# Тема 5. Природні алгоритми

Всі алгоритми, що натхненні живою природою відносяться до класу евристичних алгоритмів (Heuristic Algorithms), тобто алгоритмів, для яких збіжність до глобального вирішення не є доведеною, але експериментально встановлено, що в більшості випадків вони дають досить гарне рішення.

Дослідження, що відбуваються у цьому напрямку можна умовно поділити на

* **Природні моделі**. Це системи, які відтворюють біологічні популяції або системи і не є корисними в прикладному сенсі. Природні моделі відтворюють розвиток біологічних системи, мають складну поведінку і зазвичай не використовуються для вирішення технічних завдань. До цих систем відносять так зване «штучне життя».
* **Природні алгоритми**. Це системи, які використовують та моделюють біологічні принципи діяльності та розвитку популяції. Вони успішно використовуються для завдань евристичного пошуку, оптимізації і можуть легко бути описані математичною мовою. Дані інтелектуальні алгоритми часто називають популяційними. Популярними для вирішення різного типу завдань є:
  + Еволюційні алгоритми (генетичний алгоритм та еволюційні стратегії).
  + Алгоритми ройового інтелекту (алгоритм рою частинок, мурашиний алгоритм, бджолиний алгоритм тощо)
  + Алгоритми колективної діяльності людей (наприклад, алгоритм еволюції розуму).
  + Інші алгоритми, що запозичені з живої та неживої природи.

## 5.1. Постановка задачі оптимізації

У найпростішому випадку задача оптимізації полягає у знаходженні екстремуму (мінімуму або максимуму) цільової функції шляхом систематичного перебору вхідних значень з дозволеного набору та обчислення її значення.

Y= f(x1, x2, ..., xn)

Y називається *цільовою функцією*, що залежить від параметрів(x1, x2, ..., xn) і в залежності від задачі прагне до максимального чи мінімального значення.

Для більшої наочності можна навести приклад великого підприємства, що виробляє певні товари і пов’язане з багатьма іншими підприємствами (постачальники сировини чи комплектуючих, збут продукції, маркетинг, гуртовий продаж, партнери). Діяльність підприємства залежить від багатьох факторів (параметрів), які можуть змінюватися в широкому або обмеженому діапазоні.

Нехай, ставиться завдання збільшити дохід компанії і знайти ті чинники і їх значення, що цьому сприятимуть. У цьому випадку цільовою функцією буде «Дохід компанії», що прямо чи опосередковано залежить від багатьох параметрів, таких як число працівників компанії, обсяг виробництва, витрати на рекламу, ціни на кінцеві продукти тощо. Ці параметри пов'язані між собою - наприклад, при зменшенні числа працівників швидше за все впаде й обсяг виробництва.

В даній постановці завданням є знайти таку комбінацію значень вхідних параметрів, щоб цільова функція була як можна максимальною.

Деякі задачі оптимізації (з невеликою кількістю параметрів та діапазону їх змін) легко розв’язати, але є і такі, точне рішення яких знайти практично неможливо (знайти максимальне значення цільової функції, що залежить від значної кількості параметрів з широким діапазоном значень).

Відомо два основні шляхи вирішення таких задач - переборний та градієнтний.

* **Переборний метод.** Для пошуку оптимального рішення (максимум цільової функції) потрібно послідовно обчислити значення функції у всіх точках. Метод є простим, але потребує великої кількості обчислень і часу.
* **Градієнтний спуск.** Обираються випадкові значення параметрів, які поступово змінюються, досягаючи найбільшої швидкості зросту цільової функції. Алгоритм може зупинитись, досягнувши локального оптимуму (максимуму чи мінімуму). Градієнтні методи швидкі, але не гарантують оптимального рішення (оскільки цільова функція має як глобальний оптимум так і декілька локальних оптимумів).

Ефективним способом вирішення задач оптимізації є природні алгоритми, що представляють комбінацію переборного та градієнтного методів. Тобто, якщо на деякій множині задано складну функцію від багатьох змінних, тоді природні алгоритми є програмами, які за визначений час знаходять точку, де значення функції знаходиться достатньо близько до максимально можливого значення, це буде одним з кращих рішень.

Для кращого розуміння природніх алгоритмів потрібно роз'яснити поняття поверхні станів цільової функції. Кожному значенню параметрів кількістю N відповідає один вимір в багатовимірному просторі. N+1-ий вимір відповідає значенню цільової функції. Для різноманітних сполучень значень параметрів відповідне значення цільової функції можна зобразити точкою в N+1-вимірному просторі. Ці точки утворюють деяку поверхню - поверхню станів. Метою використання природніх алгоритмів є знаходження на багатовимірній поверхні найвищої або найнижчої точки (оптимуму), що відповідатиме значенню цільової функції.

Поверхня станів має складну будову і досить неприємні властивості, зокрема, наявність локальних оптимумів (точки, найнижчі в своєму певному околі, але вищі від глобального оптимуму), пласкі ділянки, сідлові точки і довгі вузькі яри. Аналітичними засобами неможливо визначити розташування глобального оптимуму на поверхні станів, тому робота природніх алгоритмів полягає в дослідженні цієї поверхні.

Відштовхуючись від початкової конфігурації параметрів (від випадково обраних точок на поверхні), алгоритми поступово відшукують глобальний оптимум (максимум чи мінімум) цільової функції. Обчислюється вектор напрямку та розмір кроків просування по поверхні з заданої точки (рис.5.1).

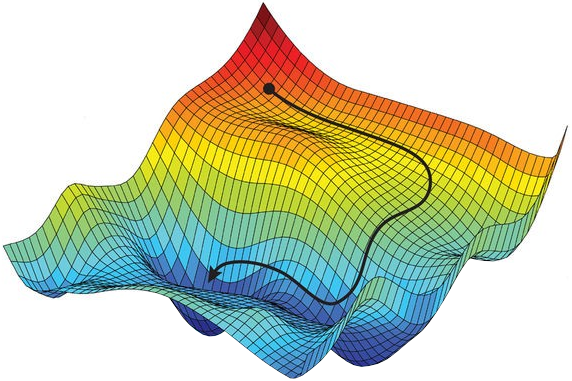


Рис.5.1. Наочний приклад пошуку глобального оптимуму

Алгоритми діють ітеративне, на кожній ітерації перевизначається знайдене значення цільової функції, що порівнюється з попередніми значеннями і обчислюється похибка. Значення похибки використовують для корекції певних коефіцієнтів (напрямок і довжина кроків просування по поверхні), і дії повторюються.

Робота алгоритму припиняється або коли пройдена визначена кількість ітерацій, або коли похибка досягає визначеного рівня мінімуму, або коли похибка перестає зменшуватись. Користувач переважно сам вибирає потрібний критерій припинення обчислень. Зрештою алгоритм зупиняється в точці оптимуму, що може виявитись лише локальним оптимумом (в ідеальному випадку - глобальним).

Складність в алгоритмах полягає у виборі довжини кроків. При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути рішення, або просуватися в невірному напрямку. При маленьких кроках, правильний напрямок буде виявлено, але зростає кількість ітерацій. На практиці розмір кроків береться пропорційним до крутизні схилу з деякою константою - швидкістю навчання. Правильний вибір швидкості навчання залежить від конкретної задачі і здійснюється дослідним шляхом. Ця константа може також залежати від часу, зменшуючись в міру просування алгоритму.

## 5.2. Еволюційні алгоритми

Еволюційні алгоритми – це загальна назва алгоритмів пошуку, оптимізації або навчання, що засновані на принципах природного еволюційного процесу. Вони призначені для ітераційного пошуку бажаних рішень. На відміну від точних математичних методів еволюційні методи дозволяють за прийнятний час знаходити рішення, наближені до оптимальних та характеризуються меншою залежністю від особливостей завдання.

До них відносяться:

* Генетичні алгоритми.
* Еволюційні стратегії.
* Генетичне програмування.
* Еволюційне програмування.

## 5.3. Генетичний алгоритм

Генетичний алгоритм - це евристичний алгоритм пошуку, який застосовується для вирішення завдань оптимізації, що нагадує біологічну еволюцію.

Еволюція - це процес постійної оптимізації біологічних видів. Еволюційна теорія стверджує, що життя на планеті виникло спочатку лише в найпростіших формах - у вигляді одноклітинних організмів. Ці форми поступово ускладнювалися, пристосовуючись до навколишнього середовища, породжуючи нові види, і через багато мільйонів років з'явилися перші тварини і люди. Кожен біологічний вид з часом вдосконалює свої якості, щоб ефективно справлятися з важливими задачами виживання, самозахисту, розмноження. Таким чином виникло захисне забарвлення в багатьох риб і комах, панцир у черепахи, отрута в скорпіона тощо.

За допомогою еволюції природа постійно оптимізує живі організми, знаходячи часом неординарні рішення, ґрунтуючись лише на двох біологічних механізмах - природного відбору і генетичного спадкування.

Природний відбір гарантує, що найбільш пристосовані особини надають більшу кількість нащадків, а завдяки генетичному спадкуванню частина нащадків не лише збереже високу пристосованість батьків, але буде мати і деякі нові властивості. Якщо нові властивості виявляються корисними, то з великою імовірністю вони перейдуть до наступного покоління. Таким чином, відбувається нагромадження корисних якостей і поступове підвищення пристосованості біологічного виду в цілому. Знаючи, як вирішується задача оптимізації видів у природі, можна застосувати схожий метод для вирішення реальних завдань.

Для ясного розуміння спадковості, потрібно дещо поглибитися в побудову природної клітини - у світ генів і хромосом. Майже в кожній клітині будь-якої особини є набір хромосом, що містять інформацію про цю особину. Основна частина хромосоми - нитка ДНК, яка визначає, які хімічні реакції будуть відбуватися в даній клітині, як вона буде розвиватися і які функції виконувати (рис.5.2).

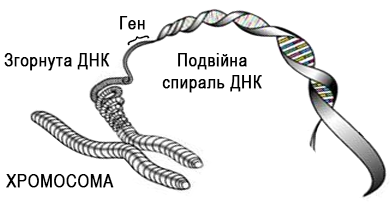


Рис.5.2. Спрощене представлення складу хромосоми

В будь якому організмі присутні два види клітин: статеві (такі, як сперматозоїд і яйцеклітина) і соматичні.

Соматичні клітини містять 23 пари (46 хромосом): 22 пари нестатевих хромосом і одна пара статевих хромосом (Х і Y хромосоми). В кожній парі одна з хромосом отримана від батька, а друга - від матері (рис.5.3).

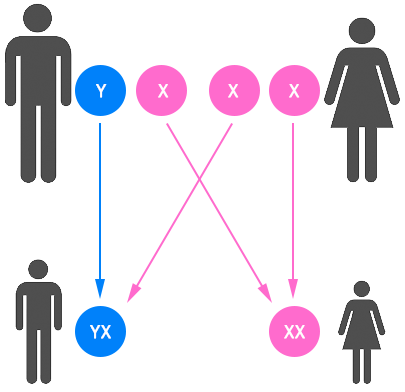


Рис. 5.3. Процес наслідування хромосом

В статевих клітинах хромосом лише 23, і вони непарні. При заплідненні відбувається об’єднання чоловічої і жіночої статевих клітин і утворюється клітина зародка, що містить саме 46 хромосом. Які властивості нащадок одержить від батька, а які - від матері? Це залежить від того, які саме статеві клітини брали участь у заплідненні.

При статевому розмноженні людини (мейоз) зливаються дві окремі статеві клітини, які продукують соматичну клітину. Чоловічі сперматозоїди містять один з двох типів статевих хромосом X або Y. Жіночі яйцеклітини містять тільки Х-хромосому. В цьому випадку клітина сперми визначає стать індивідуума. Якщо сперматозоїдна клітина, яка містить Х-хромосому, запліднює яйцеклітину, результуючою соматичною клітиною XX - жіноча стать. Якщо клітина сперми містить Y-хромосому, тоді результуючою буде XY - чоловіча стать.

Ген - це частина хромосоми (ланка ДНК), який відповідає за певну властивість особини, наприклад за колір очей, тип волосся, колір шкіри тощо. Вся сукупність генетичних ознак людини кодується за допомогою приблизно 60 тис. генів.

Парні хромосоми відповідають за однакові ознаки - наприклад, батьківська хромосома може містити ген чорного кольору око, а парна їй материнська - ген голубого кольору. Існують визначені закони, що керують участю тих чи інших генів у розвитку особини. Зокрема, у даному прикладі нащадок буде чорнооким, оскільки ген блакитних очей є "слабшим" і подавляється геном будь-якого іншого кольору.

Процес вироблення соматичних клітин в організмі піддається змінам, завдяки яким нащадки відрізняються від своїх батьків. Зокрема, відбувається наступне: парні хромосоми соматичної клітини зближаються впритул, потім їхні нитки ДНК розриваються в кількох випадкових місцях і хромосоми обмінюються своїми частинами (рис.5.4).

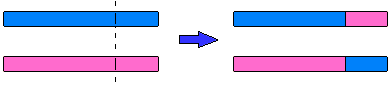


Рис.5.4. Спрощене представлення процесу схрещування

Цей процес забезпечує появу нових варіантів хромосом і зветься "кросовер" (crossover, crossingover або схрещування). Кожна з нових хромосом з'являється потім всередині однієї з статевих клітин, і її генетична інформація може реалізуватись в нащадках даної особини.

Важливим фактор, що впливає на спадковість є мутації, які виражаються в зміні деяких ділянок ДНК. Мутації є випадковими і можуть бути викликані різними зовнішніми факторами. Якщо мутація відбулася в статевій клітині, то змінений ген може передатися до нащадку і проявитися у вигляді нових властивостей. Саме мутації є причиною появи нових біологічних видів, а кросовер визначає вже змінність всередині виду.

### Загальна ідея генетичного алгоритму

Генетичний алгоритм є методом багатовимірної оптимізації, тобто методом пошуку оптимуму багатовимірної функції. Потенційно цей метод можна використовувати для глобальної оптимізації, але з цим виникають складності. Суть методу полягає в моделюванні еволюційного процесу: є певна популяція (набір векторів), яка розмножується, на яку впливають мутації і проводиться природний відбір на підставі оптимізації цільової функції.

Отже, перш за все популяція повинна розмножуватися. Основний принцип розмноження - нащадок схожий на своїх батьків. Тобто, потрібно задати механізм успадкування, що містить елемент випадковості. Швидкість розвитку таких систем дуже низька – генетична різноманітність зменшується, популяція вироджується, значення функції перестає мінімізуватися.

Для вирішення цієї проблеми використовується механізм мутації, який полягає в випадковій зміні випадкових особин. Цей механізм дозволяє ввести щось нове в генетичну різноманітність.

Наступний важливий механізм - селекція - відбір особин, які краще оптимізують функцію. Зазвичай, відбирають стільки особин, скільки було до розмноження, щоб в кожній епосі була постійна кількість особин в популяції. Також, прийнято відбирати випадкових особин, які, можливо, погано оптимізують функцію, але зате вносять різноманітність в наступні покоління.

Цих трьох механізмів може забракнути, щоб мінімізувати функцію. Так популяція вироджується - рано чи пізно популяція може застрягнути в локальному оптимумі. Коли таке відбувається, проводять процес, званий струсом (в природі аналогії - глобальні катаклізми), коли знищується майже вся популяція, і вводяться нові (випадкові) особини.

Це опис класичного генетичного алгоритму, він простий в реалізації і є місце для фантазії і досліджень.

### Практичний приклад генетичного алгоритму

Для того, щоб говорити про генетичне спадкування, потрібно наділити особини хромосомами. В генетичному алгоритмі хромосома - це числовий вектор набору параметрів, що сприяють вирішенню задачі. Які саме вектори варто розглядати в конкретній задачі, вирішує користувач. Кожна з позицій вектору хромосоми називається геном і відповідає за певний вхідний параметр.

Наприклад, в завданні знайти найкращі рішення для збільшення прибутків від діяльності компанії цільовою функцією буде «Дохід компанії», що прагне до максимуму і залежить від певних чинників

Дохід компанії = f («собівартість продукції», «витратні матеріали», «заробітна платня», «амортизаційні витрати», …)

В термінах генетичного алгоритму «Дохід компанії» є хромосомою. Чинники, що впливають на цільову функцію будуть розглядатися як гени («собівартість продукції», «витратні матеріали», «заробітна платня», «амортизаційні витрати»). Тоді хромосоми «Дохід компанії» можна представити як набір генів. Кожен ген, що відповідає за певний чинник може містити різні значення.

1. Хромосома 1 – 15, 200, 5000, 100
2. Хромосома n – 12, 260, 3800, 112

Зазвичай для спрощення обчислень в генетичних алгоритмах кожен ген представлено у двійковому коді

1. Хромосома 1 – 10001111000 00001001000 10001111100 11101111011
2. Хромосома n – 10101101100 01101001011 10110011100 11101001000

В загальному роботу генетичного алгоритму можна описати основними етапами (рис.5.5).

Схрещування

Початкова популяція

Мутація

Селекція

Формування нового

покоління

Чи досягнуто

Результат ?

Результуюча популяція

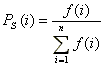
так

ні

Рис. 5.5. Генетичний алгоритм

1. Генерація початкової популяції (Ініціалізація), що складається з особин, вибраних випадково чином або згенерованою будь-яким іншим способом.

2. Кожне рішення із популяції оцінюється з використанням фітнес-функції Ps(i), яка дорівнює відношенню її пристосованості до сумарної пристосованості популяції. Функція може враховувати різні аспекти, що залежать від конкретної оптимізації задачі. Наприклад, завдання оптимізації планування маршрутів, фітнес-функція може враховувати загальну довжину маршруту, час доставки тощо.



3. Відповідно до величини фітнес-функції Ps(і) відбувається відбір найсильніших n особин для подальшої генетичної обробки. Члени популяції з високою пристосованістю будуть вибиратися з більшою імовірністю, ніж особини з низькою пристосованістю.

4. Застосовуються генетичні оператори, такі як схрещування (кросовер) та мутація до обраних рішень, щоб створити нове покоління.

5. Для кожної пари застосовується схрещування (кросовер). Кросовер- це операція, при якій із двох хромосом породжується кілька нових хромосом. Для одноточкового кросоверу випадковим образом вибирається точка розриву (ділянка між сусідніми ланками в рядку). Обидві батьківські структури розриваються на два сегменти в цій точці. Потім, відповідні сегменти різних батьків склеюються і виходять два генотипи нащадків.

Наприклад, припустимо, один батько складається з 10 нулів, а інший - з 10 одиниць. Нехай з 9 можливих точок розриву обрано точку 3. Нижче представлено батьківські особини та утворені їхні нащадки.

**Батько 1** 0000000000 000~0000000--> 111~0000000 1110000000 **Нащадок 1**

**Батько 2** 1111111111 111~1111111 --> 000~1111111 0001111111 **Нащадок 2**

6. Відповідно до алгоритму випадково обрані особини піддаються мутації. Мутація - це перетворення хромосоми, яке випадково змінює одну чи декілька її позицій (генів). Поширеним видом мутацій є випадкова зміна лише одного гену з хромосоми. В особині, що піддається мутації, випадковий біт з імовірністю Pm змінюється на протилежний.

7. В утвореній популяції відбувається обчислення пристосованості особин (п.1) і обираються найсильніші. Отримана популяція заміняє поточну і цикл одного покоління завершується.

Кроки 2-6 повторюються протягом кількох поколінь або до досягнення критерію зупинки, що залежить від конкретного завдання.

Наступні покоління обробляються подібним чином: відбір, кросовер і мутація. В кожному наступному поколінні продукуються нові рішення, серед яких будуть як погані, так і хороші, але завдяки відбору число прийнятних рішень буде зростати.

Мета полягає в тому, щоб ГА знайшов найкраще рішення (або наближення до нього) у просторі можливих рішень, максимізуючи чи мінімізуючи значення фітнес-функції, залежно від поставленого завдання оптимізації. Фітнес-функція є одним із ключових елементів, який визначає, які рішення будуть просуватися вперед у процесі еволюції, а які будуть відсіяні.

Робота генетичного алгоритму є ітераційним процесом, що продовжується, доки не виконається задане число поколінь або інший критерій зупинки. Таким критерієм може бути:

* Знаходження глобального або кращого локального рішення.
* Вичерпання числа поколінь, що відведено на еволюцію.
* Вичерпання часу, що відведено на еволюцію.

### Модифікації

Дослідники генетичних алгоритмів пропонують багато інших операторів відбору, кросоверу і мутації. Наприклад:

**Двохточковий і рівномірний кросовер** – альтернативи для одноточкового оператора. В двохточковому кросовері вибираються дві точки розриву, і батьківські хромосоми обмінюються сегментом, що знаходиться між двома цими точками. У рівномірному кросинговері, кожен біт першого батька успадковується першим нащадком із заданою імовірністю; у противному випадку цей біт передається до другого нащадку. І навпаки.

#### Недоліки у порівнянні з іншими методами оптимізації:

* Пошук оптимального рішення для складної задачі високої розмірності часто вимагає великих витрат. В реальних задачах обчислення можуть тривати від кількох годин до кількох днів.
* Генетичні алгоритми погано масштабуються під складність вирішуваної проблеми. При збільшенні області пошуку рішень збільшується число елементів, що піддаються до обробки.
* Умови зупинки алгоритму є різними для кожної проблеми.
* В багатьох завданнях генетичні алгоритми мають тенденцію сходитися до локального оптимуму замість глобального оптимуму для даної задачі.

## 5.4. Еволюційні стратегії

Серед стратегій, що використовуються в сучасних генетичних алгоритмах, можна виділити стратегії елітізма, різноманіття, "свіжої крові", зміни розміру популяції, паралельні еволюції (міграції, турнірна).

Самою поширеною є **стратегія елітизму,** коли на всіх етапах еволюційного процесу зберігається елітна група з кращих особин популяції. Вважається, що краща особина містить корисний генетичний матеріал, який може бути використаний для пошуку оптимуму. Таким чином, використання стратегії елітізму запобігає можливості видалення з популяції кращих особин при формуванні нового покоління.

Розмір елітної групи визначається особливостями алгоритму і залежить, як правило, від загальної чисельності популяції. В процесі роботи алгоритму потрібно стежити, щоб особини елітної групи не виродились в однакові або подібні генотопи.

Використання **стратегії різноманіття** вирішує проблему утворення в популяції множини однакових або подібних осіб. Ці особини звужують область пошуку рішення та приводять до більш сильного виродження наступних поколінь.

Технічно дана стратегія реалізується на кожному етапі еволюційного процесу попарним оцінюванням ступеня близькості всіх особин. Якщо між кількома особинами є подібність, яка оцінюється нижче за деяке задане порогове значення, з них залишається тільки одна особина (краща). Вакантні місця заповнюються нащадками, що отримані на поточній стадії еволюції або, за необхідності, новими випадковими рішеннями.

**Стратегія "свіжої крові"** полягає у введенні в нове покоління популяції певної кількості нових, випадково отриманих особин замість групи батьківських. Процедура може виконуватися систематично з заданим інтервалом між поколіннями або випадково на будь-якій стадії еволюції. Стратегія «свіжого крові» не повинна вступати в протиріччя з стратегією елітізма, тому, серед вилучених батьківських особин не повинно бути елітних.

**Стратегія зміни розміру популяції** передбачає нестабільність кількості особин, що беруть участь у процесі на кожній стадії еволюції. Кількість особин змінюється залежно від якісних характеристик популяції в цілому. Цими характеристиками можуть бути середня пристосованість або її зміна протягом певної кількості стадій еволюції.

Наприклад, при підвищенні середньої пристосованості популяції або динаміки її зміни об'єм популяції зростає. Відповідно, при погіршенні - зменшується. Незалежно від правила зміни розміру популяції, повинні бути встановлені нижня і верхня межі, а також початкова кількість особин.

**Стратегії паралельних еволюцій** (міграційна, турнірна) скеровані на формування незалежних або обмежено залежних груп, всередині яких протікає еволюційний процес. В кожній групі можуть застосовуватися власні налаштування генетичного алгоритму, що може сприяти знаходженню кращого рішення. В подальшому відбувається перенесення генетичної інформації з однієї групи в іншу або їх об'єднання.

* При реалізації **стратегії міграції** вся популяція поділяється на кілька груп, зазвичай, з різною кількістю особин. Оператори кросоверу застосовуються до особин строго зі своїх спільнот. До окремих особин застосовуються різні оператори мутації і інверсії. Процеси еволюції відбуваються незалежно, кожна в своїй групі. І лише в деяких випадках (з невеликою ймовірністю) відбувається обмін генетичним матеріалом окремих особин між групами. Протягом еволюції збирається банк кращих рішень, які потім об'єднуються в єдину популяцію. Для популяції кращих рішень реалізується свій еволюційний процес.
* **Турнірна стратегія** передбачає проведення повністю незалежних закінчених еволюційних процесів в кількох групах і об'єднання отриманих в кожній групі кращих рішень в одну або кілька нових груп з виконанням наступного етапу еволюції. Кількість таких етапів в загальному випадку може бути довільною і обмежується лише вимогами часу і обчислювальними можливостями. Природно, чим більше етапів, тим більша ймовірність знаходження глобального оптимуму. Розміри груп також можуть бути різними: від кількох десятків до кількох одиниць особин.

## 5.5. Генетичне програмування

Генетичне програмування - це методика машинного навчання. У загальному випадку обчислення починаються з великого набору програм (популяції), згенерованих випадковим чином або написаних вручну, про які відомо, що це досить хороші рішення. Ці програми конкурують між собою в спробі вирішити поставлене користувачем завдання. Це може бути гра, в якій програми безпосередньо змагаються між собою, або спеціальний тест, що визначає, яка програма краще. По завершенні змагання складається посортований список програм - від найкращої до найгіршої.

Далі вступає в справу еволюція - найкращі програми копіюються і модифікуються в один з двох способів. Найпростішим є мутація: деякі частини програми випадковим чином і незначно змінюються в надії, що від цього рішення стане краще (рис.5.6).

Створення початкової популяції

Відбір найкращих особин

Чи є достатньо хороша особина

Готово

Копіювати кращі особини

Мутація

Схрещування

Нова популяція

Рис.5.6. Алгоритм генетичного програмування

Інший спосіб є схрещуванням – пара відібраних програм обмінюються своїми частинами. Частина однієї з відібраних програм переміщається в іншу. В результаті процедури копіювання та модифікації створюється багато нових програм, які засновані на найкращих зразках попередньої популяції, але не збігаються з ними.

На кожному етапі якість програм обчислюється за допомогою функції пристосованості (Fitness Function). Оскільки розмір популяції не змінюється, програми, які виявилися гіршими, видаляються з популяції, звільняючи місце для нових.

Створюється нова популяція і весь процес повторюється. Оскільки зберігаються і змінюються самі кращі програми, то є сподівання, що з кожним поколінням вони будуть вдосконалюватися. Нові покоління створюються, доки не буде виконано умову завершення, яка в залежності від завдання може формулюватися в один з способів:

* Знайдено ідеальне рішення.
* Знайдено достатньо хороше рішення.
* Кількість поколінь досягло заданої межі.
* Рішення не вдається поліпшити протягом кількох поколінь.

## 5.6. Еволюційне програмування

Еволюційне програмування було запропоновано Лоуренсом Дж. Фогелем в 1960 році. У той час штучний інтелект було обмежено двома основними напрямками досліджень: моделюванням людського мозку (нейронні мережі) і моделюванням поведінки людини (евристичне програмування). Альтернативний варіант Фогеля був спрямований на моделювання процесу еволюції, як засобу отримання розумної поведінки. Фогель розглядає інтелект як складову частину здатності робити передбачення відповідно до заданої мети. На його думку прогнозування є необхідною умовою для розумної поведінки.

Гіпотези про вид залежності цільової змінної від інших змінних формулюються системою у вигляді програм. Процес побудови таких програм розглядається як еволюція в популяції програм. Якщо система знаходить програму, яка точно відтворює шукану залежність, вона починає вносити до неї невеликі модифікації і відбирає серед дочірніх програм лише ті, які підвищують точність.

Система "вирощує" кілька генетичних ліній програм, що конкурують між собою в точності знаходження шуканої залежності. Спеціальний транслюючий модуль перекладає знайдені залежності з внутрішньої мови системи на зрозумілу користувачеві мову (математичні формули, таблиці тощо), роблячи їх наочними. Для того, щоб зробити отримані результати більш зрозумілими, існує великий арсенал різноманітних засобів візуалізації виявлених залежностей.

Пошук залежності цільових змінних від інших факторів проводиться у формі функцій певного виду. Наприклад, в одному з найбільш вдалих алгоритмів цього типу - методі групового урахування аргументів (МГУА) залежність шукають у формі поліномів. Причому складні поліноми заміняються кількома простими, враховують лише деякі ознаки (групи аргументів). Отримані формули залежностей надаються до аналізу та інтерпретації.

Еволюційне програмування застосовне до різних інженерних завдань:

* Системи управління, системи ідентифікації.
* Маршрутизація трафіку.
* Військове планування.
* Обробка сигналів.
* Ігрові та навчальні програми.

Основною перевагою еволюційних методів оптимізації є можливості вирішення завдань з великою розмірністю за рахунок поєднання елементів випадковості і строго визначених правил, основним з яких є закон еволюції: «перемагає найсильніший».

Іншим важливим чинником ефективності є відтворення процесів розвитку, де знайдені рішення за певним правилами породжують нові рішення, які будуть наслідувати кращі риси попередників. Як випадковий елемент використовується моделювання процесу мутації. З її допомогою характеристики певного рішення можуть бути випадково змінені, що надає новий напрямок в процесі еволюції і може пришвидшити процес вироблення кращого рішення.

#### Переваги еволюційних алгоритмів

* Наочність схеми і базових принципів еволюційних обчислень.
* Придатність для пошуку в складному просторі рішень великої розмірності.
* Можливість підбору початкових умов, комбінування еволюційних обчислень, продовження процесу еволюції доки є необхідні ресурси.
* Відсутність обмежень на вид цільової функції.
* Широка область застосування.

#### Недоліки еволюційних алгоритмів

* Еволюційні обчислення не гарантують оптимальності отриманого рішення, але за заданий час можна отримати кілька хороших альтернативних рішень.
* Вимагається висока обчислювальна потужність.
* Відносно невисока ефективність на завершальних фазах еволюційного процесу.

## Висновки

Значну роль в розвитку природних систем відіграє еволюція, тому дослідники прагнуть розробляти моделі, що реально пояснюють принципи роботи цього механізму: відтворення структури популяції, народження і виродження особин, колективна поведінка та події між ними.

Генетичні алгоритми є алгоритмами пошуку, що побудовані на принципах, подібних до принципів природного відбору і генетики. Насамперед, це принцип виживання найбільш перспективних рішень, обмін інформацією, в якій наявний елемент випадковості. Разом ці принципи спроможні моделювати природні процеси успадкування і мутації. Додатковою властивістю цих алгоритмів є невтручання людини в ітераційний процес пошуку, впливати можна лише опосередковано, задаючи певні параметри. Зберігається біологічна термінологія в спрощеному вигляді і основні поняття лінійної алгебри

Досягнення оптимуму в багатокритеріальних задачах є вкрай складним та вимагає значної кількості ресурсів. Генетичні алгоритми скеровані на знаходження кращого задовільного значення, а не оптимального рішення задачі. І така властивість сприяє розвитку і зростанню популярності генетичних алгоритмів.

## Контрольні питання

1. Які базові принципи покладено в основу генетичних алгоритмів
2. Назвіть причини введення до генетичних обчислень механізму мутації.
3. Які критерії можна застосувати для припинення роботи генетичного алгоритму?
4. Назвіть поширені завдання, що вирішують за допомогою генетичних алгоритмів.
5. Яку дію виконує кросинговер?
6. Навіть певні модифікації генетичних алгоритмів, що сприяють отриманню кращих результатів.
7. Яке основні завдання ставляться перед мурахами в мурашиних алгоритмах?
8. Назвіть фактори, що сприяють мурахам вибрати найкраще рішення.
9. Які фактори поведінки в мурашиній колонії враховуються для моделювання?
10. Перелічіть прикладні завдання, в яких можна використати мурашиний алгоритм

## Використані джерела

1. А.П. Карпенко Сучасні алгоритми пошукової оптимізації. Алгоритми, що надихаються природою - <http://baumanpress.ru/books/474/474.pdf>
2. Еволюційні алгоритми - <https://ru.coursera.org/lecture/vvedenie-v-iskusstvennyi-intellekt/evoliutsionnyie-alghoritmy-ocaze>
3. Генетичний алгоритм - <https://uk.wikipedia.org/wiki/Генетичний_алгоритм>
4. Генетичні алгоритми. Ключові поняття і методи реалізації - <http://www.znannya.org/?view=ga_general>
5. Муравьиные алгоритмы - <https://habr.com/ru/post/105302/>
6. Чураков М., Якушев А. Мурашині алгоритми - <https://blog.bullgare.com/wp-content/uploads/2019/05/aca.pdf>
7. Мурашиний алгоритм. Опис математичної моделі - <http://smart-blog.net/post/2359>
8. Демо-програма Ant System <https://courses.cs.ut.ee/demos/visual-aco/>
9. Ant Colony Optimization (ACO) - Wolfram Demonstrations Project <https://demonstrations.wolfram.com/AntColonyOptimizationACO/>