УДК 681.324

В.Ф. Третяк[1], Д.С. Шимук[1], Ю.Д. Шимук[2]

[1]*Харківський університет Повітряних Сил ім. І. Кожедуба, Харків*
[2]*Харківський національний університет радіоелектроніки, Харків*

ПАРАЛЕЛЬНИЙ МЕТОД РОЗВ'ЯЗАННЯ ЗАДАЧІ ЦІЛОЧИСЕЛЬНОГО ЛІНІЙНОГО ПРОГРАМУВАННЯ З БУЛЕВИМИ ЗМІННИМИ

Пропонується метод розв'язання задачі цілочисельного лінійного програмування з булевими змінними, який базується на ранговому підході до розпаралелювання процесу розв'язання задачі ЦЛП із БЗ на основі принципу оптимізації по напрямку з використанням правил MAX, MIN з відсіканням безперспективних варіантів розв'язань

цілочисельне лінійне програмування, розпаралелювання процесу розв'язання сильно зв'язаних задач, паралельні алгоритми,

Вступ

Постановка проблеми. Управління засобами озброєння в реальному масштабі часу в умовах швидкої зміни обстановки потребує подальшого зменшення термінів обробки інформації в АСУ для прийняття оптимальних розв'язків, що є сталою тенденцією протягом останніх десятиліть.

Аналіз літератури. Аналіз розв'язання задач в АСУ показав, що задачі оптимального планування управління, які розв'язуються в АСУ, відносяться до класу задач цілочисельного лінійного програмування з булевими змінними (ЦЛП з БЗ) [1]. Нажаль, задачі ЦЛП з БЗ відносяться до класу NP-повних задач, які з трудом підлягають розв'язанню навіть при використанні сучасних ЕОМ (таблиця 1) [1, 2].

Таблиця 1

Вплив технічного удосконалення ЕОМ на поліноміальні та експоненціальні алгоритми

Функції часової складності	Розмір найбільшої задачі, яка вирішується за 1 годину		
	На сучасних ЕОМ	На ЕОМ, в 100 разів швидших	На ЕОМ, в 1000 разів швидших
n	N_1	100 N_1	1000 N_1
n^2	N_2	10 N_2	31,6 N_2
n^3	N_3	4,64 N_3	10 N_3
n^5	N_4	2,5 N_4	3,98 N_4
2^n	N_5	6,64+N_5	9,97+N_5
3^n	N_6	4,19+N_6	6,29+N_6

Домінуюче місце в методах розв'язання цих задач у даний час займають комбінаторні методи. До них у першу чергу можна віднести методи повного перебору, віток і границь, динамічного програмування, а також локальні алгоритми [1]. Практичне застосування даних методів ускладнено при розв'язанні задач великої розмірності (табл. 2) [1, 2].

Спроби зменшення часу розв'язання задач ЦЛП з БЗ за рахунок розпаралелювання зіштовхуються з іншою проблемою теорії паралельних обчислень – з точки зору паралельних алгоритмів даний тип задач відноситься до класу сильнозв'язаних задач і тому погано підлягає розпаралелюванню [2].

Таблиця 2

Максимальна розмірність задач, які вирішуються за даний час

Складність	Час розв'язання задачі					
	1 с	10^2 с 1,7 хв.	10^4 с 2,7 хв.	10^6 с, або 12 діб	10^8 с, або 3 роки	10^{10} с, або 3 століття
1000 n	10^3	10^5	10^7	10^9	10^{11}	10^{13}
1000 n log n	140	$7{,}7\cdot 10^3$	$5{,}2\cdot 10^5$	$3{,}9\cdot 10^7$	$3{,}1\cdot 10^9$	$2{,}6\cdot 10^{11}$
100 n^2	10^2	10^3	10^4	10^5	10^6	10^7
10 n^3	46	$2{,}1\cdot 10^2$	10^3	$4{,}6\cdot 10^3$	$2{,}1\cdot 10^4$	10^5
N log n	22	36	54	79	112	156
$2^{n/3}$	59	79	99	119	139	159
2^n	19	26	33	39	46	53
3^n	12	16	20	25	29	33

Тому при реалізації методів розв'язання задач ЦЛП з БЗ на багатопроцесорних обчислювальних системах збільшення кількості процесорних елементів призводить до зниження продуктивності системи, і що для цього класу задач необхідно визначати оптимальну кількість процесорних елементів, на яких доцільно вирішувати дану задачу [5]. Таким чином, при розробці паралельних алгоритмів для розв'язання задачі ЦЛП з БЗ крім протиріччя між точністю розв'язання задачі і часом її розв'язання, виникає ще одне протиріччя – між сильною зв'язністю властивій даній задачі і необхідністю її розпаралелювання [8].

Мета статті: розробка методу паралельних обчислень для розв'язання задач ЦЛП з БЗ з метою зменшення часу розв'язання задач оптимального планування в АСУ.

Основний матеріал

Для об'єктивної оцінки ефективності алгоритмів розв'язання задачі ЦЛП з БЗ використовуються наступні показники:

– величина показника ймовірності своєчасного виконання розрахунків в АСУ, яка обумовлена співвідношенням:

$$P = 1 - e^{-\frac{T_д}{T}}, \qquad (1)$$

де $T_д$ – допустимий час розв'язання задачі, T – час виконання розрахунків.

– обсяг пам'яті (V) для реалізації алгоритму;

– для оцінювання якості наближених алгоритмів обрані абсолютна і відносна похибки:

$$\delta f = \left| f(x) - f(x^*) \right|, \qquad (2)$$

$$\Delta f = \left| f(x) - f(x^*) \right| \big/ f(x^*), \qquad (3)$$

де $f(x)$, $f\left(x^*\right)$ – наближене і точне розв'язання задачі;

– коефіцієнт прискорення паралельних обчислень відносно послідовних:

$$K_п = T_{посл.} \big/ T_{пар.}; \qquad (4)$$

– кількість елементарних операцій (ЕО) додавання і порівняння, які виконуються алгоритмами;

– t_c – середній час розв'язання задачі. Цей параметр залежить від обраної мови програмування, технічних даних ЕОМ, кваліфікації програміста;

– r_c – середній ранг шляхів, що відповідають оптимальному розв'язанню тестової задачі, який обумовлений числом одиниць у векторі x;

– $K_н$ – кількість неточних розв'язань у залежності від загальної кількості тестових задач які вирішуються наближеними алгоритмами, що виражена у відсотках;

– коефіцієнт виграшу $K_в$ у швидкодії розроблених алгоритмів стосовно алгоритму, із яким здійснюється порівняння, рівний $K_в = EO_п/EO_р$, де $EO_п$ - кількість елементарних операцій (ЕО) алгоритму з яким здійснюється порівняння і $EO_р$ - кількість ЕО розробленого алгоритму;

Процес установлення відповідності між заданою задачею і конкретним типом паралельної обчислювальної структури (ПОС) поданий у вигляді послідовного виконання чотирьох етапів: розробки послідовного алгоритму розв'язання задачі [3, 4]; розробки алгоритму паралельних обчислень [8, 9, 10]; одержання логічного опису паралельної архітектури [11]; розробки ПОС [11].

Природно очікувати, що наскільки вдало обрано алгоритм, настільки вдалою може виявитися і відповідна апаратна реалізація. Більш того, деякі алгоритми більше пристосовані під визначені обчислювальні структури, чим під інші. Тому, для розв'язання поставленої задачі, насамперед, розробляється послідовний алгоритм.

У зв'язку з тим, що розв'язання на основі рангового підходу дозволяє ефективно розпаралелити процес розв'язання задачі ЦЛП із БЗ на ПОС, дана модель і була обрана. [4, 5, 8] Ця модель базується на представленні n-мірного одиничного кубу який визначає область допустимих розв'язань задачі

$$f\left(\vec{x}\right) = \sum_{j=1}^{n} c_j x_j \qquad (5)$$

при виконанні обмежень

$$\sum_{j=1}^{n} \alpha_{ij} x_j \le b_i, \qquad (6)$$

$$x_j \in \{0,1\}, \quad i = \overline{1,m}, \; j = \overline{1,n}. \qquad (7)$$

у вигляді графу D, який упорядковує ці розв'язання.

У основу побудови наближених алгоритмів розв'язання задач ЦЛП з БЗ покладений принцип оптимізації по напрямку в дискретному просторі станів, який заданий графом DΔ. [3, 6, 8]. Під оптимізацією по напрямку в графі DΔ до вершини p мається на увазі формування множин шляхів m_{sp}^{r+1} наступного рангу, що одержані за рахунок виділення в m_{sj}^{r} шляхів, приєднання до яких ребра (j,p) дозволить у множині m_{sp}^{r+1} отримати шляхи, що задовольняють правилам L_w на основі наступного рекурентного співвідношення

$$\begin{aligned} &\forall(\mu_{sj}^{r} \in m_{sj}^{r}) \left| \mu_{sp}^{r:=r+1} = L_w \left\{ \mu_{sj}^{r} U(j,p) \right\} \right. \\ &p = (r+1, r+2, ..., n); \quad j = (r, r+1, ..., n), \end{aligned} \qquad (8)$$

де $\mu_{sj}^{r} \cup (j,p)$ – шлях із вершини s графа DΔ у вершину p, який проходить через проміжну вершину j і задовольняє правилам L_w, тобто одержаний за рахунок приєднання до шляху m_{sj} ребра (j,p), коли таке сполучення не суперечить правилам L_w.

При побудові наближених алгоритмів із множини правил $\{L_w\}$ розглянуті такі:

$$L_1 : \mu_{sp}^{r=r+1} = \max_{\{c_j\}} \{\mu_{sj}^{r} U(j,p)\}, \qquad (9)$$

де $p = \overline{r+1,n}$, $\quad j = \overline{r,n}$, $\quad j \ne p$.

$$L_2 : \mu_{sp}^{r=r+1} = \min_{\{\alpha_j\}} \{\mu_{sj}^{r} U(j,p)\}; \qquad (10)$$

$$L_3 : \begin{cases} \mu_{sp}^{r=r+1} = \max\limits_{c_j} \left\{ \mu_{sj}^{r} \cup (j,p) \right\} \\ \mu_{sp}^{r=r+1} = \min\limits_{\alpha_j} \left\{ \mu_{sj}^{r} \cup (j,p) \right\} \end{cases} \qquad (11)$$

Відповідно до правил, вираз (9) – стратегія MAX при побудові шляхів наступного рангу обирається шлях з максимальною довжиною по вазі функціоналу, вираз (10) – стратегія MIN при побудові шляхів наступного рангу обирається шлях з мінімальною довжиною по вазі обмежень, вираз (11) – стратегія MAX-MIN являє собою комбінацію стратегій MAX та MIN.

Розроблені наближені алгоритми дозволяють визначати локальні екстремуми у вершинах графа DΔ і на основі їх визначати глобальний екстремум, що отриманий на основі принципу оптимізації по напрямку з використанням правил MAX, MIN, MAX-MIN, для відсікання безперспективних варіантів розв'язань у множинах m_{sj}^{r}.

Побудова паралельного алгоритму основана на формуванні для нього графа залежностей (ГЗ), що описує залежність обчислень, які подані у паралельному алгоритмі. Побудови ГЗ заснована на просторово-тимчасових індексах у рекурсивному алгоритмі: просторові просторово-тимчасових індексів рекурсивного алгоритму відповідає природний сітковий простір для ГЗ, у кожній точці якого розміщається один вузол. Залежності даних в алгоритмі виражені дугами, що з'єднують взаємодіючі вузли в ГЗ, у які вбудовано їхній функціональний опис [2].

З метою розробки паралельних алгоритмів абстрактна обчислювальна система розглядається як множина процесорних елементів $Q=\{q_1,q_2,...,q_n\}$. Кожний процесорний елемент характеризується ім'ям $q_i \in Q$ і алфавітом A_i. Символи алфавіту інтерпретуються як константи або змінні. Парою (a,q_i) позначений стан процесорного елемента q_i. Множина станів усіх процесорних елементів системи $S=\{(a,q_i)\}, i=1,...,n \ \ a \in A_i$, у той самий момент часу визначається глобальним станом системи, а деяка його підмножина $S' \subset S$ - локальним станом.. Виконуючи обробку даних, процесорний елемент змінює свій стан і, можливо, стан деяких інших, пов'язаних із ним, елементів, тобто робить локальне перетворення.

Паралельне обчислення являє собою сумісність виконання у часі множини операцій $\Phi=\{\theta_j\}$ $j=1,...,n$, де $\theta_j : S_i' \longrightarrow S_i''$, $S_i'=(a_i,q_i)$ – локальний стан, названий умовою готовності операції θ_j; $S_i''=(b_i,q_i)$ – локальний стан, названий умовою завершення операції θ_j. У результаті застосування θ_j відбувається зміна станів $(a_i,q_i)\in S_i$ на $(b_i,q_i)\in S_i''$ [2].

Обчислювальний процес паралельного алгоритму описується виразом [7, 10, 12]:

$$q_1 =: S'(a_m,q_m)+S'(a_p,q_p) \to S_L''(b_1,q_1);$$

$$q_2 =: S'(a_m,q_m)+S'(a_p,q_p) \to S_L''(b_2,q_2);$$

$$q_{n-1} =: S'(a_m,q_m)+S'(a_p,q_p) \to S_L''(b_{n-1},q_{n-1}),$$

де r – ранг розв'язання задачі, n - кількість вершин графа, m = (i - k) - n, p = i + r, k - кількість ітерацій, i - номер процесорного елемента, L - правило відсікання безперспективних варіантів розв'язань у множинах m_{sj}^{r}.

При зміні глобальних станів відбувається виділення із множини локальних екстремумів глобального екстремуму:

$$\text{ext}(f)=\max\left(S''(b_1,q_1)\vee S''(b_2,q_2)\vee ...\vee S''(b_{n-1},q_{n-1})\right)$$

$$\forall q_i \to S'(a_m,q_m)=:S''(b_m,q_m),$$

при цьому перемінним присвоюються значення $k:=0$, $r:=r+1$, $q_{(n-1)r}:=0$. Блок-схема алгоритму представлена на рис. 1.

Структура дослідження кожного алгоритму полягала в наступному. Вирішувалося 100 тестових задач із заданими вхідними параметрами n обраним алгоритмом із різноманітним сортуванням коефіцієнтів при функціоналі й обмеженні:

$$\alpha_1 \le \alpha_2 \le ... \le \alpha_n \qquad (12)$$

$$c_1 \ge c_2 \ge ... \ge c_n \qquad (13)$$

$$\frac{c_1}{\alpha_1} \ge \frac{c_2}{\alpha_2} \ge ... \ge \frac{c_n}{\alpha_n} \qquad (14)$$

У ході розв'язання обчислювалися показники ефективності алгоритмів. Результати рішення задачі наближеним алгоритмом порівнювались з результатом її рішення точним алгоритмом.

Як показали результати експериментального дослідження, кількісні значення обраних показників істотно залежать від рангу одержуваного рішення, що визначає число одиниць в оптимальному рішенні. Так, діапазон зміни r_c умовно розбитий на три умовно виділені зони:

$$1 \text{ зона} - r^c=\left[0\div\frac{n}{3}\right]; \qquad 2 \text{ зона} - r^c=\left[\frac{n}{3}\div\frac{2n}{3}\right];$$

$$3 \text{ зона} - r^c=\left[\frac{2n}{3}\div n\right].$$

У першій зоні алгоритм знаходить рішення швидко, оскільки за рахунок зондування дуже ефективно відсікаються вітки дерева рішень, що відповідають одиничним розгалуженням. Аналогічною пояснюється і швидке одержання рішення в 3 зоні, вектори в якій складаються з великої кількості одиниць. Прояв усієї експоненціальної складності алгоритму відбувається саме в другій зоні. Для наближених алгоритмів ріст числа ЕО зі збільшенням r_c визначається числом локальних областей W, що необхідно опрацювати.

Відповідно до розподілу на зони побудовані графіки залежності точності наближених рішень для другої зони від засобу сортування коефіцієнтів (рис. 2). Використовуючи середнє значення часу рішення задачі і допустимого часу (3 хвилини) було визначено значення показника P (рис. 3).

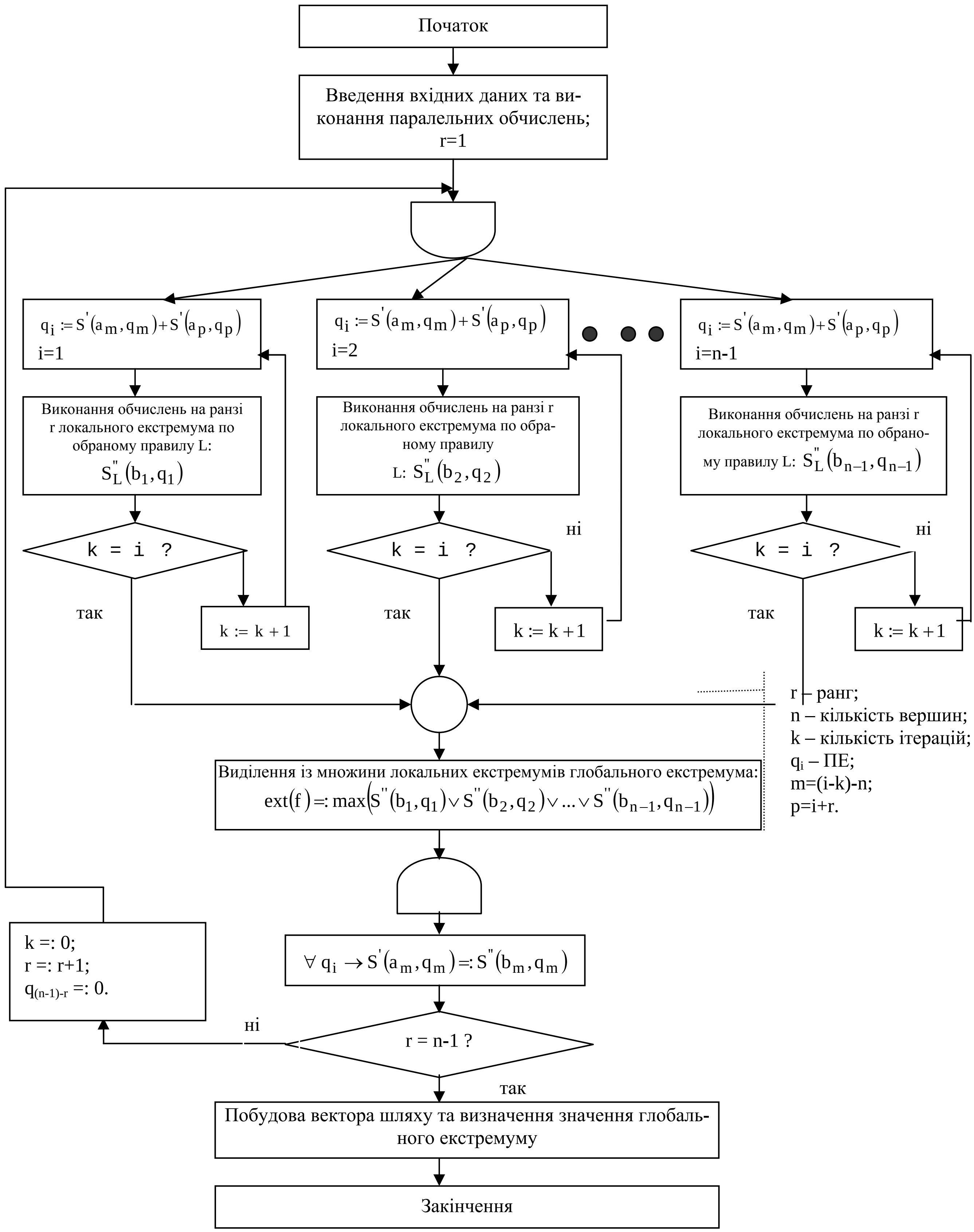

Рис. 1. Блок-схема алгоритму

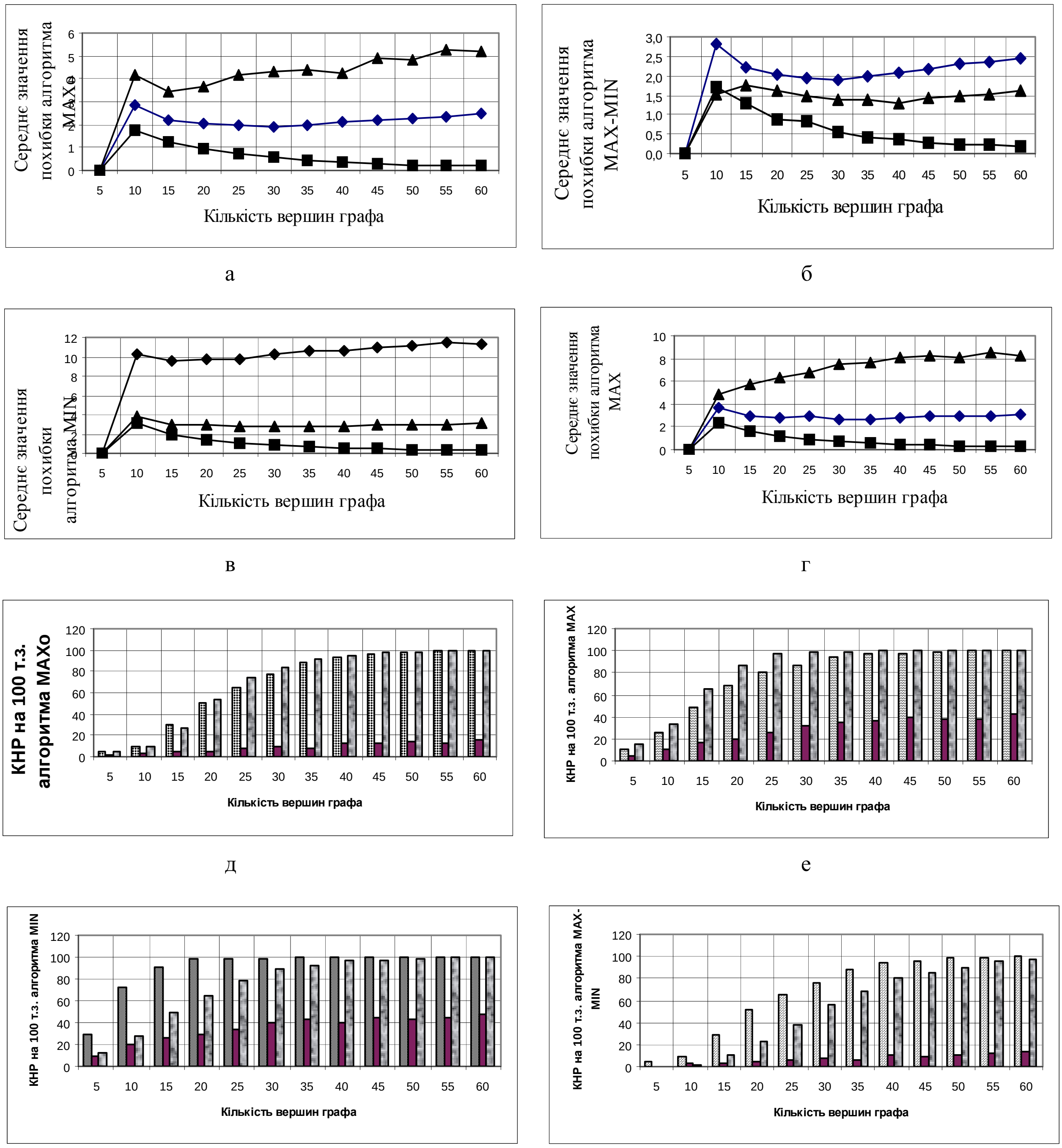

Рис. 2. Оцінка ефективності алгоритмів

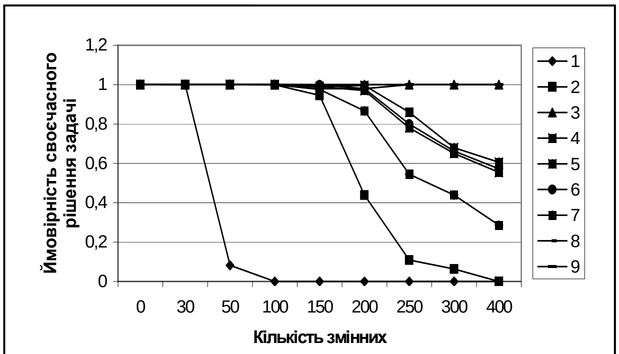

Рис. 3. Оцінка показника ймовірності своєчасного рішення задач оптимального планування в АСУ

Висновки

1. Найкращою точністю володіє алгоритм MAX-MIN, похибка якого при n>50 не перевищує 0,5%.

2. Найбільш ефективним сортуванням є сортування в порядку зменшення значень відношення коефіцієнтів при функціоналі до відповідних коефіцієнтів в обмеженнях.

У якості алгоритмів для дослідження були обрані:

- точні алгоритми:

1 – алгоритм Балаша;

2 – багатоетапний алгоритм на основі рангового підходу;

- наближені алгоритми:

1 – алгоритми з тимчасовою складністю O (n);

2 – алгоритми з тимчасовою складністю O(n^2) при сортуванні по обмеженню;

3 – алгоритми з тимчасовою складністю O (n^2) при сортуванні по функціоналу;

4 – алгоритми з тимчасовою складністю O (n^2) при сортуванні по відношенню;

5 – алгоритми з тимчасовою складністю O (n^3);

6 – паралельні алгоритми;

7 – алгоритм, що забезпечує задану точність обчислень при припустимих витратах на час та ресурси.

Аналіз показника ймовірності своєчасного рішення задач оптимального планування показав в АСУ, що забезпечення значення оперативності $P \geq 0{,}9$ можливо тільки алгоритмами з тимчасовою складністю $O(n)$ і алгоритмом, що забезпечує задану точність обчислень при припустимих витратах на час та ресурси. Застосування точних методів можливо при невеликій розмірності задачі – до 250 вершин графа – алгоритмами з тимчасовою складністю від $O\left(n^2\right)$ до 450.

Список літератури

1.Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 336 с.

2.Корректность параллельных вычислительных процессов / Ачасова С.М., Бандман О.Л. – Новосибирск: Наука, 1990. – 253 с.

3.Голубничий Д.Ю. Модификация алгоритма Балаша для задач с положительными коэффициентами // Сб.науч.трудов III-й НТК "Новые технологии в машиностроении". – Х: Основа. – 1995. – С. 43-47.

4.Королев А.В., Голубничий Д.Ю., Третьяк В.Ф. Метод решения задачи о ранце на основе рангового подхода // ІКСЗТ. – 1998. – № 2. – С. 13-17.

5. Королев А.В., Листровой С.В., Третьяк В.Ф. Эффективность параллельных алгоритмов оптимизации вычислительного процесса // ИУСЖТ. – 1997. – №1. – С. 85-91.

6. Koroljov A.V., Krjuchov O.M., Listrovoj S.V., Tretjak V.F. An optimal planning of algorithms in computer systems // Thesis of conference "Mathematical modeling and information technologies". – Belgorod (Russiya). – 1997. – P. 32-33.

7. Листровой С.В., Третьяк В.Ф., Листровая Е.С. Параллельные алгоритмы оптимизации вычислительного процесса для задач булевого программирования // Электронное моделирование. – 1998. – № 5. – С. 23-33.

8. Математичні моделі і методи в обчислювальних системах та мережах ЕОМ: Навчальн. посібн. Д.Ю. Голубничий, С.В. Лістровий, О.Ю. Гуль, В.Ф. Третяк – Х.: ХВУ, 1998. – 191 с.

9. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации. – К.: Наук. думка, 1988. – 472 с.

10. Третьяк В.Ф., Ильина Е.А., Яблочков С.В. Параллельные методы решения задачи целочисленного линейного программирования с булевыми переменными // Информатика. – 1998. – С. 25-31.

11. Третьяк В.Ф., Листорой С.В., Яблочков С. В. Алгоритм параллельных вычислений для решения задач ЦЛП с БП // Информационные системы. – 2006. – Х.: НАНУ, ПАНИ, ХВУ. – Вып. 3(11). – С. 54-57.

12. Третьяк В.Ф., Кавун С.В., Гуль А.Ю. Реализация приближенного алгоритма целочисленного программирования на систематических однородных средах // Обработка информации и обеспечения надежности систем управления: Сб. научн. тр. – Х.: НАНУ, ПАНИ, ХВУ, 1997 – С. 144-147.

Надійшла до редколегії 31.10.2006

Рецензент: д-р техн.. наук, проф. Б.Ф. Самойленко, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.