

Крачков В., Ткач І.І., Кореньков В.Н. ¹

1 - Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна.

ВИРІШЕННЯ ЗАДАЧІ ОПЕРАТИВНО-КАЛЕНДАРНОГО ПЛАНУВАННЯ НА ОСНОВІ ЕВРИСТИЧНОГО АЛГОРИТМУ

Одна із проблем календарного планування – правильний розподіл технологічних операцій по наявним верстатам виробничої ділянки для нової партії деталей. Задача ефективного планування багатостадійних систем відносяться до класу NP-важких задач, для вирішення яких використовують різні евристичні алгоритми [6]. Одним із таких методів є метод мурашиної колонії. Для дослідження актуальності даного методу при вирішенні представленої задачі, було розроблено програмний продукт опис якого разом з роз'ясненням функціоналу та можливостей представлений нижче.

Мурашиний алгоритм (алгоритм оптимізації мурашиної колонії, англ. *ant colony optimization*, ACO) — один з ефективних поліноміальних алгоритмів для знаходження наближених розв'язків задачі комівояжера, а також аналогічних завдань пошуку маршрутів на графах. У основі алгоритму лежить поведінка мурашиної колонії — маркування вдалих доріг великою кількістю феромону. Робота починається з розміщення мурашок у вершинах графа, потім починається рух мурашок — напрям визначається імовірнісним методом, на підставі формули:

$$P_{ij}^k = \left\{ \frac{(\mu_{ij})^\alpha \cdot (\tau_{ij})^\beta}{\sum_{u \in N_i^k} (\mu_{iu})^\alpha \cdot (\tau_{iu})^\beta}, j \in N_i^k, 0, j \notin N_i^k \right\}$$

де:

P - вірогідність переходу із вершини i у вершину j ;

τ - визначає апостеріорну ефективність переходу із вершини i в j ;

μ - визначає апіорну ефективність переходу із вершини i в j на основі деякої евристики;

N - множина, яка визначає набір допустимих вершин для переходу k -го агента, суміжних з i -ю вершиною графа.

Опис програми:

Головне вікно програми складається з трьох областей:

Панель управління (рис. 1)

Панель відображення поточного статусу (рис. 1)

Панель результатів (рис. 1)

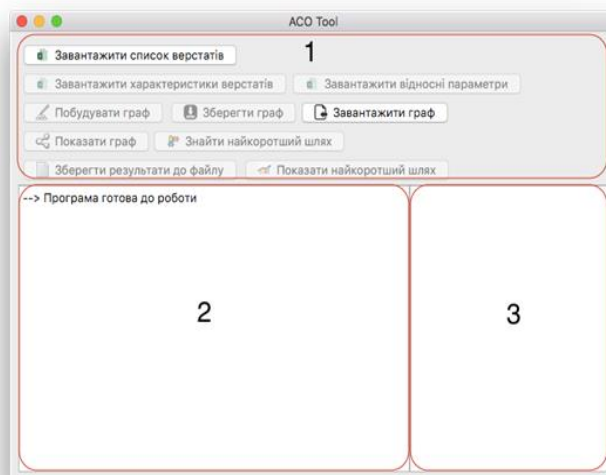


Рис. 1. Інтерфейс програми.

На панелі управління наявні наступні кнопки:

“Завантажити список верстатів” - призначена для завантаження з файлу формату CSV (Comma-separated values) списку верстатів. Приклад структури файлу можна переглянути в таблиці 1.

Таблиця 1- Приклад структури файлу зі списком верстатів.

Унікальна назва верстату
Верстат 1
...
Верстат N

“Завантажити характеристики верстатів” - призначена для завантаження власних характеристик верстатів з файлу формату CSV. Приклад структури файлу показано в таблиці 2.

Таблиця 2 - Приклад структури файлу з власними характеристиками верстатів.

Назва верстату	Потужність, (кВт)	...	Завантаженість верстату, (%)
Верстат 1	12	...	24
...	...	...	...
Верстат N	15	...	76

“Завантажити відносні параметри” - призначена для завантаження параметрів верстату які залежать від інших верстатів (наприклад відстань між верстатами). Приклад структури файлу можна побачити в таблиці 3.

Таблиця 3 - Приклад структури файлу з відносними характеристиками верстатів.

Назва першого верстату	Назва другого верстату	Відстань між верстатами, (м)
Верстат 1	Верстат 2	5
...	...	...
Верстат N	Верстат M	7

“Побудувати граф” – призначена для побудови повного направленного графу на основі інформації завантаженої з файлів.

“Зберегти граф” – дозволяє зберегти побудований граф у файл без втрати інформації про верстати.

“Завантажити граф” – призначена для завантаження графу із файлу, який був збережений за допомогою кнопки “Зберегти граф”.

“Показати граф” – у окремому вікні показує схему побудованого або завантаженого з файлу графу.

“Знайти найкоротший шлях” – знаходить найкоротший шлях на графі на основі отриманої інформації про верстати та виводить отриманий результат на панелі результатів.

“Зберегти результати до файлу” – зберігає результати розрахунку найкоротшого шляху до файлу.

“Показати найкоротший шлях” – зображає розрахований найкоротший шлях у графічному вигляді.

Приклад застосування програми

Розглянемо приклад застосування програми для розрахунку найкоротшого шляху на графі з двадцятьма верстатів. Для цього складемо список верстатів, користуючись програмним продуктом Google Sheets. Отриманий список можна побачити на рисунку 2.

	A
1	Верстат 1
2	Верстат 2
3	Верстат 3
4	Верстат 4
5	Верстат 5
6	Верстат 6
7	Верстат 7
8	Верстат 8
9	Верстат 9
10	Верстат 10
11	Верстат 11
12	Верстат 12
13	Верстат 13
14	Верстат 14
15	Верстат 15
16	Верстат 16
17	Верстат 17
18	Верстат 18
19	Верстат 19
20	Верстат 20

Рис. 2. Список верстатів.

Назва верстату	Потужність (кВт)	Завантаженість (%)
Верстат 1	7.9	30
Верстат 2	8.7	40
Верстат 3	6.5	20
Верстат 4	12.4	15
Верстат 5	15.3	60
Верстат 6	17.4	90
Верстат 7	9.6	75
Верстат 8	10.5	55
Верстат 9	11.8	43
Верстат 10	14.3	23
Верстат 11	8.9	76
Верстат 12	7.5	93
Верстат 13	13.8	73
Верстат 14	17.2	26
Верстат 15	15.0	48
Верстат 16	17.5	82
Верстат 17	18.0	75
Верстат 18	13.2	12
Верстат 19	12.5	5
Верстат 20	14.0	84

Рис. 3. Значення потужностей та завантаженості верстатів.

Отриманий файл зберігаємо на диски та завантажуюмо у програму користуючись кнопкою “Завантажити список верстатів”. Далі створюємо окремий файл для власних характеристик верстату. Для кожного верстату вкажемо його потужність та завантаженість на даний момент часу. В реальних умовах значення потужності верстату береться із паспортних даних а завантаженість отримується із систем моніторингу. В нашому випадку, обрані дані є довільними але близькими до дійсності.

Завантажуємо файл з характеристиками до програми за допомогою кнопки “Завантажити характеристики верстатів”. Після цього створюємо файл з відносними параметрами верстату. Ці параметри не залежать від конкретного верстату, але є відносними до інших верстатів. Наприклад, відстань між верстами не може бути визначена одним верстатом, а отже вона є відносним параметром. Створений файл є громіздким, тому на рисунку 4 його показано частинами.

Початок	Кінець	Відстань між верстатами (м)
Верстат 1	Верстат 2	13
Верстат 1	Верстат 3	5
...	...	...
Верстат 1	Верстат 20	23
Верстат 2	Верстат 3	15
Верстат 2	Верстат 4	16
...	...	...
Верстат 19	Верстат 20	7

Рис. 4. Відстань між верстатами

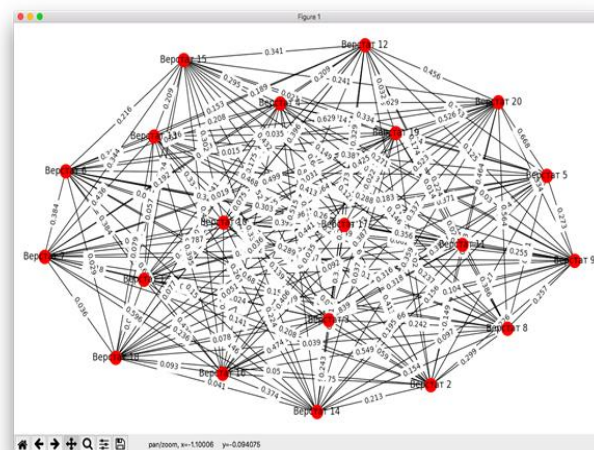


Рис. 5. Графічне зображення побудованого графу.

Після завантаження відносних характеристик до програми за допомогою кнопки “Завантажити відносні параметри”, натискаємо кнопку “Побудувати граф”. Після отримання сповіщення на панелі відображення поточного статусу про те, що граф побудований ми можемо скористатися декількома функціями програми. Для того, щоб побачити графічне представлення побудованого графу необхідно натиснути кнопку “Показати граф”. Результат цієї дії можна побачити на рисунку 5.

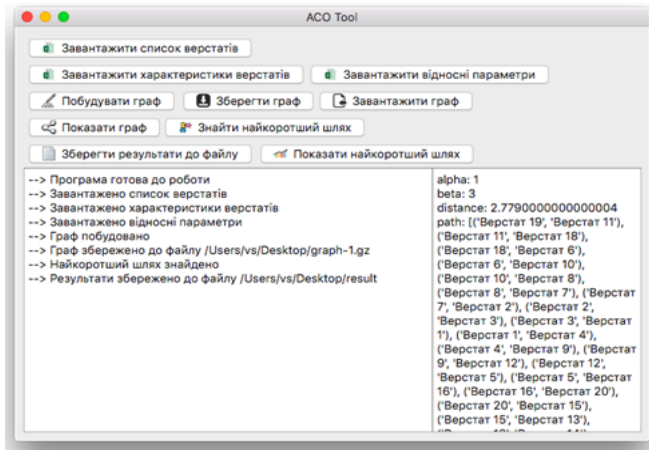


Рис. 6. Видгляд головного вікна програми після знаходження найкоротшого шляху та збереження результатів до файлу.

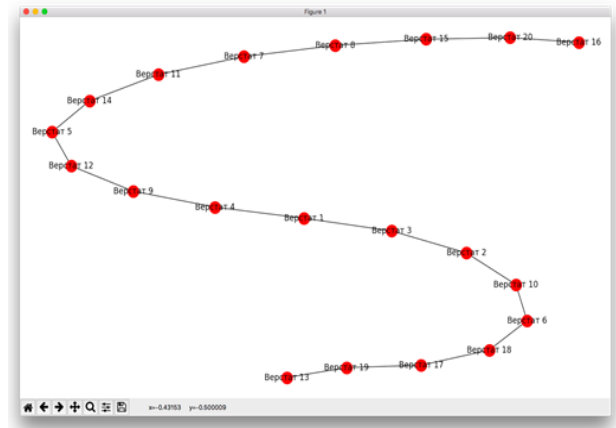


Рис. 7. Графічне зображення найкоротшого шляху.

Після натискання на кнопку “Знайти найкоротший шлях” на панелі відображення поточного статусу ми побачимо сповіщення про те, що найкоротший шлях знайдено, а на панелі результатів буде показано результати розрахунків. Результати розрахунків можна зберегти у файл за допомогою кнопки “Зберегти результати до файлу”.

Також можна переглянути графічне зображення найкоротшого шляху натиснувши на кнопку “Показати найкоротший шлях” (рис. 7).

Програма має додаткові функції збереження побудованого графу у файл та його завантаження назад у програму в майбутньому за допомогою кнопок “Зберегти граф” та “Завантажити граф”.

ВИСНОВКИ:

У статті представлено програмний продукт який втілює у собі програмну реалізацію пошуку найкоротшого шляху на графі за допомогою алгоритму мурашиної колонії. В якості вихідних даних для пошуку використовуються характеристики верстату. Представлена програма не вирішує усіх проблем календарного планування але може бути використана у якості першого кроку до автоматизації розподілу технологічних операцій на верстатах виробничої дільниці.

Список використаних джерел

1. Сачко Н.С. Организация и оперативное управление машиностроительным производством / Н.С. Сачко. – Минск: Новое знание, 2005. – 635 с.
2. Dorigo M. Swarm Intelligence, Ant Algorithms and Ant Colony Optimization // Reader for CEU Summer University Course «Complex System». – Budapest, Central European University, 2001. – P. 1–38.
3. Козаловский М.Р. Перспективные технологии информационных систем / М.Р. Козаловский — М.: ДМК Пресс; Компания АйТи, 2003. — 288 с.
4. Colnari A., Dorigo M., Maniezzo V. “Distributed optimization by ant colonies” // Proceedings of ECAL91, 1991.
5. Gutjahr W. “A graph-based ant system and its convergence” Future Generation Computer Systems 16, 2000.
6. В.Г. Секаев, П.В. Мартенин, “Использование метода колонии муравьев для решения задач календарного планирования” СБОРНИК НАУЧНЫХ ТРУДОВ НГТУ. – 2011. – № 4(66) – 109–118