

УДОСКОНАЛЕННЯ МЕТОДУ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ НА БАЗІ МУРАШИНОЇ КОЛОНІЇ

Вінницький національний технічний університет

Анотація

У роботі проаналізовано задачу комівояжера. Вдосконалено метаевристику, яка базується на використанні поведінкової моделі колонії мурашок, за допомогою зміни установки значень феромону та ймовірного вибору наступного вузла з використанням «жадібних» програмних агентів, що дозволило зменшити час розв'язання задачі комівояжера.

Ключові слова: комбінаторна оптимізація, метрична задача комівояжера, еволюційні алгоритми, метод оптимізації мурашиною колонією, агент, феромон.

Abstract

In this work has been analyzed the travelling salesman problem. Metaheuristic, which is based on the usage of the behavior of the ant colony, is improved by changing the setting of pheromone values and probabilistic selection of the next node using "greedy" software agents, which reduced the time to solve the salesman's task.

Keywords: combinatorial optimization, euclid travelling salesman problem, evolution algorithms, ant colony optimization, agent, pheromone.

Вступ

Задача комівояжера (*Travelling Salesman Problem*) – одна з найвідоміших задач комбінаторної оптимізації, яка полягає у знаходженні найвигіднішого маршруту, який проходить через вказані міста хоча б по одному разу з поверненням у вихідне місто. Фактично необхідно знайти мінімальний гамільтоновий цикл на графі. Слід зазначити, що як міра вигідності маршруту можуть служити сумарний час у дорозі, сумарна вартість шляху тощо. Дана задача надзвичайно важлива для транспортування вантажів, логістики, робототехніки, коли декілька транспортних засобів обмеженої вантажопідйомності повинні обслужити клієнтів, відвідуючи їх в задані часові проміжки, знайшовши оптимальний маршрут. Задача комівояжера займає особливе місце в комбінаторній оптимізації і дослідженні операцій. Історично вона була однією з тих задач, які послужили поштовхом для розвитку цих областей знань. Простота формулювання, скінченність множини допустимих рішень, наочність і в той же час колосальні затрати на повний перебір до цих пір підштовхують математиків до розробки все нових і нових чисельних методів пошуку оптимального розв'язку. На жаль, доведено, що не існує алгоритму розв'язання задачі комівояжера, що має степеневу складність. Це робить дану задачу (при великій розмірності) безнадійною для комп'ютера з послідовним виконанням операцій. У такому разі слід відмовитися від спроб відшукати точне рішення і зосередитися на пошуку наближеного, нехай не оптимального, але хоч би близького до нього. Тому застосовують методи, які дозволяють виконати деяку апроксимацію, визначаючи певний «оптимальний» маршрут, який, можливо, зовсім не є найкращим. Такі методи називають *метаевристичними* або *метаевристиками*. Вони дозволяють поступово покращувати деякий поточний наближений розв'язок [1-4].

До метаевристичних прийнято відносити метод імітації відпалу (*simulated annealing*), пошук із заборонами (*tabu search*), пошук з околицями, які чергуються (*variable neighborhood search*), імовірнісні жадібні алгоритми (*greedy randomized adaptive search procedure*), еволюційні методи (*evolutionary computation*): генетичні алгоритми (*genetic algorithms*), наслідування мурашиної колонії або мурашині алгоритми (*ant colony optimization*) тощо. Методи, які пов'язані з природою часто називають тими, які інспіровані живою або неживою природою (*nature inspired algorithms*). Ідея всіх цих методів заснована на припущенні, що цільова функція $f(x)$ має багато локальних екстремумів, а аналіз усіх допустимих рішень неможливий, не дивлячись їх кінцеву кількість. У такій ситуації

потрібно зосередити пошук на найбільш перспективних частинах області допустимих рішень. Таким чином, задача зводиться до виявлення таких областей і швидкого їх перегляду. Кожна з метаевристик вирішує цю проблему по-своєму. Досить популярними для розв'язання задачі комівояжера є еволюційні методи класу м'яких обчислень (soft computing), наприклад, інспірований живою природою, мурашиний алгоритм [2].

Мурашиний алгоритм (метод оптимізації мурашиною колонією) – один з найефективніших поліноміальних алгоритмів для знаходження наближених розв'язків задачі комівояжера, а також аналогічних задач пошуку маршрутів на графах. В основі мурашиних алгоритмів лежать принципи самоорганізації мурашиної колонії в природі. Не дивлячись на роз'єднану поведінку кожного зі своїх представників, вона утворює високоорганізовану систему, яка складається з великої кількості «агентів» – *мурашок*, які разом перевершують здібності кожного окремого елемента і здатні вирішувати складні задачі [5-12].

Метою роботи є аналіз метричної задачі комівояжера та вдосконалення методу, який базується на використанні поведінкової моделі колонії мурашок, за допомогою зміни установки значень феромону та ймовірнісного вибору наступного вузла з використанням «жадібних» програмних агентів, що дозволило зменшити кількість ітерацій пошуку маршрутів та відповідно час розв'язання задачі комівояжера.

Результати дослідження

Постановка задачі. Будемо розглядати метричну задачу комівояжера. Її можна представити у вигляді моделі на графі G , тобто, використовуючи вершини і ребра між ними. Таким чином, вершини графа відповідають вузлам, а ребра (i, j) між вершинами i і j – шляхи сполучення цих міст. Кожному ребру (i, j) відповідає критерій вигідності L_{ij} . Маршруту $x = \{(j_1, j_2), (j_2, j_3), (j_3, j_4), \dots, (j_N, j_1)\}$ ставиться у відповідність функція вартості (довжина)

$$f(x) = L(j_1, j_2) + L(j_2, j_3) + L(j_3, j_4) + \dots + L(j_N, j_1). \quad (1)$$

Оскільки задача розглядається на площині, то будемо обчислювати евклідову відстань між парами міст у певному маршруті. Враховуючи те, що задача замкнута, можна визначити цільову функцію як суму загальної відстані між всіма містами та відстані між останнім і першим містом. Тоді шукане рішення задачі оптимізації представимо у формі

$$f(x^*) = \min_{l=1, \dots, (N-1)!/2} \{f_l(x)\}, \quad (2)$$

де $(N-1)!/2$ – кількість усіх можливих гамільтонових циклів на графі для симетричної задачі (для асиметричної – $(N-1)!$) [1].

Алгоритм. Суть роботи мурашиного алгоритму полягає в аналізі та використанні моделі поведінки реальних мурашок, які шукають дорогу від мурашника (колонії) до джерела живлення. В основі алгоритму лежить поведінка мурашиної колонії – маркування вдалих доріг великою кількістю феромону, який залишається на поверхні. Тоді як ізольований мураха більш менш випадково рухається у просторі, мураха, який виявив помічений феромоном шлях, з певною ймовірністю піде по ньому і зміцнить його своїм власним феромоном. Таким чином, імовірність того, що в майбутньому інші мурашки будуть рухатися даним шляхом, росте з кількістю мурашок, які раніше використали цей шлях. Це приводить до виникнення найкоротших шляхів, оскільки феромон прагне акумулюватися швидше на цих шляхах. Мурашка представляє собою програмного агента колонії, який забезпечується набором простих правил, що дозволяє йому вибирати шлях на графі. Кожна мурашка вибирає наступне ребро для включення в своє часткове рішення, ґрунтуючись на евристичному оцінюванні цього ребра і кількості феромону, пов'язаного з цим елементом шляху. Безпосередній вибір наступного вузла здійснюється за принципом «колеса рулетки»: для кожної мурашки генерується маршрут руху з останнього місцезнаходження випадковим чином, з урахуванням ймовірності переходу. Потім для кожного з отриманих маршрутів розраховується цільова функція (загальна довжина маршруту) та вибирається найменше значення. Після завершення

маршруту кожний мураха залишає певну кількість ферменту на ребрах між містами. Для дослідження всього простору рішень необхідно забезпечити випаровування феромону – зменшення в часі кількості залишеного на попередніх ітераціях ферменту [8, 11, 12].

Запропоновано оновлювати рівень феромонів на ребрах не тільки в кінці чергової ітерації, але і під час кожного переходу мурах між вузлами, а в кінці ітерації підвищувати рівень феромонів тільки на найкоротшому шляху. Для збереження кращого шляху вводяться «жадібні» мурахи. Вершина, до якої мураха буде рухатися далі, вибирається з усіх суміжних невідвіданих вершин так, щоб її привабливість була найбільшою. Тобто, якщо мураха ще не відвідала всі вузли мережі, то ймовірність його переходу з вершини i в вершину j за списком міст M визначається за допомогою ймовірнісного рівняння

$$p = \begin{cases} 100\% \cdot q, \text{ де } q \in [0,1], \text{ тобто } j = \arg \max(\eta_{i,j}^\alpha \cdot \tau_{i,j}^\beta), \text{ якщо } j \in M; \\ 1 - q, \text{ тобто } \begin{cases} \frac{\eta_{i,j}^\alpha \cdot \tau_{i,j}^\beta}{\sum_{j=1}^{\text{length}(M)} \eta_{i,j}^\alpha \cdot \tau_{i,j}^\beta} \cdot 100\%, \text{ якщо } j \in M; \\ 0\%, & \text{якщо } j \notin M; \end{cases} \end{cases} \quad (3)$$

де q – ймовірність того, що поточний мураха «жадібний», а $(1-q)$ – звичайний (визначається експериментально, можна встановити на рівні 0,85-0,95); $\eta_{ij} = 1/L_{ij}$ – коефіцієнт видимості вузла; τ_{ij} – рівень феромону на ребрі (i, j) на ітерації t ; α і β – два параметри налаштування, які задають ваги видимості та феромону на ребрі.

Рівень феромону оновлюється відповідно до наведеної формули на величину

$$\Delta \tau_{ij,k} = \begin{cases} \frac{Q}{L_k}, & \text{якщо } (i, j) \in x_k; \\ 0, & \text{якщо } (i, j) \notin x_k. \end{cases} \quad (4)$$

де x_k – порядок міст, який пройдено мурахою k на даній ітерації; Q – регульований параметр (одного порядку з довжиною оптимального маршруту).

Правило поновлення феромону має вигляд

$$\tau_{ij}(t+1) = \underbrace{\tau_{ij}(t) + \Delta \tau_{ij}(t)}_{\text{новий рівень феромону випаровування}} - r \cdot \tau_{ij}(t), \quad (5)$$

де $\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij,k}(t)$; m – кількість мурах у колонії; $r \in [0,1]$ – коефіцієнт випаровування.

Основними перевагами методу оптимізації мурашиною колонією є мала похибка знайденого рішення, низькі часові витрати при роботі з задачами великої розмірності, простота модифікації, можливість розпаралелювання, використання в динамічних задачах тощо. Метод вважається одним із найефективніших і широко застосовується на практиці.

Висновки

Найбільш перспективним напрямом досліджень у вивченні мурашиних алгоритмів вважається аналіз підбору опційних параметрів, які відчутно впливають на поведінку алгоритму. У даній роботі розглянуто механізм реалізації модифікованого метаевристичного методу оптимізації мурашиною колонією із застосуванням задачі комівояжера як тестового полігону. Для покращення часових характеристик алгоритму в розрахунок вводяться «жадібні» мурахи, які підсилюють ребра

найкращого маршруту. Також застосована зміна установки значень феромону та ймовірнісний вибір наступного вузла з використанням «жадібних» і звичайних агентів. Якщо виконувати порівняння з перебірними (динамічне програмування, метод гілок та меж) та навіть метаевристичними (імітація відпалу, генетичні алгоритми) методами, то метод оптимізації мурашиною колонією знаходить близькі до оптимуму рішення за значно менший час навіть для задач невеликої розмірності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Иванов Ю.Ю. Вступ до Computer Science. Дискретна математика: цікава та не дуже: лекції, алгоритми та задачі / Ю.Ю. Иванов. – 2018. – 89 с. – Режим доступу: https://iq.vntu.edu.ua/method/read_url.php?tbl_num=2&url=/fdb/1166/Discrete_Math_by_IVANOV.djvu.
2. Ерзин А.И. Задачи маршрутизации: учебное пособие / А.И. Ерзин, Ю.А. Кочетов. – Новосибирск: РИЦ НГУ, 2014. – 95 с.
3. Щербина О.А. Метаевристические алгоритмы для задач комбинаторной оптимизации // О.А. Щербина / Таврический Вестник Информатики и Математики. – АР Крым: ТВИМ, 2014. – Выпуск 1 (24). – С. 56-72.
4. Скиена С. Алгоритмы. Руководство по разработке / С. Скиена. – СПб.: БХВ-Петербург, 2011. – 720 с.
5. Джонс М.Т. Программирование искусственного интеллекта в приложениях / М.Т. Джонс. – М.: ДМК Пресс, 2011. – 312 с.
6. Алексеев С.В. Обробка інформації в складних організаційних системах // С.В. Алексеев, В.О. Мартовицкий / Системы обробки інформації. – 2014. – Выпуск 2 (118). – С. 84-88.
7. Кирсанов М.Н. Графы в Maple / М.Н. Кирсанов. – М.: Физматлит, 2007. – 168 с.
8. Dorigo M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem // M. Dorigo, L.M. Gambardella / IEEE Transactions of Evolutionary Computing. – 1997. – V. 1. – P. 53-66.
9. Bonavear E. Swarm Intelligence: from Natural to Artificial Systems / E. Bonavear, M. Dorigo. – Oxford: Oxford University Press, 1999. – 307 p.
10. Dorigo M. Swarm Intelligence, Ant Algorithms and Ant Colony Optimization // M. Dorigo / Reader for CEU Summer University Course «Complex System». – Budapest: Central European University, 2001. – P. 1-38.
11. Штовба С.Д. Муравьиные алгоритмы // С.Д. Штовба / Математика в приложениях. – 2003. – № 4. – С. 70-75.
12. Shtovba S. Ant Algorithms: Theory and Applications // S. Shtovba / Programming and Computer Software. – 2005. – V. 31. – № 4. – P. 167-178.

Панасюк Андрій Олександрович — магістрант групи ІАКІТ-17м, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м. Вінниця.

Іванов Юрій Юрійович — канд. техн. наук, асистент кафедри автоматики та інформаційно-вимірювальної техніки, Вінницький національний технічний університет, м. Вінниця, e-mail: Yura881990@i.ua.

Науковий керівник: **Іванов Юрій Юрійович** — канд. техн. наук, асистент кафедри автоматики та інформаційно-вимірювальної техніки, Вінницький національний технічний університет, м. Вінниця.

Panasjuk Andriy O. — graduate student, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia.

Ivanov Yurii Yu. — Cand. Sc. (Eng), Assistant Professor, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: Yura881990@i.ua.

Supervisor: **Ivanov Yurii Yu.** — Cand. Sc. (Eng), Assistant Professor, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia.