

КОМП'ЮТЕРНІ НАУКИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.421.2:519.681

DOI 10.31319/2519-2884.36.2020.17

СИЗЬКО В.А., студент
БАБЕНКО М.В., к.т.н., доцент
ЛИМАР Н.М., зав. лабораторії
ГОСАЛО І.О., студент

Дніпровський державний технічний університет, м. Кам'янське

РОЗРОБКА ПАРАЛЕЛЬНОГО МУРАШИНОГО АЛГОРИТМУ НА ПРИКЛАДІ ЗАДАЧІ КОМІВОЯЖЕРА

Вступ. Задача комівояжера є однією з найвідоміших задач комбінаторної оптимізації, метою якої є пошук найвигіднішого шляху між містами, з кінцевим поверненням у початкове місто, причому, зазвичай вказується, що кожне місто повинно бути пройдено лише один раз, тоді вибір відбувається серед гамільтонових циклів. В умовах задачі зазначаються матриці відстаней, критерії вигідності маршруту, матриці вартостей і тому подібне [1]. Одним з ефективних поліноміальних алгоритмів для знаходження розв'язку задачі комівояжера є використання мурашиного алгоритму. Суть алгоритму ґрунтується на принципах поведінки мурашиної колонії та мурах, що шукають найкоротший шлях до їжі [2, 3].

Первісна ідея прийшла зі спостереження за використанням харчових ресурсів серед мурах, де мурахи, окремо обмежені в своїх пізнавальних можливостях, колективно

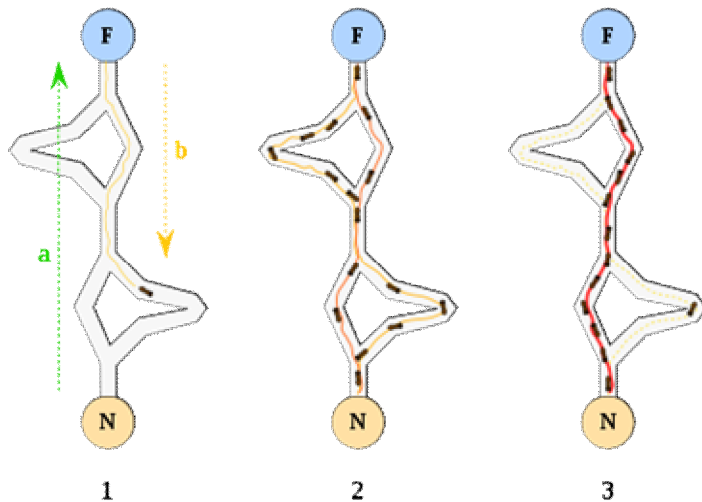


Рисунок 1 – Схематичне представлення мурашиного алгоритму

здатні знайти найкоротший шлях між джерелом їжі і гніздом (рис.1).

1. Перша мураха знаходить джерело їжі (F) через якийсь шлях (a), тоді повертається до гнізда (N), залишивши позаду слід з феромонів (b).

2. Мурахи без розбору обирають всі чотири шляхи, але підсилення основної стежки робить її привабливішою як найкоротший шлях.

3. Мурахи обирають коротший шлях, довгі відтинки втрачають щільність феромонного сліду.

В серії дослідів на колонії мурах з вибором між двома

шляхами різної довжини, які ведуть до джерела їжі, біологи спостерігали, що мурахи тяжіють до використання найкоротшого шляху.

Постановка задачі. Метою даної роботи є розробка програмного забезпечення для дослідження ефективності паралельного мурашиного алгоритму і швидкості роботи алгоритму для пошуку найкоротшої відстані маршруту.

Дослідження ефективності стохастичних алгоритмів, що працюють одночасно з великою кількістю поточних рішень на завданнях оптимізації, є актуальною науковою

проблемою сучасності. Для реалізації і дослідження ефективності мурашиного алгоритму використано оптимізаційну задачу, яка часто виникає на практиці, а саме задача комівояжера.

Припустимо, що навколишнє середовище для мурах представляє повнозв'язний неорієнтований граф. Кожне ребро має вагу, яка позначається як відстань між двома вершинами, що ним з'єднується. Граф є двоскерованим, тому мураха може подорожувати по грані в будь-якому напрямку.

Ймовірність включення ребра в маршрут окремої мурахи пропорційна до кількості феромонів на цьому ребрі, а кількість відкладеного феромону пропорційне до довжини маршруту. Чим коротший маршрут, тим більше феромону буде відкладено на його ребрах, отже, більша кількість мурах буде включати його в синтез власних маршрутів. Моделювання такого підходу, що використовує тільки додатній зворотний зв'язок, призводить до передчасної збіжності – більшість мурашок рухається по локально-оптимальному маршруту.

Уникнути цього можна, моделюючи від'ємний зворотний зв'язок у вигляді випаровування феромону. Причому, якщо феромон випаровується швидко, то це призводить до втрати пам'яті колонії і забування хороших рішень, з іншого боку, збільшення часу випарів може призвести до отримання стійкого локального оптимального рішення.

Пройдений мурахою шлях відображається, коли мураха відвідає всі вузли графа. Цикли заборонено, оскільки в алгоритм включено список табу. Після завершення довжина шляху може бути підрахована – вона дорівнює сумі довжин всіх ребер, якими подорожувала мураха.

Мураха рухатиметься від вузла i до вузла j з ймовірністю

$$P_{i,j} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum (\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)},$$

де $\tau_{i,j}$ – кількість феромонів на ребрі i, j ; α – параметр, що контролює вплив $\tau_{i,j}$; $\eta_{i,j}$ – при-вабливість ребра i, j (початкове значення, як правило, $1/d_{i,j}$, де d – відстань); β – параметр, що контролює вплив $\eta_{i,j}$.

Оновлення феромонів відбувається за формулою

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j},$$

де ρ – швидкість випаровування феромонів; $\Delta\tau_{i,j}$ – кількість відкладеного феромону, як правило розраховується як:

$$\Delta\tau_{i,j}^k = \begin{cases} 1/L_k, & \text{якщо мураха пройшла ребром } i, j \\ 0, & \text{в інших випадках} \end{cases},$$

де L_k – довжина k -го шляху мурахи.

Алгоритм показує свою працездатність і хорошу надійність при достатній кількості ресурсів. Але необхідна кількість ресурсів та, відповідно, і час роботи швидко зростають зі збільшенням розмірності задачі. Для вирішення цієї проблеми використовується розпаралелювання алгоритмів. Також воно застосовується для зниження передчасної збіжності до локального оптимуму, стимуляції різноманітності і пошуку альтернативних рішень тієї ж проблеми.

Існує кілька видів розпаралелювання еволюційних алгоритмів:

- глобальні однопопуляційні, коли є одна головна популяція, на основному ядрі, а обчислення цільової функції розподілено на кілька процесорів;
- однопопуляційні, коли є одна просторово-розподілена популяція, з деякими обмеженнями;
- багатопопуляційні, коли кілька незалежних популяцій, які періодично обмінюються індивідами, тобто здійснюється міграція, керована кількома параметрами.

При реалізації паралельного алгоритму обрано саме третій тип розпаралелювання, так як він є найбільш популярним і цікавим для вивчення. До того ж обчислення придатності в задачі комівояжера не вимагає значної кількості обчислювальних ресурсів.

Результати роботи. У розробленому авторами програмному забезпеченні на кожному ядрі комп'ютера розвивається власна незалежна колонія мурашок, яка обмінюється з іншими кращими індивідами через певну кількість поколінь. Топологія їх взаємодії має вигляд «кожен з кожним».

За допомогою програми паралельного мурашиного алгоритму на тестовій задачі (решітка 5 на 5 міст) досліджено його надійність при різних настройках параметрів і при різній кількості ядер. Головне вікно програми на початку розрахунків представлено на рис.1, а в кінці розрахунку – на рис.2.

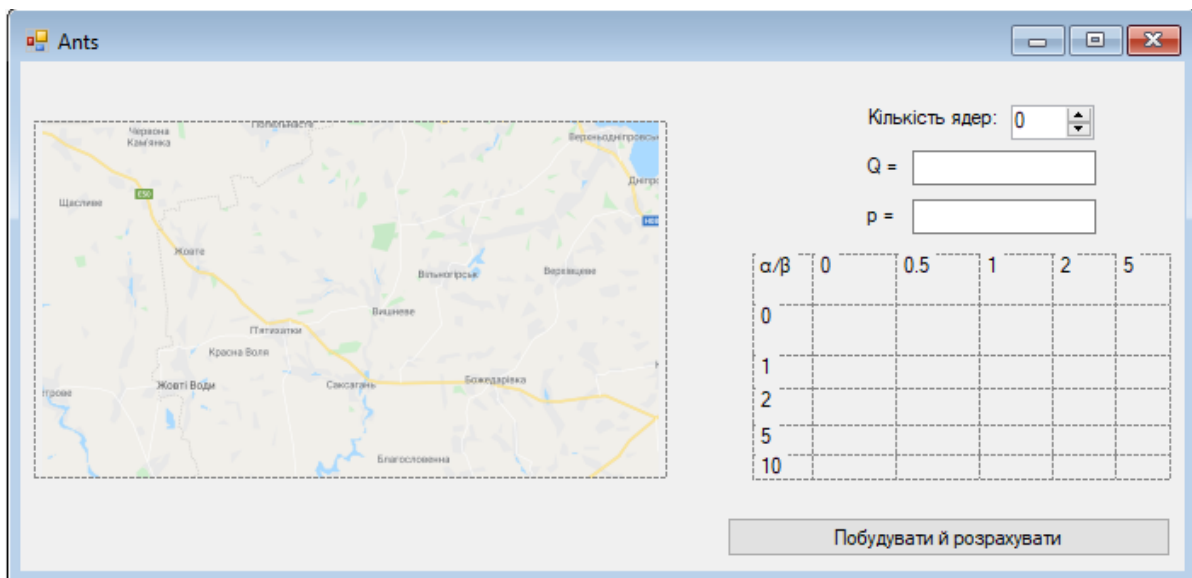


Рисунок 1 – Головне вікно програмного додатку на початку роботи

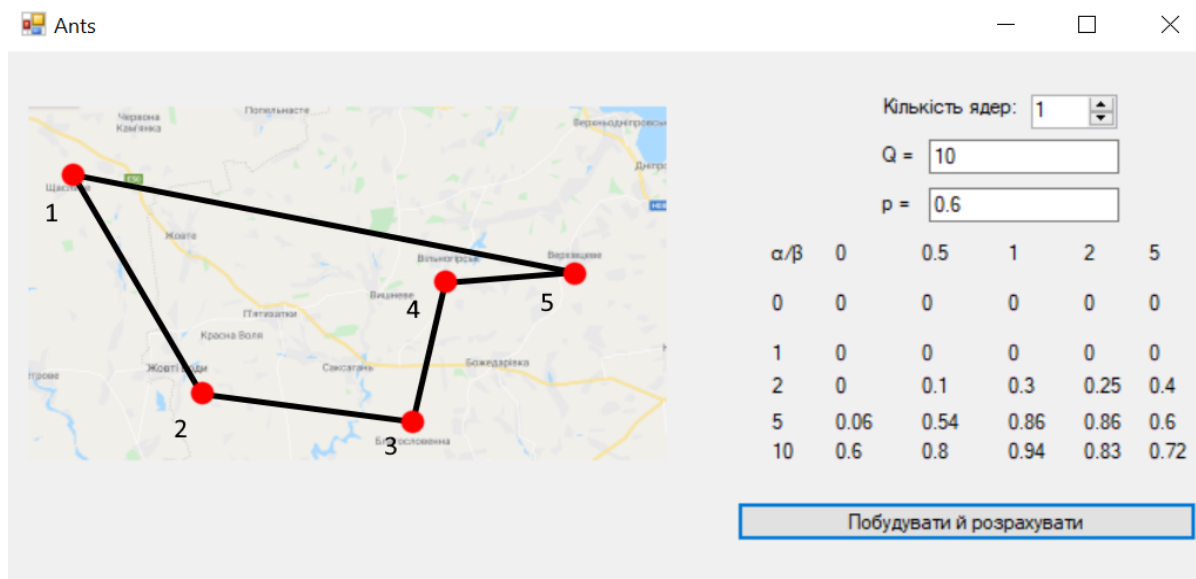


Рисунок 2 – Результати розрахунку

У табл. 1-3 представлено надійність алгоритму, усереднену по 100 прогонів для одного, двох і чотирьох ядер відповідно. Всім варіантам алгоритму давалося 10 ітера-

цій, 25 мурах і параметри $\rho = 0.6$ і $Q = 10$. При цьому в разі декількох ядер мається на увазі сумарна кількість мурах на всіх ядрах.

Таблиця 1 – Надійність алгоритму, усереднена по 100 прогонів для одного ядра

α/β	0	0.5	1	2	5
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0.1	0.3	0.25	0.4
5	0.06	0.54	0.86	0.86	0.6
10	0.6	0.8	0.94	0.83	0.72

Таблиця 2 – Надійність алгоритму, усереднена по 100 прогонів для двох ядер

α/β	0	0.5	1	2	5
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0.14	0.06	0
5	0	0.64	0.86	0.78	0.43
10	0.57	0.78	0.88	0.81	0.54

Таблиця 3 – Надійність алгоритму, усереднена по 100 прогонів для чотирьох ядер

α/β	0	0.5	1	2	5
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0.12	0.01	0
5	0.5	0.68	0.90	0.81	0.35
10	0.60	0.85	0.97	0.91	0.63

З цих таблиць можна зробити висновок, що при збільшенні кількості ядер, тобто окремих популяцій, якісно робота мурашиного алгоритму лише змінюється, хоча і спостерігається деяка зміна надійності для окремих налаштувань. В той же час, істотно підвищується швидкість роботи. Так, при одному ядрі один прогін займає близько 0.55 секунди, при двох ядрах це займає близько 0.27 секунди, а при чотирьох – близько 0.16. Тобто розпаралелювання мурашиного алгоритму дозволяє істотно знизити час виконання програмного додатку, не погіршуючи надійності. Найбільшу надійність при однакових ресурсах показує алгоритм зі значеннями параметрів $\alpha = 1$, $\beta = 10$.

Висновки. Мурашиний алгоритм застосовується для вирішення складних комплексних задач оптимізації, а типовими сферами, де можна застосувати цей алгоритм, є задача календарного планування, розподіл ресурсів та робіт, задача маршрутизації транспорту і таке інше, тому підвищення ефективності розрахунків за цим алгоритмом є досить актуальним.

ЛІТЕРАТУРА

1. Левитин А.В. Алгоритмы. Введение в разработку и анализ. М.: Вильямс, 2006. 576с.
2. Dorigo M. & Gambardella L.M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1997, 1 (1): 53-66.
3. Олейник А.А. Сравнительный анализ методов оптимизации на основе методов муравьиных колоний. *Комп'ютерне моделювання та інтелектуальні системи: збірник наукових праць*. Запоріжжя: ЗНТУ, 2007. С.147-159.

Надійшла до редколегії 16.12.20109.