

О.М. Данильченко, к.т.н., доц.
А.О. Данильченко, аспір.
С.А. Ібрагім, аспір.

Житомирський державний технологічний університет

РОЗВ'ЯЗАННЯ ОДНОГО КЛАСУ ЗАДАЧ СКЛАДАННЯ РОЗКЛАДІВ ГЕНЕТИЧНИМИ АЛГОРИТМАМИ НА КЛАСТЕРНИХ СИСТЕМАХ

Сформульовано клас задач теорії розкладів, розв'язок яких можливо задати у вигляді перестановок чисел від 1 до n . Наведена загальна схема пошуку оптимального розв'язку на базі генетичного алгоритму. Для збільшення швидкодії розроблена паралельна реалізація алгоритмів на багатопроцесорній кластерній системі.

1. Постановка задачі та мета дослідження

Розв'язки великої кількості задач теорії розкладів можуть бути задані у вигляді перестановок чисел від 1 до n . Розв'язання таких задач найчастіше проводиться або переборними методами типу гілок та меж, або наближеними алгоритмами з оцінкою отриманих розв'язків [1–4].

Існують задачі, які, на перший погляд, не є задачами, заданими на перестановках. Або для завдання повного розв'язку не досить однієї перестановки, або для одержання розв'язку варто враховувати додаткові обмеження. Наприклад, такими задачами є найбільш складні задачі складання розкладів – конвеєрна і загальна задача теорії розкладів, задача складання розкладу навчального закладу та інші. У цьому випадку, перестановку можна розглядати як деякий початковий порядок елементів (об'єктів) розв'язку, що використовується для одержання повного розв'язку, який задовольняє всім обмеженням. Тобто для одержання розв'язку варто побудувати деяку процедуру, що, згідно із заданим початковим порядком, одержувала б припустимий розв'язок задачі. Таким чином, множині перестановок ставиться у відповідність (за допомогою деякої процедури) множина усіх розв'язків задачі.

Метою даної роботи є аналіз задач теорії розкладів, заданих на перестановках, зокрема конвеєрної задачі, загальної задачі теорії розкладів та задачі складання навчального розкладу, а також розробка ефективних алгоритмів розв'язку та їх паралельної реалізації.

2. Існуючі методи та загальна схема розв'язку задач, заданих на перестановках

Для розв'язання задач, у яких цільова функція задана досить складною формулою або просто алгоритмом свого обчислення, тобто задач, у яких не представляється можливим скористатися такими властивостями цільової функції, як зростання, безперервність, диференційованість тощо, добре підходять генетичні алгоритми [5]. Фактично, при використанні модифікацій генетичних алгоритмів проектувальникові немає необхідності в докладному аналізі цільової функції. Досить задати представлення розв'язку поставленої задачі у вигляді хромосом і ввести процедуру оцінки їхньої придатності. Якщо розв'язок задачі можна описати перестановкою, то за хромосому як основу генетичного алгоритму природно вибрати таку перестановку. Цей підхід дозволяє узагальнити використання генетичних алгоритмів для всього класу задач, заданих на перестановках. Для кожної реалізації досить тільки надати процедуру (функцію) оцінки придатності хромосом, яка б переводила хромосому (перестановку) у припустиме рішення й обчислювала його оцінку.

Однак при такому підході хромосоми можуть мати дуже велику довжину, а також розрахунок процедури придатності може вимагати великих витрат часу. У той же час, зважаючи те, що для кожної хромосоми в алгоритмі повинні виконуватися аналогічні операції, то виходом з такого становища може бути використання паралельної реалізації генетичного алгоритму на кластерній обчислювальній системі [6].

Перше, з чим зіштовхується розроблювач при проектуванні власного генетичного алгоритму, – це кодування чи представлення розв'язку у вигляді хромосоми. Відомо, що при розв'язанні конвеєрної задачі теорії розкладів з не більш ніж трьома верстатами для пошуку оптимального розв'язку досить обмежитися тільки перестановочними розкладами. Тоді кодування розв'язку у вигляді хромосоми цілком очевидне. Хромосома являє собою перестановку цілих чисел від 1 до n – масив номерів робіт у порядку їхнього виконання. Якщо в конвеєрній задачі не обмежуватися тільки перестановочними розкладами, то для кодування розв'язку необхідно зберігати порядок виконання робіт на кожній машині, і довжина хромосоми збільшується до $n \cdot m$ (точніше до $n \cdot (m - 2)$). Хромосома являє собою перестановку m наборів чисел від 1 до n . Кожен ген такої хромосоми представляє номер роботи, а порядок їхнього проходження показує, на яку машину в проєктованому розкладі буде встановлюватися чергова операція цієї роботи. Для загальної задачі теорії розкладів довжина хромосоми дорівнює кількості операцій k по всіх роботах

(не більш, ніж $n \cdot m$), а кожен ген являє собою номер роботи. Установка цієї роботи в проєктований розклад здійснюється за допомогою деякої процедури, що враховує різні маршрути виконання робіт.

У загальному випадку, задачу складання розкладу навчальних занять можна сформулювати в такий спосіб: "Для заданого набору навчальних аудиторій (у даному випадку під навчальною аудиторією розуміється широке коло приміщень, у яких проводяться навчальні заняття (від комп'ютерної аудиторії до спортивного залу)) і заданого набору часових інтервалів (уроків чи навчальних пар) побудувати такий розподіл навчальних занять для всіх об'єктів (викладачі та навчальні групи), для якого обраний критерій оптимальності є найкращим".

В задачі складання навчального розкладу ми зіштовхуємося з деякими трійками (заняттями) виду – (предмет, викладач, група), яким варто знайти місце (аудиторію) і час виконання, тоді як ген хромосоми пропонується використовувати такі трійки, а точніше їхні порядкові номери. У такий спосіб усі заняття нумеруються числами від 1 до n , і хромосома являє собою деяку їх перестановку. Порядок цих чисел у хромосомі показує, в якому порядку заняття будуть встановлюватися в розклад. Для побудови самого розкладу й обчислення функції придатності хромосом розроблена процедура, що буде поміщати заняття в розклад з урахуванням виконання всіх обов'язкових вимог і обчислювати функцію придатності з обліком усіх необов'язкових вимог. Процедура повертає велике число, якщо за даною хромосомою розклад не може бути побудовано, і суму штрафу розкладу, якщо воно було побудовано.

Такий підхід дозволяє нам використовувати для розв'язання класу задач, заданих на перестановках, класичний генетичний алгоритм із високою швидкістю і перенести всю складність розв'язання й основний час розрахунку на процедуру побудови розкладу й обчислення функції придатності для кожної хромосоми, що дасть нам вигравш при реалізації паралельної версії алгоритму.

При розробці генетичного алгоритму його етапи й операції були реалізовані в такий спосіб:

- хромосоми в початковій популяції заповнювалися випадковими перестановками чисел від 1 до n ;
- вибір хромосом для схрещування проводився за методом "турнірного відбору";
- застосовувалося одномісне схрещування;
- як операція мутації використовувалася перестановка випадково обраних двох елементів у хромосомі;
- реалізовано регенераційний тип репродукції поколінь з переносом, при необхідності, кращої хромосоми з попереднього покоління в наступне;
- закінчення алгоритму проводилося або при розгляді заданої кількості поколінь, або при конвертуванні покоління.

У процесі знаходження розв'язку генетичним алгоритмом найбільш важливим є постійна підтримка правильності (допустимості) розв'язків протягом роботи алгоритму, тобто підтримка хромосом такими, щоб вони не порушували обов'язкових обмежень. Це заощаджує процесорний час, що у протилежному випадку витрачався б на розрахунок і підтримку неприпустимих розв'язків. На допустимість хромосом може вплинути оператор схрещування. Тому після виконання цього оператора потрібно перевіряти новостворену хромосому на наявність порушень обов'язкових обмежень і з їх появою виконувати процедуру виправлення хромосоми. У даній реалізації алгоритму пропонується така процедура з обчислювальною складністю не більше $O(n^2)$.

Виходячи з того, що будь-якому припустимому розкладу можна поставити у відповідність деяку перестановку чисел від 1 до n (причому одному розкладу може відповідати кілька перестановок), то, перебравши з деякою імовірністю усі перестановки, ми переберемо всі припустимі розклади. При побудові алгоритму значення найкращої функції придатності при переході від одного покоління до іншого не погіршується, тобто алгоритм розроблений правильно, і, отже, при розгляді досить великої кількості поколінь ми з імовірністю, близькою до одиниці, одержимо оптимальний розв'язок.

Усі кількісні характеристики реалізованого алгоритму наведені при описі обчислювального експерименту.

3. Паралельна реалізація алгоритму

Серед всіх операцій генетичного алгоритму найбільш тривалою є операція обчислення міри придатності, тобто побудовання оцінки кожної хромосоми. Тому було прийнято рішення розподілити між процесорами саме обчислення оцінок. Це було реалізовано шляхом поділу популяції на рівні частини і пересилання їх дочірнім процесам для подальшого обчислення оцінок. Основним завданням дочірнього процесу є одержання масиву хромосом, обчислення оцінок кожної хромосоми і повернення масиву цих оцінок батьківському процесу.

Для розрахунку навчального розкладу і проведення обчислювальних експериментів був написаний комплекс програм. Програми під ОС Windows написані на VisualFoxPro 8.0, а їх паралельна версія під Linux – мовою C.

Програми на VFP призначені для введення та формування вихідних даних для оптимізації, відображення результатів, обчислення розкладів навчального закладу генетичним алгоритмом у

лінійному режимі і ручному коректуванню отриманих результатів. У цьому блоці програм існує можливість запису вихідних даних у текстовий файл для наступного їхнього використання в паралельній реалізації, а також зчитування результатів роботи паралельних програм з текстових файлів.

У програмах на мові С було обрано стиль програмування, заснований на паралелізмі задач. Незважаючи на те, що такий підхід трудомісткий, він є більш гнучким та дозволяє різним процесам виконувати різні функції одночасно; крім того, завдяки йому спрощується процес синхронізації процесів та керування ними.

Кожна з реалізованих програм для паралельних обчислень має наступну ієрархічну структуру: є основний процес – „диспетчер” та кілька дочірніх, котрі безпосередньо займаються обчисленнями. У свою чергу, на „диспетчера” покладається генерація нових поколінь, розсилання особин дочірнім процесам та отримання і обробка результатів. Хоча „диспетчер” виконує велику кількість обов’язків, всі найбільш часомісткі операції розподілено між дочірніми процесами для досягнення найбільшої швидкодії.

Версії програм для кластерних систем використовують бібліотеку MPICH-1.2.4. Оскільки вона написана для сімейств ОС UNIX та Windows, то ці програми неможливо виконувати на інших платформах, таких як DOS, OS/2 тощо. Загальна структура основних програм співпадає, за винятком реалізації функцій обчислення оцінок хромосом.

4. Обчислювальний експеримент та висновки

Оскільки генетичні алгоритми дають наближені розв’язки, то було зроблено спробу підібрати оптимальний набір параметрів для отримання кращих результатів. У процесі дослідження виявилось, що значення параметрів алгоритму майже не залежать від типу розв’язуваної задачі. Оптимальними параметрами для класу задач на перестановках виявилися:

- кількість хромосом у популяції: 500;
- кількість кандидатів на турнірний відбір: від 2 до 5;
- ймовірність схрещування: близько 1;
- ймовірність мутації хромосоми: від 0.001 до 0.05;
- ймовірність мутації кожного гена: 0.05;
- кількість отриманих популяцій: 100.

Загальна кількість обчислених розв’язків для кожного експерименту – 50000.

Цікавим є той факт, що найгірші результати було отримано при ймовірності мутації, рівній 0 або досить великій (від 0.2), а також при малому розмірі популяції (<100 особин). Також сильно погіршує функцію цілі відсутність турнірного відбору (кількість кандидатів дорівнює 1).

На базі розробленого програмного забезпечення проведено обчислювальний експеримент, у результаті якого були отримані графіки залежності часу розв’язання задач складання розкладів від числа процесів. Приклад графіка залежності коефіцієнта прискорення (в порівнянні з лінійною версією) для конвеєрної задачі, яка вирішувалась на кластері з чотирьох машин, наведено на рис. 1.

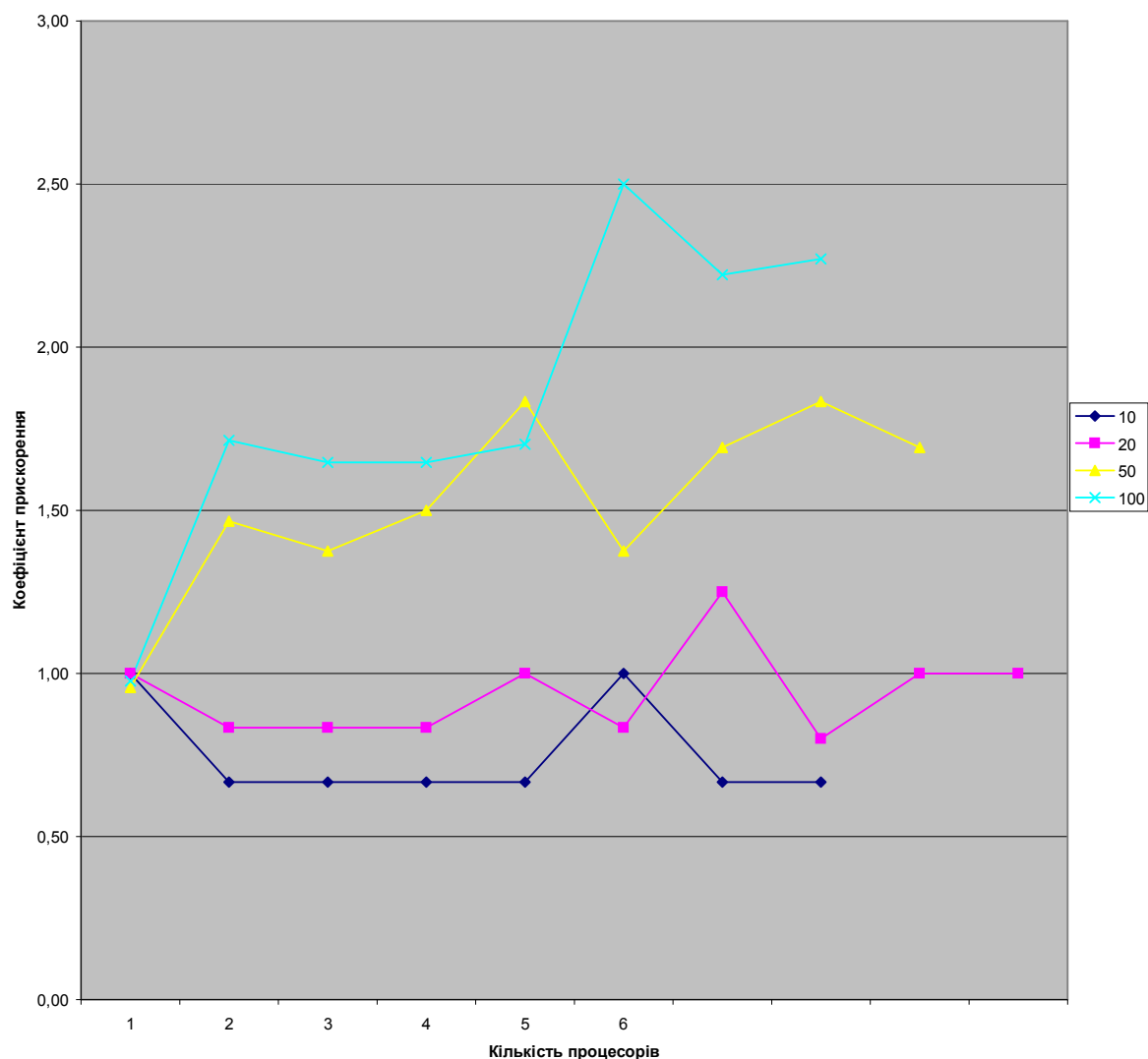


Рис. 1. Графік залежності коефіцієнта прискорення від кількості процесорів у кластері

Ефективність використання кластерних систем залежить від типу задачі, її розмірності, швидкодії мережі та інших параметрів. Так, максимальний коефіцієнт прискорення було отримано для задач великої розмірності, а для задач з довжиною хромосоми менше 20 генів прискорення немає.

Для конвеєрних систем вииграш складає 2,5 рази. Ще більший вииграш можна отримати для задачі складання навчального розкладу, що пояснюється значно більшими витратами часу для обчислення функції придатності.

В той же час було встановлено, що розв'язувати загальну задачу теорії розкладів, використовуючи генетичний алгоритм, з існуючими методами кодування хромосом та їх передачі на кластерній системі недоцільно, оскільки витрати на обмін даними між вузлами кластера займають багато часу, що нівелює вииграш від паралельного обчислення функції цілі.

Під час виконання тестів для загальної задачі на кластері з робочих станцій, пов'язаних у мережу, була помічена відносно велика кількість колізій, яка була пропорційна кількості ЕОМ у кластері. Це явище можна пояснити наступним чином: робочі станції отримують приблизно однакову кількість інформації для обробки майже одночасно; після виконання необхідних обчислень відбувається передача результатів головному процесу, тому періодично виникають проміжки часу, у котрих мережа перевантажена внаслідок одночасної передачі кількох ЕОМ інформації у мережі Ethernet. Для усунення цієї проблеми можна скористатися кластером, що містить робочі станції різної потужності, але у такому разі виникне неповне використання процесорного часу більш потужних ЕОМ. Існує також альтернатива гетерогенному кластеру – для уникнення колізій можливо використовувати мережу із передачею маркера (token ring), проте така топологія є нетиповою для локальних мереж із малою кількістю вузлів.

ЛІТЕРАТУРА:

1. Михалевич В.С., Кукса А.И. Методы последовательной оптимизации в дискретных сетевых задачах оптимального распределения ресурсов. – М.: Наука, 1984. – 334 с.
2. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. – М.: Наука, 1975. – 476 с.
3. Танаев В. С., Шурба В.В. Введение в теорию расписаний. М.: Наука, 1975. – 256 с.
4. Бабаев А.А., Друганов Б.Н. Метод ветвей и границ в задаче составления недельного расписания занятий // Кибернетика. – 1981. – № 1. – С. 63–67.
5. Holland John H. Adaptation in natural and artificial systems. Ann Arbor The University of Michigan Press. — 1975. – 97 p.
6. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – Санкт-Петербург: БХВ-Петербург, 2002. – 599 с.

ДАНИЛЬЧЕНКО Олександр Михайлович – кандидат технічних наук, доцент, завідувач кафедри програмного забезпечення обчислювальної техніки Житомирського державного технологічного університету.

Наукові інтереси:

- теорія розкладів;
- теорія складності екстремальних задач;
- паралельні обчислення.

ДАНИЛЬЧЕНКО Анна Олександрівна – аспірант Житомирського державного технологічного університету.

Наукові інтереси:

- теорія розкладів;
- паралельні обчислення.

ІБРАГІМ Саад Алла – аспірант Житомирського державного технологічного університету.

Наукові інтереси:

- теорія розкладів;
- паралельні обчислення.

Подано 20.10.2004

Данильченко О.М., Данильченко А.О., Ібрагім С.А. Розв'язання одного класу задач складання розкладів генетичними алгоритмами на кластерних системах

Данильченко А.М., Данильченко А.А., Ібрагім С.А. Решение одного класса задач составления расписаний генетическими алгоритмами на кластерных системах

Danilchenko A.M., Danilchenko A.A., Ibragim S.A. Solving of one class of tasks of compiling educational schedules on cluster systems by the means of genetic algorithms

УДК 681.3

Розв'язання одного класу задач складання розкладів генетичними алгоритмами на кластерних системах О.М. Данильченко, А.О. данильченко, С.А. Ібрагім

Сформульовано клас задач теорії розкладів, розв'язок яких можливо задати у вигляді перестановок чисел від 1 до n . Наведена загальна схема пошуку оптимального розв'язку на базі генетичного алгоритму. Для збільшення швидкодії розроблена паралельна реалізація алгоритмів на багатопроцесорній кластерній системі.

УДК 681.3

Решение одного класса задач составления расписаний генетическими алгоритмами на кластерных системах / А.М.Данильченко, А.А.Данильченко, Ибрагим С.А. // Вісник ЖДТУ. – 2004. - № / Технічні науки. – С. : ил. 1 – Библиогр.: б назв.

Сформулирован класс задач теории расписаний, решение которых можно задать в виде перестановок чисел от 1 до n . Приведена общая схема нахождения оптимального решения на базе генетического алгоритма. Для увеличения быстродействия разработана параллельная реализация алгоритмов на многопроцессорной кластерной системе.

УДК 681.3

Solving of one class of tasks of compiling educational schedules on cluster systems by the means of genetic algorithms / A.M. Danilchenko, A.A. Danilchenko, Ibragim S.A. // Вісник ЖДТУ. – 2004. - № / Технічні науки. – Р. : ill. 1 – Refs.: 6 titles.

One class of schedules theory tasks where solution can be set in the form of rearrangements of numbers from 1 to n is formulated. A general scheme for finding optimal solution on the basis of genetic algorithm is suggested. With the purpose to increase the speed the parallel realization of the algorithm on multi-processor cluster system is developed.