УДК 004.67

І.С. ДМИТРІЄВА, Н.М. КАРАСЬ
Національна металургійна академія України

КОМБІНАТОРНИЙ ТА ГЕНЕТИЧНИЙ АЛГОРИТМИ ПОШУКУ РОЗКЛАДУ ОБСЛУГОВУВАННЯ БАГАТОПРИБОРНОЇ СИСТЕМИ

Робота присвячена вирішенню задачі побудови розкладу обробки деталей приладами. Пропонується використовувати два алгоритми для вирішення задачі розкладу обслуговування багатоприборної системи: комбінаторний та генетичний.

Ключові слова: комбінаторний алгоритм, генетичний алгоритм, задачі побудови розкладу.

И.С. ДМИТРИЕВА, Н.М. КАРАСЬ
Национальная металлургическая академия Украины

КОМБИНАТОРНЫЙ И ГЕНЕТИЧЕСКИЙ АЛГОРИТМЫ ПОИСКА РАСПИСАНИЯ ОБСЛУЖИВАНИЯ МНОГОПРИБОРНОЙ СИСТЕМЫ

Работа посвящена решению задачи построения расписания обработки деталей приборами. Предлагается использовать два алгоритма для решения задачи расписания обслуживания многоприборной системы: комбинаторный и генетический.

Ключевые слова: комбинаторный алгоритм, генетический алгоритм, задачи построения расписания.

I. DMYTRIIEVA, N. KARAS
National Metallurgical Academy of Ukraine

COMBINATORIAL AND GENETIC ALGORITHMS FOR SERVICE SCHEDULE LOOKUP IN A MULTI-INSTRUMENT SYSTEM

The work is devoted to solving the problem of constructing parts processing schedules. It is proposed to use two algorithms for solving the problem of maintenance schedules in multi-instrument systems: combinatorial and genetic.

Keywords: combinatorial algorithm, genetic algorithm, the task of building schedules.

Постановка проблеми

В даний час вирішення багатьох практичних завдань складання розкладу, як і інші задачі комбінаторної оптимізації, відноситься до класу NP-повних, рішення яких пов'язано з великими часовими витратами. Побудовані в даний час алгоритми складання розкладів (планування), засновані на методі динамічного програмування, симплекс-методі, методі гілок і меж, різних евристичних підходах, дозволяють знайти прийнятні за якістю та тимчасовим витратам рішення тільки для задач невеликої розмірності.

У цій статті пропонується використовувати два алгоритми для вирішення задачі розкладу обслуговування багатоприборної різнорідної системи: комбінаторний та генетичний.

Задачі синтезу розкладів формулюються як пошук оптимального розподілу множини робіт в часі між обслуговуючими апаратами (приладами). Ці задачі мають численні додатки, до яких, перш за все, слід віднести планування виробництва, розподіл ресурсів, проектування технологічних процесів.

Комбінаторний алгоритм

Постановку побудови розкладу на паралельних машинах можна задати наступним чином: є множина деталей та набір однакових приладів для їх обробки. Кожна деталь має бути оброблена на кожному з приладів за різний час [1].

Нехай $J = \{J_1, J_2, ..., J_n\}$ – множина деталей, а $M = \{M_1, M_2, ..., M_m\}$ – множина приладів, p_{ij} – час обробки деталі J_j на приладі M_i, $i = 1..m$, $j = 1..n$.

Роботи, які необхідно виконати з деталями, складаються з операцій: $J_j = \{O_{1j}, O_{2j}, ...O_{mj}\}$. Для виконання операції O_{ij} необхідно p_{ij} j часу та може виконуватись на одному з приладів з множини M .

Під розкладом маємо на увазі призначення кожної деталі на деякий верстат в кожен момент часу. Розклад є допустимим, якщо:

- Кожна деталь оброблена повністю,
- Кожна деталь пройшла обробку на кожному з приладів,
- Ніякий верстат не виконує одночасно більше одної операції,
- Жодна з деталей не виконується одночасно більше ніж на одному приладі,

- Час виконання всіх робіт не повинен бути менше тривалості виконання робіт.

Задача побудови розкладу приймає наступний вид:

$$C_{\max} = \max(c_i) \to \min$$

$$\sum_{j=1}^{n} x_{ij} p_j \le C_{\max},\ i = 1..m;$$

$$\sum_{i=1}^{m} x_{ij} = 1,\ j = 1..n;$$

$$p_j \le C_{\max},\ j = 1..n.$$

де c_i — час виконання всіх робіт на i-му приладі;

$$x_{ij} = \begin{cases} 1, \ \textit{якщо робота } J_j \textit{ призначена на } M_i; \\ 0, \textit{ в іншому випадку}. \end{cases}.$$

Обмеження задачі говорять про те, що час виконання обробки усіх деталей має бути мінімальним, сумарна тривалість виконання робіт на кожному з приладів не перевищує $C_{\max}$, кожна робота виконана на кожному приладі.

Виконавши деякі перетворення, отримаємо оцінку:

$$C_{\max} \ge \frac{1}{m\sum_{j=1}^{n} p_j}$$

та існують допустимі розв'язки задач.

Але, як було зазначено раніше, комбінаторні задачі відноситься до класу NP-повних. Тому, для прискорення часу пошуку розв'язку даної задачі при великій кількості даних, нами було запропоновано застосування генетичного алгоритму.

Генетичний алгоритм (ГА)

Основну увагу приділено розгляду способів подання потенційних рішень для задачі синтезу розкладів і демонстрації на прикладах різних модифікацій операторів схрещування і мутації, складання плану-графіка робіт [2].

Незважаючи на те що в даний момент для вирішення різних прикладних задач вже створено велику кількість алгоритмів на основі моделювання деяких властивостей генетичного апарату пристосування живих організмів до навколишнього середовища, всім цим алгоритмам присутні загальні риси. У всіх генетичних алгоритмах можна виділити однакові основні етапи вирішення завдань. Розглянемо ці етапи більш детально.

Попередньо зауважимо, що при описі генетичного алгоритму прийнято використовувати термінологію, запозичену з молекулярної біології і генетики, що ще раз підкреслює аналогію з еволюційними процесами, що відбуваються в живій природі. Робота будь-якого ГА починається з кодування потенційних рішень, яке полягає у формуванні хромосоми. Хромосоми складаються з генів. У загальному випадку кожен ген може приймати речові, цілочисельні або бінарні значення і характеризується паралельною формою (безліччю допустимих значень). Так, наприклад, якщо вирішується завдання з n параметрами (генами), кожен з яких кодується бітами, то хромосома, представлена рядком довжиною $m-n$ бітів, кодує всі можливі потенційні рішення задачі. Кожен з варіантів вирішення завдання оцінюється за допомогою деякої функції пристосованості (цільової функції, функції фітнесу) [3].

Для скорочення кількості варіантів рішень задачі, які переглядаються алгоритмом, Голдберг висунув гіпотезу, що лежить в основі генетичних алгоритмів, про будівельні блоки. Відповідно до цієї гіпотези, генетичні алгоритми повинні одночасно виконувати дві функції: вирощування будівельних блоків рішень; змішування цих блоків з метою отримання оптимального рішення. В ГА ця гіпотеза реалізується такими механізмами і операторами:

- механізм генерації початкової популяції,
- механізм оцінки якості хромосоми з використанням функції пристосування,
- механізм селекції,
- оператори схрещування,
- оператори мутації,
- механізм зупинки ГА (еволюційного процесу).

Класичні генетичні алгоритми оперують з хромосомами постійної довжини, що складаються з генів зі значеннями 0 або 1. Закодована хромосома, в якій представлено безліч потенційних рішень, називається генотипом. Генотип задає пошуковий простір рішень. У теорії природної еволюції генотип - це програма розвитку і еволюції особини. Реалізація особини (одного з можливих варіантів вирішення задачі) у вигляді хромосоми з конкретними значеннями генів являє собою фенотип. В теорії природного відбору фенотип необхідний для селекції і переходу на наступний рівень еволюції. Популяція - його репродукційна група хромосом фіксованою чисельності, в якій будь-які дві хромосоми X_j, $X_j \in G$, $i \neq j$ можуть виступати в ролі батьків.

Схрещування батьківських хромосом моделює передачу спадковості нащадкам. В результаті схрещування утворюються дочірні хромосоми. Такий механізм в ГА реалізується оператором схрещування. Цей оператор може застосовуватися не до всіх хромосом в популяції. Його застосуванню передує вибір батьківських пар. Результат застосування оператора схрещування полягає в обміні частини генетичного коду між двома хромосомами батьків. Схрещування проводиться з метою породження з наявної безлічі рішень (популяція батьків) нової безлічі рішень (популяція нащадків) Нагадаємо, що за гіпотезою Холланда хороші схеми рішень при схрещуванні народжують нові схеми більш високої якості. Один з класичних варіантів оператора схрещування - запропоноване Холландом одноточечне схрещування, при якому випадковим чином вибирається число $n \in \{1, 2, ..., L-1\}$, де L - довжина хромосоми, n - точка схрещування. Потім утворюються дві нові дочірні хромосоми шляхом обміну всіх генів між двома батьківськими хромосомами, починаючи з $(n+1)$-го до L-го включно.

Мутація хромосом в класичному генетичному алгоритмі складається в інвертуванні символу в випадково обраному гені. Використання операторів мутації ефективно для виходу з локального екстремуму. Цей механізм може застосовуватися для популяції як батьків, так і нащадків.

Перевірка того, наскільки хорошим є i-е рішення задачі, здійснюється за допомогою обчислення функції пристосованості $FP(X_i)$ для хромосоми X_i. Зазвичай функція пристосованості в явному вигляді містить критерій оптимізації розв'язуваної задачі. Для селекції хромосом у популяційній групі розміру N часто використовують нормалізовану функцію пристосованості

$$FP_{norm}(X_i) = \frac{FP(X_i)}{\sum_{i=1}^{N} FP(X_i)}.$$

Селекція хромосом полягає в їх відборі та формуванні наступної популяції. Селекція - це випадковий процес, при якому керуються правилом: чим більше значення функції пристосованості має дана хромосома, тим вище ймовірність її вибору для репродукції. Найпростіший і найбільш популярний метод селекції хромосом - метод рулетки.

Механізм зупинки алгоритму визначається на етапі його проектування з урахуванням природних обмежень, пов'язаних з часом розрахунку на ЕОМ, або кількості популяцій, при яких значення функції пристосованості перестає поліпшуватися. Отримана в результаті n ітерацій хромосома з найбільшим значенням функції пристосованості приймається як рішення даної задачі. Однак немає гарантії, що це рішення - найкраще.

Перед застосуванням генетичного алгоритму необхідно представити завдання у вигляді хромосоми, після чого стає можливим застосування самого алгоритму.

Генетичний алгоритм включає наступні етапи:

1. Генерування початкової популяції.
2. Обчислення значень функції пристосованості FP хромосом популяції.
3. Селекція хромосом, призначених для репродукції.
4. Навчання за допомогою операторів мутації і схрещування нової популяції нащадків.
5. Популяція батьків знищується, а популяція нащадків стає новою популяцією батьків.
6. Перевірка умови кінця дії алгоритму (якщо «так» - то етап 7, «ні» - етап 2).
7. Найкраща хромосома з поточної популяції приймається як рішення задачі.

Розглянемо на прикладі основні моменти роботи генетичного алгоритму.

Припустимо, що слід зробити дві роботи J, кожна з яких складається з трьох операцій, що виконуються на трьох приладах. Введемо позначення $S_j(t_j)$, де t_j - трудомісткість операції O_j на приладі. У термінах введеного позначення завдання формулюється так:

$$J_1 \; 1(10) \; 3(7) \; 2(20)$$

$$J_2 \; 2(15) \; 1(2) \; 3(8).$$

Це означає, що операція O_1 роботи J_1 виконується на S_1 і $t_1 = 10$ і т.д.

Для подання рішення (відображення плану-графіку H) цього завдання, необхідна хромосома, що складається з 6 генів (6 - сумарна кількість операцій для всіх робіт). Кожен ген може прийняти значення 1 або 2. Нехай хромосома, яка представляє рішення, має вигляд:

1 2 2 2 2 1.

Інтерпретація цієї хромосоми наступна:

- помістити в H першу операцію O_1 першої незакінченої роботи J_1;
- помістити в H першу операцію O_1 другої незакінченої роботи J_2;
- помістити в H другу операцію O_2 другої незакінченої роботи J_2;
- помістити в H третю операцію O_3 другої незакінченої роботи J_2;
- помістити в H четверту фіктивну операцію O_4 другої незакінченої роботи J_2;
- помістити в H другу операцію O_2 першої незакінченої роботи J_1.

Спочатку, перед вставкою операції в план-графік, маємо циклічний список CL_0 робіт і хромосому X_0:

$$CL_0 = [1-(1,3,2);2-(2,1,3)],\ X_0 = [1,2,2,2,2,1].$$

Так як перший ген в хромосомі X_0 визначає роботу J_1, беремо першу операцію O_1 роботи J_1 і вставляємо її в план-графік. В результаті отримуємо:

$$CL_1 = [1-(3,2);2-(2,1,3)], \quad S_1: 1111111111$$
$$X_1 = [2,2,2,2,1], \quad S_2:$$
$$S_3:$$

Наступний ген в хромосомі X_0 (перший в X_1) визначає роботу J_2. Отже беремо першу операцію O_1, роботи J_2 і вставляємо її в план-графік. В результаті отримуємо:

$$CL_2 = [1-(3,2);2-(1,3)], \quad S_1: 1111111111$$
$$X_2 = [2,2,2,1], \quad S_2: 222222222222222$$
$$S_3:$$

Наступний ген в хромосомі X_0 (перший в X_2) визначає роботу J_2. Отже беремо першу операцію O_2, роботи J_2 і вставляємо її в план-графік. В результаті отримуємо:

$$CL_3 = [1-(3,2);2-(3)], \quad S_1: 1111111111\,22$$
$$X_3 = [2,2,1], \quad S_2: 222222222222222$$
$$S_3:$$

Наступний ген в хромосомі X_0 (перший в X_3) визначає роботу J_2. Отже беремо першу операцію O_3, роботи J_2 і вставляємо її в план-графік. В результаті отримуємо:

$$CL_4 = [1-(3,2);2-()], \quad S_1: 1111111111\,22$$
$$X_4 = [2,1], \quad S_2: 222222222222222$$
$$S_3: 22222222$$

Наступний ген визначає роботу J_2. Однак в списку незакінчених робіт друга робота вже не числиться. Застосовуємо список незакінчених робіт як циклічний, тоді цей ген означає роботу J_1; додаємо його в план-графік як роботу 1 на приладі S_3 з трудомісткістю 7 одиниць часу. Отримуємо

$$CL_5 = [1-(2);2-()], \quad S_1: 1111111111\,22$$
$$X_5 = [1], \quad S_2: 222222222222222$$
$$S_3: 1111111\,22222222$$

Останньою вставляємо 3-ю операцію роботи J_1. В результаті отримуємо:

$$CL_6 = [1-();2-()], \quad S_1: 1111111111\,22$$
$$X_6 = [], \quad S_2: 222222222222222\,111111111111111111$$
$$S_3: 1111111\,22222222$$

Список незакінчених робіт порожній. Остаточно отримуємо побудований план-графік.

Для реалізації побудови розкладу як комбінаторним, так і генетичним алгоритмом, нами було розроблено додаток. Результати роботи цього додатку представлено на рисунках 1 та 2.

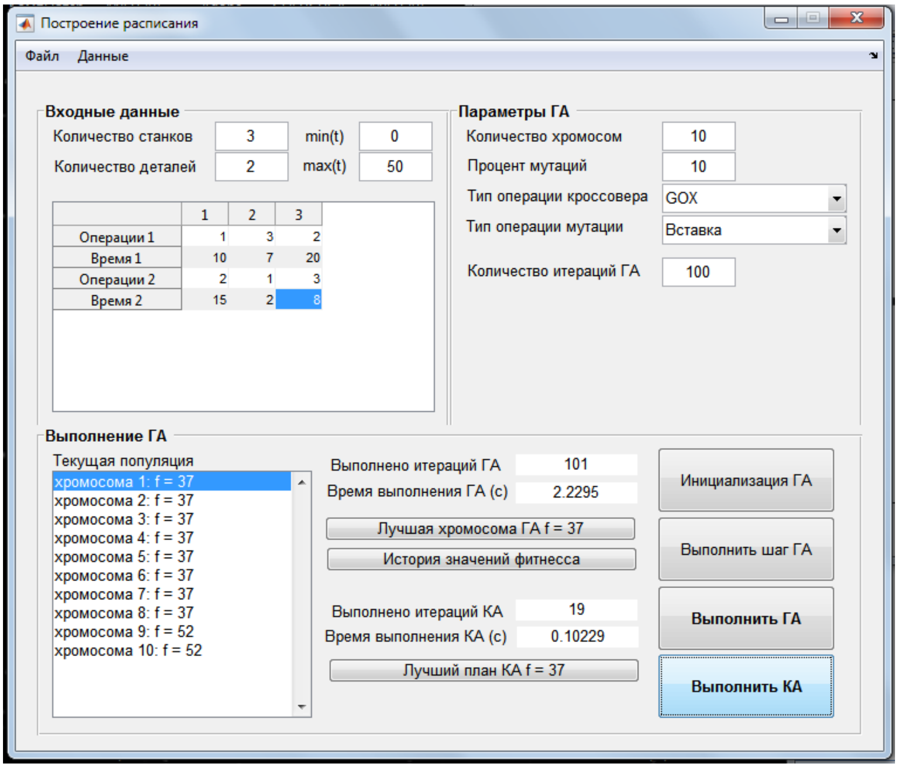

Рисунок 1 – Параметри та результати побудови розкладу за допомогою комбінаторного та генетичного алгоритмів

Побудований розклад для обох алгоритмів співпадає.

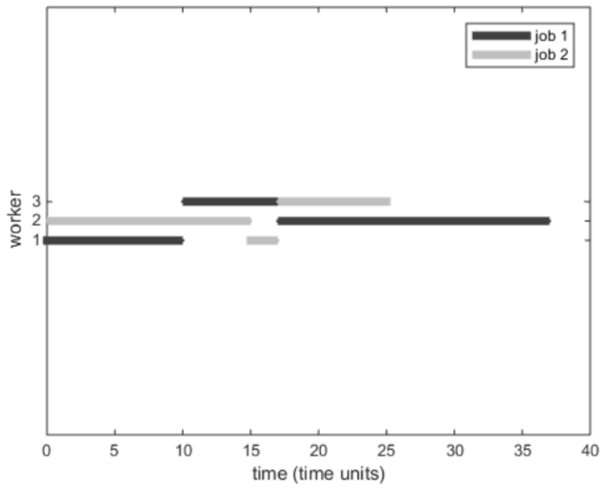

Рисунок 2 - Побудований розклад за допомогою генетичного алгоритму

Висновки

В роботі розглянуто комбінаторний та генетичний алгоритми для вирішення задачі розкладу обслуговування багатоприборної різнорідної системи. Розроблено програмний додаток, який реалізує ці алгоритми. Подальші дослідження будуть пов’язані з модифікацією генетичного алгоритму, та оптимізацією його для розв’язання конкретного типу задач.

Список використаної літератури

1. Танаев В.С., Сотсков Ю.Н., Струсевич В.А. Теория расписаний. Многостадийные системы, М.:Наука, 1989.
2. Schafer J.D. Podstawy geneticznej optymalizacij globalnej. - Krakow, Wyd. Uniwersytet Jagiellonski, 2002. – 245 p.
3. Витовски Т., Антчак А. Генетические алгоритмы – современный инструмент поиска квазиоптимальных решений // Проблемы управления и информатики. – 2003. - №5. – С.22-35.