

Мурашині алгоритми

Мурахи не можна назвати кмітливою. Окрема мураха не в змозі прийняти жодного рішення. Вона влаштована вкрай примітивно: всі її дії зводяться до елементарних реакцій на навколишнє оточення і своїх побратимів. Мураха не здатна аналізувати, робити висновки і шукати рішення.

Ці факти, однак, ніяк не узгоджуються з успішністю мурах як виду. Вони існують на планеті понад 100 мільйонів років, будують величезні споруди, забезпечують їх всім необхідним і навіть ведуть справжні війни. У порівнянні з повною безпорадністю окремих особин, досягнення мурах здаються немислимыми.

Домогтися таких успіхів мурахи здатні завдяки своїй соціальності. Вони живуть тільки в колективах - колоніях. Всі мурахи колонії формують так званий ройовий інтелект. Особини, що складають колонію, не повинні бути розумними: вони повинні лише взаємодіяти за певними простими правилами і тоді колонія цілком буде ефективною.

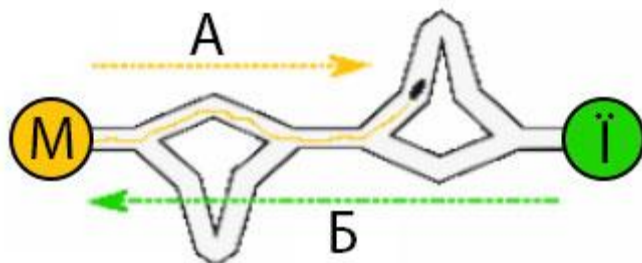
В колонії немає домінуючих особин, немає начальників і підлеглих, немає лідерів, що роздають вказівки і координують дії. Колонія є повністю самоорганізованою. Кожна з мурах володіє інформацією лише про локальну обстановку, жодна з них не має уявлення про всю ситуації в цілому - тільки про те, що дізналася сама або від своїх родичів, явно чи неявно. На неявних взаємодіях мурах засновано механізми пошуку найкоротшого шляху від мурашника до джерела їжі.

Кожен раз проходячи від мурашника до їжі і назад, мурахи залишають за собою доріжку феромонів. Інші мурахи, відчувши такі сліди на землі, будуть інстинктивно спрямовуватися до нього. Оскільки ці мурахи теж залишають за собою доріжки феромонів, то чим більше мурах проходить певним шляхом, тим більш привабливим він стає для їх родичів. При цьому, чим коротше шлях до джерела їжі, тим менше часу потрібно мурашкам на нього - а отже, тим швидше залишені на ньому сліди стають помітними.

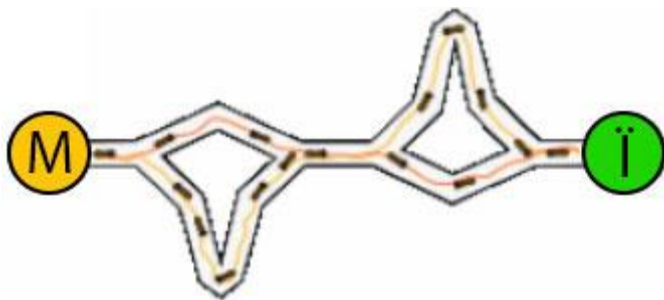
У 1992 році у своїй дисертації Марко Дориго запропонував запозичити описаний природний механізм для вирішення завдань оптимізації. Імітуючи поведінку колонії мурах в природі, мурашині алгоритми використовують багатоагентні системи, агенти яких функціонують по вкрай простим правилам. Мурашині алгоритми є ефективними при вирішенні складних комбінаторних завдань та задач оптимізації.

Ідея алгоритму

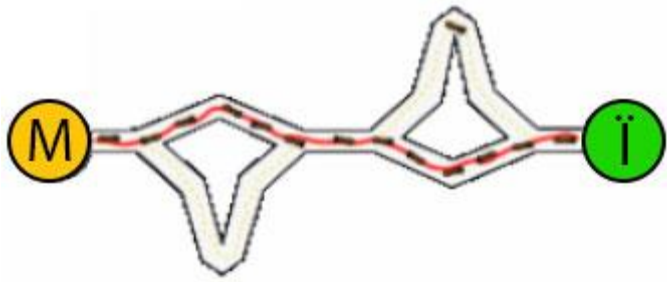
Мураха прямує від мурашника (М) у випадковому напрямку. Якщо вона знаходить їжу (Ї), то повертається до мурашника і залишає по собі слід з феромону.



Ці феромону приваблюють інших мурах, що знаходяться поруч, і скоріш за все вони рушають цим маршрутом. Повертаючись до мурашника, вони також залишають свій феромон і укріплюють доріжку.



Якщо існує два маршрути, то коротким за цей час встигне пройти більше мурах ніж по довгому і короткий маршрут стане більш вигідним. Довгі доріжки повільно зникають внаслідок випаровування феромонів.



Базова ідея алгоритму мурахи полягає в оптимізації шляхом непрямого зв'язку між автономними агентами.

Мурашиний алгоритм моделює багатоагентну систему. Її агентів надалі будемо називати мурахами. Як і справжні мурахи, вони досить просто влаштовані: для виконання своїх обов'язків вони вимагають невелику кількість пам'яті, а на кожному кроці роботи виконують нескладні обчислення.

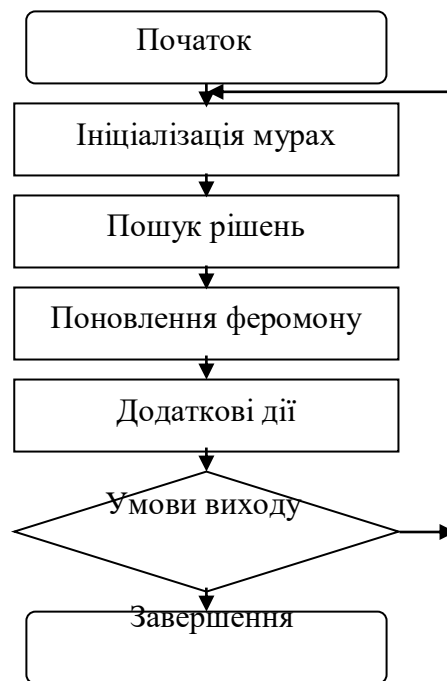
Кожна мураха зберігає в пам'яті список пройдених їм вузлів. Цей список називають списком заборон (tabu list) або просто пам'яттю мурашки.

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 мурашка знову проходить шлях - він очищується.

Окрім списку заборон, при виборі вузла для переходу мураха керується «привабливістю» ребер, які вона може пройти. Привабливість залежить, по-перше, від відстані між вузлами (тобто від ваги ребра), а по-друге, від слідів феромонів, залишених на ребрі мурахами, 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 мурахи, що проходять, навпаки, посилюють їх.

Узагальнено, базовий мурашиний алгоритм, незалежно від модифікацій, можна представити у вигляді схеми



Покроковий опис загальної схеми

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 двохскерованим, 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.

Мураха

Мураха - це програмний агент, який є членом великої колонії і використовується для вирішення певної проблеми. Мураха забезпечується набором простих правил, які дозволяють їй вибирати шлях у графі. Вона підтримує список вузлів, які вже відвідала і може пройти через кожен вузол лише один раз. Шлях між двома вузлами графа, за яким мураха відвідала кожен вузол, називається шляхом Гамільтона.

Вузли в списку "поточної подорожі" розташовуються в тому порядку, в якому їх відвідала мураха. Пізніше список використовується для визначення протяжності шляху між вузлами. Справжня мураха під час переміщення по шляху буде залишати за собою феромони. В мурашиному алгоритмі агент залишає феромони на ребрах графа після завершення подорожі.

Стартова точка, куди поміщається мураха, залежить від обмежень, накладених умовами завдання, оскільки для кожного завдання спосіб розміщення мурашок є визначальним. Або всі вони поміщаються в одну точку, або в різні з повтореннями, або без повторень.

На цьому ж етапі задається початковий рівень феромону. Він ініціалізується невеликим додатнім числом для того, щоб на початковому кроці ймовірності переходу в наступну вершину були нульовими.

Рух мурашки

Рух мурашки ґрунтується на простому ймовірнісному рівнянні. Якщо мураха ще не закінчила шлях, тобто не відвідала усі вузли мережі, для визначення наступного ребра шляху використовується рівняння


$$(1)$$

Тут $\tau(r,u)$ - інтенсивність ферменту на ребрі між вузлами r і u , $\eta(r,u)$ - функція, яка представляє вимір зворотної відстані для грані, α - вага ферменту,

а β - коефіцієнт евристики. Параметри α і β визначають відносну значимість двох параметрів, а також їх вплив на рівняння. Оскільки мураха подорожує тільки по вузлах, які ще не були відвідані (як зазначено списком табу), ймовірність обчислюється лише для ребер, які ведуть до ще не відвіданих вузлів. Ці ребра представляє змінна k .

Подорож мурашки

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. Рівняння (2) показує кількість феромону, який був залишений на кожному ребрі шляху для мурашки k . Змінна Q є константою.



$$(2)$$

Результат рівняння є засобом вимірювання шляху, - короткий шлях характеризується високою концентрацією феромонів, а більш довгий шлях - більш низькою. Далі, отриманий результат використовується в рівнянні (3), щоб збільшити кількість феромону вздовж кожного ребра пройденого мурахою шляху.



$$(3)$$

Важливо, що дане рівняння застосовується до всього шляху, при цьому кожне ребро позначається феромоном пропорційне до довжини шляху. Тому слід дочекатися, поки мураха закінчить подорож і лише потім оновити рівні феромону, в іншому випадку справжня довжина шляху залишиться невідомою. Константа p - значення між 0 і 1.

Випаровування феромонів

На початку шляху у кожного ребра є шанс бути обраним. Щоб поступово видалити ребра, які входять в гірші шляхи графа, до всіх ребер застосовується

процедура випаровування феромону. Використовуючи константу p з рівняння (3), отримуємо рівняння (4):

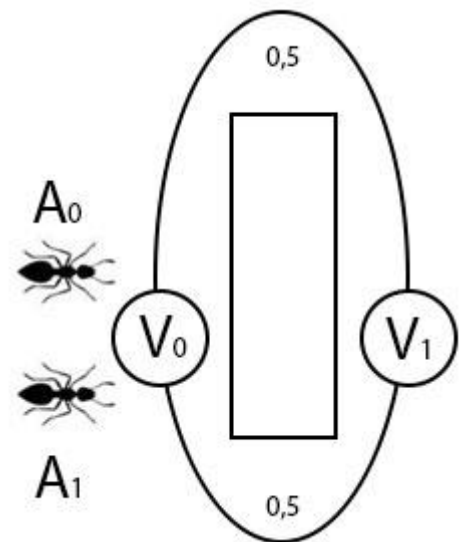


(4)

Для випаровування феромону використовується зворотний коефіцієнт оновлення шляху.

Повторний запуск

Після того, як шлях мурашки завершено, ребра оновлено відповідно до довжини шляху і сталося випаровування феромону на всіх ребрах, алгоритм запускається повторно. Список табу очищується, і довжина шляху обнулюється. Мурахам дозволяється переміщатися по графу, засновуючи вибір ребра на рівнянні (1). Цей процес може виконуватися для постійної кількості шляхів або до моменту, коли протягом кількох запусків не було відзначено повторних змін. Потім визначається кращий шлях, який і є рішенням.



Демонстраційний приклад

Щоб побачити, як працюють рівняння, розберемо функціонування розглянутого вище алгоритму на простому прикладі. Візьмемо простий сценарій з двома мурашками з прикладу яке розглянуто вище.

На рисунку показано приклад з двома ребрами між двома вузлами (V_0 і V_1). Кожне ребро ініціалізується і має однакові шанси на те, щоб бути обраним.

Два мурахи, що знаходяться у вузлі V_0 позначаються як A_0 і A_1 . Оскільки ймовірність вибору будь-якого шляху однакова, в цьому циклі проігноруємо рівняння вибору шляху.

Дані для завдання:

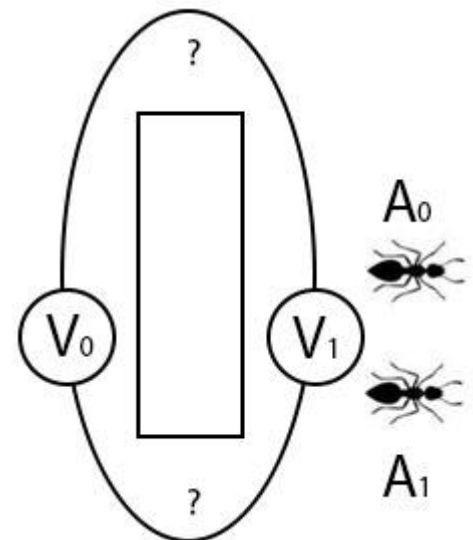
- Число пройдених кроків: для A_0 - 20, для A_1 - 10

- Рівень феромону (Q / пройдена відстань): для A_0 - 0.5, A_1 - 1.0

- $\rho = 0.6$

- $\alpha = 0.3$

- $\beta = 1.0$



На наступному рисунку показано, що кожна мураха вибирає свій шлях (мураха A_0 йде по верхньому шляху, а мураха A_1 - по нижньому). Мураха A_0 зробила 20 кроків, а мураха A_1 , - тільки 10. За рівнянням (2) розраховуємо кількість феромонів, яке має бути "нанесено".

Примітка: Роботу алгоритму можна змінити, якщо перевизначити його параметри (наприклад, α , β або ρ), наприклад надати більшу вагу феромонам або відстані між вузлами.

Далі за рівнянням (3) обчислюється кількість феромону, яка буде застосовано.

Для мурахи A_0 результат становить: $0,1 + (0,5 * 0,6) = 0,4$


Для мурахи A_1 результат становить: $0,1 + (0,5 * 0,6) = 0,4$

Далі за допомогою рівняння (4) визначається, яка частина феромонів випарується і, відповідно, скільки залишиться. Результати (для кожного шляху відповідно) становлять:


$0,4 * (1,0 - 0,6) = 0,16$

$0,7 * (1,0 - 0,6) = 0,28$

Ці значення представляють нову кількість феромонів для кожного шляху (верхнього і нижнього, відповідно). Тепер перемістимо мурах назад у вузол V_0 і скористаємося імовірнісним рівнянням вибору шляху (1), щоб визначити, який шлях повинні вибрати мурахи. Ймовірність того, що мураха вибере верхній шлях (представлений кількістю феромону 0,16), становить:



Ймовірність того, що мураха вибере нижній шлях (представлений кількістю феромону 0,28) становить:



При зіставленні двох ймовірностей обидві мурахи виберуть нижній шлях, який є найбільш оптимальним.

Характерні особливості

Для алгоритму мурашиної колонії необхідно вказати:

- Закон виділення феромону.
- Закон випаровування феромону.
- Кількість агентів.
- Місця розміщення.

Ці характеристики вибираються з врахуванням особливостей завдання на основі експериментальних досліджень (евристики).

Алгоритм:

- Реалізує пошук наближених рішень.
- Має поліноміальну складність.
- Є одним з видів імовірнісних алгоритмів (закони виділення випаровування - імовірнісні закони).

Області застосування

Алгоритм оптимізації мурашиної колонії може бути успішно застосований для вирішення складних комплексних завдань оптимізації. Мета вирішення складних комплексних завдань оптимізації - пошук і визначення

найбільш відповідного рішення для оптимізації (знаходження мінімуму або максимуму) цільової функції (ціни, точності, часу, відстані тощо) з дискретної множини можливих рішень.

Типовими прикладами вирішення такого завдання є задача календарного планування, завдання маршрутизації транспорту, різних мереж (GPRS, телефонні, комп'ютерні тощо), розподіл ресурсів та робіт. Ці задачі виникають у бізнесі, інженерії, виробництві та багатьох інших областях. Дослідження показали, що метод мурашиних колоній може давати результати, навіть кращі ніж при використанні генетичних алгоритмів і нейронних мереж.

Модифікації класичного алгоритму

Результати перших експериментів із застосуванням мурашиного алгоритму для вирішення завдання комівояжера були багатообіцяючими, проте далеко не кращими порівняно з вже існуючими методами. Однак, простота класичного мурашиного алгоритму («мурашиної системи») залишала можливість для доробок - і саме алгоритмічні вдосконалення стали предметом подальших досліджень фахівців у галузі комбінаторної оптимізації. В основному, ці вдосконалення пов'язано з великим використанням історії пошуку та більш ретельним дослідженням областей навколо вже знайдених вдалих рішень.

Elitist Ant System

Введення в алгоритм так званих «елітних мурах». Досвід показує, що проходячи ребра, що входять в короткі шляхи, мурахи з більшою ймовірністю будуть знаходити коротші шляхи. Ефективною стратегією є штучне збільшення рівня феромонів на найвдаліших маршрутах. Для цього на кожній ітерації алгоритму кожна з елітних мурах проходить шлях, який є найкоротшим із знайдених на даний момент.

Експерименти показують, що, до певного рівня, збільшення числа елітних мурах є досить ефективним, дозволяючи значно скоротити число ітерацій алгоритму. Однак, якщо число елітних мурах занадто велике, то алгоритм

досить швидко знаходить субоптимальне рішення і застряє в ньому. Як і інші змінні параметри, оптимальне число елітних мурах слід визначати дослідним шляхом.

Ant-Q

Мурашиний алгоритм, який отримав свою назву за аналогією з методом машинного навчання Q-learning. В основі алгоритму лежить ідея про те, що мурашину систему можна інтерпретувати як систему навчання з підкріпленням. Ant-Q підсилює цю аналогію, запозичуючи багато ідей з Q-навчання.

Алгоритм зберігає Q-таблицю, що співставляє кожному з ребер величину, яка визначає «корисність» переходу по цьому ребру. Q-таблиця змінюється в процесі роботи алгоритму - відбувається навчання системи. Значення корисності переходу по ребру обчислюється виходячи із значень корисності переходу за наступними ребрам в результаті попереднього визначення можливих наступних станів. Після кожної ітерації корисності оновлюються виходячи з довжин шляхів, до складу яких було включено відповідні ребра.

Ant Colony System

Для підвищення ефективності в порівнянні з класичним алгоритмом введено три основних зміни.

По-перше, рівень феромонів на ребрах оновлюється не лише в кінці чергової ітерації, але і при кожному переході мурах з вузла у вузол. По-друге, наприкінці ітерації рівень феромонів підвищується тільки на найкоротшому із знайдених шляхів. По-третє, алгоритм використовує змінене правило переходу: або, з певною часткою ймовірності, мураха безумовно вибирає краще ребро у відповідності до довжини і рівня феромонів, або робить вибір так само, як і в класичному алгоритмі.

Max-min Ant System

Мурашиний алгоритм, в якому підвищення концентрації феромонів відбувається тільки на кращих шляхах з пройдених мурахами. Така велика

увага до локальних оптимумів компенсується введенням обмежень на максимальну і мінімальну концентрацію феромонів на ребрах, які вкрай ефективно захищають алгоритм від передчасної збіжності до субоптимальних рішень.

На етапі ініціалізації, концентрація феромонів на всіх ребрах встановлюється рівною максимальній. Після кожної ітерації алгоритму тільки одна мураха залишає за собою слід - або найбільш успішна на даній ітерації, або, аналогічно до алгоритму з елітизмом, елітна. Цим досягається, з одного боку, більш ретельне дослідження області пошуку, з іншого - його прискорення.

ASrank

Модифікація класичного мурашиного алгоритму, в якому в кінці кожної ітерації мурахи ранжуються у відповідно до довжин пройдених ними шляхів. Кількість феромонів, що залишається мурахою на ребрах, таким чином, призначається пропорційно до її позиції. Для ретельного дослідження околу вже знайдених вдалих рішень, алгоритм використовує елітних мурах.

Висновок

Ефективність мурашиних алгоритмів порівнянна з ефективністю загальних мета евристичних методів, а в ряді випадків - і з проблемно-орієнтованими методами. Найкращі результати мурашині алгоритми показують для задач з великою розмірністю областей пошуку і можуть бути успішно застосовані для вирішення складних комбінаторних задач оптимізації. Мурашині алгоритми добре підходять для застосування разом з процедурами локального пошуку, дозволяючи швидко знаходити початкові точки для них.

Найбільш перспективними напрямками подальших досліджень у даному напрямку слід вважати аналіз способу вибору параметрів, що налаштовуються в алгоритмах. В останні роки пропонуються різні способи адаптації параметрів алгоритмів «на льоту». Оскільки від вибору параметрів сильно залежить

поведінка мурашиних алгоритмів, саме до цієї проблеми звернено найбільшу увагу дослідників на даний момент.