

РОЗВ'ЯЗАННЯ ТРАНСПОРТНОЇ ЗАДАЧІ НА КЛАСТЕРНИХ СИСТЕМАХ

В.В. Фальфушинський

Інститута кібернетики ім. В.М. Глушкова НАН України,
07400, Київська обл. м. Бровари, вул. Леваневського,8, E-mail: vlad_falsh@list.ru

Описано план розв'язання транспортних задач за допомогою генетичних алгоритмів. Цей метод відрізняється від класичного тим, що класичний метод потребує фрагментації та серіалізації матриці транспортної задачі, а генетичному алгоритму притаманний природний паралелізм.

An algorithm of solving of the transportation problem with the help of genetic algorithm is described. This method differs from the classic method of solving this problem. Because the classic method uses the fragmentation and serialization of the matrix and the genetic algorithm has its own parallelism.

Вступ

Швидке й просте у використанні моделювання практичних задач має бути потужним і дешевим. Нині ефективності моделювання задач досягають за допомогою паралельних обчислень, які потребують переглядання постановки задачі загалом, пошуку нових паралельних алгоритмів. Крім того, необхідно переглянути політику роботи з вихідними кодами, організацію пам'яті, схеми розрахунків, організацію обміну даними між процесами [1, 2].

Інститут кібернетики ім. В.М. Глушкова НАН України разом із НВО «Електронмаш» створив родину кластерних комплексів Інпарком-16, Інпарком-64, Інпарком-128. На цих кластерних комплексах реалізовано низку інформаційних технологій розв'язання важливих класів задач практичного застосування. Скорочення часу постановки, дослідження та розв'язування задач досягається за рахунок не тільки організації паралельних обчислень, а й інтелектуального програмування, що враховує адаптивну настройку алгоритму і синтезованої програми на властивості задачі з урахуванням технічних і математичних можливостей суперкомп'ютера [3].

Транспортні задачі, можливо, найбільше потребують потужності кластерного комплексу, їх розв'язання проходить за кілька етапів. Причому з методів розв'язання транспортних задач найцікавіший нині побудовано на генетичних алгоритмах (ГА) з огляду на притаманний їм природний паралелізм обчислень. Проте у цьому методі виникають деякі складності. Так, критерій добору хромосом і використовуваних процедур є евристичним і зовсім не гарантує відшукування «найкращого» рішення. Як і в реальному житті, еволюцію може «заклинити» на якій-небудь непродуктивній гілці. Навпаки може статися, що безперспективні батьки, яких вилучить з еволюції генетичний алгоритм, будуть здатні породити високоефективного нащадка. Отже, потрібен розвинений механізм оцінювання перспективних гілок і фактичного керування швидкістю збіжності поколінь до очікуваного результату.

На противагу ГА класичний підхід до організації паралельних обчислень покладається на фрагментацію матриці та серіалізацію обчислень матриці транспортної задачі. Тут чималу складність має склеювання результатів обчислень окремих фрагментів для отримання загального результату.

Розглянемо реалізацію ГА на прикладі оптимізації логістичних операцій доставки пошти Укрзалізницею.

Постановка задачі. На станціях відправлення $A_1, \dots, A_b, \dots, A_m$ зосереджено певну кількість однорідного поштового вантажу $a_1, \dots, a_b, \dots, a_m$ відповідно, який треба доставити залізницею на станції призначення $B_1, \dots, B_j, \dots, B_n$. Обсяг прибуття вантажу на відповідну станцію призначення дорівнює $b_1, \dots, b_j, \dots, b_n$. Із кожної i -ї станції відправлення ($i = 1, 2, \dots, m$) вантаж може бути доставлено на кожну j -ту станцію призначення ($j = 1, 2, \dots, n$). Транспортні витрати на перевезення одиниці вантажу із станції A_i на станцію B_j складає c_{ij} одиниць.

Треба визначити такий план перевезень, щоб задовольнити попит усіх споживачів вантажу, а весь вантаж вивезти з мінімальними сумарними транспортними витратами [4].

Математична модель задачі. Нехай x_{ij} кількість вантажу, що перевозиться із A_i в B_j . Слід визначити такі змінні $x_{ij} \geq 0$, що задовольняють умовам:

$$\sum_{j=1}^n x_{ij} = a_i \quad (i=1, 2, \dots, m), \quad (1)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad (j=1,2,\dots,n) \quad (2)$$

і такі, щоб цільова функція $F(x_{ij})$ досягала мінімуму

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min. \quad (3)$$

де $c_{ij} x_{ij}$ транспортні витрати на перевезення x_{ij} .

Розв'язання доцільне лише у разі додатних значень $x_{ij} \geq 0$.

Транспортна модель має дві форми:

а) замкнена модель – обсяг відправлень дорівнює обсягу призначень

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (4)$$

б) відкрита модель – умова не дотримується

$$\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j, \quad (5)$$

Для транспортної задачі відкритої модель приводять до замкненої, вводячи фіктивного відправника (надмірна модель) або фіктивного споживача (недостатня модель).

Набір x_{ij} , що задовольняє умові (1), можна зобразити матрицею, яку називають планом перевезень. План X , за якого цільова функція $F(x_{ij})$ досягає мінімуму, називають оптимальним.

$$X = \begin{pmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{pmatrix}. \quad (6)$$

Будь-яке допустиме рішення дістало назву опорного плану.

Побудова опорного плану. З кількох методів побудови опорного плану розглянемо три: метод північно-західного кута, метод найменшого елемента та метод подвійної переваги. Від обраного методу залежить, у якій мірі одержаний план буде ближче до оптимального.

Умови транспортної задачі зводять у таблицю, яка дістала назву матриці перевезень.

Кожний рядок матриці перевезень відповідає певній станції відправлення, а кожний стовпець – певній станції призначення. Клітинки матриці, поділені на дві частини – верхню та нижню, відповідають маршрутам перевезень (кількість маршрутів $m \times n$). У верхній частині клітинки записано вартість перевезень одиниці вантажу за відповідним маршрутом c_{ij} , а у нижній частині клітинки – елемент розв'язку x_{ij} , який зауважує, яка кількість вантажу планується до перевезення від i -ї станції відправлення до j -ї станції призначення.

Таблиця 1

Станції-відправники	Станції-одержувачі					Ресурси вантажу (a_i)
	B_1	...	B_j	...	B_n	
A_1	C_{11} X_{11}	...	C_{1j} X_{1j}	...	C_{1n} X_{1n}	a_1
...	·	...	·
A_i	C_{i1} X_{i1}	...	C_{ij} X_{ij}	...	C_{in} X_{in}	a_i
...	·
A_m	C_{m1} X_{m1}	...	C_{mj} X_{mj}	...	C_{mn} X_{mn}	a_m
Потреба вантажу (b_j)	b_1	...	b_j	...	b_n	$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$

Налаштування генетичного алгоритму. Генетичний алгоритм випадковим образом генерує початкову популяцію структур. Робота генетичного алгоритму уявляє собою ітераційний процес, що продовжується доти, поки не виконано задане число поколінь або будь-який інший критерій зупину. В кожному поколінні генетичного алгоритму реалізується відбір пропорційно пристосованості, одноточковий кросовер і мутація. Спочатку пропорційний відбір призначає кожній структурі ймовірність $P_s(i)$, рівну відношенню її застосованості до сумарної пристосованості популяції:

$$P_s(i) = \frac{f(i)}{\sum_{i=1}^n f(i)} \quad (7)$$

Потім відбувається відбір (із заміщенням) усіх n особин для подальшої генетичної обробки, відповідно до значення $P_s(i)$, [3].

При такому відборі члени популяції з вищою застосованістю з більшою ймовірністю частіше вибиратимуться, ніж особини з низькою застосованістю. Після відбору n обраних особин випадковим чином розбиваються на $n/2$ пари. Для кожної пари з ймовірністю P_c може застосовуватися кросовер. Відповідно з ймовірністю $1-P_c$ кросовер не відбувається і незмінні особини переходять на стадію мутації. Якщо кросовер відбувається, отримані нащадки замінюють собою батьків і переходять до мутації.

Визначимо поняття, що відповідають мутації і кросоверу в генетичному алгоритмі.

Мутація – це перетворення хромосоми, що випадково змінює одну чи декілька її позицій (генів). Найбільш розповсюджений вид мутацій - випадкова зміна тільки одного з генів хромосоми.

Кросовер – це операція, за якої з двох хромосом породжується одна чи кілька нових хромосом. Одноточковий кросовер працює в такий спосіб. Спочатку, випадковим образом вибирається одна з l -й точок розриву. Обидві батьківські структури розриваються на два сегменти за цією точкою. Потім, відповідні сегменти різних батьків склеюються і виходять два генотипи нащадків.

Після закінчення стадії кросоверу, виконуються оператори мутації. У кожному рядку, що піддається мутації, кожен біт з ймовірністю P_m змінюється на протилежний.

Популяція, отримана після мутації записує поверх старої і цим цикл одного покоління завершується. Наступні покоління обробляються подібним чином: відбір, кросовер і мутація.

У даний час дослідники генів пропонують багато інших операторів відбору, кросоверу і мутації, от лише найбільш поширені з них.

Елітні методи відбору гарантують, що при відборі обов'язково будуть виживати кращий чи кращі члени популяції сукупності. Найбільш поширена процедура обов'язкового збереження тільки одної кращої особини, якщо вона не пройшла як інші через процес відбору, кросоверу і мутації. Елітизм може бути впроваджений практично в будь-який стандартний метод відбору.

Двоточковий кросовер і рівномірний кросовер – цілком гідні альтернативи одноточковому оператору. В двоточковому кросовері вибираються дві точки розриву, і батьківські хромосоми обмінюються сегментом, що знаходиться між двома цими точками. У рівномірному кросовері, кожен біт першого батька успадковується першим нащадком із заданою ймовірністю; у протилежному випадку цей біт передається другому нащадку і навпаки.

Блок-схема генетичного алгоритму показана на рис. 2. Спочатку генерується початкова популяція особин (індивідуумів), тобто деякий набір рішень задачі. Як правило, це робиться випадковим образом. Потім ми маємо змоделювати розмноження всередині цієї популяції. Для цього випадково відбирається декілька пар індивідуумів, відбувається схрещування між хромосомами в кожній парі, а отримані нові хромосоми втілюються в популяцію нового покоління. У генетичному алгоритмі зберігається основний принцип природного відбору – чим застосованіше індивідуум (чим більше відповідне йому значення цільової функції), тим з більшою ймовірністю він буде брати участь у схрещуванні. Тепер моделюються мутації – у кількох випадково обраних особинах нового покоління змінюються деякі гени. Потім стара популяція частково або цілком знищується і ми переходимо до розгляду наступного покоління. Популяція наступного покоління в більшості реалізацій генетичних алгоритмів містить стільки ж особин, скільки початкова, але в силу відбору застосованість у ній у середньому вище. Тепер описані процеси відбору, схрещування й мутації повторюються вже для цієї популяції і т.д. [4].

У кожному наступному поколінні спостерігаємо зовсім нові розв'язки задачі. Серед них є як погані, так і гарні, але завдяки відбору число прийнятних рішень зростатиме. Зазначимо, що в природі не буває абсолютних гарантій, і найпристосованіший тигр може загинути від рушничного пострілу, не залишивши нащадків. Імітуючи еволюцію на комп'ютері, ми можемо уникати подібних небажаних подій і завжди зберігати життя кращому з індивідуумів поточного покоління – така методика називається "стратегією елітизму".

Для швидшого отримання результату на великих обсягах даних використовують паралельні генетичні алгоритми [3].

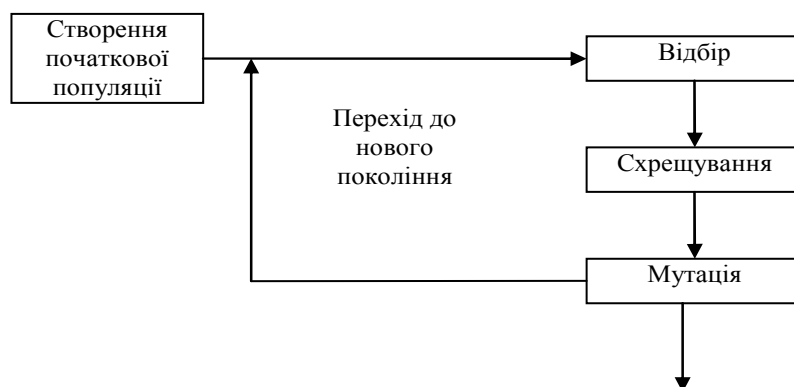


Рис. 1. Блок-схема генетичного алгоритму

Паралельні генетичні алгоритми. Головна ідея більшості паралельних алгоритмів – розділити задачу на частини та розв’язувати їх одночасно, використовуючи багато процесорів. Цей метод „розділай та володарюй” використовують багатьма способами, а тому це призводить до багатьох способів серіалізації ГА. Деякі з них змінюють поведінку ГА. Інші методи доцільніші для масово-паралельних комп’ютерних систем, а деякі – у мережі з меншою кількістю потужних вузлів. Класифікація паралельних ГА, що наведена у даному розділі, схожа на інші класифікації.

Першим способом є глобальна серіалізація ГА. В цьому типі паралельних ГА протягом їхньої роботи існує тільки одна популяція, як і в непаралельній реалізації, але генетичні оператори та функції придатності розраховуються паралельно. Загалом алгоритм концептуально не відрізняється від стандартного ГА. Порівняно з однопроцесорною версією швидкість може досить суттєво збільшитися за умови, що час комунікації не перевищує час обчислення [6].

Цікавіша ідея лежить в основі крупнозернистого паралельного ГА (рис. 2), коли популяція розділяється на багато підпопуляцій, які еволюціонують незалежно одна від одної більшість часу, періодично обмінюючись індивідами. Такий обмін кращими індивідами зветься міграцією і контролюється кількома параметрами. Крупнозернистий паралельний ГА вносить суттєві зміни в роботу ГА та має відмінну від звичайного ГА поведінку. Інколи крупнозернистий паралельний ГА називають „розподіленим” ГА, тому що його часто реалізують на MIMD-комп’ютерах з розподіленою пам’яттю. Крупнозернисті ГА також відомі як „острівкові” паралельні ГА, оскільки у популяційній генетиці описано модель структури популяції, що схожа на рис. 3.

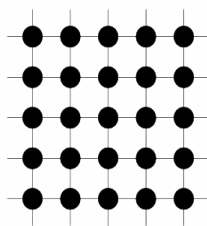


Рис. 2. Дрібнозернистий паралельний ГА

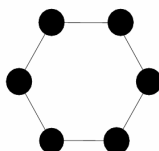


Рис. 3. Крупнозернистий паралельний ГА

Третій підхід серіалізації ГА використовує дрібнозернистий паралелізм, коли популяція розділяється на велику кількість дуже малих популяцій. Ідеальним випадком є така конфігурація, в якій на кожен процесор припадає лише одна хромосома (рис. 1). Така модель розпаралелювання призначена для масивно-паралельних комп’ютерів, але вона також корисна у будь-якому багатопроцесорному середовищі.

Зазначимо, якщо застосування методу глобальної серіалізації не впливає на поведінку алгоритму, останні два методи вносять фундаментальні зміни у спосіб роботи ГА. Зокрема при використанні методів, що розділяють популяцію на частини, парування кожної хромосоми можливе тільки з хромосомами з тієї ж

підпопуляції. Останній метод серіалізації ГА використовує певну комбінацію перших трьох, його ще називають гібридним паралельним алгоритмом [5, 10].

Розв'язання реальної задачі оптимізації перевезень Укрзалізниці на кластері. Оптимізаційні задачі, що виникають на практиці, часто відрізняються надвеликою складністю від тих, на основі яких розвивається теорія ГА. Згідно з постановкою задачі можна сказати, що матриця перевезень по Україні сягнула розміру 1200000 елементів (300 маршрутів потягів, 4000 станцій), а матриця перевезень по всьому СНГ – 5000x26000 = 130 мільйонів елементів.

Обмеження, накладені на розклад перевезень. Розглядають наступні жорсткі обмеження, порушуючи які розклад стає неприйнятним розв'язком задачі:

- поїзд не має проїздити одну і ту саму станцію більше одного разу;
- кількість станцій не змінюється;
- кількість вантажу не змінюється;
- станція може вмістити тільки задану кількість вантажу;
- координати станції не змінюються.

Програмне забезпечення для розв'язання задач оптимізації. Для розв'язання такого типу задачі можна використати низку бібліотек, які дозволяють розв'язувати задачі за допомогою генетичних алгоритмів (GALib, PGAPack, GAUL, MPIKAIA). Всі ці бібліотеки підтримують розпаралелювання обчислень такими методами, як розпаралелювання обчислень однієї популяції або одночасний розрахунок багатьох популяцій. Розроблена авторська бібліотека має стандартні класи і функції, що займаються комутацією, передачею та іншими видами керування віддалених викликів.

Реалізація алгоритму. За наявності згаданої інфраструктури у фреймворку, найпростішим для реалізації є глобально-паралельний ГА. Призначенням додаткового коду в такому алгоритмі є рівномірний перерозподіл навантаження, створюваного основними розрахунками, що веде ГА. Як видно з опису класичного генетичного алгоритму, найбільше обчислювальне навантаження припадає на розрахунки, пов'язані з роботою генетичних операторів.

Кодування хромосоми. Розглянемо особливості кодування хромосоми для задачі оптимізації перевезень. Як відомо, задавши таке кодування, а також функцію пристосованості для хромосоми, в подальшому можна абстрагуватися від задачі, оскільки ГА працюватиме з типовою задачею оптимізації. Для кодування табл. 1 у хромосому випишемо станції в рядок у тому порядку, в якому їх записано в матриці перевезень. Це будуть будівельні блоки у хромосомі. Їхня довжина залежатиме від розмірності задачі, оскільки значення цього будівельного блоку відповідатиме порядковому номеру клітини, в яку записано кількість вантажу.

Підрахунок функції придатності хромосоми. Для підрахування функції придатності хромосоми, що кодує кількість вантажу, поступово проходимо по хромосомі, беручи гени групами. Кожен ген кодує кількість вантажу, який треба перевезти зі станції відправлення в станцію призначення. Отриманої інформації достатньо, щоб відновити з хромосоми план перевезень, що у ній закодовано. Тому цю інформацію можна використовувати у розрахунках функції придатності, вживаючи поняття жорстких та нежорстких обмежень.

Результати. Розв'язавши задачу з використанням перелічених бібліотек на наведеній матриці, можна зробити висновки про швидкість роботи і порівняти результати. Згідно з тарифами на перевезення встановлені згідно «Тарифної політики країн-учасниць СНД» 1 кг багажу – \$1.8 до 5000 км, у всіх інших випадках 2.5\$ [11]. В табл. 2 отриманий результат показує вартість перевезення 1 кг багажу по певному маршруту і видно кореляцію результатів обчислень різними бібліотеками. Час виконання показує швидкість роботи бібліотеки, тобто час за який було розв'язано задачу.

Таблиця 2

Назва бібліотеки	Отримані результати, \$	Час виконання (Україна), сек	Отримані результати, \$	Час виконання (СНД), сек
GALib	8 600	2 150	46 800	14 530
PGAPack	8 320	1 830	46 380	14 210
GAUL	8 482	1 900	46 830	14 650
MPIKAIA	8 506	1 940	46 469	14 426
OGALIB	8 800	2 500	46 910	14 950

Висновки

Головні напрямки сучасних досліджень з оптимізації ГА – побудова компетентних ГА та їхня серіалізація. Саме ці напрямки стали визначальними у побудові власної реалізації комбінованого алгоритму, найвдалішими рисами якого є автоматична адаптація параметрів та механізм навчання зв'язності.

Для серіалізації ГА як перспективного шляху збільшення продуктивності алгоритму, особливо з розвитком мережних комунікацій, розглянуто чотири основні способи: глобально-паралельний, крупнозернистий, дрібнозернистий та гібридний. Перший з них реалізовано у авторському алгоритмі.

Окреме дослідження полягало у застосуванні отриманого генетичного алгоритму до реальної задачі оптимізації перевезень пошти шляхами Укрзалізниці з використанням різних бібліотек. Від стандартної реалізації алгоритму нова реалізація відрізняється насамперед новим кодуванням хромосом і способом підрахунку функції придатності хромосом.

1. *Перевозчикова О.Л., Тульчинский В.Г., Юценко Р.А.* Построение и оптимизация параллельных компьютеров для обработки больших объемов данных // Кибернетика и системный анализ. – 2006. – № 4. – С. 117 – 129.
2. *Тульчинский В.Г., Чарута А.К.* Оценка времени обработки данных в кластерных системах // Проблемы програмування. – 2006. – № 2-3. – С. 118 – 123.
3. *Глибовец Н.Н., Медведь С.А.* Генетические алгоритмы и их использование для решения задачи составления расписаний // Кибернетика и системный анализ. 2003. – № 1. – С. 95–108.
4. *Акуліч І.І.* Математичне програмування в прикладах і задачах. – К.: Вища школа, 1986. – 319 с.
5. *Holland J. H.* Adaptation in natural and artificial systems. - Ann Arbor, MI: University of Michigan Press, 1975. – 97 p.
6. *N. Takahashi:* "Optimization of die press model", TEAM Workshop problem 25.
7. *Wright A.* Genetic algorithms for real parameter optimization // Foundations of Genetic Algorithms, V. 1. – 1991. – P. 205–218.
8. *Deb K., and Kumar A.* (1995). Realcoded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. Complex Systems. – 1995. – 9(6). – P. 451–454.
9. *Markus Schwehm* Parallel Population Models for Genetic Algorithms. – 1996. – 9(6).
10. *Химич А.Н., Молчанов И.Н., Мова В.И., Перевозчикова О.Л. и др.* Численное программное обеспечение интеллектуального MIMD-компьютера ИНПАРКОМ // – Киев: Наук. думка, 2007. – 222 с.
11. *Тарифная политика железных дорог государств-участников содружества независимых государств на перевозки грузов в международном сообщении на 2008 год.*