

**ІНФОРМАТИКА, ОБЧИСЛЮВАЛЬНА ТЕХНІКА
ТА АВТОМАТИЗАЦІЯ**

УДК 681.3

**В.В. Арешкова, нач. відд.
О.М. Данильченко, к.т.н., доц.
С.А. Ібрагім, аспір.***Житомирський інженерно-технологічний інститут***ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ СКЛАДАННЯ
РОЗКЛАДІВ ДЛЯ МУЛЬТИПРОЦЕСОРНОЇ ПОСЛІДОВНО-ПАРАЛЕЛЬНОЇ СИСТЕМИ**

В даній роботі розглянуто один з методів вирішування задач складання розкладів для послідовно-паралельної системи за допомогою генетичних алгоритмів. Наведено алгоритм розв'язання задачі, описано застосування генетичних операторів для даного типу задач, а також результати обчислювального експерименту. Показано, що застосування генетичних алгоритмів для даного типу задач дає змогу отримувати оптимальні або близькі до оптимальних рішення за достатньо невеликий інтервал часу.

Задачі теорії розкладів традиційно вирішувалися або точними алгоритмами типу гілок і меж, або евристичними засобами. Розв'язання точними засобами вимагає не виправдано великих витрат часу, а застосування евристичних алгоритмів часто призводить до рішень, далеких від оптимальних.

Як альтернативу зазначеним підходам пропонується для рішення задач складання розкладів використовувати так звані генетичні алгоритми (ГА). Генетичні алгоритми – це адаптивні методи, що можуть використовуватися для розв'язання пошукових та оптимізаційних задач. Вони побудовані на основі генетичних процесів біологічних організмів. В наслідування процесу еволюції живих організмів генетичні алгоритми здатні виробляти "еволюцію" рішень реальних задач, якщо останні прийнятно закодовані. ГА використовують аналогію з природною поведінкою. Вони працюють з популяцією хромосом, кожна з яких є можливим розв'язанням деякої задачі. Кожній хромосомі привласнюється значення придатності у відповідності до того, наскільки добрим рішенням задачі вона є. Високопридатним хромосомам надається можливість розмножуватися за допомогою схрещування з іншими хромосомами в популяції. В результаті одержуються нащадки, що успадковують деякі властивості батьків. Менш придатні хромосоми в популяції мають меншу ймовірність бути вибраними для розмноження і таким чином вимирають.

Вся нова популяція можливих рішень, таким чином, створюється шляхом вибору найбільш придатних хромосом поточної популяції і схрещування їх для утворення нового набору хромосом. Такий підхід називається повною заміною популяції, саме він використовувався в нашому алгоритмі. В дійсності термін "повна заміна популяції" не є абсолютно точним, оскільки ймовірність схрещування не завжди дорівнює 1, тому для деяких пар хромосом воно може не застосовуватись, а нащадки можуть утворюватись простим копіюванням батьків.

У природі еквівалентом повної заміни популяції можуть служити всі ті види, в яких батьки не конкурують за ресурси з нащадками. Нова популяція містить більшу кількість характеристик, притаманних добрим членам попередньої популяції. Таким чином, через багато поколінь добрі якості розповсюджуються серед популяції, по ходу змішуючись та обмінюючись з іншими добрими якостями. Підтримуючи схрещування найбільш придатних хромосом, ми досліджуємо найбільш багатобіччючі області пошукового простору. Якщо ГА був розроблений добре, популяція конвертує до оптимального рішення задачі (можна сказати, що задача конвертувала, якщо не менше 95 % її хромосом співпадають).

Основні принципи ГА були вперше точно сформульовані Холландом в його роботі "Адаптація в природних та штучних системах" [1]. Класичний ГА був описаний в багатьох джерелах, наприклад, в роботі Д. Біслей, Д. Булл і Р. Мартін "Огляд генетичних алгоритмів" [2]. Наведемо цей алгоритм:

1. Згенерувати початкову популяцію.
2. Обчислити придатність для кожної хромосоми популяції.
3. Вибрати 2 хромосоми зі старої популяції для схрещування за допомогою одного з алгоритмів відбору.
4. Провести схрещування 2 хромосом для отримання нащадків.
5. Включити нащадків у нову популяцію.
6. Якщо створення нової популяції не закінчено, перейти до п. 3. Інакше перейти до п. 7.
7. Якщо популяція не конвертувала, перейти до створення нової популяції (п. 3). Інакше завершити роботу генетичного алгоритму.

В даній роботі розглянемо застосування ГА для розв'язання задачі теорії розкладів у такій

постановці.

Нехай є деяка система, призначена для обробки n завдань, кожне з яких полягає у виконанні k послідовно-паралельних етапів обробки, причому кожне завдання повинно обов'язково виконуватись на кожному етапі. Кожний послідовний етап представлений одним процесором, на якому завдання виконуються в порядку їхнього надходження, а паралельні етапи представлені m процесорами, причому завдання можуть виконуватись на будь-якому вільному процесорі (рис. 1). Вимагається визначити порядок виконання завдань на кожному процесорі, при якому загальний час обробки завдань буде мінімальним, дотримуючись при цьому наступних обмежень:

1. Жоден процесор не може виконувати водночас більше одного завдання.
2. Жодне завдання не може оброблятися водночас на більш ніж одному процесорі.
3. Кожне завдання повинно оброблятися на деякому процесорі не більш одного разу.
4. Неприпустимі переривання в обробці завдань.

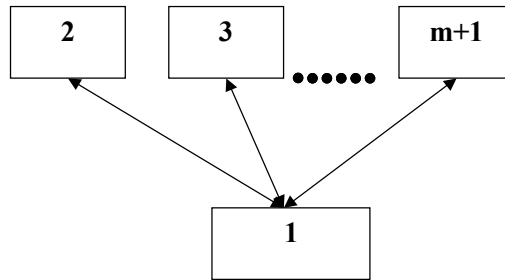


Рис. 1. Схема функціонування системи

Будемо вважати, що дана система починається з послідовного етапу, який значною мірою визначає порядок подальшої обробки завдань на інших процесорах системи.

Перш ніж ГА може бути виконаний, необхідно розробити відповідне кодування (або подання) задачі. Нам буде необхідна також функція придатності, що присвоює кожному закодованому рішенню деяке число – оцінку. Розмір популяції визначає користувач. Рекомендується задавати достатньо великі розміри популяції для того, щоб поширити простір пошуку і завдяки цьому збільшити ймовірність конвергенції популяції до оптимального рішення.

Визначимо хромосому як вектор, що визначає порядок обробки завдань на деякому процесорі. Довжина хромосоми у цьому випадку буде дорівнювати кількості завдань n , де кожний ген хромосоми містить номер завдання, що обробляється, а номер гена є номером завдання в черзі на обробку. За придатність хромосоми можна використовувати різноманітні функції, що призначають кожній хромосомі деяке число у відповідності до максимального часу завершення обробки завдань; в даному алгоритмі використовується величина, зворотна значенню загального часу обробки завдань.

Основною складністю застосування ГА для розв'язання задач цього класу є необхідність стежити за тим, щоб не порушувалось жодне з обмежень (1). Оскільки застосування генетичних операторів в їх класичному вигляді призводить до порушення цих обмежень, що, в свою чергу, робить розв'язання задачі, отримане за допомогою ГА, недопустимим, склалася необхідність модифікувати генетичні оператори так, щоб після їх застосування проводилось відповідне корегування отриманих хромосом.

Хромосоми, які задають порядок обробки на першому етапі, згенеруємо випадковим чином, стежачи за тим, щоб вони не були суперечливими (тобто щоб вони задовольняли всім обмеженням (1)). Для того, щоб хромосома задовольняла цим обмеженням, вона має містити усі можливі гени (завдання), при цьому кожен ген матиме тільки одне входження в хромосому.

Для розв'язання цієї задачі пропонується такий алгоритм.

Алгоритм 1. Загальний алгоритм розв'язання задачі.

1. Ввід користувачем початкових даних (кількість хромосом в популяції, кількість процесорів на кожному етапі, кількість завдань та час обробки завдань на відповідних процесорах, вид відбору батьківських хромосом, ймовірності застосування генетичних операторів, максимальну кількість поколінь тощо).

2. Згенерувати початкову популяцію, стежачи за тим, щоб не порушувалися обмеження (1).

3. Обчислити придатність кожної хромосоми в популяції.

4. Вибрати з популяції хромосоми для проведення над ними генетичних операцій.

5. Застосувати генетичні оператори схрещування та мутації до відібраних хромосом. Ці оператори відрізняються від класичних тим, що після їх використання застосовується корегування суперечних хромосом з метою отримання допустимих розв'язань задачі.

6. Обчислити придатність нащадків.

7. Скопіювати нащадків у нову популяцію.

8. Якщо кількість хромосом у новій популяції не дорівнює їх кількості в старому, перейти до п. 4.

Інакше перейти до п. 9.

9. Перевірити, чи конвертувала популяція. Якщо так, то алгоритм завершено. Якщо ні, скопіювати нову популяцію в стару та перейти до п. 4.

Оскільки найважливішим для нас є порядок обробки завдань на першому етапі (бо саме він в тій чи іншій мірі визначає подальший порядок обробки), обмежимося проведенням генетичних операцій тільки над хромосомами, які визначають порядок обробки завдань на першому етапі. На інших етапах завдання будуть виконуватись в порядку, який буде згенеровано порядком обробки на першому етапі.

Для отримання нового покоління хромосом в ГА відбувається відбір хромосом і їх схрещування. Батьківські хромосоми вибираються випадковим чином за допомогою схеми, що сприяє вибору найбільш придатних особин. В даному алгоритмі запропоновано три схеми відбору: турнірний відбір, відбір за допомогою колеса рулетки, пропорційний відбір [4]. Наведемо ці алгоритми.

Алгоритм 2. Турнірний відбір.

1. Визначення користувачем об'єму турніру T .
2. Вибір випадковим чином T хромосом з популяції.
3. Із відібраних хромосом вибрати одну, яка має найбільшу придатність, та скопіювати її в допоміжний пул.
4. Якщо кількість хромосом, які містяться в допоміжному пулі, не дорівнює розміру популяції, то перейти до п. 2. Інакше завершити роботу алгоритму.

Алгоритм 3. Пропорційний відбір.

1. Підрахуємо суму S придатностей всіх хромосом популяції.
2. Для кожної хромосоми підрахуємо приведену придатність, яка буде дорівнювати придатності хромосоми, поділеної на S .
3. Розіб'ємо інтервал $[0;1]$ на інтервали, відповідно до значень приведених придатностей хромосом.
4. Згенеруємо випадкове число, яке належить інтервалу $[0;1]$.
5. Хромосому, яка відповідає тому інтервалу, на який попало згенероване число, поміщуємо у допоміжний пул.
6. Якщо кількість хромосом в допоміжному пулі ще не дорівнює об'єму популяції, то перейти до п. 4. Інакше завершити роботу алгоритму.

Алгоритм 4. Відбір за допомогою колеса рулетки.

1. Обчислити ймовірності відбору всіх хромосом популяції таким чином: ймовірність хромосоми дорівнює її придатності, поділеної на найкращу придатність в популяції.
2. Вибрати випадковим чином одну хромосому.
3. Згенерувати випадкове число, яке лежить в інтервалі $[0;1]$.
4. Якщо ймовірність вибору даної хромосоми є більшою або дорівнює згенерованому числу, то хромосома буде відібрана в наступне покоління; якщо ж менше, то хромосома вибрана не буде.
5. Якщо розмір нової популяції не дорівнює розміру старої, то перехід до п. 2. Інакше завершити роботу алгоритму.

Після того, як за допомогою алгоритмів відбору були відібрані батьківські хромосоми, підберемо випадковим чином пари для застосування схрещування. В даному алгоритмі застосовується так зване одномісне схрещування, що функціонує таким чином: дві відібрані батьківські хромосоми розрізаються у деякій випадково вибраній позиції, таким чином ми отримуємо два головних та два хвостових сегменти. Після цього відбувається обмін кінцевими сегментами. Два нащадки успадковують деякі гени від кожного з батьків. Оператор схрещування застосовується не до всіх пар хромосом, відібраних для розмноження. Робиться випадковий вибір, де ймовірність застосування схрещування можна вибрати з діапазону від 0.6 до 1.0. Якщо схрещування не застосовується, то нащадки утворюються простим копіюванням батьків. Це дає хромосомам змогу передати свої гени нащадкам в незмінному вигляді. Наведемо алгоритм одномісного схрещування:

Алгоритм 5. Одномісне схрещування.

1. Ввід користувачем ймовірності схрещування (рекомендується вибирати ймовірність схрещування з інтервалу $[0,6;1]$).
2. З допоміжного пулу, який містить відібрані хромосоми, випадковим чином вибираються дві хромосоми.
3. Генерується випадкове число, що належить інтервалу $[0;1]$.
4. Якщо згенероване число є більшим, ніж ймовірність схрещування, то обидві хромосоми переносяться в нове покоління без змін (тобто без застосування оператора схрещування) і перехід до п. 6.
5. Інакше (якщо згенероване число є меншим або дорівнює ймовірності схрещування) згенеруємо випадковим чином позицію розтину і обміняємо місцями кінцеві сегменти батьківських хромосом; після чого перенесемо отримані хромосоми в нову популяцію, перехід до п. 6.
6. Видаляємо з допоміжного пулу хромосоми, з якими ми працювали.
7. Якщо в допоміжному пулі ще є хромосоми, то перейти до п. 2. Інакше завершити роботу

алгоритму.

Алгоритм двомісного схрещування відрізняється від наведеного тільки тим, що замість однієї позиції розтину генеруються дві, після чого проводиться обмін серединними сегментами батьківських хромосом.

Оператор мутації застосовується індивідуально до кожного нащадка після схрещування. Він випадковим чином змінює кожен ген з малою ймовірністю (в даному алгоритмі використовується класичне значення ймовірності мутації, яке дорівнює 0.001).

Алгоритм 6. **Мутація.**

1. Візьмемо перший ген хромосоми.
2. Згенеруємо випадкове число, яке належить інтервалу $[0;1]$.
3. Якщо це число є меншим, аніж 0.001, то змінимо відповідний ген на інший, вибраний випадковим чином.
4. Якщо залишилися ще нерозглянуті гени хромосоми, то перехід до п. 2. Інакше завершити роботу алгоритму.

Оскільки після застосування операторів схрещування і мутації ми отримуємо, в загальному випадку, суперечні хромосоми (і відповідно неприпустимі рішення), то їх необхідно скорегувати. Для цього пропонується наступний алгоритм.

Алгоритм 7. **Корегування суперечних хромосом.**

1. Виберемо одну з двох суперечних хромосом, отриманих в результаті застосування генетичних операторів. Якщо застосовувалось і схрещування, і мутація, виберемо для корегування ту частину хромосоми, в якій мала місце мутація. Якщо мутація не застосовувалась, виберемо для корегування кінцеву частину хромосоми.

2. Розглядаючи вибрану для корегування частину першої хромосоми і протилежну їй частину другої хромосоми, підрахуємо кількість подвоєних елементів та зафіксуємо, які елементи відсутні.

3. Випадковим чином змінимо кожен з подвоєних елементів на відсутні гени, стежачи за тим, щоб значення нових генів не дублювали значень, які вже є присутніми в хромосомі.

Застосування гнучкої техніки операторів схрещування і мутації в даному ГА дозволяє одержувати прийнятні рішення за невеликий час. Так, для задачі вимірності 100×10 (100 завдань, 10 процесорів) оптимальні рішення були отримані в 92 % випадків за час, що не переважає 25 хвилин. Слід помітити, що час отримання оптимального рішення суттєво залежить від вибору і розміру початкової популяції хромосом. В цьому напрямленні уявляється перспективним використання спеціальних засобів для отримання початкової популяції: наприклад, генерування частини популяції за допомогою евристичних алгоритмів, а не випадковим чином.

ЛІТЕРАТУРА:

1. *J. H. Holland.* Adaptation in Natural and Artificial Systems. MIT Press, 1975.
2. *D. Beasley, D. R. Bull, R. R. Martin.* An Overview of Genetic Algorithms: Part I, Fundamentals. University Computing, 15 (2): 58–69, 1993.
3. *D. Beasley, D. R. Bull, R. R. Martin.* An Overview of Genetic Algorithms: Part 2, Research Topics. University Computing, 15 (2): 58–69, 1993.
4. *T. Blicke, L. Tiele.* Mathematical Analysis of Tournament Selection.
5. *Hsiao-Lan Fang.* Investigating Genetic Algorithms for Scheduling. MSc Dissertation, Dept. of Artificial Intelligence, University of Edinburgh: 116pp., 1992.

АРЄШКОВА Вікторія Володимирівна – пошукач, начальник відділу міжнародних відносин Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- генетичний алгоритм;
- теорія розкладів.

ІБРАГІМ Саад Алла – аспірант Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- системи керування базами даних;
- теорія розкладів.

ДАНИЛЬЧЕНКО Олександр Михайлович – кандидат технічних наук, доцент, завідувач кафедри програмного забезпечення обчислювальної техніки Житомирського інженерно-технологічного інституту

Наукові інтереси:

- теорія розкладів;

– комп'ютерні інформаційні технології.

Подано 17.04.2002

Арешкова В.В., Данильченко О.М., Ібрагім С.А. Застосування генетичних алгоритмів для розв'язання задач складання розкладів для мультипроцесорної послідовно-паралельної системи

Арешкова В.В., Данильченко О.М., Ібрагім Саад Алла Применение генетических алгоритмов для решения задач составления расписаний для мультипроцессорной последовательно-параллельной системы.

Areshkova V.V., Daniltchenko A.M., Ibragim Caad Alla. Application genetic algorithms for problem solving of scheduling of serial-parallel multiprocessor system

УДК 681.3

Применение генетических алгоритмов для решения задач составления расписаний для мультипроцессорной последовательно-параллельной системы / Арешкова В.В., Данильченко О.М., Ібрагім Саад Алла

В данной работе рассмотрено один из методов решения задач составления расписаний для последовательно-параллельной системы при помощи генетических алгоритмов. Приведен алгоритм решения задачи, описано применение генетических операторов для данного типа задач, а также результаты вычислительного эксперимента. Показано, что применение генетических алгоритмов для данного типа задач даёт возможность получать оптимальные или близкие к оптимальным решениям за достаточно небольшой интервал времени.

УДК 681.3

Application genetic algorithms for problem solving of scheduling of serial-parallel multiprocessor system / V.V.Areshkova, A.M.Daniltchenko, Ibragim Caad Alla

One of the methods solving for scheduling tasks for consequently-parallel. System by means of genetic algorithms is considers in this work. The algorithm for problem solving is presented. The genetic operators application for such tasks with computing experiment results is described. It is shown. That application of genetic algorithms for such tasks gives the possibility of getting optimal and near optimal decisions during the short time interval.