

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, СИСТЕМНИЙ АНАЛІЗ ТА КЕРУВАННЯ

УДК 519.161

DOI: 10.20535/1810-0546.2017.2.96638

А.О. Данильченко, С.М. Кравченко*

Житомирський державний технологічний університет, Житомир, Україна

МОДИФІКАЦІЯ МУРАШИНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРО ПАРОСПОЛУЧЕННЯ ЗІ ЗНИКАЮЧИМИ ДУГАМИ

Background. The applied problem of making out the optimal schedule of reception of manipulation treatments can be reduced to the extended mathematical problem of search of maximal matchings in the dicotyledonous graph. The basic challenge of solving this problem is the necessity of taking into account the limits on the acceptance of procedures.

Objective. The aim of the paper is modification of the ant colony algorithm for solution of the problem of matching with the vanishing arcs.

Methods. The initial population forming method and modified method of analysis of way of ants are proposed.

Results. The carried out studies proved the possibility of receipt of feasible optimal solution of the problem of matching with the vanishing arcs at the use of the modified ant colony algorithm.

Conclusions. The proposed method can be used for development and application of the scheduling systems and operative management in a direct care as well as at the development of control systems by flexible CASS for enterprises with discrete type of production.

Keywords: matchings; dicotyledonous graph; ant colony optimization algorithm.

Вступ

Прикладна задача складання оптимального розкладу прийому лікувальних процедур пацієнтами санаторію може бути зведена до розширеної математичної задачі пошуку максимального паросполучення у дводольному графі [1, 2]. Для розв'язання цієї та подібних задач (розподіл за часом обмежених ресурсів, призначення виконання різних видів робіт тощо) широко використовують методи комбінаторної оптимізації [3]. У статті [4] запропоновано точний алгоритм, який дає змогу врахувати задані обмеження та має порівняно з оптимальним алгоритмом [1] меншу обчислювальну складність. Однак задача про паросполучення може бути розв'язана й іншими відомими методами: мурашиним, генетичним алгоритмом, методом гілок і меж тощо.

Мурашиний алгоритм [5, 6] – один із ефективних поліноміальних алгоритмів для знаходження наближених розв'язків задачі комівояже-ра [7], а також аналогічних задач пошуку маршрутів на графах. Суть підходу полягає в аналізі та використанні моделі поведінки мурах, які шукають шляхи від колонії до джерела живлення.

У праці [8] автори розглядають складність і наближені результати розв'язку задачі знаходження максимального паросполучення в модифікованій задачі про паросполучення і показують, що така задача є NP-повною.

У багатьох публікаціях пропонуються алгоритми розв'язання класичної задачі про паросполучення, які б дали змогу скоротити обчислювальну складність програмних реалізацій [9, 10].

Розв'язок задачі знаходження максимального паросполучення у дводольному графі розглядається у багатьох джерелах. Але постановка задачі про паросполучення зі зникаючим дугами не наводиться ніде. Відповідно, модифікація відомих алгоритмів для її розв'язання також не розглядалася дослідниками.

Постановка задачі

Мета роботи – модифікувати мурашиний алгоритм і адаптувати його до розв'язання математичної задачі про паросполучення зі зникаючими дугами для складання оптимального розкладу прийому лікувальних процедур пацієнтами санаторію.

Основний матеріал

Розклад призначених пацієнтам процедур заданий дводольним графом $G = (X, Y, E)$, для якого відомі відношення наслідку:

$$(x_i, y_j) \rightarrow C_{i,j} = \{(x_{i_1}, y_{j_1}), \dots, (x_{i_k}, y_{j_k})\}.$$

Це означає, що дуга (x_i, y_j) не може бути включена в паросполучення в парі з іншою дугою (x_{i_p}, y_{j_q}) для будь якого $1 \leq p \leq k$, $1 \leq q \leq k$, де $k = |C_{i,j}|$ – потужність множини $C_{i,j}$. Однак усі

* corresponding author: kravchenko-sveta2012@yandex.ua

дуги з множини $C_{i,j}$ можуть бути не зв'язані між собою такими відношеннями, якщо це не обумовлено окремо. Тоді з відношення наслідку передування

$$(x_i, y_j) \rightarrow \{(x_{i_1}, y_{j_1}), \dots, (x_{i_k}, y_{j_k})\}$$

слідують k відношень:

$$(x_i, y_j) \rightarrow (x_{i_1}, y_{j_1});$$

$$(x_i, y_j) \rightarrow (x_{i_2}, y_{j_2});$$

.....

$$(x_i, y_j) \rightarrow (x_{i_k}, y_{j_k}).$$

Мураха в цьому алгоритмі виступає в ролі програмного агента, який має набір визначених простих правил поведінки при пошуку шляху на графі. До цих правил належить насамперед заборона повторного відвідування вершин графа. Всі ребра початкового дводольного графа для мурахи мають однакову довжину (вагу).

Крім того, мураха повинна пам'ятати пройдений шлях з урахуванням порядку відвідування вершин графа. Цей шлях, реалізований зазвичай у вигляді списку, використовують на наступних етапах алгоритму.

Початкова популяція

Існує досить багато варіантів формування та розміщення початкової популяції мурах по вершинах графа.

У простому мурашиному алгоритмі прийнято вважати, що в кожній вершині спочатку перебуває по одній мураці. В [1] запропоновано деякі інші варіанти початкового розміщення мурах:

- “дробовик” – випадковий розподіл мурах по вершинах графа, причому необов'язково, щоб чисельності колонії та вершин збігалися;
- “фокусування” – вся колонія знаходиться в одній вершині;
- “блукаюча колонія” – у кожний момент часу, тобто на кожній ітерації, вся колонія переміщується у випадково вибрану вершину.

Однак більш прийнятним є рівний розподіл мурах між усіма вершинами. У цьому випадку вершини графа мають однакові шанси стати початковою точкою оптимального маршруту, що, з урахуванням накладених на нашу задачу обмежень щодо сумісності процедур ($C_{i,j}$), є найбільш вагомим фактором.

Виходячи з цього, розмір популяції мурах (PM) має бути кратним $\|Y\|$ – потужності мно-

жини вершин графа, що відповідають призначеним пацієнтам процедурам

$$PM = D \cdot \|Y\|,$$

де D – коефіцієнт пропорційності ($D = 1, 2, 3 \dots$).

Таким чином, у кожній вершині графа, що відповідає призначеній пацієнту процедурі, початково має знаходитися по одній мураці або по дві мурахи в кожній і т.д.

Рух мурахи

Якщо мураха не досягла кінця шляху, для визначення наступного напрямку руху використовується рівняння

$$P_j = \frac{l_j^g \cdot f_j^s}{\sum_{n=1}^N l_n^g \cdot f_n^s}, \quad (1)$$

де P_j – імовірність переходу по шляху j ; l_j – величина, обернена вазі (довжині) j -го переходу; f_j – кількість ферменту на j -му переході; g – величина “жадібності” алгоритму; s – величина “стадності” алгоритму; N – кількість можливих напрямків руху з поточної вершини.

Порядок руху мурахи визначимо таким: після переходу з поточної вершини u_i із множини Y по одному з імовірних шляхів (до вершини з множини X) мураха “телепортується” до наступної вершини u_{i+1} (доки не відвідає всі вершини множини Y). Далі відбувається модифікація графа з урахуванням сумісності пройденної дуги. Якщо доступних напрямків руху з нової вершини немає, то шлях мурахи завершується.

Аналіз шляхів мурах

Після завершення руху всіх мурах виконуємо аналіз знайдених ними шляхів. Для кожного шляху повинна обчислюватись його довжина, що дорівнює кількості пройдених ребр e_j . Причому фактично цей шлях відповідає деякому варіанту проходження процедур з урахуванням заданих обмежень $C_{i,j}$. Формально можна говорити про те, що знайдені шляхи найбільшої довжини вже є допустимим розв'язком задачі, але, враховуючи ймовірнісний характер процедури вибору напрямку руху, слід виконати всі передбачені мурашиним алгоритмом етапи.

Всі ребра, що входять до знайденого k -го шляху, повинні бути помічені деякою кількістю

ферменту. Виходячи з постановки нашої задачі, найдовший шлях має бути помічений більшою кількістю ферменту. Тоді, з урахуванням рівної довжини ребер, кількість ферменту для маркування j -ї дуги знайденого шляху може бути визначена як

$$\Delta f_j^k = F \cdot L_k, \quad (2)$$

де F – деяка фіксована порція ферменту, однакова для всіх мурах; L_k – довжина знайденого k -го шляху.

Відповідно, в результаті маркування дуг знайденого шляху кількість ферменту на кожній із них буде описуватися виразом

$$f_j = f_j + \Delta f_j^k \cdot \eta,$$

де η – коефіцієнт оновлення шляху (константа), $\eta = 0 \dots 1$.

Випаровування ферменту

Для того щоб надати можливість мурахам знайти інший імовірний шлях, застосовується процедура випаровування ферменту. Суть її полягає у поступовому видаленні ребер, що входять до гірших знайдених шляхів.

Для визначення кількості ферменту для всіх дуг використовують зворотний коефіцієнт оновлення шляху:

$$f_j = f_j \cdot (1 - \eta).$$

У відомих працях вказуються різні модифікації цього етапу алгоритму, але в цілому вони враховують саме практичні аспекти його застосування [5].

Повторний запуск й умови завершення алгоритму

Після завершення переміщення мурах, маркування знайдених шляхів ферментом і його випаровування здійснюється повторний запуск алгоритму. При цьому пам'ять мурах щодо знайдених шляхів стирається.

Таким чином, кожна з мурах знову шукає шлях, визначаючи його за виразом (1), але, на відміну від першого запуску алгоритму, деякі дуги вже по-

мічені ферментом, що спонукає до вибору привабливішого напрямку руху.

Процес повторного запуску може виконуватися до того моменту, поки не буде виявлено змін у знайдених шляхах або до знаходження деякої кількості шляхів заданої довжини тощо. Після цього визначається шлях, який і є розв'язком задачі.

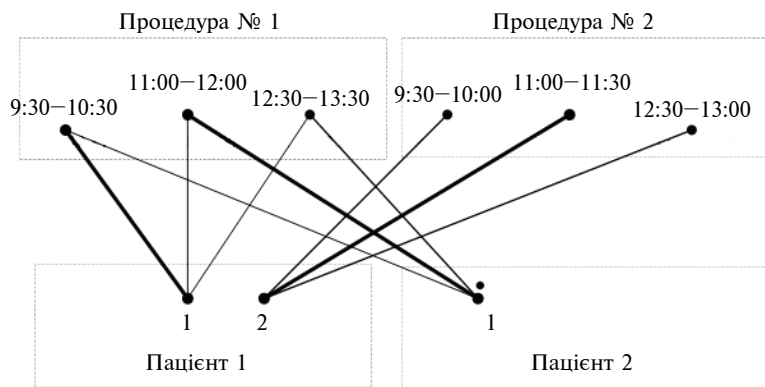
Приклад розв'язання задачі про паросполучення зі зникаючими дугами за допомогою модифікованого мурашиного алгоритму

Враховуючи ймовірнісний характер алгоритму, наведемо варіант розв'язку нашої задачі на прикладі.

Нехай потрібно побудувати розклад для 2-х пацієнтів санаторію. Першому призначено 1 і 2-гу процедуру, другому – лише 1-шу процедуру. Час роботи процедурного кабінету № 1 – 9:30–13:30, тривалість процедури – 1 год, технічна перерва – 30 хв. Час роботи процедурного кабінету № 2 – 9:30–13:00, тривалість процедури – 30 хв, технічна перерва – 1 год. (Тобто перший процедурний кабінет працює в часові інтервали 9:30–10:30, 11:00–12:00, 12:30–13:30; графік роботи другого процедурного кабінету: 9:30–10:00, 11:00–11:30, 12:30–13:00.)

Для побудованого дводольного графа $G = (X, Y, E)$ маємо: $X = \{9:30-10:30; 11:00-12:00; 12:30-13:30; 9:30-10:00; 11:00-11:30; 12:30-13:00\}$, $\|X\| = 6$; $Y = \{1; 2\}$, $\|Y\| = 3$.

На рисунку показано побудову відповідного дводольного графа для наведеного прикладу. Світлі ребра позначають можливі часові проміжки для призначення кожної процедури пацієнта. Темні ребра відповідають максимальному паросполученню в графі і позначають часові проміжки, на які призначена процедура відповідно до розкладу.



Побудова дводольного графа

Задані обмеження визначимо множиною слідства C , такою, що

$$\begin{aligned}
 (x_i, y_j) &\rightarrow C_{i,j} = \{(x_{i1}, y_{j1}), \dots, (x_{ik}, y_{jk})\}; \\
 (x_1, y_1) &\rightarrow C_{1,1} = \\
 &= \{(x_1, y_3), (x_2, y_1), (x_3, y_1), (x_4, y_2), (x_5, y_2)\}; \\
 (x_2, y_1) &\rightarrow C_{2,1} = \\
 &= \{(x_2, y_3), (x_1, y_1), (x_3, y_1), (x_5, y_2), (x_6, y_2)\}; \\
 (x_3, y_1) &\rightarrow C_{3,1} = \\
 &= \{(x_3, y_3), (x_1, y_1), (x_2, y_1), (x_6, y_2)\}; \\
 (x_4, y_2) &\rightarrow C_{4,2} = \\
 &= \{(x_5, y_2), (x_6, y_2), (x_1, y_1)\}; \\
 (x_5, y_2) &\rightarrow C_{5,2} = \\
 &= \{(x_4, y_2), (x_6, y_2), (x_1, y_1), (x_2, y_1)\}; \\
 (x_6, y_2) &\rightarrow C_{6,2} = \\
 &= \{(x_4, y_2), (x_5, y_2), (x_2, y_1), (x_3, y_1)\}; \\
 (x_1, y_3) &\rightarrow C_{1,3} = \{(x_2, y_3), (x_3, y_3), (x_1, y_1)\}; \\
 (x_2, y_3) &\rightarrow C_{2,3} = \{(x_1, y_3), (x_3, y_3), (x_2, y_1)\}; \\
 (x_3, y_3) &\rightarrow C_{3,3} = \{(x_1, y_3), (x_2, y_3), (x_3, y_1)\}.
 \end{aligned} \tag{3}$$

Відношення наслідку щодо сумісності процедур задані виразами (3).

Задамо константи. Покладемо $D = 1$; $g = 3$; $s = 1$; $F = 1$; $\eta = 0,7$. Довжини всіх дуг (їх вагу) прирівняємо до одиниці. Кількість ферменту по всіх дугах дорівнює нулю.

Перший запуск. Розмістимо трьох мурах по вершинах множини Y . Оскільки ймовірність вибору будь-якої дуги для кожної з вершин однакова, ігноруємо рівняння (1).

Перша мураха знаходиться у вершині y_1 . Була вибрана дуга (x_2, y_1) . Телепортуємо першу мурашу у вершину y_2 і модифікуємо граф. Із вершини y_2 є всього один можливий напрямок руху – дуга (x_4, y_2) . Телепортуємо першу мурашу у вершину y_3 і модифікуємо граф. Із вершини y_3 є всього один можливий напрямок руху – дуга (x_3, y_3) . Усі вершини множини Y відвідані; кінець шляху.

Друга мураха знаходиться у вершині y_2 . Була вибрана дуга (x_4, y_2) . Телепортуємо мурашу у вершину y_3 і модифікуємо граф. Із вершини y_3 є три можливих напрямки руху. Вибрана дуга (x_2, y_3) . Телепортуємо мурашу у вершину y_1 і модифікуємо граф. Із вершини y_1 залишився один мож-

ливий напрямок руху – дуга (x_3, y_1) . Усі вершини множини Y відвідані; кінець шляху.

Третя мураха знаходиться у вершині y_3 . Була вибрана дуга (x_1, y_3) . Телепортуємо мурашу у вершину y_1 і модифікуємо граф. Із вершини y_1 є два можливих напрямки руху: дуги (x_2, y_1) і (x_3, y_1) . Вибрана дуга (x_3, y_1) . Телепортуємо мурашу у вершину y_2 і модифікуємо граф. Із вершини y_2 є два можливих напрямки руху: дуги (x_4, y_2) і (x_5, y_2) . Вибрана дуга (x_5, y_2) . Усі вершини множини Y відвідані; кінець шляху.

Усі знайдені шляхи мають довжину 3. Відповідно до (2) кількість ферменту, яким кожна з мурах повинна маркувати дугу свого шляху, становитиме $\Delta f_j^k = 1 \cdot 3 = 3$.

З урахуванням $\eta = 0,7$ для всіх наявних дуг отримуємо такі значення:

$$\begin{aligned}
 (x_1, y_1) &\rightarrow f_1 = 0; \\
 (x_2, y_1) &\rightarrow f_2 = 3 \cdot 0,7 = 2,1; \\
 (x_3, y_1) &\rightarrow f_3 = 3 \cdot 0,7 + 3 \cdot 0,7 = 2,1 + 2,1 = 4,2; \\
 (x_4, y_2) &\rightarrow f_4 = 3 \cdot 0,7 + 3 \cdot 0,7 = 2,1 + 2,1 = 4,2; \\
 (x_5, y_2) &\rightarrow f_5 = 3 \cdot 0,7 = 2,1; \\
 (x_6, y_2) &\rightarrow f_6 = 0; \\
 (x_1, y_3) &\rightarrow f_7 = 3 \cdot 0,7 = 2,1; \\
 (x_2, y_3) &\rightarrow f_8 = 3 \cdot 0,7 = 2,1; \\
 (x_3, y_3) &\rightarrow f_9 = 3 \cdot 0,7 = 2,1.
 \end{aligned}$$

Здійснимо випаровування ферменту. Отримаємо:

$$\begin{aligned}
 f_1 &= 0; \\
 f_2 &= 2,1 \cdot 0,3 = 0,63; \\
 f_3 &= 4,2 \cdot 0,3 = 1,26; \\
 f_4 &= 4,2 \cdot 0,3 = 1,26; \\
 f_5 &= 2,1 \cdot 0,3 = 0,63; \\
 f_6 &= 0; \\
 f_7 &= 2,1 \cdot 0,3 = 0,63; \\
 f_8 &= 2,1 \cdot 0,3 = 0,63; \\
 f_9 &= 2,1 \cdot 0,3 = 0,63.
 \end{aligned}$$

1-й повторний запуск. Розмістимо трьох мурах по вершинах множини Y і обчислимо ймовірність вибору дуг для кожної з вершин за виразом (1).

Перша мураха знаходиться у вершині y_1 .

Імовірності переходу по напрямках становлять:

$$\begin{aligned}(x_1, y_1) &\rightarrow P_1 = \\ &= \frac{1^3 \cdot 0^1}{1^3 \cdot 0^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 1,26^1} = \frac{0}{1,89} = 0; \\ (x_2, y_1) &\rightarrow P_2 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 1,26^1} = \frac{0,63}{1,89} \approx 0,33; \\ (x_3, y_1) &\rightarrow P_3 = \\ &= \frac{1^3 \cdot 1,26^1}{1^3 \cdot 0^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 1,26^1} = \frac{1,26}{1,89} \approx 0,67.\end{aligned}$$

Була вибрана дуга (x_3, y_1) . Телепортуємо першу мурашу у вершину y_2 і модифікуємо граф. Із вершини y_2 є два можливих напрямки руху: дуги (x_4, y_2) і (x_5, y_2) .

Імовірності переходу по напрямках становлять:

$$\begin{aligned}(x_4, y_2) &\rightarrow P_4 = \\ &= \frac{1^3 \cdot 1,26^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1} = \frac{1,26}{1,89} \approx 0,67; \\ (x_5, y_2) &\rightarrow P_5 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33.\end{aligned}$$

Вибрана дуга (x_4, y_2) . Телепортуємо першу мурашу у вершину y_3 і модифікуємо граф. Із вершини y_3 є два можливих напрямки руху: дуги (x_1, y_3) і (x_2, y_3) .

Імовірності переходу по напрямках становлять:

$$\begin{aligned}(x_1, y_3) &\rightarrow P_7 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,26} = 0,5; \\ (x_2, y_3) &\rightarrow P_8 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,26} = 0,5.\end{aligned}$$

Вибрана дуга (x_1, y_3) . Усі вершини множини Y відвідані; кінець шляху.

Друга мураха знаходиться у вершині y_2 .

Імовірності переходу по напрямках становлять:

$$\begin{aligned}(x_4, y_2) &\rightarrow P_4 = \\ &= \frac{1^3 \cdot 1,26^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0^1} = \frac{1,26}{1,89} \approx 0,67; \\ (x_5, y_2) &\rightarrow P_5 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0^1} = \frac{0,63}{1,89} \approx 0,33; \\ (x_6, y_2) &\rightarrow P_6 = \\ &= \frac{1^3 \cdot 0^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0^1} = \frac{0}{1,89} = 0.\end{aligned}$$

Тепер вибрана дуга (x_4, y_2) . Телепортуємо мурашу у вершину y_3 і модифікуємо граф. Із вершини y_3 є три можливих напрямки руху з імовірностями переходу:

$$\begin{aligned}(x_1, y_3) &\rightarrow P_7 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33; \\ (x_2, y_3) &\rightarrow P_8 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33; \\ (x_3, y_3) &\rightarrow P_9 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33.\end{aligned}$$

Вибрана дуга (x_1, y_3) . Телепортуємо мурашу у вершину y_1 і модифікуємо граф. Із вершини y_1 є два можливих напрямки руху: дуги (x_2, y_1) і (x_3, y_1) .

Імовірності переходу по напрямках становлять:

$$\begin{aligned}(x_2, y_1) &\rightarrow P_2 = \\ &= \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 1,26^1} = \frac{0,63}{1,89} \approx 0,33; \\ (x_3, y_1) &\rightarrow P_3 = \\ &= \frac{1^3 \cdot 1,26^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 1,26^1} = \frac{1,26}{1,89} \approx 0,67.\end{aligned}$$

Вибрана дуга (x_3, y_1) . Усі вершини множини Y відвідані; кінець шляху.

Третя мураха знаходиться у вершині y_3 .

Імовірності переходу по напрямках становлять:

$$(x_1, y_3) \rightarrow P_7 = \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33;$$

$$(x_2, y_3) \rightarrow P_8 = \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33;$$

$$(x_3, y_3) \rightarrow P_9 = \frac{1^3 \cdot 0,63^1}{1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33.$$

Була вибрана дуга (x_2, y_3) . Телепортуємо мураха у вершину y_1 і модифікуємо граф. Із вершини y_1 є два можливих напрямки руху: дуги (x_1, y_1) і (x_3, y_1) з імовірностями переходу:

$$(x_1, y_1) \rightarrow P_1 = \frac{1^3 \cdot 0^1}{1^3 \cdot 0^1 + 1^3 \cdot 1,26^1} = \frac{0}{1,26} = 0;$$

$$(x_3, y_1) \rightarrow P_3 = \frac{1^3 \cdot 1,26^1}{1^3 \cdot 0^1 + 1^3 \cdot 1,26^1} = \frac{1,26}{1,26} = 1.$$

Вибрана дуга (x_3, y_1) . Телепортуємо мураха у вершину y_2 і модифікуємо граф. Із вершини y_2 є два можливих напрямки руху: дуги (x_4, y_2) і (x_5, y_2) .

Імовірності переходу по напрямках становлять:

$$(x_4, y_2) \rightarrow P_4 = \frac{1^3 \cdot 1,26^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1} = \frac{1,26}{1,89} \approx 0,67;$$

$$(x_5, y_2) \rightarrow P_5 = \frac{1^3 \cdot 0,63^1}{1^3 \cdot 1,26^1 + 1^3 \cdot 0,63^1} = \frac{0,63}{1,89} \approx 0,33.$$

Вибрана дуга (x_4, y_2) . Усі вершини множини Y відвідані; кінець шляху.

Проаналізуємо отримані шляхи. Всі вони мають довжину 3. Кількість ферменту для маркування становить $\Delta f_j^k = 1 \cdot 3 = 3$.

Розподіл ферменту по дугах:

$$(x_1, y_1) \rightarrow f_1 = 0; \quad (x_2, y_1) \rightarrow f_2 = 0;$$

$$(x_3, y_1) \rightarrow f_3 = 6,3; \quad (x_4, y_2) \rightarrow f_4 = 6,3;$$

$$(x_5, y_2) \rightarrow f_5 = 0; \quad (x_6, y_2) \rightarrow f_6 = 0;$$

$$(x_1, y_3) \rightarrow f_7 = 2,1; \quad (x_2, y_3) \rightarrow f_8 = 4,2;$$

$$(x_3, y_3) \rightarrow f_9 = 0.$$

Здійснимо випаровування ферменту. Отримаємо:

$$f_1 = 0; \quad f_2 = 0; \quad f_3 = 1,89; \quad f_4 = 1,89; \quad f_5 = 0;$$

$$f_6 = 0; \quad f_7 = 0,63; \quad f_8 = 1,26; \quad f_9 = 0.$$

2-й повторний запуск. Розмістимо трьох мурах по вершинах множини Y і обчислимо ймовірність вибору дуг для кожної з вершин за виразом (1).

Перша мураха знаходиться у вершині y_1 .

Імовірності переходу по напрямках становлять:

$$(x_1, y_1) \rightarrow P_1 = 0;$$

$$(x_3, y_1) \rightarrow P_3 = 1.$$

Була вибрана дуга (x_3, y_1) . Телепортуємо першу мураха у вершину y_2 і модифікуємо граф. Із вершини y_2 є два можливих напрямки руху: дуги (x_4, y_2) і (x_5, y_2) . Імовірності переходу по напрямках становлять:

$$(x_4, y_2) \rightarrow P_4 = 1;$$

$$(x_5, y_2) \rightarrow P_5 = 0.$$

Вибрана дуга (x_4, y_2) . Телепортуємо першу мураха у вершину y_3 і модифікуємо граф. Із вершини y_3 є два можливих напрямки руху: дуги (x_1, y_3) і (x_2, y_3) . Імовірності переходу по напрямках становлять:

$$(x_1, y_3) \rightarrow P_7 \approx 0,33;$$

$$(x_2, y_3) \rightarrow P_8 \approx 0,67.$$

Вибрана дуга (x_2, y_3) . Усі вершини множини Y відвідані; кінець шляху.

Друга мураха знаходиться у вершині y_2 .

Імовірності переходу по напрямках становлять:

$$(x_4, y_2) \rightarrow P_4 = 1;$$

$$(x_5, y_2) \rightarrow P_5 = 0;$$

$$(x_6, y_2) \rightarrow P_6 = 0.$$

Була вибрана дуга (x_4, y_2) . Телепортуємо мураха у вершину y_3 і модифікуємо граф. Із вершини y_3 є три можливих напрямки руху з імовірностями переходу:

$$(x_1, y_3) \rightarrow P_7 \approx 0,33;$$

$$(x_2, y_3) \rightarrow P_8 \approx 0,67;$$

$$(x_3, y_3) \rightarrow P_9 = 0.$$

(4)

Вибрана дуга (x_2, y_3) . Телепортуємо мурашу у вершину y_1 і модифікуємо граф. Із вершини y_1 залишився один можливий напрямок руху: дуга (x_3, y_1) з імовірністю переходу $(x_3, y_1) \rightarrow P_3 = 1$.

Вибрана дуга (x_3, y_1) . Усі вершини множини Y відвідані; кінець шляху.

Третя мураха знаходиться у вершині y_3 .

Імовірності переходу по напрямках відповідають значенням (4).

Була выбрана дуга (x_2, y_3) . Телепортуємо мурашу у вершину y_1 і модифікуємо граф. Із вершини y_1 є два можливих напрямки руху: дуги (x_1, y_1) і (x_3, y_1) з імовірностями переходу:

$$(x_1, y_1) \rightarrow P_1 = 0;$$

$$(x_3, y_1) \rightarrow P_3 = 1.$$

Вибрана дуга (x_3, y_1) . Телепортуємо мурашу у вершину y_2 і модифікуємо граф. Із вершини y_2 є два можливих напрямки руху: дуги (x_4, y_2) і (x_5, y_2) .

Імовірності переходу по напрямках становлять:

$$(x_4, y_2) \rightarrow P_4 = 1;$$

$$(x_5, y_2) \rightarrow P_5 = 0.$$

Вибрана дуга (x_4, y_2) . Усі вершини множини Y відвідані; кінець шляху.

Проаналізуємо отримані шляхи. Всі вони збігаються і мають довжину 3. Кількість ферменту для маркування становить $\Delta f_j^k = 1 \cdot 3 = 3$.

Розподіл ферменту по дугах:

$$(x_1, y_1) \rightarrow f_1 = 0; (x_2, y_1) \rightarrow f_2 = 0;$$

$$(x_3, y_1) \rightarrow f_3 = 6, 3; (x_4, y_2) \rightarrow f_4 = 6, 3;$$

$$(x_5, y_2) \rightarrow f_5 = 0; (x_6, y_2) \rightarrow f_6 = 0;$$

$$(x_1, y_3) \rightarrow f_7 = 0; (x_2, y_3) \rightarrow f_8 = 6, 3;$$

$$(x_3, y_3) \rightarrow f_9 = 0.$$

Здійснимо випаровування ферменту. Отримаємо:

$$f_1 = 0; f_2 = 0; f_3 = 1, 89; f_4 = 1, 89; f_5 = 0;$$

$$f_6 = 0; f_7 = 0; f_8 = 1, 89; f_9 = 0;$$

$$(x_4, y_2) \rightarrow P_4 = 1; (x_5, y_2) \rightarrow P_5 = 0.$$

3-й повторний запуск. Отримано такі самі шляхи. Змін не відбулося – кінець алгоритму.

Таким чином, підмножина $\{(x_3, y_1), (x_4, y_2), (x_2, y_3)\}$ є розв'язком задачі.

Знайдене паросполучення $M = \{(x_3, y_1), (x_4, y_2), (x_2, y_3)\}$ є одним із восьми допустимих при заданих обмеженнях оптимальних рішень, які були отримані методом повного перебору.

Тобто можемо говорити про коректну роботу наведеного алгоритму щодо пошуку оптимального розв'язку нашої задачі.

Висновки

Наведено модифікацію мурашиного алгоритму для розв'язання задачі про паросполучення зі зникаючими дугами. На відміну від відомих модифікацій мурашиного алгоритму, найбільшою кількістю ферменту маркується найдовший шлях, що дає змогу визначити паросполучення максимальної при заданих обмеженнях потужності. Крім того, всі ребра (дуги) графа мають однакову довжину (вагу).

Розглянуто всі етапи модифікованого мурашиного алгоритму для знаходження розв'язку задачі про паросполучення зі зникаючими дугами: формування початкової популяції, рух мурахи, аналіз знайдених шляхів, процедуру випаровування ферменту, повторний запуск та умови завершення алгоритму.

Проведені дослідження довели можливість отримання допустимого оптимального розв'язку задачі про паросполучення зі зникаючими дугами при застосуванні модифікованого мурашиного алгоритму.

Запропонований метод може бути використаний при розробці та застосуванні систем календарного планування і оперативного управління в лікувальному процесі та при розробці систем управління гнучкими автоматизованими системами для підприємств із дискретним характером виробництва.

Перспективним напрямом подальших досліджень є модифікація відомих методів (генетичного, методу гілок та меж тощо) для розв'язання задачі складання розкладу прийому процедур пацієнтами санаторію, а також порівняння часу розв'язку задачі цими методами з часом розв'язку мурашиним алгоритмом.

Список літератури

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. Данильченко О.М., Данильченко А.О. Інтелектуальний аналіз даних. – Житомир: ЖДТУ, 2009. – 405 с.
3. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. – М.: Мир, 1980. – 476 с.
4. Панішев А.В., Данильченко А.М., Данильченко А.А. Задача про паросполучення зі “зникаючими” дугами // Моделювання та інформ. технології. – 2012. – № 63. – С. 75–81.
5. Штовба С.Д. Муравьиные алгоритмы. Exponenta Pro // Математика в приложениях. – 2003. – № 4. – С. 70–75.
6. Dorigo M., Maniezzo V., Colomi A. Ant system: optimization by a colony of cooperating agents // IEEE Trans. Sys. Man Cybern. B, Cybern. – 1996. – 26, № 1. – P. 29–41.
7. Алгоритм для решения задачи коммивояжера / Д.Ж. Литтл, К. Мурти, Д. Суини, К. Кэрел // Экономика и мат. методы. – 1965. – 1, вып. 1. – С. 90–107.
8. Biro P., Manlove D.F., Mittal S. Size versus stability in the marriage problem // Theor. Comp. Sci. – 2010. – 411, № 16-18. – P. 1828–1841.
9. Local search approaches in stable matching problems / M. Gelain, M. Pini, F. Rossi et al. // Algorithms. – 2013. – 6, № 4. – P. 591–617.
10. Li W., Patrikeev E.M., Xiao D. A DNA Algorithm for the maximal matching problem // Automation and Remote Control. – 2015. – 76, № 10. – P. 1797–1802.

References

- [1] A. Danilchenko *et al.*, “Solving a class of problems scheduling genetic algorithms to cluster systems”, *Visnik ZhITI*, no. 4, pp. 130–135, 2004 (in Ukrainian).
- [2] A. Danilchenko and O. Danilchenko, *Data Mining*. Zhytomyr, Ukraine: *ZhITI*, 2009 (in Ukrainian).
- [3] E. Reinhold *et al.*, *Combinatorial Algorithms. Theory and Practice*. Moscow, SU: Mir, 1980 (in Russian).
- [4] A. Panishev *et al.*, “The problem of matching with the “disappearing” arcs”, *Modelyuvannya ta Informatsiyni Tekhnologii*, no. 63, pp. 75–81, 2012 (in Ukrainian).
- [5] C. Shtovba, “Ant algorithms. Exponenta Pro”, *Matematika v Prilozhenijah*, no. 4, pp. 70–75, 2003 (in Russian).
- [6] M. Dorigo *et al.*, “Ant system: Optimization by a colony of cooperating agents”, *IEEE Trans. Sys. Man Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, 1996. doi: 10.1109/3477.484436
- [7] J.D. Little *et al.*, “The algorithm for solving the traveling salesman problem”, *Jekonomika i Matematicheskie Metody*, vol. 1, no. 1, pp. 90–107, 1965 (in Russian).
- [8] P. Biro *et al.*, “Size versus stability in the marriage problem”, *Theor. Comp. Sci.*, vol. 411, no. 16-18, pp. 1828–1841, 2010. doi: 10.1016/j.tcs.2010.02.003
- [9] M. Gelain *et al.*, “Local search approaches in stable matching problems”, *Algorithms*, vol. 6, no. 4, pp. 591–617, 2013. doi: 10.3390/a6040591
- [10] W. Li *et al.*, “A DNA Algorithm for the maximal matching problem”, *Automation and Remote Control*, vol. 76, no. 10, pp. 1797–1802, 2015. doi: 10.1134/s0005117915100070

А.О. Данильченко, С.М. Кравченко

МОДИФІКАЦІЯ МУРАШИНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРО ПАРОСПОЛУЧЕННЯ ЗІ ЗНИКАЮЧИМИ ДУГАМИ

Проблематика. Прикладна задача складання оптимального розкладу прийому лікувальних процедур може бути зведена до розширеної математичної задачі пошуку максимального паросполучення у дводольному графі. Основною складністю розв'язання цієї задачі є необхідність урахування обмежень на приймання процедур.

Мета дослідження. Модифікація мурашиного алгоритму для розв'язання задачі про паросполучення зі зникаючими дугами.

Методика реалізації. Запропоновано спосіб формування початкової популяції та модифікований спосіб аналізу шляху мурах.

Результати дослідження. Проведені дослідження довели можливість отримати допустимий оптимальний розв'язок задачі про паросполучення зі зникаючими дугами при застосуванні модифікованого мурашиного алгоритму.

Висновки. Запропонований метод може бути використаний при розробці та застосуванні систем календарного планування і оперативного управління в лікувальному процесі, а також при розробці систем управління гнучкими автоматизованими системами для підприємств із дискретним характером виробництва.

Ключові слова: паросполучення; дводольний граф; мурашиний алгоритм.

А.А. Данильченко, С.Н. Кравченко

МОДИФИКАЦІЯ МУРАВЬИНОГО АЛГОРИТМА ДЛЯ РЕШЕННЯ ЗАДАЧИ ПАРСОЧЕТАННЯ С ІСЧЕЗАЮЧИМИ ДУГАМИ

Проблематика. Прикладная задача составления оптимального расписания приема лечебных процедур может быть сведена к расширенной математической задаче поиска максимального паросочетания в двудольном графе. Основной сложностью решения этой задачи является необходимость учета ограничений на принятие процедур.

Цель исследования. Модификация муравьиного алгоритма для решения задачи паросочетания с исчезающими дугами.

Методика реализации. Предложены способ формирования начальной популяции и модифицированный способ анализа пути муравьев.

Результаты исследования. Проведенные исследования доказали возможность получения допустимого оптимального решения задачи паросочетания с исчезающими дугами при использовании модифицированного муравьиного алгоритма.

Выводы. Предложенный метод может быть использованный при разработке и применении систем календарного планирования и оперативного управления в лечебном процессе, а также при разработке систем управления гибкими автоматизированными системами для предприятий с дискретным характером производства.

Ключевые слова: паросочетания; двудольный граф; муравьиный алгоритм.

Рекомендована Радою
Навчально-наукового комплексу
“Інститут прикладного системного
аналізу” КПІ ім. Ігоря Сікорського

Надійшла до редакції
31 січня 2017 року