

*Розглядається задача знаходження оптимальних навантажень енергетичних об'єктів з нелінійною негладкою функцією витрат умовного палива, яка враховує ефект «пульсації». Представлено паралельну МРІ-реалізацію алгоритму розв'язання задачі та результати обчислювальних експериментів.*

© О.П. Лиховид, О.В. Фесюк,  
2016

УДК 519.8

О.П. ЛИХОВИД, О.В. ФЕСЮК

## **МРІ-РЕАЛІЗАЦІЯ АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ ЗАВАНТАЖЕННЯ ЕНЕРГОСИСТЕМИ**

**Вступ.** Традиційно є дві основні моделі задачі завантаження енергосистеми. В першій моделі задано набір працюючих енергоблоків без можливості їх відключення, і потрібно знайти кількість електроенергії, що генерує кожний енергоблок, щоб задовольнити потребу в електроенергії, з мінімальними сумарними затратами на генерацію. В другій задачі задано набір існуючих енергоблоків з можливістю їх включення/відключення. Потрібно знайти, коли кожний енергоблок треба включити/виключити, і скільки електроенергії має генерувати кожний включений енергоблок, щоб задовольнити потребу в електроенергії з мінімальними сумарними затратами на генерацію та включення/відключення енергоблоків. Перша задача в англійській літературі має назву Economic Load Dispatch Problem, друга – Unit Commitment Problem.

В роботі [1] розглянуто одну математичну модель задачі першого типу. Ця математична модель представлена спеціальною задачею нелінійного програмування з негладкою функцією витрат умовного палива (враховано так званий ефект «пульсації»). Ця задача є багатоекстремальною і для її розв'язання було запропоновано метод мультистарту з пошуком її локальних розв'язків за допомогою  $r$ -алгоритму [2]. Метою даної роботи є

дослідження ефективності МРІ-реалізації [3] для цієї моделі.

**Математична модель задачі.** Нехай енергосистема складається з  $n$  паралельно працюючих енергоблоків. Для кожного енергоблоку  $i$  ( $i=1, \dots, n$ ) задані  $P_i^{low}$  і  $P_i^{up}$  – відповідно нижня і верхня границі його електричного навантаження. Позначимо  $T$  – тривалість добового планового періоду в годинах. Для кожного інтервалу  $t$  ( $t=1, \dots, T$ ) задано планове електричне навантаження енергосистеми  $E_t$ . Задані параметри  $DR_i$  ( $UR_i$ ) – допустимі значення на послідовне зменшення (збільшення) навантаження для  $i$ -го енергоблоку та параметр  $P_{i,0}$  – значення електричного навантаження  $i$ -го енергоблоку в момент часу перед початком планового періоду.

Нехай  $x_{it}$  – невідоме електричне навантаження  $i$ -го енергоблоку в інтервалі  $t$ . Розглянемо наступну задачу математичного програмування: знайти

$$f_C^* = \min \sum_{t=1}^T \sum_{i=1}^N f_i(x_{i,t}) \quad (1)$$

при обмеженнях

$$\sum_{i=1}^N x_{i,t} = E_t, \quad t = 1, \dots, T, \quad (2)$$

$$P_i^{low} \leq x_{i,t} \leq P_i^{up}, \quad i = 1, \dots, N, \quad t = 1, \dots, T, \quad (3)$$

$$x_{i,1} - P_{i,0} \leq UR_i, \quad i = 1, \dots, N, \quad (4)$$

$$P_{i,0} - x_{i,1} \leq DR_i, \quad i = 1, \dots, N. \quad (5)$$

$$x_{i,t} - x_{i,t-1} \leq UR_i, \quad t = 2, \dots, T, \quad i = 1, \dots, N, \quad (6)$$

$$x_{i,t-1} - x_{i,t} \leq DR_i, \quad t = 2, \dots, T, \quad i = 1, \dots, N. \quad (7)$$

Тут  $f_i(x_{i,t})$  – функція витрат умовного палива для  $i$ -го енергоблоку, яка враховує ефект «пульсації» і має наступний вигляд:

$$f_i(x_{i,t}) = a_i x_{i,t}^2 + b_i x_{i,t} + c_i + |e_i \times \sin(f_i \times (P_i^{low} - x_{i,t}))|, \quad (8)$$

де  $e_i, f_i$  – задані параметри. Така функція є негладкою та має багато локальних мінімумів в області пошуку.

**Алгоритм розв'язання.** Задача (1)–(7) з використанням функції (8) є багатоекстремальною. Для знаходження локальних екстремумів будемо використовувати метод негладких штрафних функцій [2]. Зведемо задачу (1)–(7) до задачі безумовної мінімізації наступної негладкої функції:

$$\begin{aligned}
 F(x) = & \sum_{i=1}^N \sum_{t=1}^T f_i(x_{i,t}) + Q_1 \sum_{t=1}^T \left| \sum_{i=1}^N x_{i,t} - E_t \right| + Q_2 \sum_{t=1}^T \sum_{i=1}^N \max\{0, x_{i,t} - P_i^{up}, P_i^{low} - x_{i,t}\} + \\
 & + Q_3 \sum_{i=1}^N \max\{0, x_{i,1} - p_{i,0} - UR_i, p_{i,0} - x_{i,1} - DR_i\} + \\
 & + Q_3 \sum_{t=2}^T \sum_{i=1}^N \max\{0, x_{i,t} - x_{i,t-1} - UR_i, x_{i,t-1} - x_{i,t} - DR_i\}. \quad (9)
 \end{aligned}$$

Тут  $Q_1$ ,  $Q_2$ ,  $Q_3$  – штрафні множники, які відповідають обмеженням (2), (3) та (4) – (7) відповідно. Для знаходження локальних екстремумів негладкої функції (9) будемо використовувати  $r$ -алгоритм [2]. Для знаходження глобального оптимального розв’язку будемо використовувати паралельну MPI-реалізацію методу мултистарту [4]. Далі наведемо її короткий опис.

Дана MPI-реалізація методу мултистарту запускає багаторазовий пошук локальних розв’язків за допомогою модифікації  $r$ -алгоритму [5]. Реалізація паралельного алгоритму використовує процедуру «Master-Slave» на  $(k + 1)$  процесорах. Один з них вибирається «провідним» (Master), а решта «підлеглими» (Slave). На початку обчислень в Master-процесорі генеруються  $k$  початкових точок. Ці точки генерувалися випадковим чином за допомогою функції рівномірного розподілу з інтервалу допустимих значень потужностей енергоблоків  $[P_i^{low}, P_i^{up}]$ . Потім згенеровані початкові точки пересилаються в Slave-процесори. Slave-процесор займається пошуком локального мінімуму функції (9) для отриманої їм початкової точки за допомогою  $r$ -алгоритму. Як тільки  $r$ -алгоритм закінчує роботу на будь-якому Slave-процесорі, то результат пошуку передається в Master-процесор. Якщо при цьому знайдено локальний мінімум задачі (1) – (7), то відповідне значення цільової функції  $F(x)$  порівнюється з її найкращим до цього моменту значенням («рекордом»). Якщо воно менше поточного «рекорду», то стає новим значенням «рекорду», а відповідні значення навантажень енергоблоків  $x_{it}$  запам’ятовуються. Потім Master-процесор генерує нову початкову точку, яка передається для чергового пошуку локального мінімуму в той Slave-процесор, для якого  $r$ -алгоритм закінчив роботу. Процес завершується, якщо перевищено задану кількість початкових точок або перевищено замовлений час.

**Обчислювальні експерименти.** Вищеописаний паралельний алгоритм реалізований на мові програмування C++ у програмному середовищі MPI. Проведено ряд обчислювальних експериментів на кластерному комплексі СКІТ-3 Інституту кібернетики імені В.М. Глушкова НАН України [6].

Досліджувалося два тестових приклади. Перший тестовий приклад мав наступні параметри: кількість енергоблоків – 40, кількість інтервалів в плановому періоді – 24. Загальна кількість змінних – 960, обмежень – 1944. Значення добоових потреб в електроенергії, потужності енергоблоків і параметри функції

вартості були аналогічні для тестового прикладу з роботи [1]. Другий тестовий приклад генерувався на основі даних першого, але мав більшу кількість енерго-блоків – 113. Загальна кількість змінних – 2712, обмежень – 5448.

Параметри  $r$ -алгоритму для обох прикладів були наступними: коефіцієнт розтягування простору – 2; початковий крок – 1.0;  $q_1 = 0.95$ ;  $q_2 = 1.2$ ; точність зупинки за аргументом –  $1.e-6$ ; значення штрафних коефіцієнтів – 10000.

Результати обчислювальних експериментів для першого тестового прикладу наведені в табл. 1, 2 і на рисунку. Кількість початкових точок, що генерувались, і, відповідно, запусків пошуку локального розв'язку, дорівнювала 12. В табл. 1 наведено час розв'язання, прискорення і ефективність. Тут  $k$  – загальне число процесорів,  $t$  – час розв'язання в секундах,  $S_i$  – прискорення,  $E_i$  – ефективність паралельного алгоритму. На рисунку відображена залежність часу розв'язання тестового прикладу від кількості процесорів. З цих результатів видно, що, наприклад, для 4 процесорів час виконання 12 розрахунків з різних початкових точок зменшується в 1.75 рази, для 8 процесорів – в 3.29 рази, тобто отримане прискорення значно відрізняється від «ідеального» лінійного.

В табл. 2 наведено результати пошуку локального розв'язку для прикладу 2 для 4 процесорів за різною кількістю початкових точок. Тут  $nTests$  – кількість початкових точок, що генерувались,  $t$  – час розв'язання в секундах,  $F$  – найкраще знайдене значення функції вартості. Зі зростанням кількості запусків пошуку локального розв'язку вдавалося знайти розв'язок з меншим значенням функції вартості. Найменше знайдене значення функції вартості дорівнює  $3.125937e+06$  за кількістю запусків пошуку локального розв'язку, що дорівнює 60.

Результати обчислювальних експериментів для другого тестового прикладу наведені в табл. 3. Тут  $nP$  – загальне число процесорів,  $nTests$  – кількість початкових точок, що генерувались,  $t$  – час розв'язання в секундах,  $F$  – найкраще знайдене значення функції вартості. Як і для першого прикладу, зі зростанням кількості запусків пошуку локального розв'язку знаходили розв'язок з меншим значенням функції вартості. Тут найкраще знайдене значення функції вартості дорівнює  $8.262781e+06$  за кількістю запусків пошуку локального розв'язку, що дорівнює 14.

ТАБЛИЦЯ 1. Час розв'язання, прискорення і ефективність для прикладу 1

$k$	$t$	$S_i$	$E_i$
1	1277.31	1	1
4	729.79	1.75	0.44
8	388.32	3.29	0.41

ТАБЛИЦЯ 2. Результати пошуку локальних розв'язків для прикладу 1

$nTests$	$F$	$t$
----------	-----	-----

3	3.133054e+06	179.29
12	3.126418e+06	729.79
48	3.125946e+06	2828.68
60	3.125937e+06	3454.16

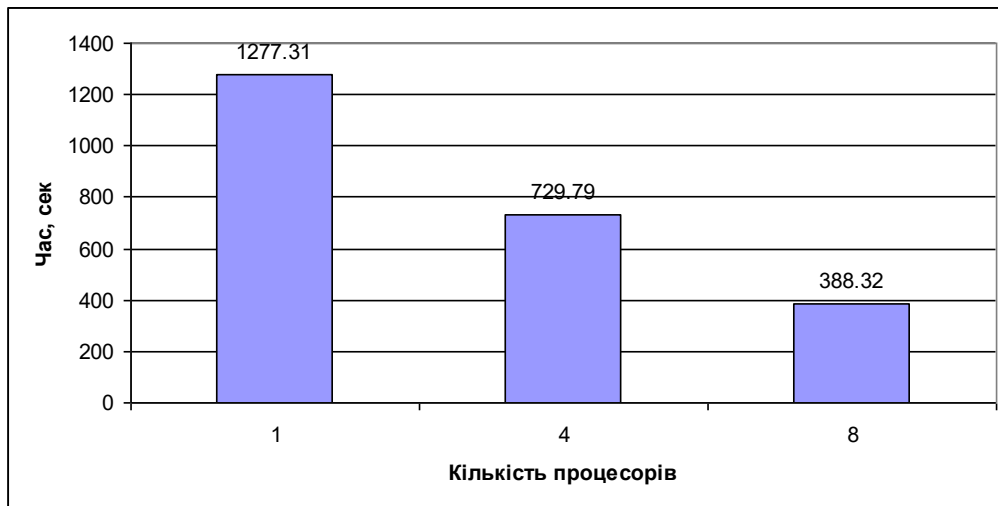


РИСУНОК. Залежність часу розв'язання прикладу 1 від кількості процесорів

ТАБЛИЦЯ 3. Результати пошуку локальних розв'язків для прикладу 2

nP	nTests	F	t
1	1	8.275091e+06	2264.86
4	3	8.270892e+06	3234.63
8	14	8.262781e+06	8048.85

Слід відзначити, що ці задачі є досить важкими і для розв'язання стандартними програмами нелінійного програмування. Наприклад, другу тестову задачу було реалізовано на мові AMPL і переслано на NEOS-сервер [7] для розв'язання відомою програмою SNOPT. Програма SNOPT витратила на пошук одного локального розв'язку 1996154 ітерацій і потім зупинилась через неможливість покращити поточний розв'язок.

**Висновки.** З результатів обчислювальних експериментів можна зробити висновок, що для практичного розв'язання великих задач добового завантаження енергосистеми з числом блоків порядку сотні та з неопуклою функцією вартості, яка враховує ефект «пульсації», для розрахунків на кластерному комплексі можна застосовувати MPI-реалізацію алгоритму на основі методу мульти-

старту з використанням  $r$ -алгоритму. В майбутньому планується покращити ефективність цієї MPI-реалізації, а також модифікувати її для врахування додаткових експлуатаційних обмежень (наприклад, обмежень на резервну потужність, заборонені зони експлуатації і т. п.).

*А.П. Лиховид, А.В. Фесюк*

#### MPI-РЕАЛИЗАЦИЯ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ ЗАГРУЗКИ ЭНЕРГОСИСТЕМЫ

Рассматривается задача нахождения оптимальных нагрузок энергетических объектов с нелинейной негладкой функцией расхода условного топлива, которая учитывает эффект «пульсации». Представлена параллельная MPI-реализация алгоритма решения задачи и результаты вычислительных экспериментов.

*О.Р. Lykhovyd, O.V. Fesyuk*

#### MPI-REALIZATION OF AN ALGORITHM FOR SOLVING THE PROBLEM OF FINDING LOAD OF POWER SYSTEM

The problem of finding optimal load of power units with a non-smooth nonlinear fuel cost function, which takes into account the valve point loading effect, is considered. Parallel MPI-realization of an algorithm for solving the problem and the results of computational experiments are presented.

1. *Лиховид А.П., Фесюк А.В.* Задачи нахождения оптимальных нагрузок энергетических объектов с нелинейными функциями стоимости // Теория оптимальных решений. – К.: Институт кибернетики имени В.М. Глушкова НАН Украины. – 2014. – С. 84 – 90.
2. *Шор Н.З.* Методы минимизации недифференцируемых функций и их приложения. – Киев: Наук. думка, 1979. – 199 с.
3. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
4. *Лиховид А.П.* О реализации параллельного алгоритма для решения многоэкстремальных задач // Теория оптимальных решений. – К.: Ин-т кибернетики имени В.М. Глушкова НАН Украины. – 2010. – № 9. – С. 3 – 9.
5. *Стецюк П.И.* Методы эллипсоидов и  $r$ -алгоритмы. – Кишинэу: Эврика; 2014. – 488 с.
6. *Кластерный комплекс Института кибернетики // Кластерный комплекс СКИТ.* <https://icybcluster.org.ua/>.
7. *Web-site of NEOS Server [Электронный ресурс]:* <http://www.neos-server.org/neos/>.

Одержано 25.04.2016