УДК 519.161

А.О. Данильченко

СКЛАДАННЯ ОПТИМАЛЬНОГО РОЗКЛАДУ ЗА НАЯВНОСТІ ЗАДАНИХ ОБМЕЖЕНЬ НА ОСНОВІ ТЕОРІЇ ГРАФІВ

The purpose of this study is to develop an algorithm for solving the nurse scheduling problem for home patients who receive treatment that will provide a lower computational complexity than the method of exhaustive search and will allow finding a solution that meets the specified limit for the procedures. The proposed new algorithm for solving the application problem of scheduling treatment of patients who receive nursing home as an extended mathematical problem of finding the maximum matching in a bipartite graph with vanishing edges. In contrast to the known ones, the algorithm can take into account the limited compatibility of treatment and has a lower computational complexity than the method of exhaustive search by reducing the number of matches to be analyzed. In addition, we conduct the comparative computational experiment relying on a series of the task random environment obtained from real home patients based on the hourly sampling. It shows that the proposed algorithm provides optimal reduction time scheduling from 4,48 times to 8,87 times compared to exhaustive search method. Time scheduling is directly proportional to the number of vertices of bipartite graphs.

Вступ

Призначення процедур у сучасних санаторних і лікувальних закладах є складним процесом, який має враховувати доволі велику кількість чинників, основними з яких є такі [1, 2]:

- перелік призначених лікарем процедур;
- час роботи процедурного кабінету;
- пропускна здатність процедурного кабінету (одну процедуру одночасно можуть приймати кілька пацієнтів);
- тривалість прийому процедури (для різних процедур тривалість прийому різна);
- тривалість часу технічної перерви між прийомами процедур;
- сумісність процедур (пацієнт не може одночасно приймати кілька процедур, але, крім цього, на розклад накладається додаткове обмеження – пацієнт не може приймати наступну процедуру менш ніж через деякий час після прийняття попередньої, для кожної пари процедур значення часу сумісності може різнитися).

Зрозуміло, що зазначена конкретна ситуація може бути поширена на велику кількість споріднених задач складання оптимальних розкладів при наявності деякої множини обмежень. Наприклад, розподіл за часом обмежених ресурсів, призначення виконання різних видів робіт (операцій, завдань, процесів).

Такі задачі розв’язуються математичним апаратом теорії розкладів – розділу прикладної математики, що вивчає моделі упорядкування робіт і методи складання розкладів. При розв’язанні цього класу задач застосовують різні методи комбінаторної оптимізації [3], але наявність додаткових обмежень істотно їх ускладнює.

Задача складання оптимального розкладу може бути розв’язана за допомогою теорії графів, зокрема, модифікацією (а саме, врахуванням додаткових обмежень) класичної задачі про паросполучення, варіанти розв’язання якої в різних постановках наведено в [4–6].

У статті [1] розглянуто задачу складання розкладу проходження процедур пацієнтами санаторію. Сформульована задача зведена до розширеної задачі пошуку максимального паросполучення у дводольному графі. Розроблено оптимальний алгоритм її розв’язання.

Із NP-повноти задачі “про паросполучення зі зникаючими дугами”, яку доведено в [1], випливає, що вона не піддається ефективним точним методам [7].

Відомо, що будь-яка NP-повна задача може бути розв’язана методом повного перебору. Але при цьому, залежно від розмірності задачі, потрібні обчислювальні ресурси та час її розв’язання можуть бути неприпустимо великими з практичної точки зору.

Для оптимізації процесу повного перебору застосовують метод гілок і меж, який дає можливість зменшувати множину допустимих розв’язків за допомогою ефективного алгоритму пошуку, а також розпаралелювання обчислень.

Варто зазначити, що для методу гілок і меж найбільшу складність мають саме процедури розгалуження та знаходження оцінок верхніх і нижніх меж для оптимального значення на підмножині припустимих розв'язків.

Розпаралелювання обчислень не звужує кількість варіантів, що аналізуються, а лише скорочує потрібний на це час.

Таким чином, найбільш доцільною схемою зменшення обчислювальної складності знаходження точного розв'язання NP-повних задач залишається скорочення повного перебору.

Постановка задачі

Метою статті є розроблення точного алгоритму, який, згідно із запропонованою в [1] математичною моделлю, має забезпечити меншу обчислювальну складність порівняно з методом повного перебору за рахунок скорочення кількості паросполучень, що аналізуватимуться, та гарантоване знаходження найбільшого паросполучення, яке відповідатиме заданим обмеженням.

Оптимальний алгоритм розв'язання задачі про паросполучення зі зникаючими дугами

Нехай задано дводольний граф $G=(X,Y,E)$, в якому X – множина вершин графа, що відповідають усім можливим проміжкам прийому процедур (згідно із встановленим графіком роботи відповідного процедурного кабінету) $\|X\|=m$; Y – множина вершин графа, які відповідають процедурам, призначеним пацієнтам санаторію, $\|Y\|=n$ (при цьому кожна вершина множини Y має ознаку приналежності до певного пацієнта); E – множина ребер графа. Ребро $(x_i, y_k)\in E$, $x_i\in X$, $y_k\in Y$, $i=1,...,m$, $k=1,...,n$, в тому випадку, коли процедура y_k може бути призначена для прийому визначеним пацієнтом у проміжок часу x_i.

Відомі обмеження задані множиною наслідків C, таких, що $(x_i,y_j)\rightarrow C_{i,j}=\{(x_{i_1},y_{j_1}),...,(x_{i_k},y_{j_k})\}$. Ці обмеження враховують, насамперед, неможливість призначення тієї самої процедури різним пацієнтам на однаковий час (із врахуванням пропускної здатності процедурного кабінету), а також сумісність процедур. Крім того, можна врахувати за необхідності й деяку послідовність прийняття процедур (наприклад, деяка процедура має обов'язково передувати заданій множині інших процедур) тощо.

Скорочення кількості паросполучень, що аналізуватимуться, можна досягнути за рахунок врахування наслідків $C_{i,j}$ при виборі деякого паросполучення (x_i,y_j) на кожному з послідовних кроків алгоритму. Тоді при виборі будь-якої дуги (x_i,y_j) для включення її до множини паросполучень на ітерації t множина ребер графа для подальшого аналізу може бути виражена як $E^{t+1}\Big|_{(x_i,y_j)}=E^t-C_{i,j}$. При цьому отримаємо по кожній з ітерацій множину E^{t+1} меншої потужності, що й звужує область аналізу порівняно з методом повного перебору.

Для гарантованого знаходження найбільшого паросполучення M в алгоритмі, що розробляється, необхідно реалізувати послідовний перебір усіх можливих варіантів побудови паросполучень на заданому графі $G=(X,Y,E)$ із врахуванням заданих обмежень C. З цією метою потрібно організувати цикл по всіх вершинах множини Y і вкладений цикл по підмножині інцидентних поточній вершині y_i дуг.

Достатньою умовою оптимальності знайденого варіанту розв'язання задачі буде виконання рівності $\|M\|=\|Y\|$. При знаходженні такого найбільшого паросполучення виконання алгоритму може бути завершене.

Необхідною умовою оптимальності знайденого варіанту розв'язання задачі є виконання умови $\|M\|=\max$, тобто знаходження паросполучення M максимальної потужності.

Якщо після закінчення циклу перебору всіх вершин множини Y достатня умова оптимальності не виконується, то при заданих обмеженнях може бути знайдений оптимальний розв'язок, який забезпечить призначення найбільшої можливої кількості (але не всіх призначених) процедур при заданих обмеженнях.

Оптимальний алгоритм розв'язання задачі про паросполучення зі зникаючими дугами (рис. 1) включає такі кроки:

1. Будуємо дводольний граф $G=(X,Y,E)$, в якому X – множина вершин графа, що відповідають можливим проміжкам прийому процедур, $\|X\|=m$; Y – множина вершин графа, які відповідають процедурам, що призначені пацієнтам санаторію, $\|Y\|=n$; P – множина пацієнтів; E – множина ребер графа. Ребро $(x_i, y_k)\in E$, $x_i \in X$, $y_k \in Y$, $i=1,...,m$, $k=1,...,n$, в тому випадку, коли процедура пацієнта y_k може бути призначена для прийому в проміжок часу x_i. Задано відношення наслідку $C_{i,j}$. Множина ребер, які входять до максимального паросполучення, порожня множина. Нехай $k=q$, $q=1$.

2. Послідовно перебираємо, починаючи з $k=q$, вершини множини Y. Якщо всі вершини були вибрані, переходимо до пункту 5.

3. Для вершини y_k послідовно досліджуємо ребра $(x_i, y_k)\in E$, $i=1,...,m$. Нехай знайдене деяке ребро $(x_i, y_k)\in E$, $y_k \in P_j$, претендент на включення до M, тобто жодна з вершин x_i та y_k не інцидентна ребрам, що вже входять до паросполучення M. Враховуємо задані обмеження, для чого тимчасово виключаємо з графа $G=(X,Y,E)$ дуги, $(x_i, y_k)\to C_{i,k}$.

4. Перевіряємо наявність дуг $(x_i, y_r)\in E$, $x_i \in X$, $y_r \in Y$, $i=1,...,m$, $r=k+1,...,n$, тобто інцидентних наступним вершинам у множині

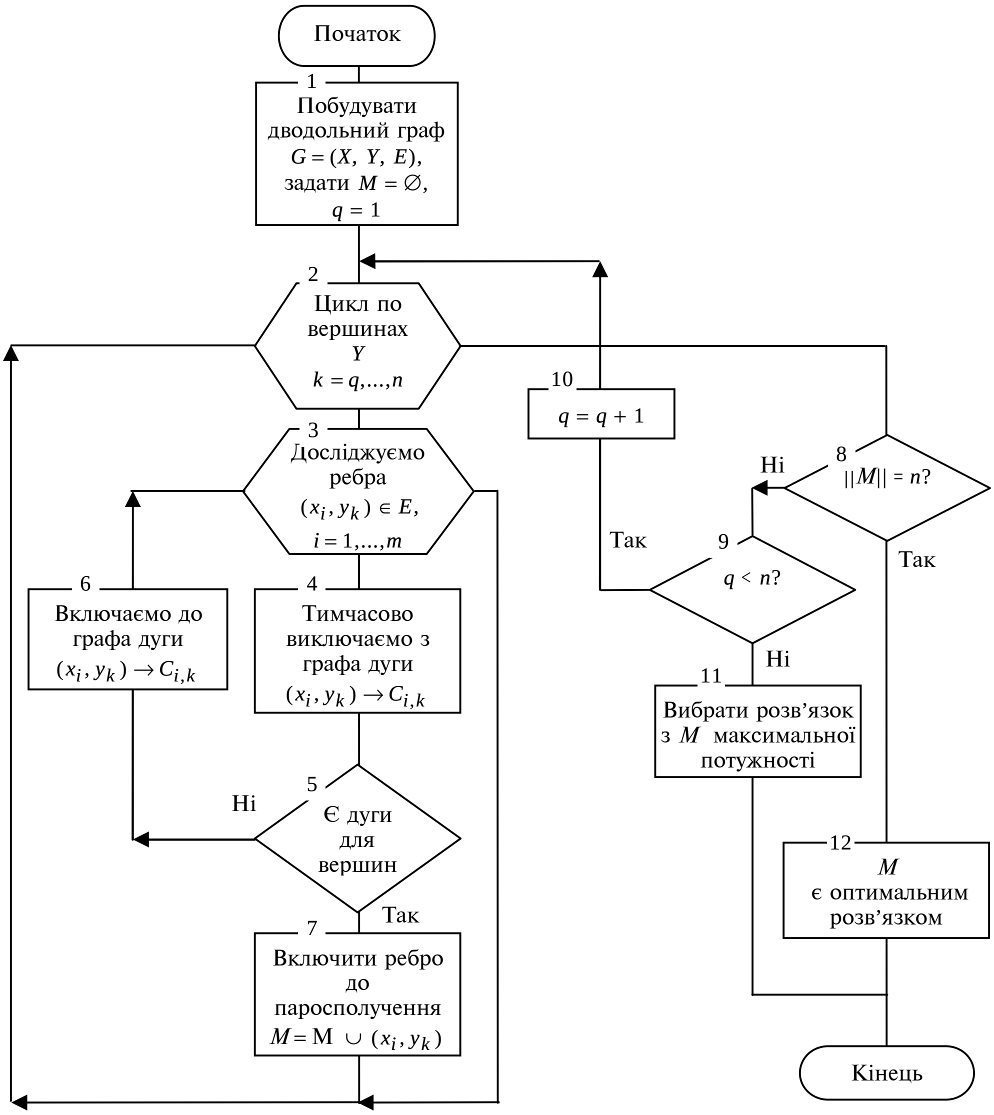

Рис 1. Схема оптимального алгоритму розв'язання задачі про паросполучення зі зникаючими дугами

Y (ще є процедури, для яких не призначено час їх проходження). Якщо по кожній з цих вершин є хоча б одна інцидентна їй дуга, то $M = \mathrm{M}\,\mathbf{U}(x_i, y_k)$. Якщо $k = n$, перехід на пункт 2; інакше $k = k + 1$ і переходимо до пункту 3.

В іншому випадку (немає інцидентних дуг, тобто неможливо призначити час для всіх процедур) включаємо до графа $G = (X, Y, E)$ дуги, $(x_i, y_k) \to C_{i,k}$ та шукаємо наступне ребро: якщо $i = m$, перехід на пункт 2; інакше $i = i + 1$ і переходимо до пункту 3.

5. Перевіряємо, що в знайденому паросполученні присутні всі призначені процедури.

Якщо $\|\mathrm{M}\| = n$, то розв'язок знайдено, кінець алгоритму.

Якщо $\|\mathrm{M}\| \neq n$, то $q = q + 1$. Якщо $q < n$, перехід на пункт 2; інакше необхідно вибрати зі знайдених розв'язків паросполучення максимальної потужності.

Приклад розв'язання задачі про паросполучення зі зникаючими дугами за допомогою запропонованого оптимального алгоритму

Розглянемо приклад.

Крок 1 (рис. 2, *а*). Побудований дводольний граф $G = (X, Y, E)$ містить множину вершин графа, що відповідають можливим проміжкам прийому процедур: $X = \{9.30 - 10.30,$ $11.00 - 12.00, 12.30 - 13.30, 9.30 - 10.00, 11.00 - 11.30,$ $12.30 - 13.00\}$, $\|\mathrm{X}\| = 6$; множину вершин графа, які відповідають процедурам, що призначені пацієнтам санаторію $Y = \{1_1, 2_1, 1_2\}$, $\|\mathrm{Y}\| = 3$; E – множину ребер графа, які відповідають випадку, коли процедура пацієнта y_k може бути призначена для прийому в проміжок часу x_i.

Введемо обмеження:

- неможливість призначення тієї самої процедури різним пацієнтам на однаковий час;
- друга процедура не може призначатися після першої раніше ніж через одну годину.

Задані обмеження визначимо множиною наслідків C, таких, що

$$(x_i, y_j) \to C_{i,j} = \{(x_{i_1}, y_{j_1}), \ldots, (x_{i_k}, y_{j_k})\}:$$

$$(x_1, y_1) \to C_{1,1} =$$
$$= \{(x_1, y_3), (x_2, y_1), (x_3, y_1), (x_4, y_2), (x_5, y_2)\};$$

$$(x_2, y_1) \to C_{2,1} =$$
$$= \{(x_2, y_3), (x_1, y_1), (x_3, y_1), (x_5, y_2), (x_6, y_2)\};$$
$$(x_3, y_1) \to C_{3,1} = \{(x_3, y_3), (x_1, y_1), (x_2, y_1), (x_6, y_2)\};$$
$$(x_4, y_2) \to C_{4,2} = \{(x_5, y_2), (x_6, y_2), (x_1, y_1)\};$$
$$(x_5, y_2) \to C_{5,2} = \{(x_4, y_2), (x_6, y_2), (x_1, y_1), (x_2, y_1)\};$$
$$(x_6, y_2) \to C_{6,2} = \{(x_4, y_2), (x_5, y_2), (x_2, y_1), (x_3, y_1)\};$$
$$(x_1, y_3) \to C_{1,3} = \{(x_2, y_3), (x_3, y_3), (x_1, y_1)\};$$
$$(x_2, y_3) \to C_{2,3} = \{(x_1, y_3), (x_3, y_3), (x_2, y_1)\};$$
$$(x_3, y_3) \to C_{3,3} = \{(x_1, y_3), (x_2, y_3), (x_3, y_1)\}.$$

Множина ребер, що входять до максимального паро сполучення, порожня множина. Нехай $k = q$, $q = 1$.

Крок 2. Беремо вершину y_1. Ця вершина відповідає призначенню першої процедури для першого пацієнта.

Крок 3. Починаємо послідовне дослідження інцидентних вершині y_1 дуг. Беремо перше ребро $(9.30 - 10.30,\ 1_1)$.

Тимчасово виключаємо з графа $G = (X, Y, E)$ такі дуги: $(11.00 - 12.00,\ 1_1)$, $(12.30 - 13.30,\ 1_1)$, $(9.30 - 10.30,\ 2_1)$ – за умовою неможливості призначення тієї самої процедури різним пацієнтам на однаковий час; $(9.30 - 10.00,\ 1_2)$, $(11.00 - 11.30,\ 1_2)$ – за умовою призначення другої процедури після першої не раніше ніж через одну годину.

Крок 4. Перевіряємо на отриманому графі (рис. 2, *б*) наявність дуг $(x_i, y_r) \in E$, $x_i \in X$, $y_r \in Y$, $i = 1, \ldots, m$, $r = k + 1, \ldots, n$, тобто інцидентних наступним вершинам у множині Y (ще є процедури, для яких не призначено час їх проходження). Оскільки по кожній з цих вершин (y_2, y_3) є інцидентні їм дуги, то $M =$ $= \mathrm{M}\,\mathbf{U}(9.30 - 10.30,\ 1_1)$, $k = k + 1$; переходимо до пункту 3.

Крок 5. $(k = 2)$ Беремо вершину y_2. Вона відповідає призначенню другої процедури першому пацієнту. Починаємо дослідження інцидентних їй дуг.

Крок 6. Беремо ребро $(12.30 - 13.00,\ 1_2)$. Виключення дуг із графа $G = (X, Y, E)$ у цьому випадку проводити непотрібно.

Крок 7. Перевіряємо на отриманому графі наявність дуг $(x_i, y_r) \in E$, $x_i \in X$, $y_r \in Y$, $i =$

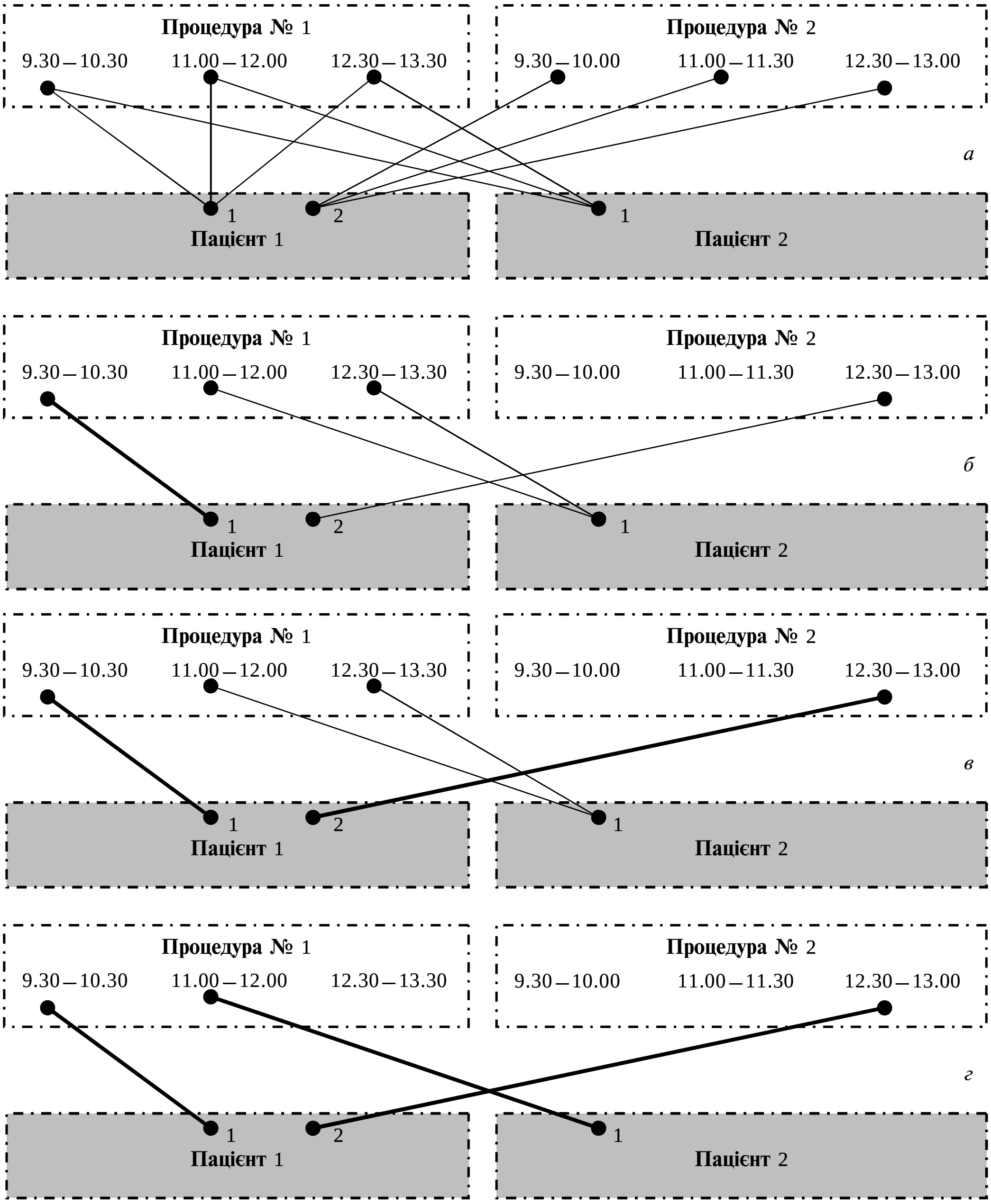

Рис. 2. Розв'язання задачі про максимальні паросполучення зі зникаючими дугами за оптимальним алгоритмом

$=1,\ldots,m$, $r=k+1,\ldots,n$, тобто інцидентних наступній вершині (2_1). Оскільки для неї є більш ніж одна інцидентна дуга (рис. 2, *в*), то $M = \mathrm{M}\,\mathbf{U}(12.30-13.00,\ 1_2)$, $k=k+1$ і переходимо до пункту 3.

Крок 8. ($k=3$) Беремо вершину y_3. Вона відповідає призначенню першої процедури для другого пацієнта. Починаємо дослідження інцидентних їй дуг.

Крок 9. Беремо ребро ($11.00-12.00,\ 2_1$). Тимчасово виключаємо з графа $G=(X,Y,E)$ дугу ($12.30-13.30,\ 2_1$).

Крок 10. Наступних вершин у множині Y немає; $M = \mathrm{M}\,\mathbf{U}(11.00-12.00,\ 2_1)$ (рис. 2, *г*); переходимо до пункту 2.

Крок 11. Оскільки всі вершини у множині Y були вибрані, переходимо до пункту 5.

Крок 12. Перевіряємо, чи у знайденому паросполученні присутні всі призначені процедури.

Оскільки $\|M\| = 3$, то оптимальний розв'язок при заданих обмеженнях знайдений, кінець алгоритму.

Результати порівняльного обчислювального експерименту розв'язання задачі про паросполучення зі зникаючими дугами

Основною метою порівняльного обчислювального експерименту є оцінка часових витрат на виконання розрахунків різними алгоритмами на обчислювальних платформах, які мають розбіжні характеристики (кількість ядер мікропроцесора, тактову частоту, об'єм пам'яті тощо) та вибір найбільш ефективного алгоритму розв'язання задачі складання розкладу приймання лікувальних процедур пацієнтами санаторію.

Критерієм ефективності є мінімізація часу виконання розрахунків з пошуку найбільшого паросполучення у дводольному графі зі зникаючими дугами.

Експеримент проводився на базі санаторію "Деніш" з використанням авторської програми ICS_DENISH, яка може розв'язувати задачі складання розкладу приймання процедур пацієнтами санаторію методом повного перебору й оптимальним алгоритмом вирішення задачі пошуку максимального паросполучення у дводольному графі зі зникаючими дугами.

Порівняльний обчислювальний експеримент проведено на серії випадкових умов задачі, отриманих від реальних пацієнтів санаторію за часовою вибіркою. Характеристики комп'ютерів, що були використані при проведенні експерименту, наведено в табл. 1.

Таблиця 1. Характеристики комп'ютерів, використаних при проведенні порівняльного обчислювального експерименту

Номер конфігурації	Назва мікропроцесора	Кількість ядер	Тактова частота, ГГц	Об'єм оперативної пам'яті
1	Pentium 4	1	2,42	512 Мб
2	Intel Celeron E3300	2	2,50	2 Гб
3	Intel Core i5-3570K	4	3,40	4 Гб

Вхідні параметри експерименту (табл. 2) були детермінованими, тобто залежними від спостерігача, і випадковими, тобто реєстрованими, але некерованими.

Відомі обмеження прийому лікувальних процедур (див. табл. 2) задаються множиною наслідків C, таких, що $(x_i, y_j) \to C_{i,j} = \{(x_{i_1}, y_{j_1}), ..., (x_{i_k}, y_{j_k})\}$. Ці обмеження враховують, насампе-

Таблиця 2. Перелік вхідних параметрів обчислювального експерименту

Номер параметра	Позначення	Назва параметра	Тип параметра
1	$k_{ЛП}$	Кількість лікувальних процедур	Детермінований
2	$k_{П}$	Кількість пацієнтів	Випадковий
3	$k_{ПП}$	Кількість призначених процедур	Випадковий
4	$C_{i,j}$	Обмеження прийому процедур	Детермінований

Таблиця 3. Результати окремих випробувань обчислювального експерименту

Номер випробування	Вхідні параметри			Вихідний параметр ($t_{ВР}$, с)					
				Конфігурація 1		Конфігурація 2		Конфігурація 3	
	$k_{ЛП}$	$k_{П}$	$k_{ПП}$	$t_{МП1}$	$t_{ОА1}$	$t_{МП2}$	$t_{ОА2}$	$t_{МП3}$	$t_{ОА3}$
1	74	1584	2974	511,24	57,63	365,17	41,16	170,41	19,21
2	74	1312	3217	612,75	96,17	437,68	68,69	204,25	32,06
3	80	1723	3343	712	118,8	508,57	84,86	237,33	39,60
4	80	1784	3578	811,2	141	579,43	100,71	270,40	47,00
5	85	1837	3724	909,2	152,3	649,43	108,79	303,07	50,77
...									
86	129	8719	51622	2634,68	587,43	1881,91	419,59	878,23	195,81

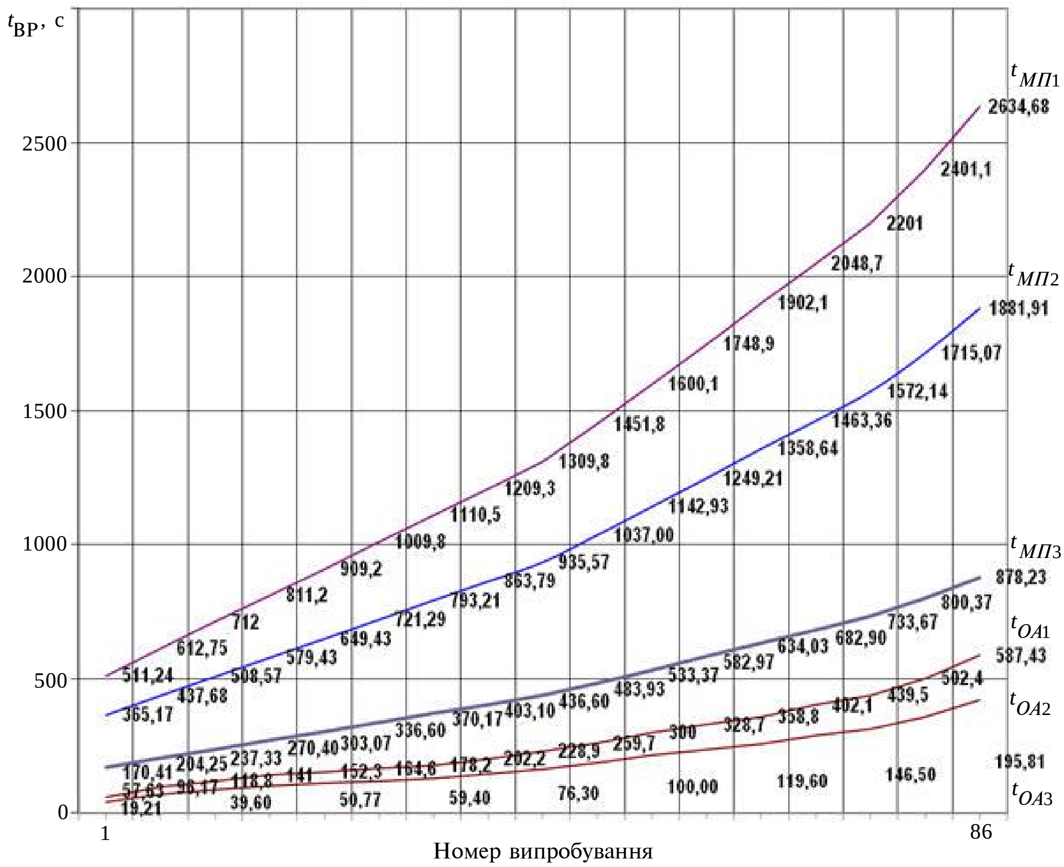

Рис. 3. Графіки залежності часу від розмірності вхідних даних (згідно з номером випробування) для порівнюваних методів

ред, неможливість призначення тієї самої процедури різним пацієнтам на однаковий час, а також сумісність процедур.

Вихідним параметром експерименту є час $t_{\text{ВР}}$, що витрачається на виконання розрахунків зі складання розкладу приймання лікувальних процедур пацієнтами санаторію за допомогою програми ICS_DENISH.

Точність експериментальних даних істотно залежить від кількості проведених випробувань: чим їх більше, тим вища достовірність результатів [8]. При невеликій кількості факторів достатньо провести 50–100 випробувань на кожній обчислювальній конфігурації (див. табл. 1).

У кожному з 86-ти проведених випробувань (табл. 3) проводився вимір часу на виконання розрахунків: $t_{\text{МП}}$ – методом повного перебору; $t_{\text{ОА}}$ – оптимальним алгоритмом розв'язання задачі зі зникаючими дугами.

Отримані результати обчислювального експерименту (рис. 3) свідчать, що час складання розкладу прямо пропорційно залежить від кількості вершин дводольного графа (тобто від сумарної кількості процедур і пацієнтів), кількості призначених процедур та обмежень $C_{i,j}$; для зібраних вхідних даних час складання розкладу з використанням методу повного перебору від 8,87 до 4,48 разу більший від часу розв'язання цієї ж задачі оптимальним алгоритмом.

Висновки

Як показали попередні дослідження, задача складання оптимального розкладу може бути зведена до розширеної задачі пошуку максимального паросполучення у дводольному графі. Але відомі алгоритми, у класичному випадку, використовують для знаходження оптимального розв'язку метод повного перебору.

Оскільки клас задач, що розглядається, відноситься до класу NP-повних, найбільш доцільною схемою зменшення обчислювальної складності знаходження їх точного розв'язку залишається скорочення повного перебору.

Запропонований алгоритм, на відміну від відомих, дає можливість врахувати задані обмеження та має порівняно з алгоритмом [1] (заснованим на методі повного перебору) меншу обчислювальну складність за рахунок скорочення кількості паросполучень, що аналізуються.

У результаті проведеного обчислювального експерименту встановлено, що запропонований оптимальний алгоритм забезпечує зменшення часу складання розкладу від 4,48 до 8,87 разу порівняно з методом повного перебору.

Перспективним напрямом подальших досліджень є розв'язання задачі складання оптимального розкладу іншими відомими методами (а саме, їх модифікація з метою врахування заданих обмежень) та порівняння із запропонованим методом.

1. *Данильченко О.М., Данильченко А.О., Ібрагім С.А.* Розв'язання одного класу задач складання розкладів генетичними алгоритмами на кластерних системах // Вісник ЖІТІ, 2004. – № 4. – С. 130–135.
2. *Панішев А.В., Данильченко А.М., Данильченко А.А.* Задача про паросполучення зі "зникаючими" дугами" // Моделювання та інформ. технол.: Зб. наук. пр. – К., 2012 – С. 75–81.
3. *Рыбников К.А.* Введение в комбинаторный анализ. – М.: Изд-во МГУ, 1985. – 312 с.
4. *Оре О.* Теория графов. – М.: Наука, Гл. ред. физ.-мат. лит., 1980. – 336 с.
5. *Майника Э.* Алгоритмы оптимизации в сетях и графах: Пер. с англ. – М.: Мир, 1981. – 324 с.
6. *Харари Ф.* Теория графов / Пер. с англ. и предисл. В.П. Козырева; под ред. Г.П. Гаврилова. – 2-е изд. – М.: Едиториал УРСС, 2003. – 296 с.
7. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985. – 512 с.
8. *Данильченко О.М., Данильченко А.О.* Інтелектуальний аналіз даних (Data Mining): Навч. посібник. – Житомир: ЖДТУ, 2009. – 406 с.

Рекомендована Радою
факультету прикладної математики
НТУУ "КПІ"

Надійшла до редакції
9 жовтня 2012 року