УДК 681.3

С.А. Ібрагім, аспір.
Житомирський державний технологічний університет

РОЗВ'ЯЗУВАННЯ ЗАДАЧ ТЕОРІЇ РОЗКЛАДІВ ГЕНЕТИЧНИМИ АЛГОРИТМАМИ

(Представлено д.т.н. Панішевим А.В.)

Розглянуто класичні задачі теорії розкладів. Проведено огляд алгоритмів та методів їх розв'язування. Пропонується для рішення сформульованих задач використовувати модифікацію стандартного генетичного алгоритму. Наведені алгоритми виправлення неприпустимих хромосом після схрещування та алгоритми обчислення функцій придатності для конвеєрної та загальної задач теорії розкладів.

1. Постановка загальної задачі складання розкладів

Загальна задача теорії розкладів (ЗЗТР) може бути сформульована наступним чином. Є множина робіт $T = \{T_1,\ldots,T_n\}$, пов'язаних між собою логічними, технологічними та виробничими умовами передування. Роботи виконуються на машинах $M_1, M_2, \ldots, M_m$, причому процес виконання робіт в залежності від постановки задачі може бути послідовним, паралельним або загального вигляду (послідовно-паралельним). Маршрути або вважаються відомими, або їх необхідно знайти. Виконання робіт може вимагати додаткових ресурсів (що складаються або не складаються) з заданої множини ресурсів, при цьому відомо, в яких межах можуть змінюватись обсяги споживання даних ресурсів. Тривалість операції по виконанню роботи T_i на машині M_j при залученні підмножини ресурсів передбачається або заданою, або легко вираховується за допомогою відомих характеристик технологічного процесу. Необхідно знайти маршрути проходження робіт по машинах, призначити строки виконання всіх операцій по виконанню робіт на машинах і розподілити при цьому за часом споживання ресурсів, що використовуються, таким чином, щоб сягав екстремуму деякий критерій оптимальності розкладу, який залежить від строків початку та завершення операцій.

У зв'язку з тим, що умови задач теорії розкладів можуть змінюватися в дуже широких межах, виникає велике розмаїття постановок цих задач, що відрізняються числом машин і робіт, видом обслуговування, типом обмежень, видом та кількістю критеріїв оптимальності, характером представлення вихідних даних. Тому зручно користуватися так званою "класифікаційною формулою", яка дозволяє в компактному вигляді представляти умови задач [2], [3].

Отже, класифікаційна формула представлення задач теорії розкладів має вигляд:

$$\alpha = \alpha_1, \alpha_2 \mid \beta_1, \beta_2, \beta_3, \beta_4, \beta_5 \mid \gamma \quad \delta \quad \nu \quad \mu .$$

Масив (α_1, α_2) описує характеристики обслуговування: α_1 означає число машин, α_2 – вид обслуговування (послідовний, паралельний, послідовно-паралельний), $\alpha_2 \in \{\varnothing, P, Q, R\}$ в однооопераційному випадку та $\alpha_2 \in \{O, F, O, C\}$ в багатоопераційному, де

$\varnothing$ – означає, що роботи виконуються на одній машині;

P – роботи виконуються ідентичними, паралельно працюючими машинами;

Q – роботи виконуються однотипними, паралельно працюючими машинами;

R – тривалості операцій довільні;

O – довільний маршрут послідовного проходження кожної роботи через машини;

F – задано фіксований маршрут проходження кожної роботи через машини, цей маршрут однаковий для всіх робіт, причому попередня операція повинна закінчитися не пізніше, ніж почнеться наступна;

J – для кожної роботи задано свій маршрут проходження машин, також попередня операція повинна закінчитися не пізніше, ніж почнеться наступна;

C – всі роботи проходять однаковий маршрут, але в цьому маршруті деякі машини можуть брати участь більше одного разу і попередня операція повинна закінчитися не пізніше, ніж почнеться наступна;

масив $(\beta_1,\ldots\beta_5)$ – описує характеристики робіт:

β_1 – визначає число робіт;

β_2 – фіксує, чи дозволені в системі переривання обслуговування;

β_3 – характеризує відношення передування робіт;

β_4 і β_5 – фіксують наявність директивних строків надходження робіт та їх закінчення;

масив γ – вказує на критерій оптимальності розкладу (або кілька критеріїв у разі багатокритеріальної задачі);

масив δ – вказує на характер використаних ресурсів;

параметр ν – означає характер представлення вихідних даних (детерміновані, вірогідні, задані в термінах розмитих систем);

масив μ – визначає режим складання розкладу.

2. Методи розв'язання загальної задачі упорядкування робіт

Швидкий погляд на вихідні дані класичних задач теорії розкладів дозволяє зробити висновки про те, що шукані оптимальні значення цільових функцій можуть бути знайдені тільки в результаті перебору деякої кінцевої множини допустимих варіантів розв'язків, а це пред'являє підвищені вимоги до ефективності алгоритмів розв'язання задач.

Серед задач теорії розкладів можна виділити поліноміально розв'язні та NP-складні. Для кожної поліноміально розв'язної задачі відомо хоча б один ефективний алгоритм її розв'язку, в той час як для NP-складних задач такі алгоритми не визначені.

Той факт, що деяка екстремальна задача належить до класу NP-складних, говорить про складність її розв'язання за прийнятний час та служить обґрунтуванням на користь розробки негарантуючих оптимуму, наближених поліноміальних процедур. Спроба систематизації результатів та створення основ теорії наближених алгоритмів дискретної оптимізації почата у роботах [7], [8], [9]–[15]. Можливою альтернативою наближеним алгоритмам служать методи, що реалізують різні універсальні схеми скорочення перебору рішень.

Методи теорії розкладів розподіляються на універсальні та спеціальні.

Універсальні (загальні) методи застосовуються для розв'язання широкого класу задач і являють собою відомі обчислювальні методи дискретної оптимізації, адаптовані до задач складання розкладів. Серед них найбільш поширеними є методи:

- неявного перебору варіантів (особливо метод "гілок і меж" та динамічне програмування);
- еквівалентних перетворень;
- локальної оптимізації;
- двоїсті методи;
- наближені методи з гарантованими оцінками.

Спеціальні методи, навпаки, не претендують на розв'язання широкого класу задач дискретної оптимізації, вони орієнтовані лише на розв'язання окремих задач теорії розкладів або їх вузьких класів. У багатьох випадках така спеціалізація виправдана, оскільки вона дозволяє скоротити повний перебір варіантів за рахунок максимального використання комбінаторних особливостей теорії розкладів. Як приклад можна назвати "перестановочний прийом", пріоритетні методи, складання циклограм [1], [4–6], [16].

Методи неявного перебору. Ця група методів займає одне з центральних місць в дискретній оптимізації і, зокрема, в теорії розкладів. Головна ідея цих методів – скорочення повного перебору всіх розв'язків задачі за рахунок інтенсивного відсіювання множин безперспективних рішень. До цього сімейства методів належать:

- метод гілок і меж [1], [4–6], [16];
- динамічне програмування;
- метод послідовного аналізу варіантів, запропонований В.Міхалевічем та Н.Шором [5], [17];
- метод побудови послідовних рішень В.Ємелічева [18];
- метод послідовних вирахувань та інші [19].

Метод еквівалентних перетворень – переформування задачі теорії розкладів в інших термінах (наприклад, в термінах цілочислового лінійного програмування (ЦЛП) чи теорії графів) і представлення її у вигляді "стандартної" задачі, добре вивченої та ефективно розв'язної у відповідній області дискретної математики. У багатьох випадках цей метод є простим і в той же час ефективним методом розв'язання вихідної задачі теорії розкладів. Практично будь-яка задача теорії розкладів може бути переформульована частково в термінах

ЦЛП, однак одержані таким чином задачі можуть бути надзвичайно великих розмірів. Практичне застосування цей метод знаходить для розв'язання задач теорії розкладів, що зводяться до спеціальних задач ЦЛП, наприклад, транспортних, потокових або задач про призначення [20–21].

Загальна ідея *методів локальної оптимізації* полягає в послідовному знаходженні "рекордних рішень" $x_0, x_1, x_2, \ldots$, кожне з яких, починаючи з x_1, надає краще значення цільовій функції в окрузі попереднього розв'язку. Характерна особливість цих методів – простота алгоритмічної реалізації.

Для задач дискретного програмування існують два головних напрямки використання *двоїстого підходу* при побудові чисельних методів. Перший пов'язаний з побудовою оцінок знизу (для задач мінімізації) оптимального значення вихідної задачі. Другий базується на узагальненій теоремі Еверетта [15] і пов'язаний з використанням властивості оптимальних рішень зберігати оптимальність і по відношенню до вихідної задачі із зміненою деяким чином правою частиною обмежень.

Необхідність в *наближених методах* з гарантованими оцінками пов'язана, в першу чергу, з різноманітними ситуаціями на транспорті, коли вихідні дані при складанні розкладів наближені. По-друге – це обчислювальні складності, формалізовані в теорії NP-повноти, бо більшість задач теорії розкладів – складновирішуємі, тобто NP-складні.

Рекурсивні методи. Рекурсивною є процедура, яка в ході її реалізації прямо чи побічно звертається сама до себе. Сама по собі вона не приводить до більш ефективних алгоритмів, однак разом з іншими прийомами (перестановочним, еквівалентних перетворень, динамічним програмуванням) рекурсія сприяє підвищенню ефективності алгоритмів та спрощенню програм. Наприклад, алгоритм Джексона для задачі $2F|n|V_{max}$, по суті, є комбінацією рекурсії та перестановочного прийому.

3. Генетичні алгоритми розв'язання задач теорії розкладів

В якості альтернативи вказаним методам розглянемо клас генетичних алгоритмів.

Генетичні алгоритми (ГА) – це адаптивні методи, що можуть використовуватися для розв'язання пошукових і оптимізаційних задач. Вони побудовані на генетичних процесах біологічних організмів.

ГА використовують природну аналогію з природним поводженням. Вони працюють з *популяцією* «особин», кожна з яких являє собою можливе розв'язання деякої задачі. Кожній особині привласнюється значення придатності відповідно до того, наскільки гарним розв'язком задачі вона є.

Сила ГА полягає в тім, що це надійна техніка, що успішно працює з великою кількістю типів задач, включаючи оптимізаційні задачі, але вони гарні і для отримання «гарного прийнятного» розв'язання задачі за «прийнятно малий» час. Для розв'язання деяких окремих випадків таких задач існують алгоритми, що перевершують ГА як за швидкістю, так і за точністю отриманого розв'язку. Основна область застосування ГА, таким чином, – це складні задачі, для яких таких алгоритмів не існує. Але навіть там, де існують подібні алгоритми, їх можна поліпшити гібридизацією з ГА.

Стандартний ГА може бути представлений наступним чином.

```
BEGIN /* GENETIC ALGORITHM */
    створити початкову популяцію
    обчислити функцію придатності для кожної особини

WHILE NOT завершено DO
BEGIN /* створити нове покоління */

    FOR розмір_популяції/2 DO
    BEGIN /* цикл розмноження */
    вибрати 2 особини з старого покоління для схрещування
    схрестити 2 особини для одержання 2 нащадків
    обчислити придатність нащадків
    включити нащадків в нове покоління
```

```
    END

    IF популяція конвергувала THEN
    finished := TRUE
END

END
```

Перед тим, як використовувати ГА, необхідно розробити представлення (чи кодування) розв'язку задачі. Також потрібно визначити функцію придатності, яка кожному закодованому розв'язку присвоює деяке число – оцінку.

В даній реалізації ГА, рішення (хромосоми) будуть представлені деякими перестановками чисел від 1 до n, а якості функції придатності будемо використовувати цільову функцію задачі. Причому для різних задач величина n різна. Наприклад, для конвеєрної задачі n – це кількість робіт, а для загальної задачі теорії розкладів n – це кількість операцій з усіх робіт. Таке кодування хромосом дозволяє використовувати однакову структуру ГА для розв'язання різноманітних задач теорії розкладів. Звісно для кожної задачі слід змінювати функцію придатності. Далі розглянемо алгоритм виправлення недопустимих хромосом та алгоритми обчислення функцій придатності для конвеєрної та загальної задач теорії розкладів.

Алгоритм виправлення недопустимих хромосом після схрещування, де

n – довжина хромосоми;

k – номер елемента для схрещування (1 < k < n);

Hr[n], Hr1[n] – хромосоми для схрещування, масиви перестановок чисел від 1 до n;

Mas[n]=0, Mas1[n]=0 – допоміжні масиви, в яких відмічається повторення елементів у хромосомі;

Nom[n/2]=0, Nom1[n/2]=0 – допоміжні масиви, в яких зберігаються номери елементів, що повторюються;

S0. i=1, j=1, j1=1.
S1. Якщо Mas[Hr[i]]=0, то Mas[Hr[i]]=1, інакше Nom[j]=i, j=j+1.
S2. Якщо Mas1[Hr1[i]]=0, то Mas1[Hr1[i]]=1, інакше Nom1[j1]=i, j1=j1+1.
S3. i=i+1, якщо <=n, то перехід до S1.
S4. i=1, a=Hr[Nom[i]], Hr[Nom[i]]=Hr1[Nom1[i]], Hr1[Nom1[i]]=a.
S5. i=i+1, якщо i<=j то перехід до S4.
S6. Кінець.

Алгоритм обчислення функції придатності для конвеєрної задачі теорії розкладів, де

det – кількість деталей;

stank – кількість верстатів;

matr[det,stank] – матриця тривалостей виконання операцій;

sttime[stank] – масив часу закінчення роботи верстатів.

S0. i=1.
S1. j=1, sttime[1]=matr[hr[i],1].
S2. j=j+1, sttime[j]=max(fr[j-1],sttime[j])+matr[hr[i],j].
S3. Якщо j<=stank то перехід до S2.
S4. i=i+1, якщо i<=det то перехід до S1
S5. Rez=sttime[stank], кінець.

Алгоритм обчислення функції придатності для загальної задачі теорії розкладів, де

n – довжина хромосоми (<det*stank);

det – кількість деталей;

stank – кількість верстатів;

matr[det,stank] – матриця тривалостей виконання операцій;

sttime[stank] – масив часу закінчення роботи верстатів;

detfree[det] – масив часу закінчення останньої операції по деталях;
currop[det] – масив номерів текучих операцій по деталях;
seq[det?stank] – матриця послідовностей виконання операції.

S0. i=1.
S1. currdet=hr[i], currst=seq[currdet,currop[currdet]].
S2. Якщо sttime[currst]>detfree[currdet] то
sttime[currst]=sttime[currst]+dettime[currdet,[currop[currdet]].
інакше sttime[currst]=detfree[currdet]+dettime[currdet,[currop[currdet]].
S3. detfree[currdet]=sttime[currst], currop[currdet]=currop[currdet]+1.
S4. i=i+1, якщо i<=n то перехід до S1.
S5. Rez=max(detfree[i], i=1,det+), кінець.

4. Висновки

На базі розробленого програмного забезпечення проведено обчислювальні експерименти, у результаті яких було підтверджено збільшення швидкодії знаходження розв'язків задач теорії розкладів генетичними алгоритмами.

ЛІТЕРАТУРА:

1. *Grudmann W., Krause H.* Eine Verallgemeinerung des Verfahrens von Little u. A. Zur Losung speceller Touren probleme // 21 Int. Wiss. Kolog. Techn. Hochsch. Ilmenau. 1976. - Th. 4. S. 1, S.A. - P. 52–56.
2. *Конвей Р.В.,Максвелл В.Л., Миллер Л.В.* Теория расписаний. – М.: Наука, 1975. - 476 с.
3. Теория расписаний и вычислительные машины / Под ред. Э.Г. Коффмана. – М.: Наука, 1984. – 336 с.
4. *Танаев В.С., Шкурба В.В.* Введение в теорию расписаний. – М.: Наука, 1975. – 256 с.
5. *Михалевич В.С., Кукса А.И.* Методы последовательной оптимизации в дискретных сетевых задачах оптимального распределения ресурсов. – М.: Наука, 1984. – 334 с.
6. *Танаев В.С., Гордон В.С., Шафранский Я.М.* Теория расписаний. Одностадийные системы. – М.: Наука, 1984. – 365 с.
7. *Трауб Дж., Вожьньяковский Х.* Общая теория оптимальных алгоритмов. – М.: Мир, 1983. – 328 с.
8. *Гери М.Р., Джонсон Д.С.* Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.
9. *Пападмитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность: Пер. с англ. – М.: Мир, 1985. – 512 с.
10. *Foulds L.R.* The heuristic problem solving approach // I. Oper. Res. Soc. – 1983. – V.34, № 10. – P. 927–934.
11. *Stainton R.S., Papoulians D.B.* A fine facets frame for the design of heuristics – a rejoinder // Ibid. – P. 317–319.
12. *Muller-Merbatch H.* A fine facets frame for the design of heuristics – a rejoinder // Eur. I. Oper. Res. – 1984. – V.17, № 3. – P. 313–316.
13. *Golden B.L., Assad A.A.* A decision-theoretic framework for comparing heuristics // Eur. I. Oper. Res. – 1984. – V.18, № 2. – P. 167–171.
14. *Rinnoy Kan A.H.* An introduction to the analysis of approximation algorithms // Dser. Appl. Match. – 1986. – V.14, № 2. – P. 171–185.
15. *Levner E.* Complexity of heuristics for knapsack type problems: a survey / Operational Research-81. Proc. Of IX IFORS Internat. Confer. On Oper. Res. Hamburg, Germany. 1981. Amsterdam: North-Holland, 1981. Part II. P. 639–654.
16. Методы решения задач математического программирования и оптимального управления / Л.Г. Ащенков, Б.И. Белов, В.П. Булатов и др. – Новосибирск: Наука. Сиб. отд-ие, 1984. – 234 с.

17. *Михалевич В.С., Волкович В.Л.* Вычислительные методы исследования и проектирования сложных систем. – М.: Наука, 1982. – 286 с.
18. *Емеличев В.А., Комлик В.И.* Метод построения планов для решения задач дискретной оптимизации. – М.: Наука, 1981. – 208 с.
19. *Воеводин В.В.* Математические модели и методы в параллельных процессах. – М.: Наука, 1986. – 296 с.
20. *Титов В.В.* Оптимизация принятия решений в управлении производством. – Новосибирск: Наука, 1981. – 206 с.
21. *Сергиенко И.В., Рощин В.А.* О решении одной задачи целочисленного программирования специального вида // Оптимальное управление и методы оптимизации. – 1996. – № 5. – С. 16–22.

ІБРАГІМ Саад Алла – аспірант Житомирського державного технологічного університету.
Наукові інтереси:
- теорія розкладів;
- паралельні обчислення.

Подано 15.03.2004