

УДК 519.688

К.т.н., доцент Зорін Ю. М., магістрант Ажнюк Я. Б.

Національний технічний університет України
«Київський політехнічний інститут»

ГЕНЕТИЧНИЙ АЛГОРИТМ ДЛЯ ПЛАНУВАННЯ ПРОЕКТНИХ РОБІТ

Abstract

Yuriy Zorin, assoc. prof., PhD; Yaroslav Azhnyuk, student
Genetic Algorithms for Job-Shop Scheduling problem

A Job-Shop Scheduling problem is a NP-hard class problem for which there are no precise solution methods. So for the most part, different heuristic algorithms are used to solve this kind of problems – they allow to reduce the solution space and obtain satisfactory results within a rather short timeframe. After studying different heuristic approaches to the problem, such as, branch and bound method, simulated annealing algorithm, dynamic programming method, evolutionary approaches and genetic algorithm, the last one was chosen as a most promising approach for this class of problems.

Вступ

У роботі розглядається задача планування робіт, яка належить до класу NP-складних [1] і не має відомих точних методів розв'язку. Як правило для розв'язку цих задач використовують евристичні алгоритми, які дозволяють звужити область перебору і дають можливість отримати задовільний результат за розумний час.

Вивчивши різні підходи до розв'язку задачі за допомогою евристичних алгоритмів було вибрано генетичний алгоритм, як найбільш перспективний для розв'язку задач даного класу.

Постановка задачі

Метою роботи є дослідження та порівняння сучасних методів вирішення задачі планування робіт і побудова теоретичних моделей розв'язку, таких, що дозволяють безпосереднє відображення на програмну модель. Для досягнення мети в роботі вирішуються такі задачі:

1. Дослідження проблемної задачі та побудова її достовірної математичної моделі.

2. Розробка модифікацій генетичного алгоритму (ГА) з метою покращення отримуваних результатів.

Термінологія

Проект – це певний захід, що включає в себе визначений набір дій, виконання яких необхідно для досягнення цілей проекту. Проектом може бути розробка нового пристрою, випуск продукції, зведення будівлі та багато іншого.

Планування робіт — планування, координація та контроль робіт по проекту для досягнення його цілей в рамках установленого бюджету та термінів з відповідною якістю. Крім бюджету та термінів на проект можуть накладатися інші обмеження, такі як технологічні залежності між роботами, обмеженість ресурсів та бюджету, директивні терміни робіт та зміни, що виникають в ході реалізації проекту.

Робота у рамках проекту представляє певну діяльність, необхідну для досягнення конкретних результатів (скінчених продуктів нижнього рівня). Таким чином, робота є основним елементом проекту на нижньому рівні деталізації

Опис алгоритму

Розробка будь-якого ГА починається з вибору способу кодування розв'язку задачі, тобто виду хромосоми. Для задачі з n робіт, що виконуються на m машинах, всього існує $n \times m$ операцій, отже їх можна представити за допомогою списку довжиною $n \times m$. Кожен елемент списку містить вектор $\langle j, m, d \rangle$, де j є номер роботи, m – номер машини, а d – тривалість виконання роботи j на машині m . Таким чином, хромосома з варіантом розв'язку може бути представлена перестановкою порядкових номерів операцій. Всі можливі перестановки складають множину розв'язків.

Повний план робіт можна отримати шляхом декодування хромосоми. Переглядаючи її зліва направо спочатку отримуємо номер операції, а потім точну інформацію про цю операцію. Після цього, операція призначається до виконання у найближчий часовий проміжок на машині, що виконує відповідну роботу. Повний план робіт буде отримано після повторення цієї процедури для всіх операцій. Очевидно, що ресурсне обмеження можна легко задовільнити, призначаючи операції на часовий проміжок, коли і робота і машина готові до цього. Втім, технічне обмеження порівняно складно задовільнити. Тому, при декодуванні хромосоми, перевіряється дотримання ресурсного обмеження і не розглядається технічне обмеження.

Тут використана модифікація відомого алгоритму Гіффлера і Томпсона [2]. За цим алгоритмом, коли виконання деякої роботи призначається машині, роботи із коротшим часом виконання і скорішим початком виконання мають перевагу. Крім того беруться до уваги так звані критичні роботи. Якщо зміщення цієї роботи у часі вперед чи назад впливає на час закінчення виконання інших робіт, ця операція називається критичною.

Мутація є важливою операцією у генетичному алгоритмі. Вона використовується, щоб викликати пертурбації у хромосомах для підтримування різноманіття популяції. У роботі було використано три типи мутації - мутація інверсією, мутація обміном і мутація вставкою.

Цільова функція зі штрафом складається з оригінальної цільової функції $F(\bar{x})$ та зі штрафної функції $P(\bar{x})$. У задачі планування робіт метою є мінімізація загальної тривалості робіт. Для операції \bar{x} можна отримати повний план робіт π шляхом декодування хромосоми. Таким чином, оригінальну $F(\bar{x})$ можна визначити як

$$F(\bar{x}) = \max \{T[1], T[2], \dots, T[m]\}$$

де $T[i]$ – час закінчення роботи на i -ій машині, а $F(\bar{x})$ – час закінчення виконання всіх робіт.

Для задачі планування робіт розміру $n_c \times m$, існує n_c технічних обмежень. Перевіряючи π , можна легко отримати деталі роботи $\pi(j)$ для кожної з робіт j . Тоді можна обчислити функцію порушення обмеження $g_j(\bar{x})$ за формулою:

$$g_j(\bar{x}) = \sum_{k=1}^m (\delta_p(k) + \delta_s(k))$$

$\delta_p(k) = 1$ якщо порушення виникає через попередню операцію 0, у іншому випадку, $\delta_s(k) = 1$ якщо порушення виникає через наступну операцію 0, у іншому випадку.

Алгоритм закінчує роботу, коли отримано розв'язок, або коли виконано максимальну встановлену кількість ітерацій. Тобто, цільову функцію зі штрафом можна описати таким чином:

$$\tilde{F}(\bar{x}) = F(x) + v \cdot k^\alpha \cdot \left(\frac{n_{infeasible}}{n}\right)^\beta \cdot w \cdot \sum_{j=1}^m g_j^2(\bar{x})$$

Для перевірки запропонованого алгоритму, його було випробувано на наборі з 23 стандартних задач з OR-library [3].

Висновки

Завдяки застосуванню теоретичних та емпіричних даних вдалося розробити алгоритм, який працює на рівні з кращими відомими алгоритмами і експериментально показати дієвість таких підходів, як метод клональної селекції, метод стратегії довголіття, пошук у полі нежиттєздатних розв'язків з використанням штрафної функції, використання відбору за методом критичного шляху при генерації початкового розв'язку.

Під час досліджень були встановлені найкращі параметри для запропонованого алгоритму, а саме: розмір популяції – 50; параметр кроссоверу c – 15; параметр мутації m – 30; мінімальна тривалість життя minimum life – 20; розмір клональної бібліотеки – 30.

Література

1. *E.H.L. Aarts, P.J.M. van Laarhoven, J.K. Lenstra, N.L.J. Ulder.* A computational study of local search algorithms for job shop scheduling. *ORSA J. on Comput.* - 1994.- vol. 6, № 2, P. 118-125.
2. *J. Giffler, G.L. Thompson.* Algorithms for Solving Production Scheduling Problems.// *Operations Research.* - 1960.- Vol. 8, № 7, P. 487-503.
3. *J.E. Beasley.* OR-library: distributing test problems by electronic mail. // *Jornal on Operational Research.*- 1990. - Vol. 11, № 4, P. 1069–1072.