

Еволюційні алгоритми

Визначення

Еволюційні алгоритми — напрям в штучному інтелекті (розділ еволюційного моделювання), що використовує і моделює біологічну еволюцію. Розрізняють різні алгоритми: генетичні алгоритми, еволюційне програмування, еволюційні стратегії, системи класифікаторів, генетичне програмування тощо. Всі вони моделюють базові положення в теорії біологічної еволюції — процеси відбору, мутації і відтворення. Поведінка агентів визначається довкіллям. Множину агентів прийнято називати популяцією. Така популяція еволюціонує відповідно до правил відбору відповідно до цільової функції, що задається довкіллям. Таким чином, кожному агентові (індивідуумові) популяції призначається значення його придатності в довкіллі. Розмножуються лише найпридатніші види. Рекомбінація і мутація дозволяють агентам змінюватись і пристосовуватися до середовища. Такі алгоритми відносяться до адаптивних пошукових

Класифікація алгоритмів

Системи, які використовують лише еволюційні принципи. Вони успішно використовувалися для завдань виду функціональної оптимізації і можуть легко бути описані на математичній мові. До них відносяться еволюційні алгоритми, такі як еволюційне програмування, **генетичні алгоритми**, еволюційні стратегії.

Системи, які є біологічно реалістичніші, але які не виявилися корисними в прикладному сенсі. Вони більше схожі на біологічні системи і менш направлені на вирішення технічних завдань. Вони володіють складною і цікавою поведінкою, і, мабуть, незабаром отримають практичне вживання. До цих систем відносять так зване штучне життя.

Приклади

Генетичні алгоритми

Мурашині алгоритми

Еволюційні стратегії

Еволюційне програмування

Генетичне програмування

Еволюційні методи призначені для пошуку бажаних рішень і засновані на статистичному підході до дослідження ситуацій та ітераційному наближенні системи до шуканого стану. На відміну від точних методів математичного програмування еволюційні методи дозволяють знаходити рішення, близькі до оптимальних, за прийнятний час, а на відміну від відомих евристичних методів оптимізації характеризуються істотно меншою залежністю від особливостей додатку (більш універсальні) і в багатьох випадках забезпечують кращу ступінь наближення до оптимального рішення.

Переваги еволюційних алгоритмів

Широка область застосування.

Можливість проблемно - орієнтованого кодування рішень, підбору початкових умов, комбінування еволюційних обчислень з не еволюційними алгоритмами, продовження процесу еволюції до тих пір, поки є необхідні ресурси.

Придатність для пошуку в складному просторі рішень великої розмірності.

Відсутність обмежень на вид цільової функції.

Ясність схеми і базових принципів еволюційних обчислень.

Інтегрованість еволюційних обчислень з іншими неklasичними парадигмами штучного інтелекту, такими як штучні нейромережі та нечітка логіка.

Недоліки еволюційних алгоритмів

Евристичний характер еволюційних обчислень не гарантує оптимальності отриманого рішення (правда, на практиці, найчастіше, важливо за заданий час отримати одне або кілька субоптимальних альтернативних рішень, тим більше, що вихідні дані в задачі можуть динамічно змінюватися, бути неточними або неповними).

Відносно висока обчислювальна трудомісткість, яка проте долається за рахунок розпаралелювання на рівні організації еволюційних обчислень і на рівні їх безпосередньої реалізації в обчислювальній системі.

Відносно невисока ефективність на заключних фазах моделювання еволюції (оператори пошуку в еволюційних алгоритмах не орієнтовані на швидке попадання в локальний оптимум).

Невирішеність питань самоадаптації.

Генетичний алгоритм. Історія.

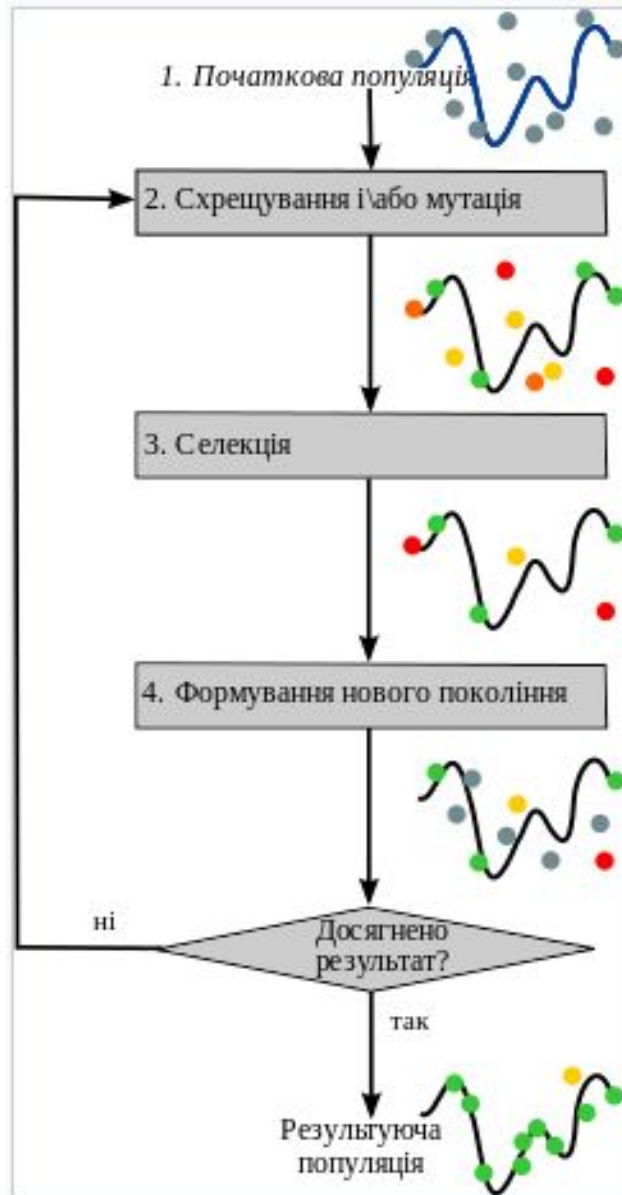
Перші спроби симуляції еволюції були проведені у 1954 році Нільсом Баричеллі на комп'ютері, встановленому в Інституті перспективних досліджень Принстонського університету. Його робота, що була опублікована у тому ж році, привернула увагу громадськості. З 1957 року, австралійський генетик Алекс Фразер опублікував серію робіт з симуляції штучного відбору серед організмів з множинним контролем вимірюваних характеристик. Це дозволило комп'ютерній симуляції еволюційних процесів та методам, які були описані у книгах Фразера та Барнела(1970) та Кросбі(1975), з 1960-х років стати більш розповсюдженим видом діяльності серед біологів. Симуляції Фразера містили усі найважливіші елементи сучасних генетичних алгоритмів. До того ж, Ганс-Іоахім Бремерман в 1960-х опублікував серію робіт, які також приймали підхід використання популяції рішень, що піддаються відбору, мутації та рекомбінації, в проблемах оптимізації. Дослідження Бремермана також містили елементи сучасних генетичних алгоритмів. Також варто відмітити Річарда

Генетичний алгоритм

Задача кодується таким чином, щоб її вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи, допущені до схрещення (*селекція*). До осіб застосовується «генетичні оператори» (в більшості випадків це оператор схрещення (crossover) і оператор мутації (mutation)), створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (*поколінь*), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

- знаходження глобального, або надоптимального вирішення;
- вичерпання числа поколінь, що відпущені на еволюцію;
- вичерпання часу, відпущеного на еволюцію.

Етапи генетичного алгоритму



Створення початкової популяції:
Обчислення функції пристосованості для осіб популяції (оцінювання)
Повторювання до виконання критерію зупинки алгоритму:
Вибір індивідів із поточної популяції (селекція)
Схрещення або/та мутація
Обчислення функції пристосовуваності для всіх осіб
Формування нового покоління

Застосування ГА

Оптимізація функцій

Оптимізація запитів в базах даних

Різноманітні задачі на графах (задача комівояжера, розфарбування)

Налаштування і навчання штучної нейронної мережі

Задачі компоновки

Створення розкладів

Ігрові стратегії

Апроксимація функцій

Штучне життя

біоінформатика (згортання білків)

Завдання на практичну роботу

Вихідні дані

ch₁, = [111001100101]
ch₂ = [001100111010]
ch₃ = [011101110011]
ch₄ = [001000101000]
ch₅ = [010001100100]
ch₆ = [010011000101]
ch₇ = [101011011011]
ch₈ = [000010111100]

Мета



Chx=[111111111111]