DOI: 10.31319/2519-8106.2(39)2018.154226
УДК 519.85
**А.І. Косолап,** д-р. фіз.-мат. наук, професор, (зав. кафедри спеціалізованих комп'ютерних систем), anivkos@ua.fm
**Н.С. Волинець,** викладач, natalia.wolynec@gmail.com
Український державний хіміко-технологічний університет, м. Дніпро

# ОПТИМАЛЬНИЙ РОЗПОДІЛ РЕСУРСІВ У БАГАТОПРОЦЕСОРНИХ СИСТЕМАХ

*В статті розглянута задача оптимального розподілу ресурсів в багатопроцесорних системах. Побудовано декілька математичних моделей для різних постановок задач. Описано існуючі методи і алгоритми для побудови розкладів. Запропоновано для розв'язання задач оптимального розподілення ресурсів метод точної квадратичної регуляризації. Проведені чисельні експерименти, які підтверджують ефективність цього методу.*
***Ключові слова***: *багатопроцесорні системи, метод точної квадратичної регуляризації.*

*The problem of optimal distribution of resources in multiprocessor systems is considered in the article. Several mathematical models for various tasks have been constructed. Existing methods and algorithms for constructing schedules are described. The method of exact quadratic regularization is proposed for solving optimal resource allocation problems. Numerous experiments have been conducted to confirm the effectiveness of this method.*
***Keywords***: *multiprocessor systems, exact quadratic regularization method.*

## Постановка проблеми

В сучасних обчислювальних системах виникає необхідність розв'язувати велику кількість задач обробки інформації. Сучасні тенденції збільшення продуктивності комп'ютерів пов'язані з розпаралелюванням обчислювальних процесів: з'явились багатоядерні процесори та потужні кластерні системи — модульні багатопроцесорні системи пов'язані високошвидкісною комутаційною середою. Ефективність роботи кластерної системи залежить від оптимального розподілу завдань по обчислювальним вузлам. Існує багато класів таких задач, які ще потребують оптимального розв'язування.

Актуальні питання які виникають на етапі експлуатації багатопроцесорних систем:

- Рівномірне завантаження процесорів
- Мінімальний час виконання завдань
- Створення паралельних алгоритмів
- Використання загальних ресурсів
- Планування виконання завдань з перериваннями

## Аналіз останніх досліджень та публікацій

Постановка задачі розподілення і впорядкування завдань в багатопроцесорній системі є відомою в теорії розкладів і розглядається в роботах Р.В. Конвея, В.Л. Максвелла, Л.В. Міллера [1]. Відомо, що такі задачі відносяться до класу NP-складних, що означає експоненціальне зростання обчислювальної складності розв'язування при лінійному зростанні розмірності задачі. В теорії обчислювальної складності існує два напрямки розв'язання NP-складних комбінаторних задач, націлені на пошук точних і наближених розв'язків відповідно. В рамках першого напряму використовуються методи перебору, які характеризуються експоненціальною обчислювальною складністю. Другому напрямку відповідають наближені алгоритми, що дозволяють порівняно швидко отримувати задовільні рішення, які не є точними. Їх іноді називають конструктивними і оптимізаційними евристиками [2]. Застосування наближених алгоритмів виправдане при необхідності швидкої оцінки здійсненності обчислень в системі, розробці прототипів обчислювальних систем, а також для отримання опорних розв'язків при пошуку оптимальних результатів.

***Точні алгоритми.*** Найочевиднішим і при цьому найбільш витратним серед оптимальних алгоритмів по часу виконання є алгоритм повного перебору, в якому послідовно досліджується вся множина розв‘язків (варіантів призначення) на предмет оптимальності по заданому критерію.

***Методи обмеженого перебору*** здійснюють пошук в дереві можливих рішень, складеному базуючись на залежностях між завданнями. Перебір усіх гілок дерева відбувається до тих пір, поки не буде знайдено розклад, який задовольняє заданій умові. Серед його недоліків — необхідність внесення змін в алгоритм у випадку зміни обмежень та велике збільшення дерева у випадку збільшення розмірності задачі.

Для усунення недоліків, пов'язаних з значним часом виконання, існують різні паралельні варіанти цього алгоритму, а так само ряд інших алгоритмів. Серед останніх найбільшого поширення набув алгоритм ***розгалужень та границь*** [3—4]. В основі цього алгоритму лежать дві процедури: розгалуження і обмеження. Процедура розгалуження полягає в розбитті всієї множини розв‘язків на підмножини і їх подальше вивчення. Підмножину розв‘язків або відбраковують, коли воно не містить розв‘язок краще відомого, або відбувається його подальше розбиття на підмножини до тих пір, поки не буде знайдено оптимальний розв’язок задачі. Цей алгоритм при розв‘язанні конкретної задачі може в гіршому випадку здійснювати перегляд всіх розв‘язків.

Для задач розподілення процесів по процесорам великої розмірності точні методи стають непрактичними при збільшенні розмірності задачі.

***Наближені алгоритми***. До оптимізаційних евристичних методів можна віднести генетичні алгоритми, метод імітації відпалу, пошук із заборонами (taboo search), жадібні алгоритми, підхід на основі мурашиних колоній.

***Генетичний метод*** скоріше являє собою популяційно-генетичний підхід до вирішення завдання пошуку, ніж єдиний алгоритм розв‘язування дискретних оптимізаційних задач [5—6].

***Імітація відпалу***, на відміну від генетичного алгоритму, є універсальним методом пошуку глобального мінімуму цільової функції і має як переваги, так і недоліки. До переваг відносять: можливість пошуку розв‘язків для складних нелінійних задач, можливість роботи з даними з великою кількістю шумів і перешкод; здатність виходу з локальних мінімумів, універсальність методу, відносну легкість модифікації, адаптації та технічної реалізації. Серед недоліків, як правило, відзначають наступні: необхідність адаптації параметрів для кожного конкретного завдання, залежність якості розв‘язку від часу його отримання [7].

***Пошук із заборонами (taboo search)*** — модифікація локального пошуку [8]. Дозволяє продовжити пошук, здійснюючи перехід до розв‘язку з околиці, навіть якщо значення цільової функції при новому розв‘язку гірше. Це дозволяє «вискакувати» з локальних екстремумів. Недолік — можливість зациклення, коли відбувається повернення до вже переглянутого раніше розв‘язку. Щоб уникнути зациклення, деякі розв‘язки або переходи від рішення до вирішення оголошуються забороненими (табу). Для цього створюється і зберігається «список заборон», в який поміщається інформація про останні рішення або останні їх зміни.

***Жадібні алгоритми*** ґрунтуються на властивості оптимальності для підзадач [9]. Якщо оптимальний розв‘язок задачі містить оптимальний розв‘язок її підзадач, то задача має властивість оптимальності для підзадач. Жадібні алгоритми, роблячи на кожному кроці локально-оптимальний вибір (розв’язуючи підзадачу), припускають, що в результаті буде отримано оптимальний розв‘язок задачі. Жадібні алгоритми мають низьку обчислювальну складність, але область їх застосування обмежується окремими задачами, які мають властивість оптимальності для підзадач. В роботі [10] пропонується використовувати комбінацію жадібних методів та обмеженого перебору.

Алгоритми, засновані на схемі ***мурашиних колоній***, мають властивість автоматичного налаштування в ході роботи на конкретну задачу. Для застосування мурашиних алгоритмів потрібно зведення розв'язуваної задачі до задачі знаходження на графі маршруту, який має певні властивості. Крім того, як і в ітераційних алгоритмах, виникає проблема налаштування параметрів алгоритму на конкретну задачу.

Ефективність розглянутих алгоритмів значно погіршується при збільшенні розмірності задачі.

Наявність великої кількості різних алгоритмів свідчить що проблема оптимального розподілу ресурсів в багатопроцесорних системах ще далека від свого завершення. Практичне використання потребує розробки нових, більш ефективних методів.

**Формулювання мети дослідження**

У найбільш загальному формулюванні завдання складання розкладу полягає в наступному. За допомогою деякої кількості ресурсів або обслуговуючих обладнань повинна бути виконана деяка фіксована система завдань. Мета полягає в тому, щоб при заданих властивостях завдань і ресурсів і накладених на них обмеженнях знайти ефективний алгоритм упорядкування завдань, який оптимізує бажаний критерій ефективності. У якості критеріїв ефективності звичайно розглядаються довжина розкладу, середній час перебування завдань у системі, завантаженість системи. Моделі цих завдань є детермінованими в тому розумінні, що вся інформація, на основі якої ухвалюються рішення про впорядкування, відома заздалегідь.

**Постановка задачі**

Багатопроцесорні системи обробляють велику кількість завдань, при цьому виникає проблема рівномірного завантаження процесорів [11]. Якщо відомий час $t_{ij}$ обробки $j$-го завдання на $i$-му процесорі, то виникає задача рівномірного розподілення завдань по процесорам за умови, що всі ці завдання будуть оброблені за мінімальний час. Для розв‘язання цієї задачі вводимо змінні:

$$x_{ij} = \begin{cases} 1, & j\text{ - е завдання виконується } i\text{ - м процесором} \\ 0, & \text{в інншому випадку} \end{cases}$$

і отримуємо оптимізаційну задачу розподілення ресурсів

$$\min\{T \mid \sum_{j=1}^{n} t_{ij}x_{ij} \le T, i=1,...,m, \sum_{i=1}^{m} x_{ij} = 1, j=1,...,n, x = 0 \vee 1\}. \quad (1)$$

**Виклад основного матеріалу**

Задача (1) буде мати велику кількість розв’язків з однаковим значенням цільової функції. Це пов‘язано з тим, зо завдання, які виділяється кожному процесору, можуть виконуватися у довільному порядку. Задача (1) відноситься до класу лінійних задач з бульовими змінними.

Чисельні експерименти показали, що швидкість розв‘язання задачі (1) значно зростає якщо перетворити її до вигляду

$$\min\{\sum_{i=1}^{m}(\sum_{j=1}^{n} t_{ij}x_{ij})^2 \mid \sum_{j=1}^{n} t_{ij}x_{ij} \le T, \sum_{i=1}^{m} x_{ij} = 1, j=1,...,n, x = 0 \vee 1\}, \quad (2)$$

в якій мінімізується сума квадратів роботи кожного процесора. Ця сума буде мінімальною, якщо завантаження процесорів буде рівномірним. В задачі (2) значення $T$ фіксовано і на кожній ітерації розв‘язку задачі (2) це значення зменшується.

Задача (1) може містити інші обмеження, наприклад вартості роботи кожного процесора. При проектуванні таких систем частина процесорів може бути надлишковою. Для розв‘язання задачі введемо нові змінні $z_i$, які дорівнюють 1, якщо $i$-й процесор використовується і 0 — в протилежному випадку. Змінні необхідно пов‘язати з основними змінними $x_{ij}$ наступними умовами:

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

Необхідно врахувати обмеження на вартість системи,

$$\sum_{i=1}^{m} z_i (P_i + s_i \sum_{j=1}^{n} t_{ij} x_{ij}) \le S,$$

де $P_i$ — вартість $i$-го процесора, $s_i$ — вартість одиниці часу його експлуатації, а $S$ — обмеження на вартість багатопроцесорної системи, що проектується.

Крім розподілених ресурсів, багатопроцесорні системи містять загальні ресурси з послідовним доступом. Це може бути диск, принтер і т.д. Після завершення роботи на процесорі воно проходить обробку на пристрої загального доступу. Процесор і зовнішній пристрій утворюють двохмашинний комплекс, для якого відомий ефективний алгоритм Джонсона. За цієї умови задача (2) розв‘язується в два етапи. На першому етапі завдання розподіляються оптимальним чином між процесорами, а на другому етапі використовується алгоритм Джонсона, який перевпорядковує послідовність обробки завдань на процесорах.

Розглянуті задачі розв‘язувались з використанням пакетів OpenSolver, Open Studio и других. Вони дозволяють розв‘язувати такі задачі до 100 змінних. Але сучасні багатопроцесорні системи інколи містять тисячі процесорів.

Прогрес в розв‘язанні даного класу задач може бути досягнутий використанням методу точної квадратичної регуляризації [12]. Точна квадратична регуляризація дозволяє перетворити задачу (2) до вигляду:

$$\max\{\|z\|^2 \mid \sum_{i=1}^{m}(\sum_{j=1}^{n} t_{ij}x_{ij})^2 + s + (r-1)\|z\|^2 \le d, \sum_{j=1}^{n} t_{ij}x_{ij} \le T,$$
$$\sum_{i=1}^{m} x_{ij} = 1, j = 1,...,n, \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}(1-x_{ij}) + r\|z\|^2 \le d, 0 \le x \le 1\}, \quad (3)$$

де $z = (x_1,\ldots,x_{nm}, x_{nm+1})$, $r$ — параметр квадратичної регуляризації, який перетворює обмеження задачі (3) до опуклих, $s$ — параметр, який забезпечує активність першого обмеження задачі (3). Задача (3) містить лише неперервні змінні. Для розв‘язування оптимізаційних задач з неперервними змінними використовується ефективний прямо-двоїстий метод внутрішньої точки [13]. В цій задачі необхідно визначити мінімальне значення змінної $d$, для якої виконується умова $r\|z\|^2 = d$, де $z$ — розв‘язок задачі (3) при фіксованому значенні $d$. Значення $d$ визначається методом дихотомії. Особливістю методу точної квадратичної регуляризації є те, що його можна застосовувати до розв‘язання розглянутих задач великої розмірності. Це пов’язано з тим, що прямо-двоїстий метод внутрішньої точки дозволяє розв’язувати задачі з сотнями тисяч змінних, а метод дихотомії по одній змінній швидко збігається. Для таких розмірностей існуючі методи непридатні. Проведені чисельні експерименти засвідчують значну перевагу методу точної квадратичної регуляризації над існуючими методами.

**Висновки та перспективи подальших досліджень**

Описані сучасні методи та алгоритми, які використовуються для розв‘язування оптимізаційних задач з булевими змінними, що виникають при оптимальному розподілу ресурсів в багатопроцесорних системах.

Побудовані математичні моделі для розв‘язання задачі побудови розкладів багатопроцесорної системи. Ці моделі дозволяють зробити завантаження процесорів більш рівномірним, врахувати та мінімізувати вартість багатопроцесорної системи, а також розташувати завдання для виконання на процесорах в найкращому порядку.

Для розв‘язання описаних задач застосований метод точної квадратичної регуляризації та перевірена його ефективність.

В подальшому планується перевірити адекватність цих моделей на реальних прикладах, вдосконалити алгоритм розв‘язання, а також побудувати нові моделі з можливістю переривань.

**Список використаної літератури**

1. Конвей Р.В., Максвелл В. Л., Миллер Л. В. Теория расписаний. / Р. В. Конвей, В. Л. Максвелл, Л. В. Миллер. – М.: Наука, 1975. – 320 с.
2. Stochastic global optimization. Techniques and applications in chemical engineering [editor G. P. Rangaiah]. – National University of Singapore, 2010. – 722 p.

3. Григорьева Н.С. Алгоритм ветвей и границ для задачи составления расписания на параллельных процессорах / Н.С. Григорьева // Вестник Санкт-Петербургского университета. – 2009. – № 1(10). – С 44–55.
4. Crainic T.G. Parallel branch-and-bound algorithms. / T.G. Crainic, B. Le Cun, C. Roucairol; In E.-G. Talbi, editor // Parallel Combinatorial Optimization, Wiley, Hoboken NJ. – 2006. – pp. 1–28.
5. Goldberg, D.E. Genetic Algorithms in Search, Optimization and Machine Learning. / D.E. Goldberg // Addison Wesley, Reading, MA. – 1989. – 432 p.
6. Ashish Sharma and Mandeep Kaur An Efficient Task Scheduling of Multiprocessor using Genetic Algorithm based on Task Height / Ashish Sharma and Mandeep Kaur // International Journal of Hybrid Information Technology. – Vol.8, No.8. – 2015, pp.83-90.
7. Троценко Р.В., Посашенко А.В. Обзор метода имитации отжига и его модификаций в аспекте применимости к решению задачи комплектации вычислительной системы минимальной стоимости в условиях дефицита времени / Троценко Р.В., Посашенко А.В. // Наука вчера, сегодня, завтра: сб. ст. по матер. XI междунар. науч.-практ. конф. – 2014. – № 4(11). – Новосибирск: СибАК, 2014. – 29–35 с.
8. Glover, F. Tabu Search. / F. Glover, M. Laguna. // Kluwer Academic, Boston. –1997. – 382 p.
9. Кротов К. В. Жадный алгоритм построения расписаний обработки данных в конвейерных системах / К. В. Кротов // вестник ВГУ, серия: системный анализ и информационные технологии. – 2015. – № 1. – 44–60 с.
10. Костенко В. А. Алгоритмы комбинаторной оптимизации, сочетающие Жадные стратегии и ограниченный перебор / В. А. Костенко // Известия РАН. Теория и системы управления. – 2017. – № 2. – С. 48–56.
11. Таненбаум, Э. Современные операционные системы. Изд. третье /Э. Тененбаум. – Киев: 2010. – 1116 с.
12. Косолап, А. И. Глобальная оптимизация. Метод точной квадратичной регуляризации / А. И. Косолап. – Днепропетровск: ПГАСА, 2015. – 164 с.
13. Nocedal, J. Numerical optimization / J. Nocedal, S.J. Wright. – Springer, 2006. – 685 p.
14. Cook W. Fifty-Plus Years of Combinatorial Integer Programming/W. Cook.- Georgia Institute of Technology, 2009. – 39 p.
15. Kenneth V.P. Differential Evolution. A Practical Approach to Global Optimization / V.P. Kenneth, R.M. Storn, J.A. Lampinen. – Berlin, Heidelberg: Springer-Verlag, 2005. – 542 p.

**OPTIMAL RESOURCES ALLOCATION IN MULTIPROCESSORS SYSTEMS**

**Kosolap A.I., Volynets N.S.**

**Abstract**

The efficiency of a multiprocessor system depends on the optimal allocation of tasks to the computing nodes. The problem of distributing and organizing tasks in a multiprocessor system belongs to a class of NP-complex. This means an exponential increase in the computational complexity of the solution with a linear increase in the dimension of the problem. In the theory of computational complexity there are two directions for solving such problems - exact and approximate.

Exact methods are characterized by exponential computational complexity. Therefore, they are almost not used to solve real problems.

Optimization heuristic methods include genetic algorithms, annealing imitation method, taboo search, greedy algorithms, ant colony-based approach. Over the past years, scientists have been trying to find new and improve existing methods and algorithms. This indicates the relevance of the task.

In the most general formulation, the problem of scheduling is as follows: with some resources or service equipment, some fixed system of tasks must be executed. The goal - taking into account the

properties of tasks and resources, imposed restrictions on them, to find an efficient algorithm for ordering tasks, which optimizes the desired performance criterion. Criteria of efficiency - length of the schedule, average time of the tasks in the system, system load. Models of these tasks are deterministic. That is, all the information on the basis of which decisions are made on ordering is known in advance.

The authors compiled mathematical models for solving the problem of constructing schedules of a multiprocessor system. These models make loading processors more uniform, take into account and minimize the cost of a multiprocessor system, and arrange tasks for execution on processors in the best order.

To solve the described problems, the method of exact quadratic regularization was applied and its efficiency proved.

In the future it is planned to check the adequacy of these models on real examples, to improve the algorithm of solution, as well as to build new models with the possibility of interruptions.

**References**

[1] Conway R.V., Maxwell V. L., Miller L. V. *Teoriya raspisaniy* [Schedule Theory]. Moskva, 1975. – 320 p.

[2] Stochastic global optimization. Techniques and applications in chemical engineering, editor G. P. Rangaiah, National University of Singapore. 722 p., 2010.

[3] Grigorieva N. S. Algoritm vetvey i granits dlya zadachi sostavleniya raspisaniya na parallel'nykh protsessorakh [Algorithm of branches and boundaries for the problem of scheduling on parallel processors] *Vestnik Sankt-Peterburgskogo universiteta*, 2009, no 1 (10), pp 44–55.

[4] Crainic T.G., Le Cun B., Rcairooul C.; In E.-G. Talbi, editor “Parallel branch-and-bound algorithms” Parallel Combinatorial Optimization, Wiley, Hoboken NJ., pp. 1–28, 2006

[5] Goldberg, D.E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading, MA, 432 p., 1989.

[6] Ashish Sharma and Mandeep Kaur “An Efficient Task Scheduling of Multiprocessor using Genetic Algorithm based on Task Height” International Journal of Hybrid Information Technology, vol.8, no.8., pp.83–90, 2015.

[7] Trotsenko R.V., Posashenko A.V. [A review of the method of simulating annealing and its modifications in terms of applicability to solving the task of completing a minimum cost computer system under time pressure]. *Nauka vchera, segodnya, zavtra: sb. st. po mater. XI mezhdunar. nauch.-prakt. konf* [Science yesterday, today, tomorrow: dig. of art. on mater. XI Intern. scientific-practical conf.] Novosibirsk, 2014, pp 29–35 (In Russian)

[8] Glover F., Laguna M Tabu Search, Kluwer Academic, Boston. 382 p., 1997.

[9] Krotov K. B. Zhadnyy algoritm postroyeniya raspisaniy obrabotki dannykh v konveyyernykh sistemakh [A greedy algorithm for scheduling data processing in conveyor systems] *vestnik VGU, seriya: sistemnyy analiz i informatsionnyye tekhnologii.* [Vestnik VSU, series: system analysis and information technologies], no 1, pp 44–60, 2015

[10] Kostenko V. A. Algoritmy kombinatornoy optimizatsii, sochetayushchiye Zhadnyye strategii i ogranichennyy perebor [Algorithms of combinatorial optimization, combining greedy strategies and limited search] Izvestiya RAN. Teoriya i sistemy upravleniya. [News of the Russian Academy of Sciences. Theory and control systems] no 2., pp. 48–56, 2017.

[11] Tanenbaum, E. *Sovremennyye operatsionnyye sistemy* [Modern operating systems. Ed. Third], Kiev, 1116 p., 2010.

[12] Kosolap, A. I. *Global'naya optimizatsiya. Metod tochnoy kvadratichnoy regulyarizatsii* [Global optimization. The method of exact quadratic regularization], Dnipropetrovsk, 164 p. 2015

[13] Nocedal J., Wright S.J., Numerical optimization. Springer, 685 p., 2006.

[14] Cook W. Fifty-Plus Years of Combinatorial Integer Programming, Georgia Institute of Technology, 39 p., 2009.

[15] Kenneth V.P., Storn R.M., Lampinen J.A. Differential Evolution. A Practical Approach to Global Optimization, Berlin, Heidelberg, 542 p., 2005.