.Ж. Алгоритм для решения задачи коммивояжера / Д.Ж. Литтл, К. Мурти, Д. Суини, К. Кэрел // Экономика и математические методы. – 1965. – Т. 1, вып. 1. – С. 90-107.
20. Colorni A. Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle / A. Colorni, M. Dorigo, V. Maniezzo. – Paris, France, Elsevier Publishing, 1991. – 134-142.
21. Dorigo M. Optimization by a Colony of Cooperating Agents / M. Dorigo, V. Maniezzo, A. Colorni // IEEE Transactions on Systems, Man and Cybernetics-Part B. – 1996. – 26 (1). – P. 29-41.

Надійшла до редколегії 18.12.2013

Рецензент: д-р техн. наук, проф. І.В. Рубан, Харківський університет Повітряних Сил ім. Івана Кожедуба, Харків.

АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ ЗАНЯТИЙ

С.В. Алексеев, В.А. Мартовицкий

В статье приведены результаты анализа основных методов решения задачи составления расписания: метод имитации отжига, метод ветвей и границ, генетические алгоритмы, муравьиный алгоритм. Показано, что для решения задач составления расписания учебных занятий целесообразно использовать модификации генетического алгоритма, которые за счет возможности выполнения заданного количества итераций позволят получить приближенное к оптимальному или допустимое решение.

Ключевые слова: *расписание занятий, метод имитации отжига, метод ветвей и границ, генетический алгоритм, муравьиный алгоритм.*

ANALYSIS METHOD FOR SOLVING THE PROBLEM OF CLASS SCHEDULES

S.V. Alekseev, V.A. Martovytskyi

The results of the analysis of the main methods of solution of the problem of scheduling: the method of simulated annealing, branch and bound method, genetic algorithms, the ant algorithm. It is shown that problem solving scheduling classes it is advisable to use a amendment of genetic algorithm, that due to the possibility of the specified number of iterations will provide close to optimal or acceptable solution.

Keywords: *schedule method simulated annealing, branch and bound method, genetic algorithms, the ant algorithm.*