Використання паралельних обчислень у методі «Р-алгоритм»

Яремчук С.І., Морозов А.В., Кондратовець С.Л., Криворучко Р.М.

Житомирський державний технологічний університет, вул. Черняховського 103, м. Житомир, Україна

yaremchuk_s@mail.ru, morozov.andriy@gmail.com, s.kondratovec@gmail.com, kryvoruchko.roma@gmail.com

Анотація. *У статті розглядається Р-алгоритм розв'язання дискретної мінімаксної задачі розміщення джерел фізичного поля на фіксовані посадкові місця. Наведено формальну постановку задачі та загальну схему алгоритму. Оскільки на кожному кроці алгоритму розглядається K задач про призначення з різними функціями цілі, але з однаковою множиною допустимих розв'язків, певні кроки алгоритму пропонується виконувати паралельно, зокрема, обчислення потенціалів та оцінок (для отримання чергового опорного плану) тощо. Для програмної реалізації використано мову програмування C# та вбудовані механізми реалізації паралельних обчислень .NET-платформи. Отримані результати обчислювального експерименту свідчать про досягнення виграшу у швидкодії при використанні двох процесорів у 1,6 рази, чотирьох – у 3,4.*

Ключові слова

Р-алгоритм, дискретна мінімаксна задача розміщення, транспортна задача, паралельні обчислення.

1 Вступ

У різних сферах діяльності виникають задачі, пов'язані з розміщенням об'єктів у заданих областях. Серед них існує багато задач, для яких розроблені і широко використовуються точні та наближені методи розв'язання. Стрімке вдосконалення комп'ютерної техніки та стрімкий розвиток технологій програмування дозволяє підвищити ефективність використання програмних реалізацій алгоритмів та методів. Зокрема, поява багатоядерних процесорів дозволяє задіяти механізми паралельних обчислень, а використання кількох багатоядерних обчислювальних машин – розподілене виконання обчислювальних процедур, дає можливість значно зменшити час розв'язання задачі. У даній роботі пропонується схема використання паралельних обчислень в *P*-алгоритмі, розробленому для розв'язання дискретної мінімаксної задачі розміщення джерел фізичного поля на задані посадкові місця.

Задача відшукання оптимального призначення досліджувалася у багатьох роботах. Публікація [1] присвячена двухрівневій задачі про призначення, яка розв'язується з використанням методу гілок та меж. У роботі [2] розглядається узагальнення мінімаксної задачі про призначення і для її розв'язання пропонується алгоритм псевдо поліноміальної складності. Дослідження [3] присвячене задачі знаходження циклів, які не перетинаються, на мережі з двома вагами дуг. У роботі наводяться основні етапи поліноміального алгоритму для розв'язання задачі у загальному вигляді. Для збільшення швидкодії програм (зокрема програм для розв'язання задачі про призначення) у [4] розглядається ефективна реалізація девевовидних структур, що дозволяє значно збільшити швидкість роботи програм.

У даній статті розглядається мінімаксна задача про призначення спеціального вигляду, що виникає при розміщенні джерел фізичного поля [5], яка є малодосліджена.

2 Постановка задачі

Розглянемо задачу розміщення джерел фізичного поля на фіксовані посадкові місця [6]. Є область розміщення $\Omega \subset R^n$, N джерел фізичного поля $D_i, i \in [1:N]$ та M посадкових місць $m^j \in \Omega, j \in [1:M]$. Необхідно розмістити джерела фізичного поля на посадкові місця таким чином, щоб максимальне із значень поля у фіксованих точках було найменшим. На розміщення джерел накладаються такі обмеження:

1) на кожне посадкове місце може бути поставлено не більше одного джерела:

$$m\left[\left\{m^j\right\}\bigcap\left(\bigcup_{i=1}^{N}\left\{p^i\right\}\right)\right]\le 1,\, j\in[1:M], \tag{1}$$

де m – потужність множини, p^i – полюс i -го джерела, $i\in[1:N]$;

2) кожне джерело може бути поставлено не більше, ніж на одне посадкове місце

$$m\left[\left\{p^i\right\}\bigcap\left(\bigcup_{j=1}^{M}\left\{m^j\right\}\right)\right]\le 1, i\in[1:N]. \tag{2}$$

3) кількість розміщених джерел повинна бути максимально можливою

$$m\left[\left(\bigcup_{i=1}^{N}\left\{p^i\right\}\right)\bigcap\left(\bigcup_{j=1}^{M}\left\{m^j\right\}\right)\right]=\min\left\{N,M\right\}. \tag{3}$$

Посадкові місця розташовані таким чином, що будь-яке розміщення джерел в області, яке задовольняє обмеженням (1)-(3), не призводить до їхнього взаємного перетину або до виходу за межі області. Необхідно знайти таке допустиме розміщення, при якому заданий критерій якості досягає екстремального значення.

Можливими є три випадки: 1) $N>M$; 2) $N<M$; 3) $N=M$.

В першому випадку будуть розміщені M джерел з N, при цьому всі посадкові місця будуть зайнятими. В другому випадку будуть зайняті N посадкових місць з M, при цьому всі джерела будуть розміщені. В третьому випадку будуть розміщені всі джерела і всі посадкові місця будуть зайнятими.

Критерієм якості є функція

$$f(Z)=\max_{k\in[1:K]}u\left(y^k,Z\right),$$

де $y^k\in\Omega, k\in[1:K]$ – задані точки області, вектор $Z=\left(Z^1,Z^2,...,Z^N\right)$ задає розміщення всіх джерел, Z^i визначає положення i -го джерела в області і співпадає з координатами його полюсу $p^i, i\in[1:N]$, функція $u(y,Z)$ визначається з наступної крайової задачі [7]:

$$Lu=\tilde f(y,Z),$$

$$B_j u=\varphi_j,\, j\in[1:J],$$

в якій L – заданий лінійний диференційний оператор, B_j – задані лінійні оператори, які визначають граничні умови, φ_j – задані функції; y – поточна точка області Ω , $\tilde f$ – функція, яка має наступний вигляд

$$\tilde f(y,Z)=\begin{cases}A^i\left(y-Z^i\right), & якщо\ y\in D_i,\\ 0, & якщо\ y\notin\bigcup_{i=1}^{N}D_i,\end{cases}$$

де $A^i\left(y-Z^i\right)$ – інтенсивність i -го джерела.

Введемо керовані змінні таким чином

$$x_{ij}=\begin{cases}0, & якщо\ i-те\ джерело\ не\ призначається\ на\ j-те\ місце\\ 1, & якщо\ i-те\ джерело\ призначається\ на\ j-те\ місце\end{cases},$$

$i\in[1:N], j\in[1:M]$. Тоді математична модель сформульованої задачі матиме наступний, зручний для її розв’язання вигляд:

$$f(x) = \max_{k\in[1:K]} f_k(x) \to \min,$$

$$\begin{cases} \sum_{i=1}^{L_{\max}} x_{ij} = 1, j \in [1 : L_{\max}], \\ \sum_{j=1}^{L_{\max}} x_{ij} = 1, i \in [1 : L_{\max}], \end{cases}$$

$$x_{ij} \in \{0,1\}, i \in [1 : L_{\max}], j \in [1 : L_{\max}],$$

де $L_{\max} = \max\{N, M\}$, $f_k(x) = \sum_{i=1}^{L_{\max}}\sum_{j=1}^{L_{\max}} c_{ij}^k x_{ij}$,

3 Обчислювальна схема методу

1. Обирається початковий базис $\overline{x}_0^0$, якому відповідає точка x^0. Його можна отримати, наприклад, за допомогою методу північно-західного кута. $s = 0$, $r = 0$.
2. Нехай є базис $\overline{x}_s^r$, якому відповідає точка x^r. Знаходиться наступний опорний план.
   - 2.1. Будується множина $K_{\max}(x^r) = \left\{k \in [1:K] \,\middle|\, f_k(x^r) = f(x^r)\right\}$. Для $\overline{x}_s^r$ знаходяться потенціали та оцінки для всіх $k \in [1:K]$.
   - 2.2. Знаходиться множина комірок $I(\overline{x}_s^r)$, кожен елемент якої задовольняє умові $\Delta_{ij}^k(\overline{x}_s^r) > 0, \forall k \in K_{\max}$,. Якщо вона порожня, то здійснюється перехід до пункту 4.
   - 2.3. Серед елементів множини $I(\overline{x}_s^r)$ обирається такий, що задовольняє умові $f_k(x^r) - \Delta_{i^* j^*}^k(\overline{x}_s^r) < f(x^r)\ \forall k \notin K_{\max}$. Якщо таких не існує, то здійснюється перехід до пункту 4. Якщо таких елементів декілька, то в першу чергу обирається той, що призводить до перевезення по циклу одиниці. Позначимо його через (i^*, j^*).
   - 2.4. Знаходиться наступний опорний план. Для цього комірка (i^*, j^*), знайдена в попередньому пункті, вводиться в базис.
3. Якщо значення перевезення дорівнює одиниці, то отримуємо нову точку x^{r+1}, якій відповідає базис $\overline{x}_0^{r+1}$. При цьому r збільшуємо на одиницю, а s присвоюємо нуль. У випадку фіктивного перевезення отримаємо ту ж точку x^r, але інший базис $\overline{x}_{s+1}^r$. При цьому r не змінюється, а s збільшується на одиницю. Здійснюється перехід до пункту 2.
4. За розв’язок обирається $x^* = x^r$. Якщо $\Delta_{ij}^k(x^*) \le 0$, то x^* – глобальний мінімум задачі [8]. В іншому випадку x^* – стаціонарна точка методу. Кінець роботи алгоритму.

4 Реалізація паралельних обчислень

P-алгоритм реалізовано у вигляді програмного продукту, який розроблено з використанням платформи .NET та мови програмування C# 5.0. Для паралельного виконання обчислень використано клас Parallel .NET Framework 4 та стандартні засоби організації багатопотоковості. Для створення інтерактивного інтерфейсу користувача використано клас *BackgroundWorker*. Ієрархію класів зображено на рис. 1.

Обчислювальна схема методу реалізована у класі *PAlgo*, який містить наступні методи:

- *PAlgo* – конструктор класу;

- *Compute* – пошук розв’язку задачі за допомогою *P*-алгоритму;

- *ComputeFirstApproximation* – для кожної матриці знаходить початкове наближення методом північного-західного кута;

- *GetComputingStatus* – повертає істину, якщо процес розв'язання задачі завершено;

- *FindNextBasicCell* – визначає клітинку, яка буде вводитися у базис.

Паралельні обчислення здійснюються за допомогою класу *Parallel*:

1) побудова початкового наближення у методі *ComputeFirstApproximation:*

```
Parallel.For(0, _countPoints, i => {
   _points[i].ComputeFirstApproximation();
});
```

2) пошук максимальної оцінки у методі *FindNextBasicCell*:

```
Parallel.For(0, length, k => {
   bool isAdd = true;
   for (int j = 0; j < _countPoints; j++)
     if (_kMax != j && _points[j].GetMark(cells[k].RowIndex, cells[k].CollIndex) < 0)
       isAdd = false;

     if (isAdd && cells[k].Mark > maxMark) {
        cell = cells[k];
        maxMark = cells[k].Mark;
     }
});
```

3) обчислення потенціалів та оцінок у методі *Compute*:

```
Parallel.For(0, _countPoints, i => {
   _points[i].ComputePotentials();
   _points[i].ComputeMarks();
});
```

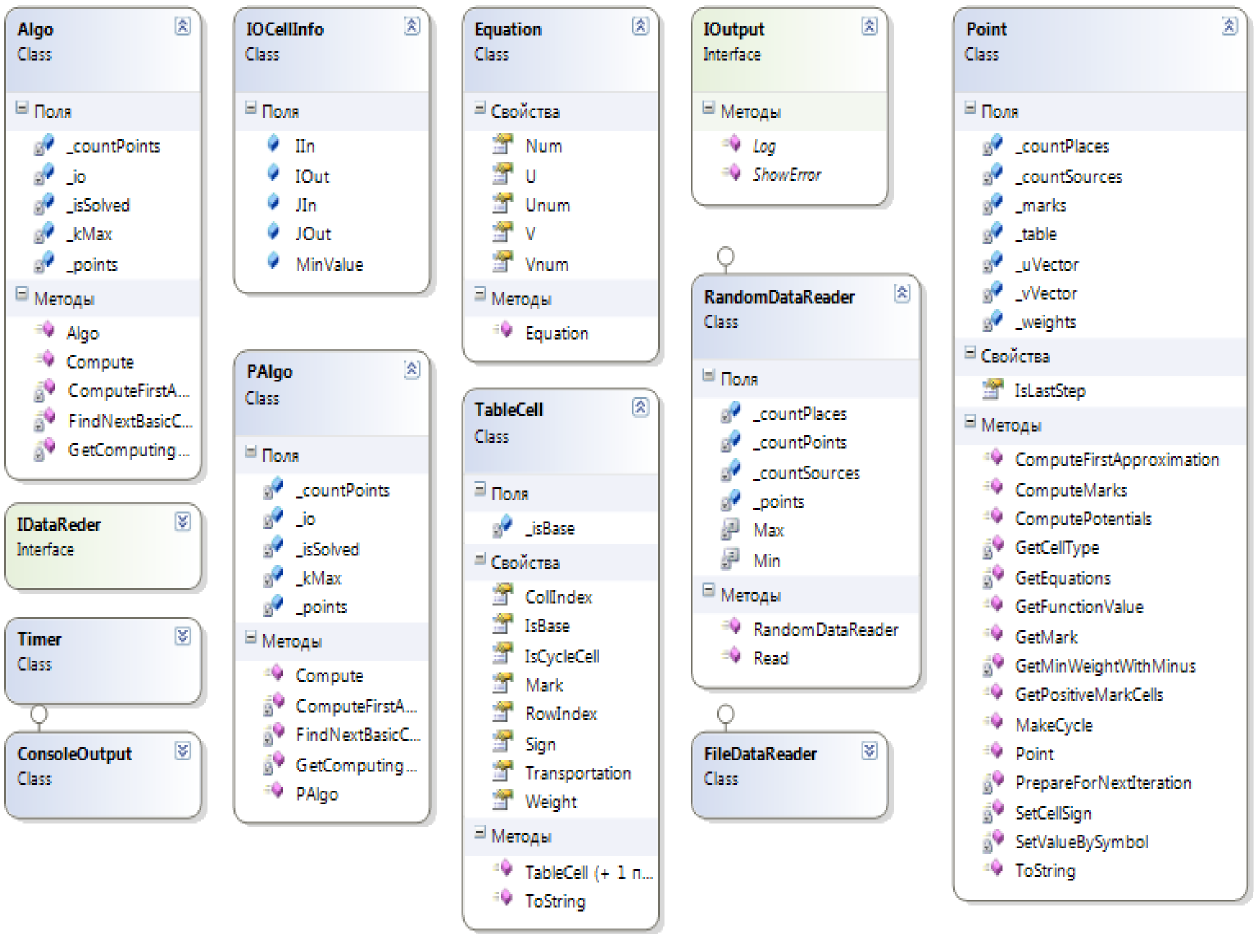

Рис. 1. Ієрархія класів для реалізації методу «*P*-алгоритм»

5 Обчислювальний експеримент

Для проведення обчислювального експерименту використовувався комп'ютер Intel Core 2 Quad з тактовою частотою 2,83 ГГц з об'ємом оперативної пам'яті 4 ГБ.

На першому етапі експерименту досліджувалася залежність часу розв'язання класичної транспортної задачі методом потенціалів від розмірності задачі. Випадковим чином генерувалося по 100 задач для кожної розмірності від 10 до 200. Матриця заповнювалась випадковими числами від 1 до 200. Кожна задача розв'язування методом потенціалів.

Другий етап експерименту полягав у дослідженні залежності часу розв'язання дискретної мінімаксної задачі розміщення джерел фізичного поля на фіксовані посадкові місця від розмірності задачі та кількості використаних процесорів. Випадковим чином генерувалося по 100 задач (K = 10) для кожної розмірності від 10 до 200. Матриця заповнювалась випадковими числами від 1 до 200. Кожна задача розв'язування методом «P-алгоритм».

Результати обчислювального експерименту наведено на рис. 2 та рис. 3.

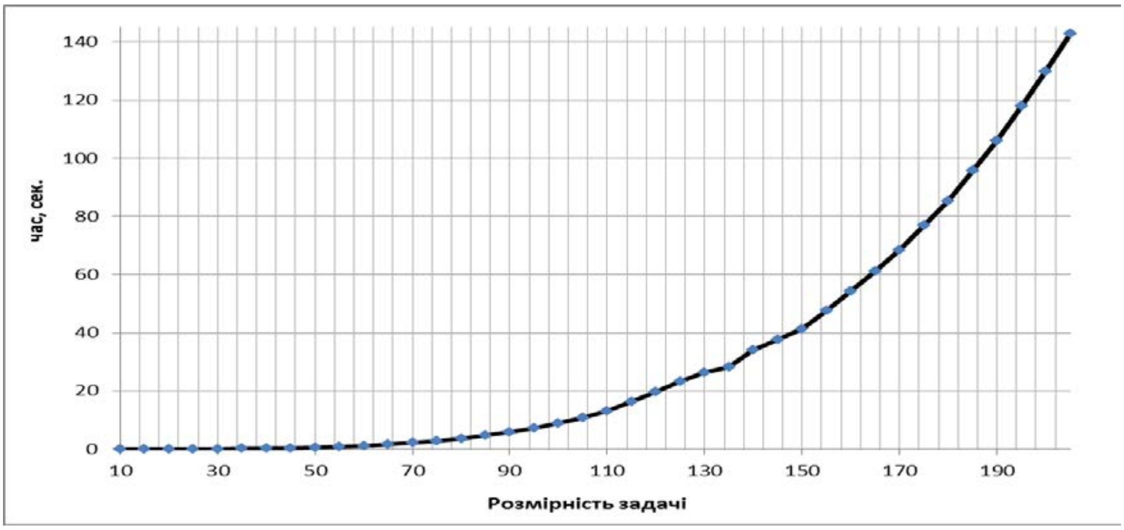

Рис. 2. Залежність часу розв'язання транспортної задачі методом потенціалів від розмірності задачі (з використанням одного процесора)

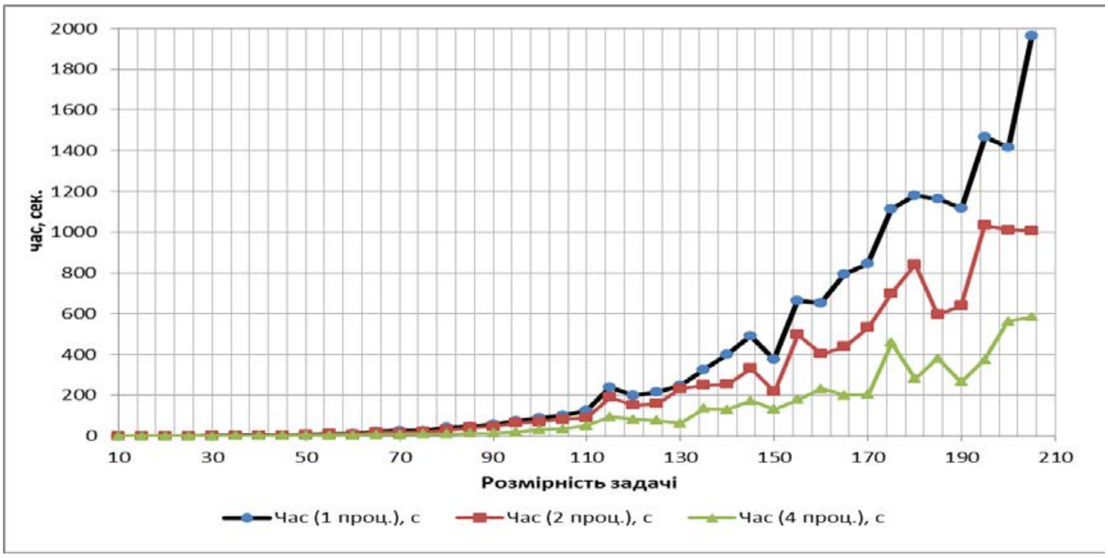

Рис. 3. Залежність часу розв'язання дискретної мінімаксної задачі розміщення джерел фізичного поля на задані посадкові місця за допомогою методу «Р-алгоритм» від розмірності задачі та кількості використаних процесорів

6 Висновки

У статті наведено змістовну та математичну постановки дискретної мінімаксної задачі розміщення джерел фізичного поля на фіксовані посадкові місця та запропоновано метод «*P*-алгоритм» її розв'язання. Для підвищення швидкодії методу пропонується використовувати механізм паралельних обчислень. За допомогою обчислювального експерименту показано, що у випадку використання двох процесорів, у середньому, досягається виграш у 1,6 рази, чотирьох – у 3,4 рази.

У подальшому планується реалізувати алгоритм розв'язання задачі на багатопроцесорних системах із загальною та розподіленою пам'яттю, що значно зменшить час розв'язання задач великих розмірностей та провести експериментальне дослідження залежності часу розв'язання задачі від значення *K*.

Література

[1] Ларин. Р. М. Двухуровневая задача о назначених / Р.М. Ларин, А.В. Пяткин // Дискретный анализ и исследование операцій. – Июль-декабрь 2001. Серия 2. Том 8. – № 2. – С. 42-51.
[2] Глебов Н. И. Об одном обобщении минимаксной задачи о назначених // Дискретный анализ и исследование операций / Н.И. Глебов.– Октябрь-декабрь 2004. Серия 1. Том 11. – № 4. – С.36–43.
[3] Шарифов Ф. А. Задача нахождения непересекающихся и несовпадающих циклов на сети / Ф. А. Шарифов // Теорія оптимальних рішень, – №2. – 2003. – С155-161.
[4] Шарифов Ф. А. Об эффективности алгоритмов решения сетевых задач на древовидных структурах / Ф.А. Шарифов // Кибернетика и системный анализ. – №3. – 2003. – С179-184.
[5] Стоян Ю. Г. Оптимизация технических систем с источниками физических полей / Ю.Г. Стоян, В. П. Путтин.– Киев: Наукова думка. – 1988. – 192с.
[6] Burda R.V. The optimization methods of solution of minimax problems of geometric objects allocation / S.I. Yaremchuk, Y.O.Shapovalov, R.V.Burda, O.B.Morgalyuk // Wspolczesne problemy informatyki. Seria:zeszyty i monografie. - Legnica: Wydawnictwo Wyzszej Szkoly Menedzerskiej, Poland, 2011. - P. 257-267.
[7] Стоян Ю. Г., Путятин В. П. Оптимизация технических систем с источниками физических полей. – Киев: Наукова думка. – 1988. – 192с.
[8] Яремчук С. І., Бурда Р. В., Матущенко С. С. Алгоритм решения дискретной минимаксной задачи размещения источников физического поля // Кибернетика и системный аналіз №5, 2009, С. 153-163.