# Тема 6. Ройовий інтелект

Здавна люди дивувалися колективній поведінці живих істот - яким чином птахи летять на південь величезними зграями, не збиваючись з курсу. Як бджоли можуть так точно визначати джерела їжі і здобувати її в необхідному об’ємі для всієї колонії.

Винахідники прагнуть базуватися на надійних та перевірених принципах. Архітектори при проектуванні систем вентиляції висотних будівель використовують хитрощі термітів. Фахівці в області інформаційних технологій вивчають принципи поведінки зграй птахів та риб для того, щоб розробити максимально ефективні системи управління потоками автомобілів і транспорту.

Великі групи тварин/комах можна об'єднати одним загальним словом - рій. З розвитком комп’ютерних технологій дослідники та вчені почали моделювати «ройовий інтелект» у спробах запозичити унікальні можливості колективної поведінки. Хоча не всі спроби були успішними, але вони поклали початок створенню напрямку «ройовий інтелект», заклавши до його основи деякі фундаментальні правила.

Термін «Ройовий інтелект» (Swarm intelligence) введено вченими Ван Цзінь і Херардо Бені в 1989 році. Модель ройового інтелекту представляє «багатоагентну систему», що складається з множини інтелектуальних агентів - програм, здатних самостійно протягом деякого часу виконувати поставлене завдання, самонавчатися та самоорганізовуватися.

Біонічний напрямок має за мету вирішувати різноманітні технічні завдання за допомогою принципів, властивостей і структури живих організмів. Еволюція відсіяла найкращі та ефективні природні механізми. Зацікавленість в природніх алгоритмах можна пояснити багатьма факторами:

* Високий рівень розвитку сучасних технологій, що дозволяє реалізувати принципи функціонування біологічних явищ, систем та організмів.
* Зацікавленість у ефективних та енергозберігаючих технологіях.
* Винайдення нових підходів для простішого вирішення складних оптимізаційних та комбінаторних задач.

Біонічні принципи впроваджуються в метаевристичні алгоритми, і найчастіше використовують наступні концепції:

* **Ройовий інтелект**. Описує колективну поведінку децентралізованої самоорганізованої системи, як метод оптимізації. Системи ройового інтелекту, як правило, складаються з множини агентів (багатоагентна система), що локально взаємодіють між собою і з навколишнім середовищем. Самі агенти, зазвичай, є достатньо простими, але всі разом, локально взаємодіючи, створюють так званий ройовий інтелект. Прикладом в природі може служити рій бджіл, зграя птахів, риб.
* **Колективний людський інтелект.** Людське суспільство представляє реальний інструмент обробки інформації, який добре піддається спостереженню. Якщо ройовий інтелект орієнтовано на отримання складної поведінки в системі з простих елементів, цей підхід, навпаки, досліджує побудову простих і спеціальних структур на базі складних і універсальних об'єктів: «спільнота є менш розумною, ніж більшість її членів».

## 6.1. Метаевристичні алгоритми

Метаевристичні алгоритми - це загальний клас оптимізаційних алгоритмів, які використовуються для знаходження наближених розв'язків складних завдань оптимізації у великих просторах пошуку. Вони засновані на евристиках і прагнуть знайти хороші рішення в розумний час, навіть якщо точне оптимальне рішення недосяжно у прийнятні терміни.

#### Ключові характеристики метаевристичних алгоритмів:

* Метаевристичні алгоритми не гарантують знаходження оптимального рішення, але намагаються знайти прийнятне рішення, близьке до оптимального. Їхня мета - знайти хороший компроміс між якістю рішення та витратами обчислювальних ресурсів.
* Метаевристичні алгоритми працюють ітеративно, послідовно покращуючи поточне рішення кожної ітерації. Вони можуть багаторазово досліджувати та модифікувати рішення, щоб наблизитися до оптимального результату.
* Метаевристичні алгоритми є узагальненими методами, які можуть застосовуватися до широкого спектра оптимізації задач без необхідності адаптації для кожної конкретної задачі.
* Багато метаевристичних алгоритмів використовують випадкові компоненти у своїй роботі. Це дозволяє уникати застрягання в локальних оптимумах та забезпечує різноманітність у дослідженні простору рішень.
* Метаевристичні алгоритми зазвичай легко реалізуються і можуть працювати на практиці в реальних умовах, навіть для складних завдань.

До метаевристичних алгоритмів можна віднести алгоритми ройового інтелекту. Ідеї ​​поведінки запозичено від природи, а особливо, від біологічних систем. Кожен агент слідує простим правилам. В системі відсутній будь-який центр керування поведінкою, що вказує кожному з агентів на те, що слід робити, але локальні і, в деякій мірі, випадкові взаємодії призводять до виникнення інтелектуальної глобальної поведінки, що не контролюється окремими агентами.

* **Алгоритм рою частинок** (Particle Swarm Optimization, PSO). Імітація узгодженої поведінки птахів у зграї.
* **Алгоритм мурашиної колонії** (Ant Colony Optimization, ACO). Імітація самоорганізації мурах у виборі найкоротших шляхів.
* **Алгоритм бджолиної сім'ї** (Artificial Bee Colony, ABC). Імітує поведінку бджіл у пошуку та повідомленні про джерело нектару.
* **Алгоритм пошуку зграєю вовків** (Wolf Pack Search, WPS). Відтворює поведінку зграї вовків у пошуку здобичі.
* **Алгоритм кажанів** (Bat Algorithm, BA). Імітує процес пошуку їжі кажанами, враховуючи їх здатність до ехолокації.
* **Алгоритм світлячків** (Firefly Algorithm, FFA). Використовує властивість світіння світлячків, що є засобом комунікації між ними.
* **Алгоритм мавп** (Monkey Search, MS). Відтворює поведінку мавп, що шукають їжу на деревах.
* **Алгоритм пошуку зозуль** (Cuckoo Search Algorithm, CSA). Імітує гніздовий паразитизм деяких видів зозуль, що підкладають свої яйця в гнізда інших птахів.
* **Алгоритм зграї риб** (Fish School Search, FSS). Імітація поведінки зграї риб в реальних умовах: пошук їжі та розмноження.

Більшість алгоритмів ройового інтелекту розроблено для вирішення завдань оптимізації зі змінними, що мають набір параметрів, які необхідно підбирати при вирішенні різних завдань. Вони знаходять застосування в багатьох областях, таких як оптимізація розкладів, розв'язання задач комівояжера, параметричний пошук, оптимізація в машинному навчанні та інших задачах, де потрібне знаходження оптимальних рішень в умовах обмежених ресурсів.

#### Основні терміни метаевристичних алгоритмів

В метаевристичних алгоритмах для назви членів популяції використовується термін агент (agent). В різних алгоритмах агентів позначають термінами – особини, індивіди тощо.

Загальна схема метаевристичних алгоритмів містить наступні етапи:

1. **Ініціалізація популяції.** В області пошуку в різний спосіб розміщується певна кількість початкових точок для пошуку рішення задачі - ініціалізація популяції агентів.
2. **Міграція агентів популяції.** За допомогою певного набору операторів переміщення, специфічних для кожного з популяційних алгоритмів, агенти в області пошуку переміщаються таким чином, щоб в кінцевому результаті наблизитися до шуканого екстремуму цільової функції.
3. **Завершення пошуку.** Перевіряється виконання умов закінчення ітерацій, і якщо їх виконано, обчислення завершуються, приймаючи краще із знайдених станів агентів популяції за наближене рішення задачі. Якщо зазначені умови не виконано, етап 2 повторюється.

При ініціалізації популяції можуть бути використані детерміновані і випадкові алгоритми. Істотно скоротити час вирішення завдання може формування початкової популяції, агенти якої знаходяться поблизу глобального оптимуму цільової функції. Однак, зазвичай, апріорна інформація про місцезнаходження цього екстремуму є відсутньою, тому, агентів початкової популяції розподіляють рівномірно по всій області пошуку.

Як умову завершення пошуку використовують, як правило, досягнення заданого числа ітерацій (поколінь). Часто використовують також умову стагнації (stagnation) алгоритму, коли краще досягнуте значення цільової функції не змінюється протягом заданого числа ітерацій. Можуть бути використані й інші умови, наприклад, умова вичерпання часу, що відведено на вирішення завдання.

Більшість популяційних алгоритмів мають яскраво виражену модульну структуру, що дозволяє легко отримати велику кількість варіантів кожного з алгоритмів шляхом комбінування правил ініціалізації популяції, операторів переміщення і умов завершення пошуку.

Агентам популяції притаманні основні властивості.

* **Автономність** - агенти пересуваються в просторі пошуку, зазвичай, незалежно один від одного.
* **Стохастичність** - процес пересування агентів містить випадкову компоненту.
* **Обмеженість уявлення** - кожен з агентів популяції має інформацію лише про частину області пошуку, в якій він просувається і, можливо, про оточення деяких інших агентів.
* **Децентралізація** - відсутність агентів, які керують процесом пошуку в цілому.
* **Комунікабельність** - агенти в різний спосіб можуть обмінюватися між собою інформацією про топологію (ландшафт) поверхні станів цільової функції, що виявлено в процесі дослідження своєї частини області пошуку.

Навіть, якщо стратегія поведінки кожного з агентів популяції досить проста, зазначені властивості агентів забезпечують формування колективного інтелекту, який проявляється в самоорганізації і складній поведінці популяції в цілому.

## 6.2. Алгоритм рою частинок

Наочним прикладом колективної поведінки тварин є зграя птахів. Літаючи великими групами, вони майже ніколи не стикаються в повітрі. Зграя рухається плавно і скоординовано, немов нею хтось керує. Якщо розсипати корм, то за кілька хвилин злітаються всі птахи в окрузі.

В багатьох видах птахів немає ватажків, що віддають накази, провідну роль грає колективний інтелект. Птахи в зграї діють відповідно до визначених простих правил. Кружляючи в небі, кожен з птахів стежить за своїми родичами і координує свій рух згідно їх положення, а знайшовши джерело їжі, він сповіщає про це.

Причини такого «альтруїзму» птахів були предметом дослідження багатьох соціобіологів. Одним з найбільш популярних пояснень є те, що переваги від такої поведінки для кожної особини є більшими за очевидні недоліки (необхідність боротьби за знайдену їжу з іншими особинами).

Джерела їжі, зазвичай, розташовані випадковим чином, тому, на самоті птах цілком може загинути, не знайшовши їжі протягом довгого часу. Однак, якщо всі птахи будуть «грати за правилами», повідомляючи родичам інформацію про знахідки, то шанси кожного з них на виживання різко підвищуються. Таким чином, будучи невигідною для окремої істоти, така стратегія є запорукою ефективності зграї і виду в цілому.

#### Класичний алгоритм рою частинок

Метою алгоритму є вирішення задач глобальної пошукової оптимізації. Алгоритм моделює багатоагентну систему, де агенти - частинки просуваються до оптимальних рішень, обмінюючись при цьому інформацією з іншими частинками – «сусідами».

Агенти алгоритму (частинки) в рої ставлять для себе три досить простих завдання:

* Всі агенти повинні уникати перетинання з оточуючими агентами.
* Кожний агент має коригувати свою швидкість відповідно до швидкостей оточуючих агентів.
* Кожен агент повинен намагатися зберігати досить малу відстань між собою і оточуючими агентами.

Алгоритм рою частинок є ітеративним процесом, який постійно перебуває в зміні (рис.6.1).

Створення рою частинок

Знаходження рішення для кожної частинки

Знаходження кращого рішення серед всіх частинок

Корекція швидкості кожної частинки

Переміщення кожної частинки

Чи виконано

критерій станову

Вивід результату

ні

Рис. 6.1. Класичний алгоритм рою частинок

Область пошуку представлено у вигляді багатовимірного простору з агентами. Спочатку всі агенти знаходяться в випадкових місцях простору і з випадковим вектором швидкості. В кожній з точок, яку частинка відвідує, вона обчислює задану функцію і фіксує оптимальне значення шуканої функції. Імітується миттєвий обмін інформацією між птахами.

Всі частинки знають місце розташування найкращого результату пошуку у всьому рої і з кожної ітерацією агенти коригують вектори своїх швидкостей та їх напрямки, намагаючись наблизитися до найкращої точки рою і при цьому бути ближче до свого індивідуального максимуму. При цьому постійно відбувається обчислення шуканої функції і пошук найкращого значення.

Характеристиками частинки є її координати в просторі рішень, а також вектор швидкості пересування. Кожна частинка зберігає координати кращого рішення, що знайдено за весь час пошуку, а також краще рішення зі всіх рішень, пройдених всіма частинками (рис.6.2).

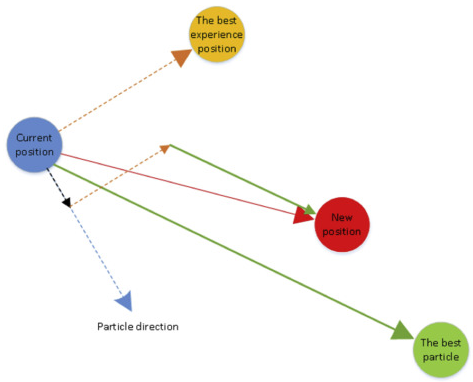


Рис. 6.2. Ітерації алгоритму рою частинок

На кожній ітерації алгоритму напрям і довжина вектору швидкості кожної частинки змінюються відповідно до відомостей про знайдені оптимуми.

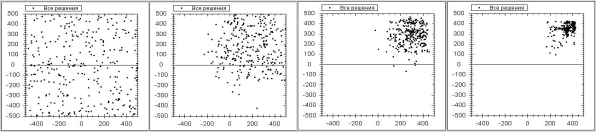


Рис. 6.3. Наочний приклад роботи рою частинок

Спочатку алгоритм рою частинок застосовувався для досліджень соціальних психологів, але найбільше поширення цей алгоритм зміг отримати в машинному навчанні при рішеннях завдань оптимізації.

* **Переваги.** Невисока складність в реалізації алгоритму. Ефективний для глобальної оптимізації.
* **Застосування.** Машинне навчання. Оптимізація функцій багатьох параметрів, форм, розмірів і топологій. Область проектування. Біоінженерія, біомеханіка, біохімія.
* **Розвиток.** Використання як багатоагентної обчислювальної системи. Можливості додавання інших, більш складних методів ройового інтелекту.

## 6.3. Алгоритми колективної поведінки мурашиної колонії

Мурашиний алгоритм — один з ефективних алгоритмів для знаходження наближених розв'язків задачі комівояжера, а також аналогічних завдань пошуку маршрутів на графах. Суть підходу полягає в аналізі та використанні моделі поведінки мурах, що шукають найкоротші шляхи від колонії до їжі.

Основу мурашиних алгоритмів оптимізації складає імітація самоорганізації мурашиної колонії. Колонія мурах може розглядатися як багатоагентна система, в якій кожен агент (мураха) функціонує автономно за дуже простими правилами. Але не зважаючи на примітивну поведінку агентів, поведінка всієї системи є розумною та ефективною.

Окрема мураха 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, не узгоджуються з успішністю мурах як виду. Вони існують на планеті понад 100 млн років, будують величезні житла та забезпечують себе всім необхідним.

Домогтися таких успіхів мурахі здатні завдяки своїй соціальності. Вони живуть лише в колективах – колоніях і повинні строго взаємодіяти за певними правилами, і тоді колонія цілком буде ефективною.

Колонія є повністю самоорганізованою і не має домінуючих особин, які роздають вказівки і координують дії. Кожна з мурах володіє інформацією тільки про локальну обстановку і не має уявлення про всю ситуації в цілому - тільки про те, що дізналася сама або явно чи неявно від своїх родичів. На неявних взаємодіях мурах засновано механізми пошуку найкоротшого шляху від мурашника до джерела їжі (рис.6.3).

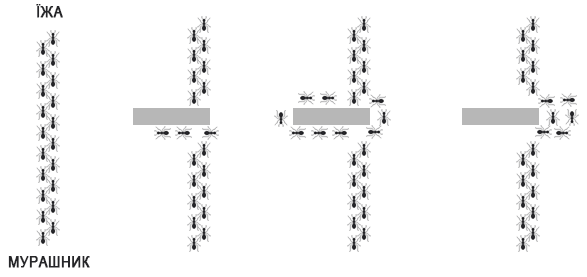


Рис.6.3. Наочний приклад поведінки мурах при доланні перешкод

На рисунку показано випадок, коли на шляху пересування мурах до їжі виникає перешкода. У цьому випадку потрібно прокласти новий оптимальний шлях. Після того, як мурахі дійшли до перешкоди, вони з рівною ймовірністю будуть її обходити як зліва так й справа. Кожен раз, проходячи від мурашника до джерела їжі і назад, мурахі залишають за собою доріжку феромонів.

Інші мурахі, відчувши такі сліди на землі, будуть інстинктивно скеровуватися до джерела їжі. Мурахі, які обрали коротший шлях, будуть його проходити швидше (туди/назад), і за кілька проходжень шлях буде більш насичений феромоном. Чим більше мурах проходить за певним шляхом і залишає феромон, тим привабливішим він стає для їх родичів. Але феромон має властивість й випаровуватися, тобто на довшій ділянці, по який пройде менша кількість мурах концентрація феромону буде значно меншою.

Така поведінка в мурашиній колонії має корисні властивості, що можна використати для моделювання

**Випадковість.** Для пошуку їжі мурахі орієнтуються на різні варіанти побудови свого шляху і на першому етапі напрямок руху обирають випадковим чином. Навіть, коли щільність феромону допомагає вибрати оптимальний (найкоротший) маршрут, залишаються "недовірливі особи", які не залишають спроби знайти більш оптимальні маршрути.

**Багаторазовість.** Безумовно, з одною або кількома мурахами навряд чи вийде здійснити вибір оптимального шляху. Потрібно багато спроб знайти найкоротший шлях або хоча б попередити інших, що не потрібно рухатися в цьому напрямку. В якості такого попередження виступає феромон і позитивний зворотний зв'язок.

**Позитивний зворотний зв'язок.** У зоологів це називається стигмерією (stigmergy). Стигмерія - це механізм непрямої координації, через навколишнє середовище між агентами чи діями, коли один суб'єкт взаємодії змінює деяку частину навколишнього середовища, а інші використовують інформацію про цей стан пізніше, коли знаходяться в околі даної частини. Таким чином, позитивний зворотний зв'язок це певна "колективна пам'ять" на основі феромону. Використовуючи цю "пам'ять" (на основі спроб і помилок) можна знайти правильне рішення.

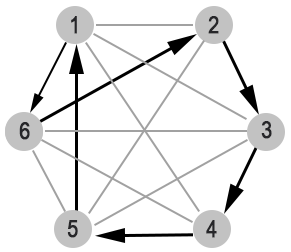
Приклади стигмерії:

* Бджоли залишають феромони у вуликах, щоб вказати іншим бджолам, які завдання потрібно виконати. Наприклад, вони можуть залишити сліди про необхідність зібрати більше нектару.
* Мурахи залишають феромонові сліди, щоб позначати шляхи до джерел їжі. Це допомагає іншим мурахам слідувати цим шляхам і збирати їжу.
* У робототехніці стигмерія використовується для розробки алгоритмів колективної поведінки роботів. Роботи можуть залишати сліди (наприклад, маркери) на поверхні та спілкуватися з іншими роботами на основі цих слідів.

**Негативний зворотний зв'язок.** Згодом феромон випаровується, що дозволяє мурахам адаптувати свою поведінку під зміни зовнішнього середовища. Розподіл феромону по простору пересування мурах є динамічною змінною глобальної пам'яті мурашника. Будь-яка мураха в фіксований момент часу може сприймати і змінювати лише одну локальну комірку цієї глобальної пам'яті.

**Цільова функція.** Найважливішим елементом алгоритму є цільова функція (фітнес-функція), яку потрібно оптимізувати (для мурахі це короткий маршрут). Але можна з успіхом вирішувати і інші подібні завдання (складання розкладу тощо) Кінцевою метою алгоритму є оптимізація (пошук глобального або прийнятного максимуму чи мінімуму). Алгоритм є евристичним, тобто не гарантує точного рішення, а тільки наближене або прийнятне.

### Класичний мурашиний алгоритм

Мурашиний алгоритм моделює систему з простими агентами, яким для виконання дій потрібно невелику кількість пам'яті та нескладні обчислення. В основі алгоритму лежить поведінка мурашиної колонії — маркування доріжок певною кількістю феромону, який може з часом або накопичуватися або випаровуватися.

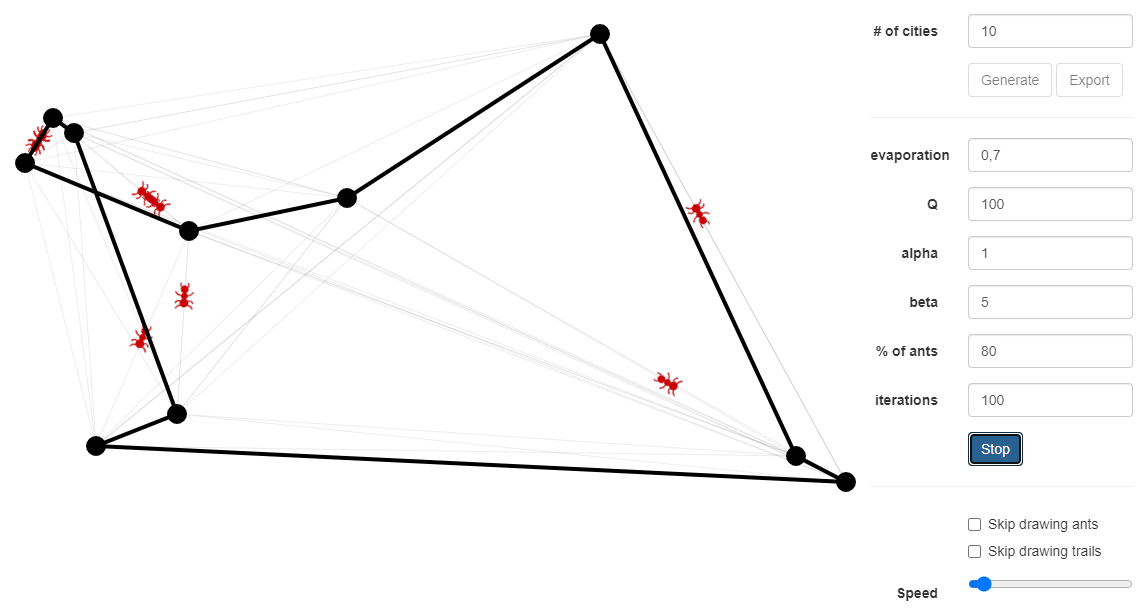
Наочним поясненням мурашиного алгоритму є завдання пошуку найкоротшого шляху у задачі комівояжера.

Робота починається з розміщення мурашок у вершинах графа, потім починається рух мурашок, що мають пройти всі вершини в різний спосіб. Кожна мураха зберігає в пам'яті список пройдених їм вузлів (список заборон). Вибираючи вузол для наступного кроку, мураха «пам'ятає» про вже пройдені вузли і не розглядає їх як можливі для переходу. На кожному кроці список заборон поповнюється новим вузлом.

Крім списку заборон при виборі вузла для переходу мураха керується «привабливістю» ребер, які вона може пройти. Привабливість залежить, по-перше, від відстані між вузлами (вага ребра), а по-друге, від слідів феромонів, які залишили мурахі, що пройшли 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 мурахі, що проходять цим шляхом, навпаки, посилюють їх.

Перед новою ітерацією алгоритму, тобто перед тим, як мураха знову проходить вершини, список заборон очищується і мураха може знов обрати попередній шлях, ґрунтуючись на його привабливість. Результати однієї ітерації звісно не нададуть хорошого рішення, проте, повторення алгоритму може видавати досить точний результат.

Наочний приклад мурашиного алгоритму при пошуку найкоротшого шляху можна спостерігати у демо-програмі Ant System (<https://courses.cs.ut.ee/demos/visual-aco/#/visualisation>).



Ant System застосовано до задачі комівояжера, де мурахи використовуються для пошуку найкоротшого шляху між набором міст, де кожне місто відвідується лише один раз. Основними компонентами алгоритму є встановлення та оновлення рівня феромону сліду, а також вибір наступного міста для відвідування для кожної мурахи.

Мурашині алгоритми використовують для складних комбінаторних задач, таких як задача комівояжера, маршрутизація, оптимізація, календарне планування тощо. Особливо ефективними мурашині алгоритми є при динамічній оптимізації процесів в розподілених нестаціонарних системах, наприклад, трафіки у телекомунікаційних мережах.

## 6.4. Алгоритм бджолиної колонії

Алгоритм бджолиної колонії (бджолиний алгоритм) - алгоритм ройового інтелекту, що заснований на імітації поведінки бджіл. Незважаючи на колективний характер і різні схеми цієї поведінки, окрема бджола здатна виконувати різні добре організовані складні дії.

Для збору нектару в бджолиної колонії застосовується два види бджіл: бджоли-розвідники і робочі бджоли. Перші проводять дослідження території навколо вулика на предмет наявності нектару. Після повернення у вулик, бджоли-розвідники повідомляють інформацію про кількість нектару, напрямок та відстані його розташування.

В привабливі ділянки вилітають робочі бджоли, причому, чим більше нектару в даній ділянці, тим більше туди вилітає бджіл. Крім збору меду, в їх завдання входить оновлення інформації про дану та прилеглі ділянки.

Бджоли-розвідники шукають в радіусі польоту область з найвищою щільністю квітів, як нові джерела нектару. Без жодного уявлення про поле апріорі, бджоли починають пошук квітів з випадкових позицій з випадковими векторами швидкості. Кожна бджола може пам'ятати позиції, де вона знайшла найбільшу кількість квітів і порівнювати знайдені джерела найбільшої щільності квітів з іншими, які виявили інші бджоли.

Вибираючи між поверненням до місця, де бджола сама виявила найбільшу кількість квітів, або дослідженням місця, визначеного іншими, як місце з найбільшою кількістю квітів, бджола спрямовується в напрямку між двома точками в залежності від того, що надасть більший вплив на її рішення — персональний спогад або соціальний рефлекс.

Бджоли перевіряють місця, які пролетіли, порівнюючи з знайденими раніше розвідниками місцями сподіваючись знайти кращі поля. В підсумку, бджола закінчує рух на місці поля з найбільшою концентрацією квітів. Інформація до інших бджіл передається всередині вулика, за допомогою «бджолиного танцю». Незабаром всі робочі бджоли сім'ї зосереджується навколо цієї позиції, що позначається як нове місце найбільшого скупчення квітів.

Подібну поведінку бджіл покладено в основу цього методу оптимізації.

### Класичний бджолиний алгоритм

Штучна колонія використовує алгоритм схожий з видобутком нектару медоносними бджолами. Замість поля з квітами розглядається область рішень. Замість нектару використовуються критерії задачі оптимізації та цільова функція.

На кожній ітерації алгоритму вибирається nb областей з кращим значенням цільової функції, вони називаються «кращими», з решти вибирається ще ng кращих, що називаються «перспективними». Можна задати певну мінімальну відстань між двома сусідніми ділянками. В цьому випадку, при виникненні нашарування, ділянка з гіршим значенням цільової функції видаляється. Замість неї вибирається інша ділянка. Дані ділянки запам'ятовуються і при наступній ітерації до них надсилається певна кількість бджіл.

Схему описаного алгоритму представлено ​​на рис. 6.4.

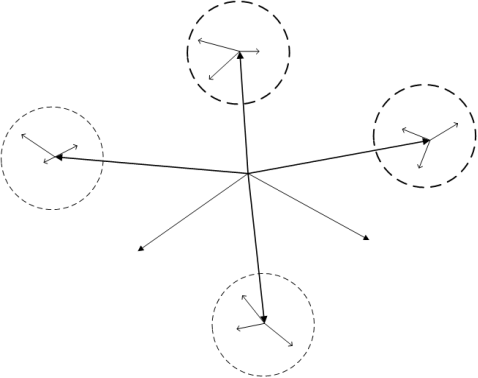


Рис. 6.4. Схематичне зображення стратегії бджолиного алгоритму

Кроки алгоритму рою бджіл

* Створення початкової популяції штучних бджіл (рішень). Кожна бджола є потенційним розв'язанням задачі оптимізації. При ініціалізації для n розвідників генеруються початкові положення, у найпростішому випадку використовується метод випадкового перебору.
* Обчислення значення цільової функції кожної бджоли. Цільова функція визначається залежно від конкретного завдання оптимізації.
* Розвідка шукачами їжі (Employed Bees phase). Бджоли-розвідники шукають нові рішення навколо своїх поточних позицій. Вони можуть виконувати локальні пошуки або випадкові переміщення у просторі рішень.
* Вибір шукачів їжі (Onlooker Bees phase). Робочі бджоли вибирають бджіл-розвідників, ґрунтуючись на якості їх рішень. Робочі бджоли вибирають рішення з вищими значеннями цільової функції для подальшого вивчення.
* Після формування списків кращих і перспективних областей, в їх околи відправляються робочі бджоли. Робочі бджоли виконують локальний пошук навколо вибраних рішень для уточнення їхньої якості.
* В деяких варіантах алгоритму число робочих бджіл залежить від якості області, з точки зору цільової функції. Ця залежність може бути лінійною або визначатися за більш складними правилами.
* Якщо бджола-розвідник не знаходить кращого рішення протягом деякого числа ітерацій, вона стає "зниклою" і замінюється новою випадковою бджолою, щоб забезпечити різноманітність у популяції.
* Перевірка досягнення заданої умови зупинки: досягнення максимальної кількості ітерацій або досягнення необхідної точності рішення.
* Повернення оптимального рішення, знайденого в результаті виконання алгоритму Artificial Bee Colony.

В даному алгоритмі використовується кілька параметрів: кількість розвідників, кількість кращих і перспективних ділянок, радіус локальної розвідки, кількість бджіл для кожного класу області, мінімально можлива відстань між сусідніми областями. Якість одержуваних рішень і швидкість роботи алгоритму значно залежить від вибору даних параметрів.

Алгоритм рою бджіл можна ефективно розподілити на кілька паралельних процесів, за рахунок чого значно збільшується його швидкість. Бджолині алгоритми підтверджують свою ефективність в якості методів випадкового пошуку для оптимізації управління. До недоліків даного методу варто віднести велике число вільних параметрів, від значення яких часто залежить результат, з іншого боку, відсутні підстави для вибору цих значень.

## 6.5. Алгоритм сірих вовків

Для вовків типовим є сімейний спосіб життя: вони живуть зграями - сімейними групами, що складаються з пари ватажків, їх родичів, а також одиноких вовків. Всередині зграї спостерігається строга ієрархія, на вершині якої знаходиться ватажок зграї - альфа-вовк, що обирається зграєю.

* **Альфа-вовк** є домінуючим вовком, оскільки його наказам слід дотримуватися всім вовкам. Альфа не обов'язково є найсильнішим учасником зграї, але найкращим з точки зору управління, оскільки організація та дисципліна в угрупуванні є значно важливішою, ніж його сила. Рішення Альфи продиктовано зграєю, проте, спостерігається і певна демократична поведінка, в якій альфа слідує за іншими вовками.
* **Бета** - це підлеглі вовки, які допомагають альфа в процесі прийняття рішень або інших заходів. Бета-вовк командує вовками нижчого рівня, доглядає дисципліну в зграї та здійснює зворотний зв'язок з альфа.
* **Дельта-вовки** повинні підкорятися альфам і бетам.
* **Омега-вовки** завжди повинні підкорятися всім домінуючим вовкам і допомагають підтримувати структуру домінування.

Крім соціальної ієрархії, цікавою поведінкою сірих вовків є групове полювання. Альфа-вовк скеровує інших вовків на пошук здобичі. Вовки «досліджують» місцевість на її наявність, якщо хтось відчує запах жертви, починається її пошук. Чим сильніше відчувається запах, тим ближче вовки до жертви. Таким чином, вони переміщаються в напрямку посилення запаху здобичі. Вовки поділяються на невеликі групи, і кожна група здійснює пошук в певному напрямку, відмінному від напрямків інших груп. У підсумку, коли один з вовків знайде жертву, він подає сигнал ватажкові і іншим, щоб поділитися здобиччю з вовками зі зграї.

Основними етапами полювання є:

* Відслідковування, переслідування та наближення до здобичі.
* Оточити та переслідувати здобич, поки вона не припине чинити опір.
* Напад на здобич.

### Класичний алгоритм зграї вовків

Суть алгоритму побудовано на основі моделі полювання групи вовків. Вважається, що до вовка, котрий знаходиться найближче до здобичі підтягуються інші вовки, створюючи кільце. Надалі інші вовки знову пересуваються до вовка, який знаходиться ще ближче до цілі. Процес пересування проходить доки вовки не зберуться у групу. Коли дану ціль буде досягнуто – це і буде оптимальна відстань до здійснення атаки на здобич (рис.6.5).

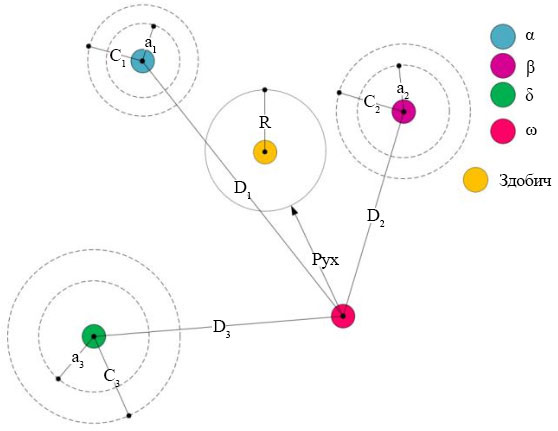


Рис. 6.5. Схематичне зображення стратегії алгоритму зграї вовків

На рисунку представлено етап оточення жертви. Основні положення роботи даного алгоритму полягають в наступному. Передбачається існування чотирьох типів сірих вовків за ієрархією, таких як Альфа - лідери зграї, Бета - помічники Альфа, Дельта - старійшини та дозорні і Омега-все решта. Три основних етапи полювання: пошук, оточення і напад на жертву, що виконуються для виконання оптимізації. Процес перелаштування відбувається доки вовки не збираються в зграю, це і буде оптимальним напрямком для атаки з мінімальною відстанню до здобичі (рис.6.6).

Ініціалізація вовків

Знаходження головних 3 вовків

Обчислення координат переміщення до цих 3 вовків

Обчислення нових координат для вовків

Вивід координат альфа-вовка

Кількість повторень дорівнює макс. кількості повторень?

Ні

Так

Рис. 6.6. Кроки алгоритму зграї вовків

Передбачається, що Альфа (кращий кандидат рішення), Бета і Дельта мають більш чітке представлення про потенційне знаходження здобичі. За допомогою розроблених математичних моделей, пошукові агенти (Омега-вовки) оновлюють свої позиції відповідно до Альфа, Бета і Дельта-вовків в N-вимірному просторі пошуку. Остаточна позиція буде у випадковому місці всередині кола, який визначається положеннями Альфа, Бета і Дельта в просторі пошуку. Іншими словами Альфа, Бета і Дельта оцінюють стан здобичі, і інші вовки оновлюють свої позиції випадковим чином навколо жертви.

* **Переваги.** Гнучкий та простий для реалізації алгоритм.
* **Застосування.** Проблеми оптимізації процесів. Вирішення векторних задач.
* **Розвиток.** Розроблено спеціальні математичні моделі, що дозволяють імітувати процес поведінки зграї: соціальну ієрархію, стеження, оточення і атаку.

## 6.6. Алгоритм кажанів

До особливостей поведінки кажанів (летючих мишей) під час полювання відносять спроможності активно збирати інформацію і дуже швидко приймати рішення. Для орієнтації в просторі кажани використовують ехолокацію - вони випускають ультразвук, який досягає жертви і відбивається назад. Більшість видів кажанів володіє досконалими засобами ехолокації, які використовуються ними для виявлення здобичі і перешкод, а також для забезпечення можливості розміститися на поверхні в темряві.

Завдяки ехолокації кажан розуміє, де є жертва і атакує її. Кажан приймає адекватне рішення і витрачає на це лише близько 100 мілісекунд. Здатність так швидко приймати рішення на основі отриманої інформації із зовнішнього середовища розвинулася в кажанів в процесі еволюції і стала їх перевагою (рис.6.7).

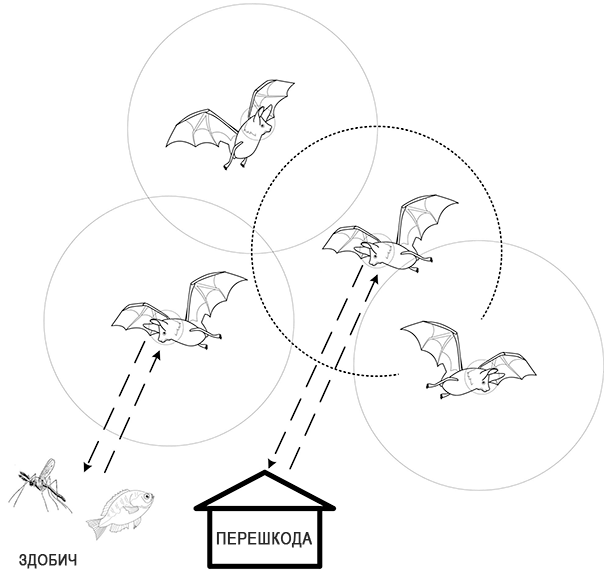


Рис. 6.7. Схематичне зображення стратегії алгоритму кажанів

#### Основні етапи алгоритму кажанів:

* Створення початкової популяції кажанів. Кожний кажан представляється вектором параметрів, що визначає її становище у пошуковому просторі.
* Обчислення значення цільової функції для кожної кажана. Цільова функція може бути визначена залежно від конкретної оптимізації задачі.
* Кажани оновлюють свої позиції в пошуковому просторі на основі поточних положень та швидкостей. Це дозволяє їм досліджувати нові області та наближатися до оптимальних рішень.
* Кажани генерують звукові імпульси, які допомагають їм орієнтуватися в пошуковому просторі і залучати інших кажанів. Все кажани використовують ехолокацію, щоб аналізувати відстань, а також відрізняти їжу (здобич) і природні перешкоди.
* Частота та гучність звукових імпульсів регулюються залежно від успішності пошуку. Кажани з кращими результатами мають велику ймовірність залучити інших кажанів до своєї позиції.
* Кажани можуть тимчасово зосередитися на локальному пошуку навколо найкращих рішень, щоб уточнити їх точність.
* Перевірка досягнення заданої умови зупинки: досягнення максимальної кількості ітерацій або досягнення необхідної точності рішення.
* Повернення оптимального рішення, знайденого в результаті виконання алгоритму Bat Algorithm.

Однією з основних переваг алгоритму кажанів є швидкість виконання і потенційно більша потужність, ніж в алгоритмі рою частинок. Алгоритм може здатися складнішим, ніж більшість інших алгоритмів ройового інтелекту, проте, він може бути досить ефективно застосований до проблем оптимізації і надавати добрі результати, витрачаючи меншу кількість часу.

Алгоритм Bat Algorithm може бути модифікований та налаштований залежно від конкретного завдання оптимізації. Він широко використовується для вирішення різних задач оптимізації, включаючи функціональну оптимізацію, завдання машинного навчання та інші.

## 6.7. Алгоритм світлячків

Алгоритм світлячків імітує природну поведінку світлячків, що випромінюють світло, яке є механізмом комунікації між особинами. З його допомогою вони приваблюють осіб протилежної статі, повідомляють про наближення хижаків тощо. Менш яскраві світлячки пересуваються до більш яскравих. Якщо світлячок не бачить більш яскравого представника рою, він переміщається хаотично (рис.6.8).

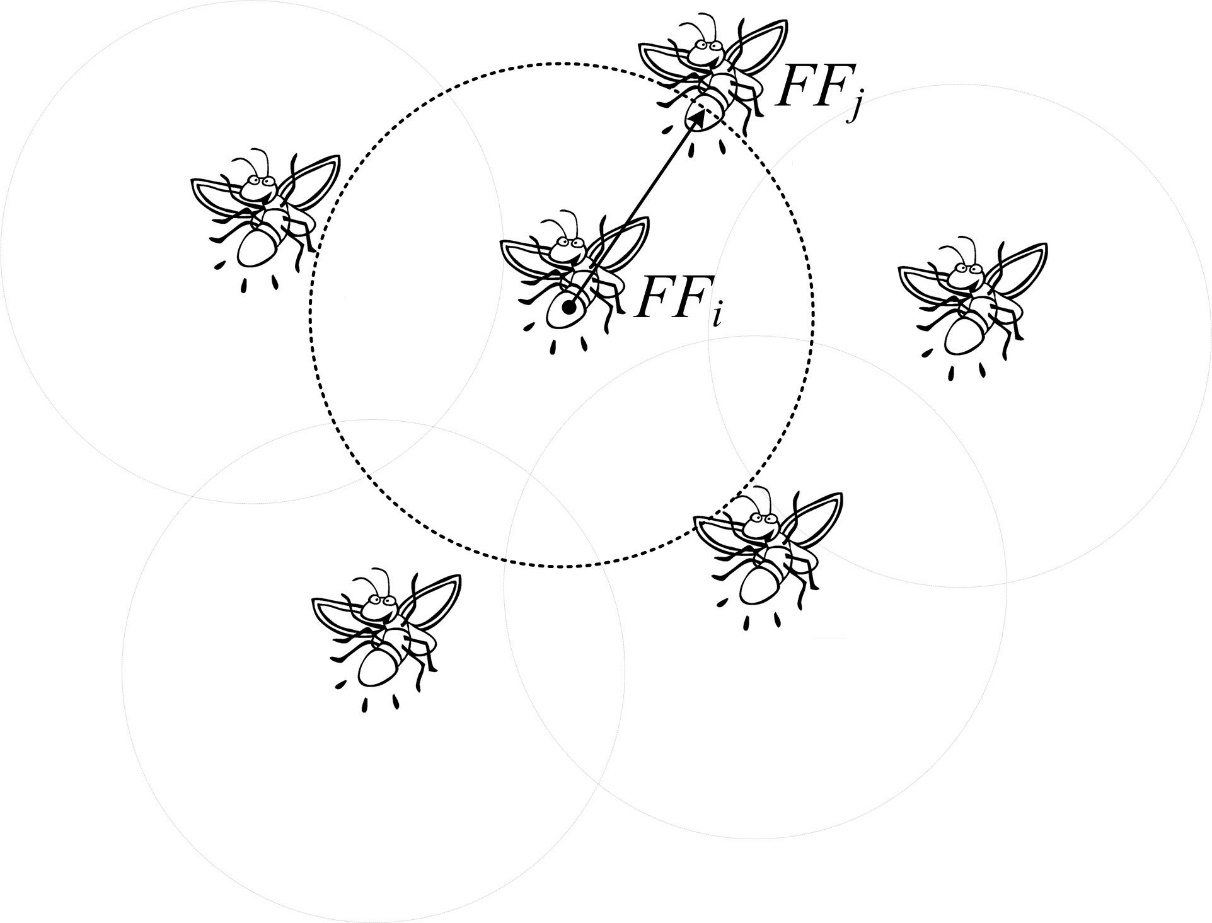


Рис. 6.8. Схематичне зображення стратегії алгоритму світлячків

#### Основні ідеї та принципи роботи алгоритму світлячків:

* Кожен світлячок є потенційним розв'язанням задачі оптимізації і має певне положення в просторі рішень.
* Яскравість світлячка відповідає якості рішення. У задачах оптимізації, які потрібно мінімізувати, світлячки з більш високою яскравістю позначають найкращі рішення.
* Світлячки рухаються до яскравіших світлячків у просторі. Положення світлячків оновлюється на кожній ітерації залежно від яскравості та відстані між ними.
* Інтенсивність світла (яскравість) зменшується зі збільшенням відстані між світлячками. Це означає, що світлячки стають менш яскравими, коли знаходяться далеко один від одного. Якщо світлячок не бачить біля себе світлячка більш яскравого, ніж він сам, то він переміщається випадковим чином.
* Алгоритм містить елемент випадковості у переміщенні світлячків, що допомагає уникати застрягання в локальних оптимумах.

Переваги алгоритму світлячків полягають у його здатності до знаходження оптимальних рішень у багатовимірних просторах та його відносну простоту реалізації. Він може застосовуватись у різних задачах оптимізації, включаючи завдання параметричної оптимізації та функціональної оптимізації. Розроблено багато модифікацій алгоритму світлячків для вирішення багатьох практичних завдань.

## 6.8. Мавпячий алгоритм

Алгоритм мавп (Monkey Search) – це метаевристичний алгоритм оптимізації, який натхненний поведінкою мавп у процесі пошуку їжі. Алгоритм моделює поведінку мавп, яка лазить по деревах у пошуках їжі, чим вище дерево – тим більше їжі. Зі свого поточного положення кожна з мавп рухається вгору доки не досягне вершини дерева. Потім мавпа робить серію локальних стрибків у випадковому напрямку в надії знайти більш високу гілку, і рух вгору повторюється.

Після виконання деякого числа підйомів і локальних стрибків мавпа вважає, що в достатній мірі дослідила ландшафт в околиці свого початкового положення. Для того щоб обстежити нову область простору пошуку, мавпа виконує довгий глобальний стрибок і перестрибує на інше дерево.

Зазначені дії повторюються задану кількість разів. Рішенням завдання буде найвища з вершин, знайдених даною популяцією мавп.

#### Схему алгоритму можна представити в наступному вигляді:

* Створення початкової популяції мавп. Кожна мавпа представляється вектором параметрів, що визначає її становище у пошуковому просторі.
* Обчислення значення цільової функції для кожної мавпи. Цільова функція може бути визначена залежно від конкретної оптимізації задачі.
* Мавпи оновлюють свої позиції у пошуковому просторі на основі поточних станів та швидкостей. Це дозволяє їм досліджувати нові області та наближатися до оптимальних рішень.
* Мавпи виконують випадкові рухи, щоб досліджувати ширші області пошукового простору.
* Запам’ятовування найкращого рішення, знайденим на поточній ітерації, та оновлення його значення за потреби. Оскільки мавпячий алгоритм не гарантує, що краще рішення буде отримано на останній ітерації, кожен раз необхідно зберігати поточне краще рішення.
* Мавпи можуть тимчасово зосередитись на локальному пошуку навколо найкращих рішень, щоб уточнити їхню точність.
* Перевірка досягнення заданої умови зупинки: досягнення максимальної кількості ітерацій або досягнення необхідної точності рішення.
* Повернення оптимального рішення, знайденого в результаті виконання алгоритму Monkey Search.

Алгоритм Monkey Search може бути модифікований та налаштований залежно від конкретного завдання оптимізації. Він може застосовуватися до різних завдань оптимізації, таких як пошук параметрів, навчання нейронних мереж, аналіз даних та інші.

## 6.9. Алгоритм пошуку зозулі

Алгоритм, що імітує поведінку зозуль під час відкладання яєць, а саме процесу гніздового паразитизму. Цілий ряд видів зозуль підкладає свої яйця в гнізда інших птахів як свого виду, так і, часто, інших видів. Зозулі можуть викидати яйця конкурентів, щоб збільшити ймовірність вилуплення їх власних пташенят.

Інші птахи можуть конфліктувати з вторгненням зозуль, тобто часом таке «вторгнення» зустрічає відсіч у деяких птахів. Наприклад, якщо господар гнізда виявить в ньому яйця іншого виду, то він або викине ці яйця, або просто покине дане гніздо і спорудить інше на новому місці.

В алгоритмі пошуку зозуль кожне яйце в гнізді є рішенням, в той час як яйце, що належить до зозулі, представляє нове рішення. Мета полягає в використанні нових і потенційно кращих рішень (зозулиних), щоб замінити менш гарні рішення в гніздах. У найпростішому варіанті алгоритму в кожному гнізді по одному яйцю (рис.6.10).

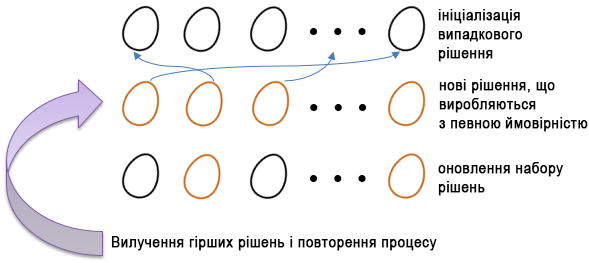


Рис. 6.10. Наочне представлення алгоритму пошуку зозулі

Нехай задано деяку цільову функцію f (x), яку необхідно максимізувати. В основу евристики закладено наступні правила.

1. Кожна зозуля відкладає одне яйце за один раз в випадково обране гніздо.
2. Кращі гнізда з яйцями «високої якості» (тобто з кращими значеннями цільової функції) переходять в наступне покоління.
3. Яйце зозулі, що відкладене у чуже гніздо, може бути виявлено господарем гнізда з певною ймовірністю і видалено з гнізда.

ініціалізація популяції хазяйських гнізд H випадковим чином

ініціалізація зозулі cuckoo випадковим чином

визначення кращого знайденого рішення GBest

виконати переміщення зозулі

визначити нові координати зозулі

випадковим чином вибрати гніздо Hi

f (cuckoo)> f (Hi)

замінюємо яйце в гнізді на яйце зозулі

з ймовірністю Ра видаляємо деяке число гірших гнізд

створення нових гнізд замість видалених

оновлення значення GBest

Чи виконано критерій зупинки?

Виведення GBest

Ні

Ні

Рис. 6.11. Схема алгоритму пошуку зозуль.

#### Переваги алгоритму зозулі

* Алгоритм простий у реалізації та може бути ефективним для вирішення різноманітних завдань оптимізації.
* Завдяки випадковим переміщенням зозуль та їх здатності відкладати яйця в кращих гніздах, алгоритм зазвичай може уникати застрягання в локальних мінімумах.
* Алгоритм може адаптуватися до різних видів оптимізації і може бути налаштований з використанням різних параметрів для поліпшення його продуктивності.

В цілому, алгоритм Cuckoo Search є цікавим методом для вирішення задач оптимізації, який може давати хороші результати в різних сценаріях, але вимагає правильного налаштування та оцінки його застосовності для конкретних завдань.

## 6.10. Алгоритм зграї риб

Зграєю риб (Fish School) називають скупчення риб, які пересуваються приблизно з однієї швидкістю і орієнтацією, підтримуючи приблизно постійну відстань між собою. Пошук їжі, міграція та боротьба з небезпеками відбувається в формі соціальної поведінки. Взаємодія між всіма рибами в групі призводить до інтелектуальної соціальної поведінки. Об'єднання риб грають важливу роль в підвищенні ефективності пошуку їжі, захисті від хижаків, а також у зменшенні енергетичних витрат.

В природі риба може виявити більш поживну область шляхом індивідуального пошуку або слідуванням за іншою рибою. Площа, яку покриває велика кількість риб, зазвичай, є найбільш поживною. Навколишнє середовище, в якому живе риба, в першу чергу є простором рішень і станом для інших риб. Наступна поведінка риби залежить від її поточного стану і локального стану навколишнього середовища (включаючи якість прийнятих рішень в даний час і стану сусідніх компаньйонів). При цьому риба впливає на навколишнє середовище через свою діяльність і діяльність своїх компаньйонів.

Оператори алгоритму зграї риб об'єднані в дві групи:

а) оператор харчування, формалізує успішність дослідження агентами тих чи інших областей акваріума.

б) оператори пересування, що реалізують алгоритми міграції агентів.

В алгоритмі риби пересуваються в акваріумі (області пошуку) в пошуках їжі (рішення задачі оптимізації). Вага кожної риби формалізує її індивідуальний успіх в пошуку рішення і відіграє роль її пам'яті. Саме наявність ваги у агентів популяції є головною особливістю парадигми у порівнянні, наприклад, з парадигмою оптимізації роєм частинок. Ця особливість алгоритму зграї риб дозволяє відмовитися від необхідності відшукувати і фіксувати глобально кращі рішення, як це робиться в алгоритмі рою частинок.

#### Основні ідеї та принципи роботи алгоритму "Fish School":

* В алгоритмі кожна риба є потенційним розв'язанням задачі оптимізації. Разом вони утворюють зграю риб, яка досліджує простір рішень.
* Риби в зграї мають певну поведінку, включаючи рух у просторі рішень, взаємодію з іншими рибами та оновлення свого становища на основі досвіду.
* Кожна риба має свою поточну позицію у просторі рішень. Вони можуть переміщатися та змінювати свою позицію з метою покращення якості рішення.
* Риби взаємодіють між собою, обмінюючись інформацією щодо якості рішень. Це допомагає їм спільно досліджувати простір рішень та швидше знаходити оптимальні рішення.
* Алгоритм "Fish School" може адаптуватися до змінних умов завдання, включаючи динамічні зміни в просторі рішень.

Переваги алгоритму "Fish School" включають його здатність знаходити оптимальні рішення у багатовимірних та складних просторах рішень, а також його здатність до спільного дослідження простору рішень багатьма агентами. Він може використовуватись у різних задачах оптимізації, включаючи завдання функціональної оптимізації та комбінаторної оптимізації.

## Висновки

Зацікавленість у біонічному напрямку пов'язана з досить привабливими перспективами використання «патентів природи» в науково-технічній практиці. Напрямок із застосуванням ідей біологічних механізмів та принципів їх функціонування сприяє створенню нових алгоритмів роботи технічних пристроїв.

Біоніка внесла в сучасну науку і техніку широке використання біологічних аналогій. Саме в біонічному напрямку отримані найбільш цікаві та ефективні алгоритми випадкового пошуку. Вони моделюють різні рівні біологічної організації - від популяційного, пов'язаного з властивостями біологічної еволюції, до субклітинного, де найтонші регуляторні механізми управління клітиною надають багато цікавих ідей для синтезу алгоритмів випадкового пошуку.

## Контрольні питання

1. Які прикладні завдання можна вирішити із застосуванням ройових алгоритмів?
2. Які переваги притаманні природнім алгоритмам у порівнянні з традиційними алгоритмами вирішення оптимізаційних задач.
3. Назвіть основні кроки в класичному алгоритмі рою частинок.
4. Що називають найкращою локальною та глобальною позицією в ройових алгоритмах?
5. Яку особливість випадкового пошуку закладено у бджолиному алгоритмі?
6. Назвіть основні етапи реалізації бджолиного алгоритму.
7. Назвіть основні типи особин в алгоритмі зграї вовків.
8. Які особливості кажанів та світляків покладено в основу однойменних алгоритмів?
9. Назвіть основні принципи, що покладено в основу алгоритму зозулі.
10. Назвіть сфери, в яких природні алгоритми будуть ефективними.

## Використані джерела

1. Еволюційний алгоритм - <http://ru.knowledgr.com/00136704/ЭволюционныйАлгоритм>
2. Алгоритм рою частинок - <https://habr.com/ru/post/105639/>
3. Алгоритм рою частинок. Опис та реалізації на мовах Python и C# - <https://jenyay.net/Programming/ParticleSwarm>
4. Природні алгоритми. Алгоритм поведінки бджолиного рою - <https://habr.com/ru/post/104055/>
5. Бджолиний алгоритм для оптимізації функції - <https://jenyay.net/Programming/Bees>
6. Алгоритм зозулі - <https://uk.wikipedia.org/wiki/Алгоритм_зозулі>
7. Популяційний алгоритм, що заснований на поведінці зграї риб - <https://habr.com/ru/post/178309/>
8. Алгоритм світляка - <http://ru.knowledgr.com/08799926/АлгоритмСветлячка>
9. Основи штучного інтелекту - https://fkti5301.github.io/exam\_tickets\_ai\_2018\_novakova/