

С.В. Шандюк, студ., V курс, гр. ПІ-43м, ФІКТ
Науковий керівник – ст. викл. В.О. Скачков
Житомирський державний технологічний університет

КЛАСИЧНИЙ МУРАШИНИЙ АЛГОРИТМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА

Мурашиний алгоритм (алгоритм оптимізації мурашиної колонії, англ. *ant colony optimization*, АСО) — один з ефективних поліноміальних алгоритмів для знаходження наближених розв'язків задачі комівояжера, а також аналогічних завдань пошуку маршрутів на графах. Підхід запропонований бельгійським дослідником Марко Доріго (англ. *Marco Dorigo*). Суть підходу полягає в аналізі та використанні моделі поведінки мурах, що шукають дороги від колонії до їжі. У основі алгоритму лежить поведінка мурашиної колонії — маркування вдалих доріг великою кількістю феромону. Робота починається з розміщення мурашок у вершинах графа (містах), потім починається рух мурашок — напрям визначається імовірнісним методом, на підставі формули. У реальному світі, мурахи (спочатку) ходять у випадковому порядку і по знаходженню продовольства повертаються в свою колонію, прокладаючи феромонами тропи. Якщо інші мурахи знаходять такі стежки, вони, найімовірніше, підуть за ними. Замість того, щоб відстежувати ланцюжок, вони зміцнюють його при поверненні, якщо в кінцевому підсумку знаходять джерело живлення. Згодом феромонна стежка починає випаровуватися, тим самим зменшуючи свою привабливу силу. Чим більше часу потрібно для проходження шляху до мети і назад, тим сильніше випарується феромонна стежка. На короткому шляху, для порівняння, проходження буде більш швидким і як наслідок, щільність феромонів залишається високою. Випаровування феромонів також має властивість уникнення прагнення до локально-оптимального рішення. Якби феромони не випаровувалися, то шлях, обраний першим, був би найпривабливішим. У цьому випадку, дослідження просторових рішень були б обмеженими. Таким чином, коли один мураха знаходить (наприклад, короткий) шлях від колонії до джерела їжі, інші мурахи, швидше за все підуть цим шляхом, і позитивні відгуки в кінцевому підсумку призводять всіх мурах до одного, найкоротшого, шляху.

Мурахи використовують навколишнє середовище як засіб спілкування. Вони обмінюються інформацією непрямым шляхом, через феромони, в ході їх «роботи». Обмін інформацією має локальний характер, тільки ті мурахи, які знаходяться в безпосередній близькості, де залишилися феромони - можуть дізнатися про них. Даний механізм вирішення проблеми дуже складний і є хорошим прикладом самоорганізації системи. Мурашиний алгоритм моделює багатоагентну систему. Кожна мураха зберігає в пам'яті список пройдених їм вузлів. Цей список називають списком заборон (*tabu list*) або просто пам'яттю мурашки. Вибираючи вузол для наступного кроку, мураха «пам'ятає» про вже пройдені вузли і не розглядає їх як можливих для переходу. На кожному кроці список заборон поповнюється новим вузлом, а перед новою ітерацією алгоритму - тобто перед тим, як мураха знову проходить шлях - він очищується. Крім списку заборон, при виборі вузла для переходу мураха керується «привабливістю» ребер, які він може пройти. Вона залежить, по-перше, від відстані між вузлами (тобто від ваги ребра), а по-друге, від слідів феромонів, залишених на ребрі від мурах що пройшли по ньому раніше. Природно, що на відміну від ваг ребер, які є константними, сліди феромонів оновлюються на кожній ітерації алгоритму: як і в природі, з часом сліди випаровуються, а проходять мурахи, навпаки, посилюють їх.

Варіації мурашиного алгоритму: елітарна мурашина система; пропорційні псевдовипадкові правила; рангова мурашина система; тривала ортогональна колонія мурашок.

Для порівняння алгоритмів побудуємо повний граф з п'ятьма вершинами і довільно виберемо в ньому початкову вершину. Значення інших трьох параметрів будемо змінювати. Оцінимо ефективність алгоритму на основі двох параметрів: число необхідних ітерацій, довжина знайденого шляху. Оскільки агенти вибирають шлях виходячи з імовірнісних характеристик, отримані значення даних параметрів в кожній ітерації будуть різними, тому для оцінки ефективності алгоритмів знайдемо математичні очікування параметрів для вибірки з 500 дослідів. Зафіксуємо Q і P і будемо варіювати кількість агентів. Залежність кількості ітерацій від числа агентів представлена на рисунку 1. З ростом числа агентів зростає і число ітерацій. Пояснюється це тим, що алгоритм завершується в момент, коли всі агенти вибрали один і той же шлях, але з ростом числа агентів зростає вплив на число ітерацій невеликих ймовірностей переходів.

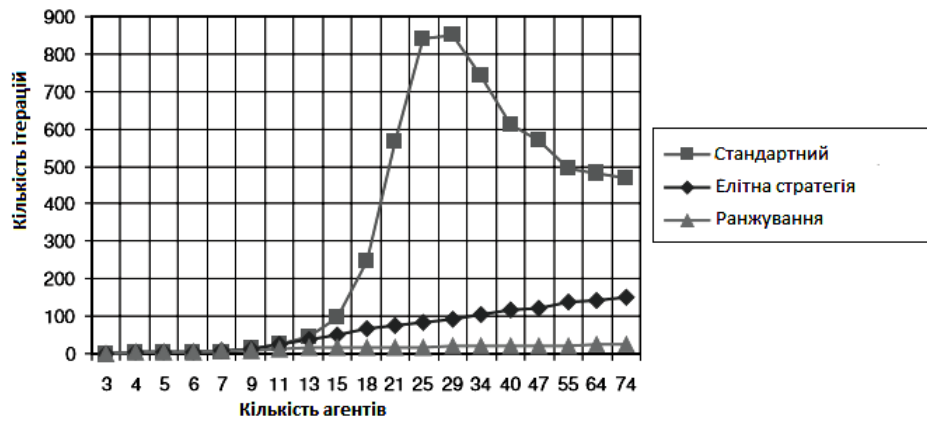


Рис. 1

Алгоритм, заснований на елітній стратегії, показав більш швидку збіжність, оскільки конкретний шлях визначається вже на першій ітерації, і він буде, скоріше за все, використаний багатьма клієнтами на наступній ітерації. В результаті алгоритм сходиться до вирішення швидше. Швидка збіжність алгоритму, заснованого на ранжируванні, обумовлена дуже великою кількістю феромонів, внесеного мурахами, що знайшли найкоротший шлях. Крім того, відразу, виключаючи з розгляду довгі дуги, алгоритм включає в розгляд короткі дуги.

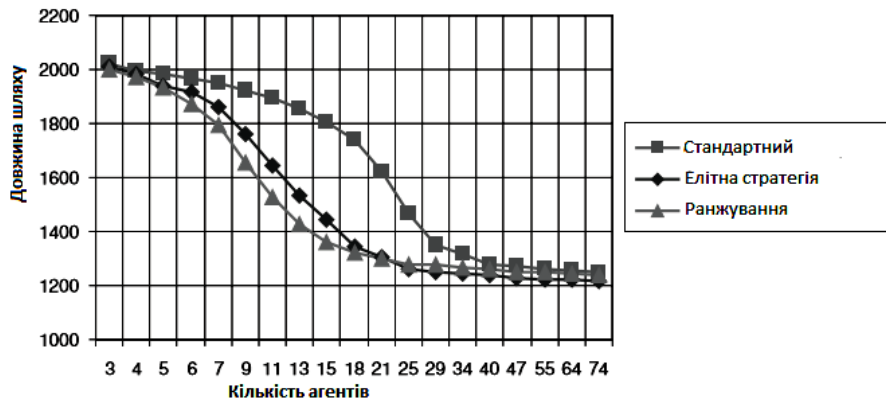


Рис. 2

На рисунку 2 показано, як змінюється довжина знайденого шляху в залежності від числа агентів. Модифікації стандартного алгоритму при однаковому числі агентів дозволяють визначати коротші шляхи, так як відбувається виділення коротких дуг з більшою кількістю феромона, ніж при використанні стандартного алгоритму. З наведених малюнків очевидно, що після досягнення деякого порогового числа агентів подальше його збільшення не призводить до зростання числа ітерацій і зменшення довжини шляху. Насичення обумовлено кінцевим числом дуг графа. На наведених графіках насичення настає при числі агентів, що перевищує 40. При зростанні числа агентів вище порога насичення, агенти нових шляхів вже не знаходять, і тому величина найкоротшого шляху, знайденого агентами, не збільшується. Число ітерацій починає зменшуватися в результаті того, що більша кількість агентів починають вибирати найкоротший шлях, збільшуючи на ньому кількість виділеного феромону. В результаті агенти знаходять шлях швидше. Найкраще визначати таке число агентів, яке призведе до насичення графа, але воно дуже велике і вимагає багато ресурсів. Завдання комівояжера є, можливо, найбільш досліджуваним завданням комбінаторної оптимізації. Не дивно, що саме воно було обране першим для вирішення з використанням підходу, запозиченим з механізму поведінки мурашиної колонії. Пізніше цей підхід знайшов застосування у вирішенні багатьох інших комбінаторних проблем, в числі яких задачі про призначення, завдання розмальовки графа, завдання маршрутизації, завдання з областей data mining і розпізнавання образів та інші. Ефективність мурашиного алгоритму можна порівняти з ефективністю загальних метаевристичних методів, а в ряді випадків - і з проблемно-орієнтованими методами. Найкращі результати мурашині алгоритми показують для задач з великими розмірами областей пошуку. Мурашині алгоритми добре підходять для застосування разом з процедурами локального пошуку, дозволяючи швидко знаходити початкові точки для них.