

Генетичний алгоритм

Як відомо оптимізаційні задачі полягають в знаходженні мінімуму (максимуму) заданої функції. Таку функцію називають цільовою. Як правило цільова функція залежить від кількох вхідних параметрів. В задачі оптимізації необхідно знайти значення вхідних параметрів при яких цільова функція отримує мінімальне (максимальне) значення. В попередньому розділі описано цілий клас оптимізаційних методів. Як результат всі методи можна поділити на методи, що використовують похідні (градієнтні методи) та стохастичні методи (наприклад методи групи Монте-Карло). З їх допомогою можна знайти екстремальне значення цільової функції, але не можна бути переконаним, що знайдено глобальне значення екстремуму. Знаходження локального екстремуму замість глобального називається передчасною збіжністю. Крім проблеми передчасної збіжності існує ще одна проблема – час процесу обчислень. Дуже часто точніші методи оптимізації працюють дуже довго.

Для розв'язку поставлених проблем і проводиться пошук нових оптимізаційних алгоритмів. Генетичні алгоритми, які розглядаються в даній роботі відносяться до стохастичних методів. Ці алгоритми успішно використовуються в різних галузях діяльності (економіка, фізика, технічні науки, та ін.). Вже створено цілу низку модифікацій цих алгоритмів і розроблено тестові функції. Розглянути як працюють генетичні алгоритми (ГА) в багатопроцесорному середовищі – мета даної роботи.

ГА відносяться до галузі м'яких обчислень. Це поняття об'єднує такі галузі, як нечітка логіка, нейронні мережі, імовірнісні судження, мережі довіри та еволюційні алгоритми, які доповнюють одна одну і використовуються в різних комбінаціях або самостійно для створення гібридних інтелектуальних систем.

Генетичний алгоритм був отриманий в процесі узагальнення і імітації в штучних системах таких якостей живої природи, як природний відбір, пристосованість до умов навколишнього середовища, наслідування нащадками життєво-важливих якостей від батьків і т.д. Так як алгоритм в процесі пошуку використовує деяке кодування множині параметрів замість самих параметрів,

тобто він може ефективно застосовується для вирішення задач дискретної оптимізації, визначених як на числових множинах, так і на кінцевих множинах вільної природи. Оскільки для роботи алгоритми використовують лише значення функції оптимізації в точках, і не вимагається обчислення ні похідних, ні будь-яких інших характеристик, то даний алгоритм застосовується в широкому класі функцій, в випадках, коли нема аналітичних описів. Використання набору початкових точок дозволяє застосовувати для їх формування різні способи, які залежать від специфіки задачі, яка вирішується, в тому числі можливе задання такого набору виключно людиною [2].

Перевага генетичних алгоритмів в тому, що цей метод дуже гнучкий, і, будучи побудованим на передбаченні, алгоритм успішно справляється з широким колом проблем, особливо в задачах, в яких не існує загальних алгоритмів вирішення або високий ступінь апріорної невизначеності.

Головні труднощі з можливістю побудови обчислювальних систем, заснованих на принципах природного відбору і застосуванням цих систем у прикладних задачах, полягає в тому, що природні системи досить хаотичні, а всі наші дії, фактично, носять чітку спрямованість. Ми використовуємо комп'ютер як інструмент для розв'язку поставлених задач, що ми самі і формулюємо та акцентуємо увагу на максимально швидкому виконанні при мінімальних витратах. Природні системи не мають ніяких таких цілей чи обмежень, у всякому разі, нам вони не відомі. Виживання в природі не спрямовано до деякої мети, замість цього еволюція робить крок вперед у будь-якому доступному напрямку. Можливо це велике узагальнення, але я думаю, що зусилля, спрямовані на моделювання еволюції за аналогією з природними системами, до точного часу можна розбити на дві великі категорії:

- Системи, що змодельовані на біологічних принципах. Вони успішно використовуються для задач функціональної оптимізації і можуть бути легко описані небіологічною мовою;
- Системи, що є біологічно більш реалістичними, але які не виявилися особливо корисними в прикладному змісті. Вони більше схожі на біологічні системи і менш спрямовані (чи не спрямовані зовсім). Вони

мають складну і цікаву поведінку, і, скоріше, незабаром набудуть практичного застосування

Генетичний алгоритм — це алгоритм, який дозволяє знайти задовільний розв’язок проблем, які аналітично розв’язати не можливо, за допомогою послідовного підбору і комбінування шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Загальний вид еволюційних алгоритмів — це оптимізаційний метод, який базується на еволюції популяції „особин”. Кожна особина характеризується пристосованістю $f(S)$ — багатомірною функцією її „генів” $S = (S_1, S_2, \dots, S_N)$. Задача оптимізації використовує максимізацію функції пристосованості $f(S)$. У процесі еволюції в результаті відбору, рекомбінації і мутації геномів особин проходить пошук особин з високим пристосуванням.

Основні еволюційні алгоритми:

- генетичний алгоритм, призначений для оптимізації функцій дискретних змінних і акцентує увагу на рекомбінації геномів;
- еволюційне програмування, орієнтоване на оптимізацію неперервних функцій без використання рекомбінації;
- еволюційна стратегія, орієнтована на оптимізацію неперервних функцій з використанням рекомбінації;
- генетичне програмування, яке використовує еволюційний метод для оптимізації комп’ютерних програм.

У порівнянні з звичайними оптимізаційними методами еволюційні алгоритми мають такі особливості: паралельний пошук, випадкові мутації і рекомбінації уже знайдених розв’язків. Вони підходять як простий евристичний метод оптимізації багатомірних, погано визначених функцій. Формуються навіть області застосування еволюційних алгоритмів.

В розрізі генетичних алгоритмів можна відзначити їх використання для розв’язку таких задач:

- Оптимізація функцій
- Різноманітні задачі на графах (задача комівояжера, розфарбовування, знаходження відповідностей пар)

- Налаштування і навчання штучної нейронної мережі
- Складання розкладів
- Ігрові стратегії
- Апроксимація функцій
- Штучне життя

У 1859 році Чарльз Дарвін опублікував "Походження видів", де було проголошено основні принципи еволюційної теорії:

- спадковість
- мінливість
- природний відбір

У Барічеллі Н.А. - перші публікації, що відносяться до ГА:

- "Symbiogenetic evolution processes realised by artificial methods" (1957)
- "Numerical testing of evolution theories" (1962)

Роботи були направлені перш за все на розуміння природного феномену спадковості.

Батьком сучасної теорії генетичних алгоритмів вважається Д.Х. Холланд (J. Holland). Проте спочатку його цікавила, перш за все, здібність природних систем до адаптації, а його мрією було створення такої системи, яка могла б пристосовуватися до будь-яких умов навколишнього середовища.

У 1975 році Холланд публікує свою найзнаменитішу роботу «Adaptation in Natural and Artificial Systems». В ній він вперше ввів термін «генетичний алгоритм» і запропонував схему класичного генетичного алгоритму (canonical GA). Надалі поняття «генетичні алгоритми» стало широковживаним, і часто до них відносяться алгоритми, що дуже відрізняються від класичного ГА.

Учні Холланда - Кенет Де Йонг (Kenneth De Jong) і Девід Голдберг (David E. Goldberg) - внесли величезний внесок до розвитку ГА. Найбільш відома робота Голдберга - «Genetic algorithms in search optimization and machine learning» (1989) [3].

1.2 Класичний ГА

Постановку задачі ГА можна сформулювати так: нехай перед нами стоїть задача оптимізації. Переформулюємо її як задачу знаходження максимуму деякої функції $f(x_1, x_2, \dots, x_n)$, що називається функцією пристосованості (fitness function). Вона повинна:

- бути визначена на обмеженій області визначення
- набувати невід'ємних значень
- при цьому абсолютно не потрібно, щоб функція була безперервною і диференційованою.

Основна ідея генетичних алгоритмів – організація „боротьби за існування” і „природного добору” серед вибраних випадковим чином початкового набору розв’язків (популяції). Випадковий вибір розв’язків показано на малюнку. Пробні розв’язки можна записати в двійковій формі: {010, 011, 101, 100}. Оскільки генетичний алгоритм використовує біологічні аналогії, то і термінологія нагадує біологічну. Таким чином особиною чи хромосомою називатиметься рядок, що є конкатенацією рядків впорядкованого набору параметрів, наприклад:

1010 10110 101 ... 10101

| x_1 | x_2 | x_3 | ... | x_n |

А набір всіх пробних розв’язків називатиметься популяцією. Як відомо принцип природного добору полягає в тому, що в конкурентній боротьбі виживає найпристосованіший. В нашому випадку пристосованість особини визначається цільовою функцією: чим менше значення цільової функції, тим більш пристосована особина, тобто пробний розв’язок, що використовувався як аргумент цільової функції.

Розглянемо процес розмноження: спробуємо на основі вихідної популяції створити нову, так щоб пробні розв’язки в новій популяції знаходилися б ближче до глобального мінімуму цільової функції. Для цього сформуємо з вихідної популяції пати для схрещування. Поставимо в відповідність кожній особині випадкове ціле число. Будемо розглядати ці числа як номери членів популяції. При такому виборі певні члени популяції не будуть брати участі в процесі розмноження, бо утворять пару самі з собою. Інші ж члени популяції приймуть

участь в процесі розмноження не один раз з різними особинами популяції. Процес розмноження (рекомбінація) полягає в обміні частинами хромосоми між батьками. Наприклад, якщо схрестити дві хромосоми 111111 і 000000, при цьому випадковим чином визначити точку розриву хромосоми (нехай це буде 3): 111/111 000/000 і здійснити обмін частинами, що знаходяться після точки розриву, то отримаємо двох нових нащадків: 111000 та 000111.

Наступним кроком в роботі генетичного алгоритму є мутації – тобто випадкові зміни отриманих в результаті схрещування хромосом. Нехай ймовірність мутації дорівнює 0,3. Для кожного нащадка виберемо випадкове число на відрізку $[0;1]$, і якщо це число буде менше 0,3, то інвертуємо випадково вибраний ген (замінімо 0 на 1 або навпаки).

Мутації можуть покращити чи погіршити пристосованість особини-нащадка. В результаті схрещування особини обмінюються „хвостами”, тобто молодшими розрядами в війковому представленні числа. В результаті мутацій зміни може піддатись будь-який розряд (в тому числі і старший). Таким чином, якщо схрещування призводить до відносно незначних змін пробних розв’язків, то мутації можуть призвести до значних змін значень пробних розв’язків.

Тепер з особин-батьків і отриманих особин нащадків необхідно сформувати нову популяцію. В нову популяцію відберемо найбільш пристосованих осіб (із „старих” особин і особин-нащадків). Для наочності популяцію до другого відбору показано на малюнку.

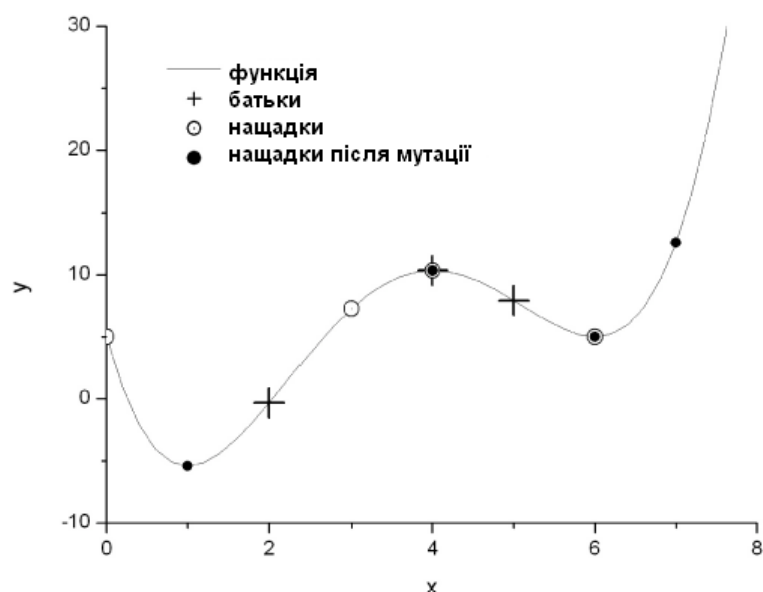


Рисунок 2.1 – Зміна популяції в процесі природного добору

В результаті було отримано нове покоління, в якому можна буде знову здійснити кросинговер, мутації і відбір особин в нове покоління. Таким чином через кілька поколінь ми отримаємо популяцію із схожих і найбільш пристосованих особин. Значення пристосованості найкращої особини (або середня пристосованість по популяції) і буде розв'язком задачі.

Таким чином принцип роботи ГА можна описати так:

- Популяція – сукупність всіх «особин», що є рядками, які кодують один з розв'язків задачі.
- Використання функції пристосованості:
 - найбільш пристосовані (більш відповідні розв'язки) отримують можливість схрещуватися і давати нащадків
 - найгірші (погані розв'язки) видаляються з популяції і не дають нащадків.
- Таким чином, пристосованість нового покоління в середньому вище попереднього.
- У класичному ГА:
 - початкова популяція формується випадковим чином
 - розмір популяції (кількість особин N) фіксується і не змінюється протягом роботи всього алгоритму
 - кожна особина генерується як випадковий L -бітний рядок, де L — довжина кодування особини
 - довжина кодування для всіх особин однакова.

Введемо основні поняття, що використовуються в генетичному алгоритмі.

Вектор – впорядкований набір чисел, що називається компонентами вектора. Так як вектор можна представити в вигляді рядка його координат, то в подальшому поняття вектора і рядка вважатимемо ідентичними.

Булевий вектор – вектор, компоненти якого приймають значення з двох елементної (булевої) множини, наприклад, $\{0, 1\}$ чи $\{-1, 1\}$.

Відстань Хемінга – використовується для булевих векторів і дорівнює кількості компонент, що відрізняються в обох векторах.

Простір Хемінга – простір булевих векторів, з введеною на ньому відстанню (метрикою) Хемінга. У випадку булевих векторів розмірності n цей простір представляє собою множину вершин n -мірного гіперкуба з метрикою хемінга. Відстань між двома вершинами визначається довжиною найкоротшого шляху між цими вершинами вздовж ребер.

Хромосома – вектор (чи рядок) з певних чисел. Якщо цей вектор представлено бінарним рядком з нулів та одиниць, наприклад 1010011, то його було отримано або за допомогою двійкового кодування, або кода Грея. Кожна позиція (біт) хромосоми називається *геном*.

Відстань – хемінгова відстань між бінарними хромосомами.

Кросингвер (кросовер) – операція при якій дві хромосоми обмінюються своїми частинами. Наприклад, 1100 та 1010 \rightarrow 1110 та 1000.

Мутація – випадкова зміна однієї або кількох позицій в хромосомі. Наприклад: 1010011 \rightarrow 10100011.

Інверсія – зміна порядку слідування бітів в хромосомі або в її фрагменті. Наприклад, 1100 \rightarrow 0011.

Популяція – сукупність індивідів.

Пристосованість – критерій чи функція, екстремуму якої необхідно знайти.

Локус – позиція гена в хромосомі.

Алель – сукупність генів, що йдуть підряд.

Епістаз – вплив гена на пристосованість індивіда в залежності від значення гена, що знаходиться в іншому місті. Ген вважається епістатичним, коли, його наявність подавляє вплив гену в іншому локусі. Епістатичні гени через свій вплив на інші гени інколи називають інгібуючими. Подавлення проявів гену неалельним до нього геном називається гіпостазом, а подавлений ген – гіпостатичним.

З наведених визначень випливає, що термінологія ГА представляє собою синтез генетичних і штучних понять. Так для розуміння, запозиченого з генетики, можна представити його штучний (символічний) аналог. Наприклад, хромосома і рядок. В біологічних системах повний генетичний пакет занимається генотипом. В штучних системах повний генетичний пакет рядків називається

структурою. В біологічних системах в процесі індивідуального розвитку організму взаємодія генотипу з оточуючим середовищем формує сукупність зовнішніх ознак і властивостей, що називається фенотипом. В математичному моделювання структура, що розглядається декодується за допомогою *множини параметрів*, які в літературі інколи називають альтернативним розв'язком або точкою. Можливі значення параметрів утворюють *простір розв'язків*. В штучній генетичній системі можна використовувати як числові так і нечислові параметри [4].

В термінології ГА говорять, що рядки утворені значеннями функції або детекторами. Значення функції можуть бути локалізовані в різних позиціях рядка. Зв'язок між біологічною і штучною термінологією наведено в таблиці.

Генетичний алгоритм – це проста модель еволюції у природі, реалізована у вигляді комп'ютерної програми. У ньому використовуються як аналог механізм генетичної спадковості, так і аналог природного відбору. При цьому зберігається біологічна термінологія в спрощеному вигляді.

Табл.2.1. Зв'язок між біологічною та штучною термінологіями.

Хромосома	Рядок
Ген	Значення функції, характеристика, детектор
Алель	Можливі значення генів
Локус	Позиція в рядку
Генотип	Закодована хромосома. Фактична структура
Фенотип	Множина параметрів, альтернативний розв'язок, декодована структура.
Епістаз	Взаємодія генів. Нелінійність.
Кросинговер	Операція, при якій дві хромосоми обмінюються своїми частинами.
Мутація	Випадкова зміна однієї або декількох позицій в хромосомі.

Щоб змоделювати еволюційний процес, генеруємо спочатку випадкову популяцію - декількох індивідуумів з випадковим набором хромосом (числових

векторів). Генетичний алгоритм імітує еволюцію цієї популяції як циклічний процес схрещування індивідуумів і зміни поколінь.

Життєвий цикл популяції – це декілька випадкових схрещувань і мутацій, в результаті яких до популяції додається деяка кількість нових індивідуумів. Відбір в генетичному алгоритмі – це процес формування нової популяції із старої, після чого стара популяція гине. Після відбору до нової популяції знову застосовуються операції кросинговеру і мутації, потім знову відбувається відбір, і так далі.

Табл.2.2. Відображення природного відбору в природі

Пристосування індивідуума	Значення цільової функції в цій точці (індивідуумі).
Виживання найбільш пристосованих	Популяція наступного покоління формується в відповідності з цільовою функцією. Чим більша пристосованість індивідуума, тим більша ймовірність його участь в кросинговері, тобто розмноженні

Як правило, ймовірність участі індивідуума в схрещуванні береться пропорційно його пристосованості. Часто використовується так звана стратегія елітизму, при якій декілька кращих індивідуумів переходять в наступне покоління без змін, не беручи участь в кросинговері і відборі. У будь-якому випадку кожне наступне покоління буде в середньому краще попереднього. Коли пристосованість індивідуумів перестає помітно збільшуватися процес зупиняють і як рішення задачі оптимізації беруть якнайкращого із знайдених індивідуумів.

Критерієм закінчення процесу може бути задана кількість поколінь, або збіжність (convergence) популяції.

Збіжність називається такий стан популяції, коли всі рядки популяції майже однакові і знаходяться в області екстремуму. В такій ситуації кросинговер практично не змінює популяцію так як утворені нащадки представляють собою

копії батьків із зміненими ділянками хромосом. Ті особини, які виходять з цієї області за рахунок мутацій схильні вимирати, так як в переважній більшості випадків вони матимуть гіршу пристосованість, особливо якщо даний екстремуму буде глобальним. Таким чином збіжність популяції зазвичай означає, що знайдено найкращий або близький до нього розв'язок.

Популярність генетичних алгоритмів зумовлена тим, що вони дозволяють знайти більш "раціональні" рішення задач оптимізації за менший час, ніж інші методи, які застосовуються у цих випадках. Переваги генетичних алгоритмів стають більш очевидними, якщо розглянути основні їх відмінності від традиційних методів. Основних відмінностей п'ять:

- Генетичні алгоритми працюють з кодами, в яких представлений набір параметрів, безпосередньо залежних від аргументів цільової функції.
- Для пошуку генетичний алгоритм використовує декілька точок пошукового простору одночасно, а не переходить від точки до точки, як це робиться в традиційних методах. Тобто ГА оперує одночасно всією сукупністю припустимих рішень.
- Генетичні алгоритми в процесі роботи не використовують ніякої додаткової інформації, що підвищує швидкість їх роботи.
- ГА використовує як імовірнісні правила для виникнення нових точок пошуку, так і детерміновані правила для переходу від одних точок до інших.
- Генетичні алгоритми здійснюють пошук оптимального рішення за однією й тією ж стратегією, як для унімодальних, так і для багатьох екстремальних функцій.

ГА працює з кодовими послідовностями (КП). Тому сама КП і її структура описуються поняттям генотип, а його інтерпретація, з точки зору задачі, що вирішується, поняттям фенотип. Кожна КП є точкою пошуку. Екземпляр КП називають хромосомою.

В процесі роботи алгоритму генерація нових КП відбувається на базі моделювання процесу розмноження. При цьому, природно, КП, що породжують,

називаються батьками, а породжені КП – нащадками. Батьківська пара, як правило, породжує пару нащадків. Безпосередня генерація нових рядків з двох обраних відбувається за рахунок роботи оператора схрещування (випадково-детермінованого обміну), який в процесі роботи алгоритму може застосовуватися не до всіх пар батьків.

Частина цих пар може переходити до популяції наступного покоління безпосередньо. Те, як часто виникатиме така ситуація, залежить від імовірності застосування оператора схрещування, яка є одним із параметрів ГА.

Моделювання процесу генерації нових точок пошуку відбувається за рахунок роботи оператора мутації, який задається визначеною імовірністю. Оскільки еволюційний процес біологічних видів супроводжується загибеллю останніх, то породження нащадків повинно супроводжуватися знищенням інших безперспективних КП. Вибір пар батьків з популяції для породження нащадків виконує оператор добору, а вибір КП для знищення - оператор редукції.

Характеристики генетичного алгоритму обираються таким чином, щоб забезпечити невеликий час роботи, з одного боку, та пошук як можна кращого рішення, з іншого. Загальна структурна схема генетичного алгоритму при біологічній інтерпретації зображена на Рис.2.2.

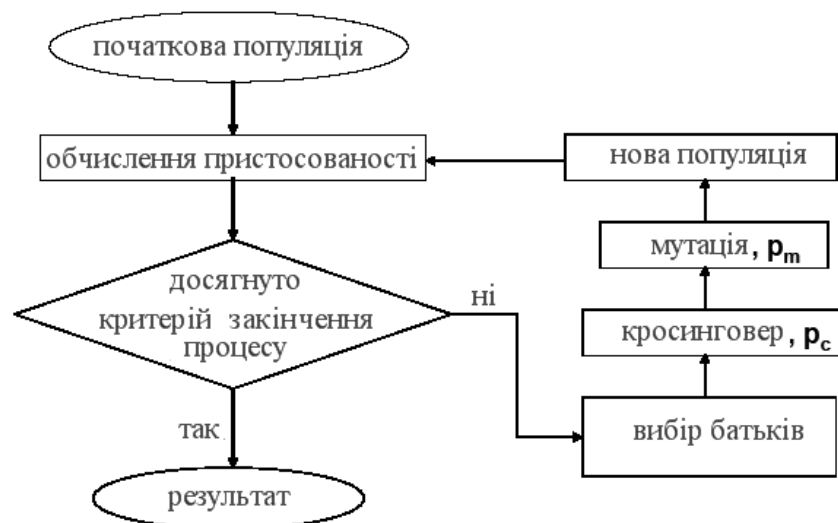


Рисунок 2.2 – Схема простого ГА

Формування вихідної популяції K відбувається за допомогою випадкового закону, наприклад, рівномірного, на базі якого обирається необхідний пошук.

Вихідна популяція може також бути результатом роботи будь-якого іншого алгоритму оптимізації.

В основі оператора відбору, який використовується для вибору батьківських пар та знищення КП, лежить принцип "виживає найсильніший".

Відбір і схрещування можна зобразити так:



Рисунок 2.4 – Відбір та схрещування

Проміжна популяція – це набір особин, які отримали можливість розмножуватись.

В класичному ГА ймовірність кожної особини потрапити в проміжну популяцію пропорційна її пристосованості, тобто здійснюється пропорційний відбір (proportional selection).

Критерієм зупинення генетичного алгоритму може бути одна з трьох подій:

- сформовано задане число поколінь;
- популяція досягла заданого рівня якості;
- досягнуто якийсь рівень збіжності (convergence) , при якому покращання популяції не відбувається.

Збіжністю називається стан популяції, коли всі рядки знаходяться в області деякого екстремуму і майже однакові. Таким чином, збіжність популяції означає, що досягнуто розв'язку, який близький до оптимального. Підсумковим розв'язком задачі може служити найбільш пристосована особина останнього покоління (див. рисунок 2.5)

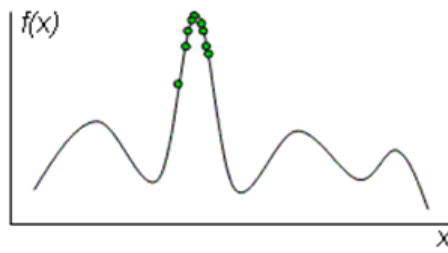


Рисунок 2.5 – Сходження алгоритму

Після роботи генетичного алгоритму з кінцевої популяції обирається та КП, яка дає мінімальне (або максимальне) значення цільової функції і є, у підсумку, результатом роботи генетичного алгоритму [5].

Ефективність ГА при вирішенні конкретної задачі оптимізації визначається видом генетичних операторів і вибором відповідних значень параметрів, а також способу представлення задачі на КП-хромосомі. Оптимізація цих факторів призводить до підвищення швидкості та стабільності пошуку, що є суттєвим для застосування генетичних алгоритмів.

Таким чином основні принципи роботи ГА можна записати такою послідовністю дій:

1. Генеруємо початкову популяцію з n хромосом
2. Обчислюємо для кожної з хромосом її пристосованість
3. Вибираємо пару хромосом-батьків за допомогою одного із способів відбору.
4. Проводимо кросинговер батьків з певною ймовірністю та отримуємо нащадків.
5. Проводимо мутацію нащадків з певною ймовірністю.
6. Повторюємо кроки 3-5, доки не буде згенеровано нову популяцію з n хромосоми.
7. Повторюємо кроки 2-5 доки не буде досягнуто критерію закінчення процесу.

Основними операторами ГА є кросинговер, мутація, вибір батьків і селекція (вибір хромосом в нову популяцію). Вид оператора відіграє важливу роль в реалізації і ефективності ГА. Існують основні форми операторів, використання

їх чистому чи модифікованому вигляді призводить до отримання ГА, які підходять для розв'язку конкретної задачі.

1.3 Оператори вибору батьків

Існує декілька підходів до вибору батьківської пари. Розглянемо найпопулярніші з них.

Панміксія – найпростіший оператор вибору. В відповідності з ним кожному члену популяції ставиться у відповідність випадкове ціле число на відрізку $[1;n]$, де n – кількість особин в популяції. Будемо розглядати ці числа як номери особин в популяції, які можуть приймати участь у схрещуванні. Про такому виборі певні з членів популяції не будуть приймати участь процесі розмноження, так як утворюють пару самі із собою. Інші ж члени популяції будуть брати участь кілька разів з різними особинами. Не дивлячись на простоту, такий підхід універсальний для розв'язку різних класів задач. Але він залежить від чисельності популяції, оскільки ефективність алгоритму, що реалізує такий підхід зменшується зі збільшенням чисельності популяції.

Інбридинг – це метод при якому перший батько вибирається випадковим чином, а другим буде член популяції, що знаходиться найближче до першого. «Найближчий» тут може розумітись як той, що має найменшу відстань Хемінга, що дорівнює кількості розрядів, що відрізняються (для бінарних рядків) або евклідову відстань між двома дійсними векторами.

При *аутбридингу* теж використовується поняття схожості особин. Але на цей раз пари формуються із максимально «далеких» особин.

Останні два способи по різному впливають на поведінку генетичного алгоритму. Так інбридинг можна охарактеризувати властивістю концентрації пошуку в локальних вузлах, що фактично призводить до розбиття популяції на окремі локальні групи навколо підозрілих на екстремум ділянок ландшафту. Аутбридинг в свою чергу націлений на попередження збіжності алгоритму до вже знайдених розв'язків, змушуючи алгоритм проглядати нові області, що не досліджувались. Інбридинг та Аутбридинг бувають генотипними (коли за

відстань береться різниця значень цільової функції для відповідних особин) та фенотипним (за відстань береться відстань Хемінга).

Селекція полягає в тому, що батьками можуть стати тільки ті особини, значення пристосованості яких не менше порогової величини, наприклад, середнього значення пристосованості по популяції. Такий підхід забезпечує кращу збіжність алгоритму. Але через швидку збіжність селективний відбір батьківської пари не підходить тоді, коли необхідно знайти декілька екстремумів, оскільки для таких задач алгоритм, як правило сходиться до одного із розв'язків. Крім того для деяких багатомірних задач зі складним ландшафтом цільової функції швидка збіжність може перетворитись в передчасну збіжність до квазіоптимального розв'язку. Цей недолік може бути частково компенсований використанням підходящого механізму відбору, який би „гальмував” швидку збіжність алгоритму.

Порогова величина в селекції може бути обчислена різними способами. Тому в літературі по ГА виділяють різні варіації селекції. Найбільш відомі з них це – турнірний відбір та відбір рулеткою.

При *турнірному відборі* з популяції, що включає N особин обираються випадковим чином t особин, і краща з них записується в проміжний масив. Ця операція повторюється N разів. Особини в отриманому проміжному масиві потім використовуються для схрещування (теж випадковим чином). Розмір групи рядків, що вибираються для турніру часто дорівнює 2. В цьому випадку кажуть про здійснюється двійковий (парний) турнір. Взагалі t називають чисельністю турніру. Перевагою цього способу є те, що він не вимагає додаткових обчислень. На рисунку зображено турнірний відбір.

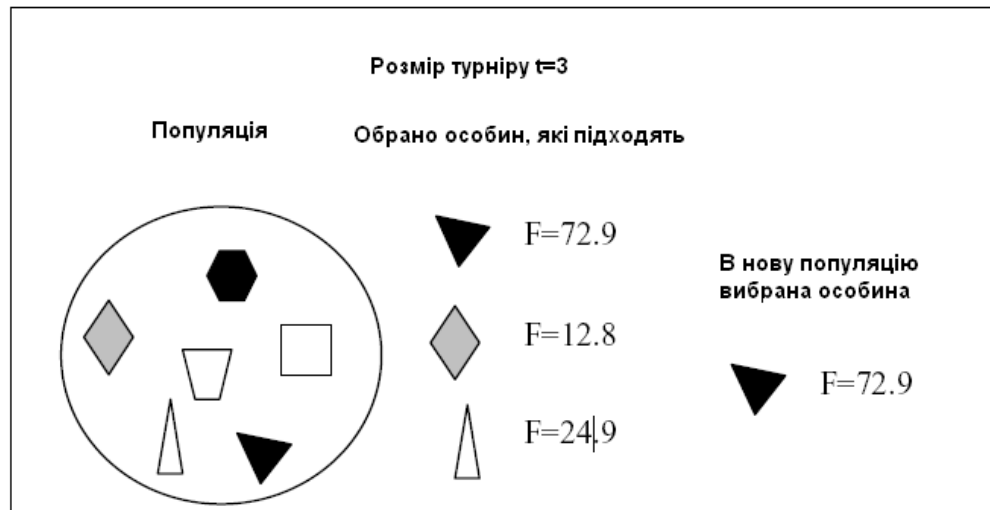


Рисунок 2.6 – Турнірний відбір

В методі рулетки особини відбираються за допомогою N „запусків” рулетки, де N – розмір популяції. Колесо рулетки має сектори для кожного члена популяції. Розмір i -го сектора пропорційний ймовірності переходу в нову популяцію $P(i)$.

Існує декілька способів реалізації такого відбору:

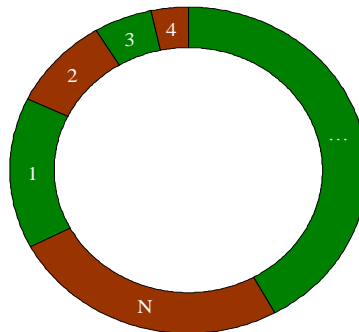


Рисунок 2.6 – Розміщення особин на рулетці відповідно до пристосованості.

- **stochastic sampling.** Особини розташовуються на колесі рулетки так, щоб розмір сектора кожної особини був пропорційним її пристосованості. N разів запускаючи рулетку, вибираємо необхідну кількість особин для запису в проміжну популяцію.
- **remainder stochastic sampling.** Особини розташовуються на рулетці так само, як і раніше. Але тепер в рулетки не одна стрілка, а N , причому вони відсікають однакові сектори. За один запуск рулетки вибираємо відразу всі N особин.

Імовірність участі i -ї КП в процесі обміну обчислюється за формулою:

$$P_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

(2.1)

де n - розмір вихідної множини K ,

i - номер КП (номер особини),

f_i - значення цільової функції для i -ї КП (пристосованість особини).

Очікувана кількість копій i -тої хромосоми після оператора рулетки визначається формулою:

$$N_i = P(i)N.$$

(2.2)

При такому відборі члени популяції з кращою пристосованістю будуть обиратись частіше ніж особини гіршою.

Можна отримати й інші способи на основі модифікації вказаних. Так, наприклад, в відборі „рулеткою” можна змінити формулу для ймовірності потрапляння особини в нову популяцію.

1.4 Рекомбінація (відтворення)

Оператор рекомбінації використовують відразу після оператора відбору батьків для отримання нових особин-нащадків. Зміст рекомбінації полягає в тому, що створені нащадки повинні наслідувати генну інформацію від обох батьків. Розрізняють *дискретну рекомбінацію* та *кросинговер*.

1.4.1. Дискретна рекомбінація

Дискретна рекомбінація в основному використовується для хромосом з дійсними генами. Основними способами дискретної рекомбінації є власне дискретна рекомбінація, проміжна, лінійна та розширено лінійна рекомбінації.

Дискретна рекомбінація відповідає обміну генами між особинами. Для ілюстрації цього оператора порівняємо дві особини з трьома генами:

Табл. 2.3

Особина 1	12	25	7
Особина 2	116	4	34

Для створення двох нащадків з однаковою ймовірністю випадково виберемо номер особини для кожного гена:

Табл. 2.4

Схема 1	2	2	1
Схема 2	1	2	1

Відповідно схеми створимо нащадків:

Табл. 2.5

Нашадок 1	116	4	7
Нашадок 2	12	4	7

Дискретна рекомбінація використовується для будь-якого типу генів (війкові, дійсні і символічні).

Проміжна рекомбінація використовується лише до дійсних змінних, але не до бінарних. В цьому методі попередньо визначається числовий інтервал значень генів нащадків, який повинен мати значення генів батьків. Нашадки створюються по такому правилу:

$$\text{Нашадок} = \text{Батько 1} + \alpha(\text{Батько 2} - \text{Батько 1}),$$

(2.3)

де множник α – випадкове число на відрізку $[-d, 1+d]$, $d \geq 0$. Як відмічають прихильники цього методу, найоптимальніше відтворення виходить при $d = 0,25$. Для кожного гена нового нащадка вибирається окремий множник α . Розглянемо використання оператора на прикладі. Нехай для батька мають такі значення генів:

Табл. 2.5

Особина 1	12	25	7
Особина 2	116	4	34

Випадково виберемо значення $\alpha \in [-0, 25; 1,25]$ для кожного гена обох нащадків:

Табл. 2.6

Схема 1	0,5	1,1	- 0,1
Схема 2	0,1	0,8	0,5

Обчислимо значення генів нащадків по запропонованій формулі:

Табл. 2.7

Нащадок 1	$12 + 0,5(116 - 12) = 64$	$25 + 1,1(4 - 25) = 1,9$	4,3
Нащадок 2	$12 + 0,1(116 - 12) = 22,4$	$25 + 0,8(4 - 25) = 8,2$	20,5

При проміжній рекомбінації виникають значення генів, що відрізняються від генів особин-батьків. Це призводить до виникнення нових особин, пристосованість яких може бути кращою, ніж пристосованість батьків. В літературі такий оператор рекомбінації інколи називають *диференціальним схрещуванням*.

Лінійна рекомбінація відрізняється від проміжної тим, що множник α вибирається для кожного нащадка один раз. Розглянемо гени обраних раніше батьків. Нехай значення α визначається таким чином:

Табл. 2.8

Схема 1	0,5
Схема 2	0,1

Тоді гени створених нащадків приймуть такі значення:

Табл. 2.9

Нащадок 1	$12 + 0,5(116 - 12) = 64$	$25 + 0,5(4 - 25) = 14,5$	20,5
Нащадок 2	$12 + 0,1(116 - 12) = 22,4$	$25 + 0,1(4 - 25) = 22,9$	9,7

Якщо розглядати особини популяції як точки в k -мірному просторі, де k – кількість генів в одній особині, то можна сказати, що при лінійній рекомбінації згенеровані точки нащадків знаходяться на прямій, яка задається двома точками – батьками.

1.4.2. Кросинговер (бінарна рекомбінація)

Рекомбінацію бінарних рядків прийнято називати кросинговером (кросовером) або схрещуванням.

Особини проміжної популяції випадковим чином розбиваються на пари, які з деякою вірогідністю

- схрещуються, внаслідок чого виходять два нащадки, які записуються в нове покоління;
- не схрещуються, тоді в нове покоління записується сама пара;

В класичному ГА застосовується одноточковий оператор кросинговера (1-point crossover): для батьківських рядків випадковим чином вибирається точка розділу, нащадки виходять шляхом обміну відсіченими частинами.

Одноточковий оператор працює наступним чином. Спочатку, довільно обирається одна з точок розриву (точка розриву - ділянка між сусідніми бітами в рядку). Обидві батьківські структури розриваються на два сегменти в цій точці. Потім відповідні сегменти різних батьків склеюються та утворюється два генотипи нащадків. Наприклад, припустимо, що один з батьків складається з 10 нулів, а інший - з 10 одиниць. Нехай з 9 можливих точок розриву обрана точка 3, тоді робота одноточкового оператора схрещування виглядає так:

```
011010.01010001101 => 111100.01010001101  
111100.10011101001 => 011010.10011101001
```

Імовірність застосування оператора схрещування обирається достатньо великою, у межах від 0.9 до 1. Робиться це для того, щоб забезпечити постійну появу нових КП, які розширюють простір пошуку. Для операції схрещування можуть застосовуватися також двоточкові та рівномірні оператори схрещування. В двоточковому операторі обираються дві точки розриву, та батьківські хромосоми обмінюються сегментом, який знаходиться між двома цими точками. В рівномірному операторі схрещування кожен біт першого батька спадкується першим нащадком із заданою імовірністю, у протилежному випадку цей біт передається другому нащадку.

Для **багатоточкового кросинговеру**, вибираємо m точок розрізу $k_i \in \{1, 2, \dots, N_{\text{var}}\}, i = 1:m, N_{\text{var}}$ - кількість змінних (генів) в особині. Точки розрізу

вибираються випадково без повторень і сортуються в порядку зростання. При кросинговері відбувається обмін ділянками хромосом, обмежених точками розрізу і таким чином виходить два нащадки. Ділянка особини з першим геном до першої точки розрізу в обміні не приймає участі. Порівняємо дві особини по 11 двійковим генам:

Табл. 2.10

Особина 1	0	1	1	1	0	0	1	1	0	1	0
Особина 2	1	0	1	0	1	1	0	0	1	0	1

Оберемо точки розрізу кросинговера:

Табл. 2.11

Точка розрізу ($m = 3$)	2	6	10
---------------------------	---	---	----

Створимо двох нових нащадків:

Табл. 2.12

Нащадок 1	0	1	1	0	1	1	1	1	0	1	1
Нащадок 2	1	0	1	1	0	0	0	0	1	0	0

Використання багато точкового кросинговеру вимагає введення кількох змінних (точок розрізу), і для відтворення вибираються особини з найбільшою пристосованістю.

Одно точковий і багатоточковий кросинговери визначають точки розрізу, які розділяють особини на частини. Однорідний кросинговер створює маску (схему) особини, в кожному локус якої знаходиться потенційна точка кросинговера. Маска кросинговера має таку ж довжину, що і батьківські особини. Створити маску можна таким чином: введемо деяку величину $0 < p_0 < 1$, і якщо випадкове число більше за p_0 , то на n -у позицію маски ставимо 0, інакше – 1. Таким чином, гени маски представляють собою випадкові бітові числа (0 чи 1). Відповідно до цих значень, геном нащадка стане перша (якщо ген маски = 0) або друга (якщо ген маски = 1) особина-батько.

Однорідний кросинговер дуже схожий на багатоточковий, але рядок випадкових бітових значень в ньому довший. Це гарантує, що у нащадків будуть чергуватись короткі рядки особин-батьків.

Алгоритм однорідного кросинговеру для війкових рядків ідентичній дискретному відтворенню для дійсних хромосом. Такий вид кросинговера ще називається уніфікованим.

Тріадний кросинговер. Цей різновид кросинговера відрізняється від однорідного тим, що після вибору пари батьків з решти членів популяції випадковим чином вибирається особина, яка потім буде використовуватись замість маски. Потім 10% маски піддається мутації. Потім гени першого батька порівнюються з генами маски: якщо гени однакові, то вони передаються першому нащадку, в іншому випадку на відповідні позиції хромосоми нащадка переходять гени другого батька. Генотип другого нащадка відрізняється від генотипу першого тим, що на тих позиціях, де у першого нащадка знаходяться гени першого батька, а у другого стоять гени другого батька і навпаки.

Перетасований кросинговер. В цьому алгоритмі особини, які були відібрані для кросинговеру, випадковим чином обмінюються генами. Потім обирають точку для одно точкового кросинговеру і проводять обмін частинами хромосом. Після схрещування створені нащадки знову тасуються. Таким чином, при кожному кросинговері створюються не лише нові нащадки, але і модифікуються батьки (старі батьки видаляються), що дозволяє зменшити кількість операцій в порівнянні з однорідним кросинговером.

Кросинговер зі зменшенням заміни. Оператор зменшення заміни обмежує кросинговер, щоб завжди, коли це можливо, створювати нових особин. Це здійснюється за рахунок обмеження на вибір точки розрізу: точки розрізу повинні з'являтися лише там, де гени відрізняються.

Як було сказано, кросинговер генерує новий розв'язок (у вигляді особини-нащадка) на основі двох батьків комбінуючи їх частини. Тому кількість різних розв'язків, які можна отримати при використанні однієї і тієї ж пари готових розв'язків обмежена. Відповідно простір, який ГА може покрити використовує лише кросинговер, що жорстко залежить від генофонду популяції. Чим різноманітніші генотипи розв'язків, тим більший і простір покриття. При виявленні локального оптимуму відповідний йому генотип буде намагатись зайняти всі позиції в популяції, і алгоритм може передчасно збігтись. Тому в

генетичному алгоритмі важливу роль відіграють мутації. Існує декілька способів мутування генів.

1.5 Мутація

Після того, як закінчиться етап схрещування, виконуються оператори випадкової зміни КП (мутації). Оператор мутації служить для моделювання природного процесу мутації. Суть даного процесу полягає у наступному:

- До отриманого в результаті відбору і схрещування нового покоління застосовується оператор мутації, необхідний для "вибивання" популяції з локального екстремуму і сприяючий захисту від передчасної збіжності.
- Кожен біт кожної особини популяції з деякою малою вірогідністю (зазвичай менше 1%) інвертується. (1110001010110 -> 1110001110110)

В процесі роботи алгоритму усі зазначені вище оператори використовуються багатократно та ведуть до поступової зміни вихідної популяції. Оскільки всі оператори спрямовані на покращання кожної окремої КП, то результатом їх роботи є поступове покращання - еволюція до локально-оптимального (оптимального) рішення.

Ймовірність мутації може бути випадковим числом, а може бути і певною функцією від якоїсь з характеристик задачі, що розв'язується. Наприклад, можна припустити, що ймовірність мутації генів обернено пропорційна кількості всіх генів в особині (розмірність).

Таким чином мутації з фіксованою ймовірністю призводять до хороших результатів для унімодальних функцій. Для мультимодальних функцій використовують оцінку ймовірності, що самоадаптується.

Розглянемо основні варіанти мутування.

Мутація для дійсних особин. На малюнку показана можлива мутація для особин з дійсними генами в двовірному просторі.

Для мутації особин з дійсним и числами необхідно визначити розмір кроку мутації – число, на яке зміниться значення гена при мутуванні.

Зазвичай визначити крок мутації буває складно. Оптимальний розмір кроку повинен змінюватись протягом всього процесу пошуку. Найкраще використовувати невеликі кроки, але інколи великі кроки можуть призвести до прискорення процесу. Гени можуть мутувати відповідно до такого правила:

$$\text{Нова змінна} = \text{стара змінна} \pm \alpha \cdot \delta, \quad (2.4)$$

де знаки $+$ чи $-$ вибираються з однаковою ймовірністю, $\alpha = 0,5 \times$ пошуковий простір (інтервал зміни даної змінної),

$$\delta = \sum_{i=1}^m a(i)2^{-i},$$

$$(2.5)$$

$a(i) = 1$ з ймовірністю $\frac{1}{m}$, в іншому випадку $a(i) = 0$, m – параметр.

Нова особина, яку було отримано при такій мутації, в більшості випадків не набагато відрізнятиметься від старої. Це пов'язано з тим, що ймовірність маленького кроку в мутації вище, ніж ймовірність великого кроку. При $m = 20$, цей алгоритм мутації придатний лише для локалізації оптимуму з точністю $\alpha \cdot 2^{-19}$.

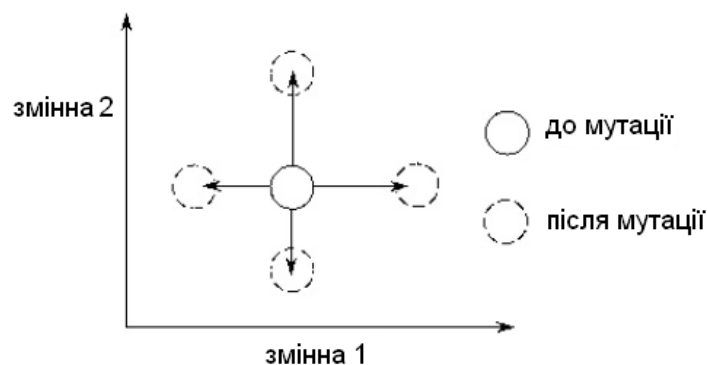


Рисунок 2.7 – Мутація особини

Двійкова мутація. Для особин, які кодувались двійковим кодом, чи кодом Грея, мутація полягає в випадковому інвертуванні гена (0 замінюється на 1 і навпаки). Ефект мутації залежить від використаного способу кодування генів. Так, в певних задачах при мутації найкращий ефект буде отримано у випадку, коли особини закодовані кодом Грея, в інших – за допомогою двійкового коду.

Густина мутації. Стратегія мутації з використанням поняття густина полягає в мутуванні кожного гена нащадка з заданою ймовірністю. Таким чином, крім ймовірності використання мутації до нащадка, використовується ще ймовірність використання мутації до кожного з його генів величину якої вибираються з таким розрахунком, щоб в середньому мутувало від 1 до 10% генів.

Інші види мутацій. Нехай особина t представлена такою послідовністю генів t_i : $t = t_1, \dots, t_k$. Тоді можна використати такі оператори мутації:

1. *Приєднання* випадкового гену з сукупності можливих значень генів в кінець послідовності $t \rightarrow t_1, \dots, t_s$.
2. *Вставка* випадкового гена з сукупності можливих значень генів в випадково вибрану позицію в послідовності: $t \rightarrow t_1, \dots, t_{i-1}, s, t_i, \dots, t_k$.
3. *Видалення* випадково вибраного гену з послідовності: $t = t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_k$.
4. *Обмін* місцями в послідовності двох сусідів одного випадково вибраного гена: $t \rightarrow t_1, \dots, t_{i+1}, s, t_{i-1}, \dots, t_k$.

Необхідно відзначити, що мутація 1 – це частковий випадок мутації 2. Для особин з фіксованим розміром (кількість генів в послідовності) можна використовувати в чистому вигляді лише 4 мутацію, а 1 і 2 мутації повинні використовуватись в поєднанні з мутацією 3.

1.6. Оператори відбору особин в нову популяцію

Для створення нової популяції можна використовувати різні методи відбору особин.

Відбір усіченням. При такому відборі використовують популяцію, яка складається як з особин-батьків, так і з особин-нащадків, відсортовану по зростанню значень функції пристосованості особин. Кількість особин для схрещування вибирається в відповідності з пороговим значенням $T \in [0; 1]$. Поріг визначає, яка частина особин, починаючи з найпершої (з найбільш пристосованої), буде приймати участь у відборі. В принципі, поріг можна задати числом більше 1, тоді від буде просто дорівнювати кількості особин з поточної

популяції, яких було допущено до схрещування. Серед особин, які потрапили „під поріг”, випадковим чином вибирається одна і записується в нову популяцію. Процес повторюється N разів, доки розмір нової популяції не буде таким самим як і вихідна популяція. Нова популяція складається лише з особин з високою пристосованістю, причому одна і та сама особина може зустрічатись кілька разів, а деякі особини, що матимуть пристосованість вище порогової можуть не потрапити в нову популяцію.

Враховуючи той факт, що в цій стратегії використовується відсортована популяція, час її роботи може бути більшим для популяції більшого розміру і до того ж залежати від алгоритму сортування.

Елітарний відбір. Створюється проміжна популяція, яка включає в себе, як батьків, так і їх нащадків. Члени цієї популяції оцінюються, а потім з них обирається N найкращих особин, які складатимуть наступне покоління. Часто цей метод комбінують з іншими – вибирають в нову популяцію, наприклад, 10% „елітних” особин, а решта 90% - одним з традиційних методів селекції. Інколи ці 90% особин створюють випадково, як при створенні початкової популяції перед запуском роботи генетичного алгоритму. Використання стратегії елітизму виявляється дуже корисним для ефективності ГА, оскільки не допускає втрати кращих розв’язків. Наприклад, якщо популяція зійшлась в локальному максимуму, а мутація вивела один з рядків в область глобального, то при попередній стратегії ймовірно, що ця особина в результаті схрещування буде втрачена і розв’язок задачі так і не буде отримано. Якщо ж використовувати елітизм, то отриманий хороший розв’язок буде залишатись в популяції до тих пір, доки не буде знайдено ще кращий.

Відбір витісненням. В цьому відборі вибір особини в нову популяцію залежить не лише від величини її пристосованості, але і від того, чи є вже в популяції, що формується особина з алогічним хромосомним набором. Відбір проводиться з батьками та їх нащадками. З усіх особин з однаковою пристосованістю перевага спочатку віддається особинам з різними генотипами. Таким чином, досягається дві мети: по-перше, не втрачаються кращі знайдені розв’язки, які мають різні хромосомні набори, по-друге, в популяції постійно

підтримується генетична різноманітність. Витіснення в цьому випадку швидше формує нову популяцію з видалених особин, замість особин, які групуються біля поточного знайденого розв'язку. Цей метод найкраще використовувати для багатоекстремальних задач, при цьому крім визначення глобальних екстремумів з'являється можливість виділити і локальні максимуми, значення яких близькі до глобальних.

Метод Больцмана, або метод відпалювання. В цьому методі ймовірність відбору в нову популяцію залежить від параметра керування – температури T .

Зазвичай ймовірність потрапляння в нову популяцію обчислюється по формулі:

$$p = \frac{1}{1 + \exp \frac{f(i) - f(j)}{T}},$$

(2.6)

де $f(i)$, $f(j)$ – значення цільової функції i і j особин, відповідно. Номера особин i і j вибираються випадково. Якщо значення p виявиться більше випадкового числа на інтервалі $(0;1)$, в нову популяцію потрапить особина $f(i)$, інакше $f(j)$.

В деяких випадках використовується альтернативна формула:

$$p = \frac{\exp(f(i)/T)}{\langle \exp(f(i)/T) \rangle},$$

(2.7)

де $\langle \rangle$ - середнє значення по популяції на ітерації з номером t . Якщо p виявиться більше випадкового числа на інтервалі $(0;1)$, то в нову популяцію потрапить особина $f(i)$.

В цьому методі першим поколінням відповідають високі температури і ймовірність відбору особин велика (підтримується різноманітність в новій популяції). Поступово зі збільшенням кількості поколіть ГА температура зменшується, ймовірність відбору зменшується і в нову популяцію потрапляють ті особини, пристосованість яких мінімальна. Цей метод відбору використовується для вузького класу задач (наприклад, знаходження основного стану спінових скелець) [6].

1.7 Різноманітність генетичних алгоритмів

Варто відмітити, що сьогодні ГА – це цілий клас алгоритмів, направлений на розв’язок різноманітних задач. Розглянемо деякі приклади різноманітних генетичних алгоритмів.

1.7.1. Канонічний ГА

Це класична модель ГА. Вона була запропонована Джоном Холладном в його роботі [8]. Відповідно до неї, популяція складається з N хромосом з фіксованою розрядністю генів. За допомогою пропорційного відбору формується проміжний масив, з якого випадковим чином вибираються для батьки. Потім здійснюється одно точковий кросинговер, і створені два нащадки мутують (одноточкова мутація) з заданою ймовірністю. Мутовані нащадки займають місця своїх батьків. Процес продовжується до тих пір, доки не буде досягнуто критерію закінчення алгоритму.

1.7.2. Генітор

В цій моделі використовується специфічний спосіб відбору [4]. На початку ініціалізується популяція і її особини оцінюються. Потім вибираються випадковим чином дві особини, схрещуються, причому отримуємо лише одного нащадка, який оцінюється і займає місце менш пристосованої особини в популяції (а не одного з батьків!). Після цього знову випадковим чином вибираються дві особини, і їх нащадок займає місце батьківської особини з найнижчою пристосованістю. Таким чином на кожному кроці в популяції обновлюється лише одна особина. Процес продовжується до тих пір, доки пристосованість хромосом не буде однаковою. В цей алгоритм можна додати мутацію нащадка після його створення. Критерій закінчення процесу, як і вид кросинговера і мутації можна вибирати різними способами.

1.7.3. Метод переривчатої рівноваги

Даний метод базується на палеонтологічній теорії переривчатої рівноваги, яка описує швидку еволюцію за рахунок вулканічних а інших змін земної кори. Для використання цього метода в технічних задачах пропонується після кожної генерації проміжного покоління випадковим чином перемішували особини в популяції, а потім використовувати основний ГА. В цій моделі для вибору батьківських па використовується панміксія. Отримані в результаті кросингвера нащадки і найбільш пристосовані батьки випадковим чином змішуються. З загальної кількості в нове покоління потраплять особини лише з вищою за середню пристосованість. Цим досягається керування розміром популяції в залежності від наявності кращих особин. Така модифікація метода перервного розмноження може дозволити зменшити неперспективні популяції і розширити популяції і яких знаходяться кращі індивідууми. Метод перервного розмноження – це потужний стресовий метод зміни навколишнього середовища, який використовується для ефективного виходу з локальних ям.

1.7.4. Гібридний алгоритм (Hybrid algorithm (Davis))

Використання гібридного алгоритму дозволяє об'єднати переваги ГА з перевагами класичних методів.

Річ у тому, що ГА - це *робастні* алгоритми, тобто дозволяють знаходити хороший розв'язок, але знаходження оптимального часто виявляється набагато важчим завданням через стохастичність принципів роботи алгоритму. Тому виникла ідея використовувати ГА на початковому етапі для ефективного звуження простору пошуку околу глобального екстремуму, а потім, узявши кращу особину використати один з «класичних» методів оптимізації.

Проте можна використовувати "класичні" методи (hill-climbing, наприклад) і всередині самих ГА. На кожному поколінні кожен отриманий нащадок оптимізується цим методом, таким чином, кожна особина досягає локального максимуму, поблизу якого вона знаходиться, після чого піддається відбору, схрещуванню і мутації. Такий метод погіршує здатність алгоритму шукати розв'язок за допомогою відбору гіперплощин, але зростає ймовірність того, що

одна з особин потрапить в область глобального максимуму і після оптимізації виявиться розв'язком задачі.

1.7.5. CHC (Eshelman)

CHC – це Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation. Для нового покоління вибираються N кращих різних особин серед батьків і дітей. Дублювання рядків не допускається. Для схрещування всі особини розбиваються на пари, але схрещуються лише ті пари, між якими відстань Хеммінга більше деякого порогового (також можливі обмеження на мінімальну відстань між крайніми бітами, що відрізняються). При схрещуванні використовується так званий Нух-оператор (Half Uniform Crossover), різновид однорідного кроссовера - кожному нащадкові переходить рівно половина бітів кожного батька. Розмір популяції невеликий. Цим виправдано використання однорідного кроссовера. Даний алгоритм досить швидко сходиться через те, що в немає немає мутацій. В цьому випадку CHC застосовує cataclysmic mutation: всі рядки, окрім найпристосованішої, піддаються сильній мутації (змінюється біля третини бітів). Таким чином, алгоритм перезапускається і далі продовжує роботу, застосовуючи лише кроссовер

1.7.6. ГА з нефіксованим розміром популяції

В генетичному алгоритмі з нефіксованим розміром популяції кожній особині приписується максимальний вік, тобто кількість поколінь, по закінченні яких особина вимиратиме. Впровадження в алгоритм нового параметра – віку дозволяє виключити оператор відбору в нову популяцію. Вік кожної особини індивідуальний і залежить від її пристосованості. В кожному поколінні t на етапі відтворення звичайним чином створюється довільним чином податкова популяція з нащадків. Розмір додаткової популяції ($AuxPopsizе(t)$) пропорційний розміру основної і дорівнює ($Popsizе(t)$) :

$$AuxPopsizе(t) = [Popsizе(t)p_c],$$

(2.8)

де p_c – ймовірність відтворення. Для відтворення особини вибираються з основної популяції з однаковою ймовірністю незалежно від пристосованості. Після використання мутації і кросинговера нащадки отримують вік відповідно значенню пристосованості. Вік – це константа протягом всієї еволюції особини (від народження до загибелі). Потім з основної популяції видаляються ті особини, термін життя яких закінчився і додаються нащадки з проміжної популяції. Таким чином розмір після однієї ітерації алгоритму обчислюється по формулі:

$$Popsiz e(t+1) = Popsiz e(t+1) + AuxPopsiz e(t) - D(t),$$

(2.9)

де $D(t)$ – кількість особин, які вимирають в поколінні t [7].

1.8 Паралельне виконання ГА

Генетичні алгоритми використовуються і при паралельних обчисленнях (Parallel implementations). При цьому формується кілька популяцій, що живуть окремо. На етапі формування нового покоління по певному правилу вибираються особини з різних популяцій. Згенерована таким чином популяція в деяких випадках замінить всіх, що залишились. Таким чином, так чи інакше, відбувається міграція особин однієї популяції в інші популяції. Тому паралельні ГА називають міграціями.

1.8.1 Паралельний ГА

Для початку розглянемо створення найпростішого паралельного ГА з класичної моделі Холланда. Для цього будемо використовувати турнірний відбір. Зведемо $N/2$ процесі (тут і далі процес розглядається як, певний механізм, процесор, який може працювати незалежно). Кожен з них буде вибирати випадково з популяції 4 особини, проводити 2 турніри і схрещувати переможців. Отримані нащадки будуть записуватись в нове покоління. Таким чином, за один цикл роботи одного процесу пробуде змінюватись ціле покоління [8].

1.8.2. Міграція

Модель міграції представляє собою популяцію як множину під популяцій. Кожна під популяція обробляється окремим процесором. Ці під популяції розвиваються незалежно один від одного протягом однакової кількості поколінь T (час ізоляції). По закінченню часу ізоляції відбувається обмін особинами між під популяціями (міграція). Кількість особи, які були замінені (ймовірність міграції), метод відбору особин для міграції і схема міграції визначає частоту виникнення генетичного різноманіття в під популяціях і обмін інформацією між під популяціями.

Відбір особин для міграції може здійснюватись таким чином:

- Випадкова одноманітна вибірка з кількості особин;
- Пропорційний відбір: для міграції беруться найбільш пристосовані особини.

Окремі підпопуляції в паралельних ГА можна умовно прийняти за вершини певного графа. В зв'язку з цим можна розглядати топологію графа міграційного ГА. Найбільш розповсюдженою топологією міграцією є повний граф (див. мал), при якій особини з будь-якої підпопуляції можуть мігрувати в будь-яку іншу підпопуляцію. Для кожної підпопуляції повна кількість потенційних іммігрантів будується на основі всіх під популяцій. Особина, що мігрує випадковим чином вибирається з тієї ж загальної кількості.

При використанні в необмеженій міграції пропорційного відбору спочатку формуємо маси з найбільш пристосованих осіб, відібраних по всіх підпопуляції. Випадковим чином з цього масиву вибираємо особину і замінюємо нею найменш пристосовану в підпопуляції 1. Аналогічні дії здійснюємо з рештою підпопуляцій. Ймовірно, що одна з популяцій отримає дублікат своєї „хорошої” особини.

Інша основна міграційна схема – це топологія кільця (див. мал.). Тут особини передаються між сусідніми (по напрямку обходу) під популяціями. Таким чином, особини з однієї підпопуляції можуть мігрувати лише в одну – сусідню підпопуляцію.

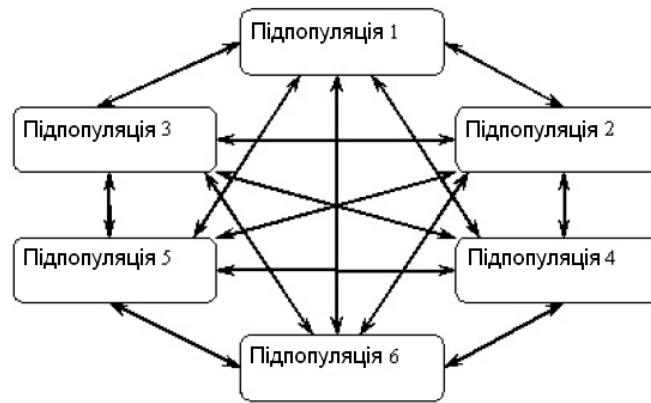


Рисунок 2.7 – Міграція з топологією повної мережі

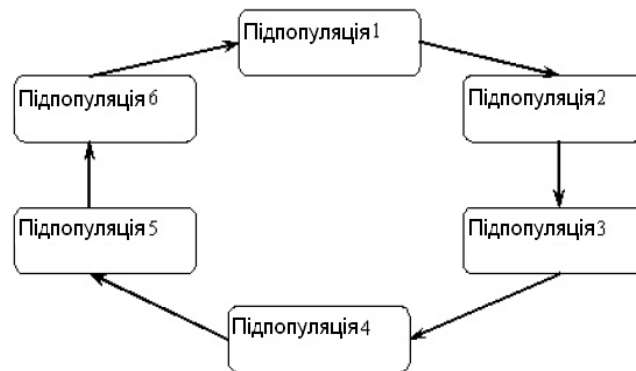


Рисунок 2.8 – Міграція з топологією кільця

На малюнку представлена стратегія міграції, схожа на топологію кільця. Як і при топології кільця, міграція здійснюється між найближчими сусідами. Але міграція в цій моделі теж можлива між крайніми під популяціями (тороїд альні краєві міграції).

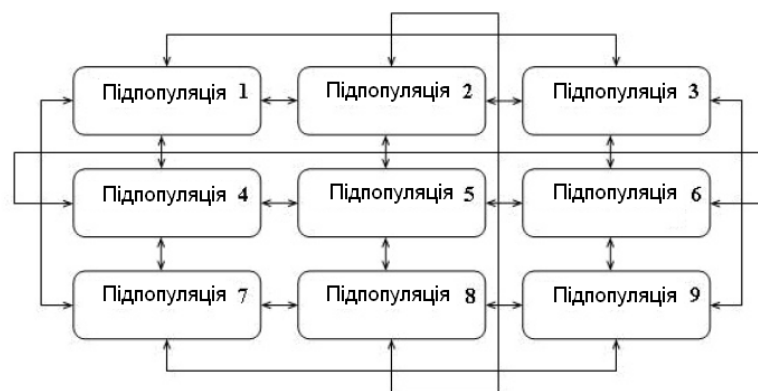


Рисунок 2.9 – Міграція з модифікованою топологією кільця

1.8.3 Глобальна модель „Робочий і Господар”

Алгоритм „Робочий і Господар” (Global model – worker/farmer) реалізується з використанням декількох комп’ютерів, що працюють одночасно (див. рис). Серед всіх комп’ютерів виділяють „Господаря” і „Робочих”. Комп’ютери – „Робочі” відповідають за процеси відтворення, мутації і обчислення функції пристосованості особин для їх відбору в нове покоління. Всі створені і оцінені „Робочими” особини передаються комп’ютеру „Господарю”, який потім проводить відбір особин відповідно оцінки їх пристосованості в нову популяцію. Відібрані особини передаються „Господарем” до комп’ютерів – „Робочим” [9].

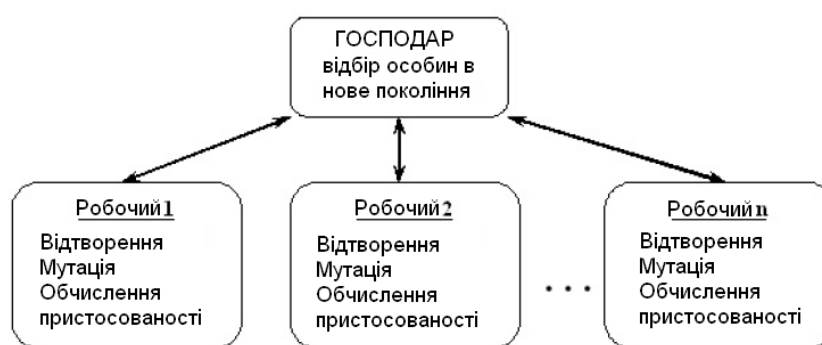


Рисунок 2.10 – Модель „Робочий і Господар”

1.8.4. Модель дифузії, або острівна модель ГА (Island Models)

Острівна модель (island model) — модель паралельного генетичного алгоритму. Розіб'ємо популяцію на декілька підпопуляцій. Кожна з них розвиватиметься окремо за допомогою певного генетичного алгоритму. Таким чином, можна сказати, що ми розселили особини по декількох ізольованих островах.

Зрідка (наприклад, кожні 5 поколінь) відбувається міграція – острови обмінюються кількома хорошими особинами.

Оскільки населеність островів невелика, то підпопуляції будуть схильні до передчасної збіжності. Тому важливо правильно встановити частоту міграції:

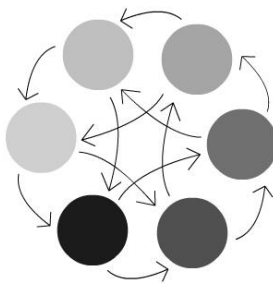


Рисунок 2.11 – Острівна модель

занадто часта міграція (або міграція дуже великої кількості особин) призведе до змішування всіх підпопуляцій, і тоді острівна модель не буде дуже відрізнятися від звичайного ГА; якщо міграція буде відбуватись рідко, то вона не зможе запобігти передчасному сходженню підпопуляцій.

Генетичні алгоритми стохастичні, тому при різних запусках популяція може сходитися до різних хороших розв'язків. Острівна модель дозволяє запустити алгоритм відразу кілька разів і поєднати «досягнення» різних островів для отримання якнайкращого розв'язку [10].

1.9. Параметри ГА

Ймовірність кросинговера зазвичай обирається доволі високою 80–95%. Але в деяких задачах найкращий результат досягається при кросинговері з ймовірністю 60%.

Ймовірність мутацій повинна бути незначною: 0,5–1%.

Як не дивно, але дуже великий розмір популяції зазвичай не призводить до хороших результатів (швидкість збіжності алгоритму не збільшується). Оптимальний розмір популяції 20-30 особин, але в деяких задачах, як пишуть дослідники необхідно 50-100 особин. Дослідження показують, що оптимальний розмір популяції залежить від розміру кодових рядків (хромосом). Так, для алгоритму з 32-бітовими хромосомами розмір популяції буде більшим ніж для алгоритму з 16-бітовими хромосомами.

В основному використовується відбір рулеткою, але інколи краще використовувати відбір усіченням. Існує багато інших методів, що змінюють параметри відбору протягом всієї роботи ГА. Елітизм використовується, якщо не використовуються інші методи, що зберігають знайдений хороший розв'язок.

Вибір способу кодування зумовлений поставленою задачею і розміром об'єкта пошуку.

Оператори життєвого циклу (кросинговера і мутацій) визначаються вибраним кодуванням.

1.10. Модернізація ГА

Однією з серйозних проблем, що виникають при використанні генетичних алгоритмів, є передчасна збіжність. Не рекомендується використовувати класичні ГА на маленьких популяціях, оскільки в популяціях з малим розміром гени розповсюджуються дуже швидко: всі особини стають схожими (популяція вироджується) ще до того, як буде знайдено розв'язок. Тобто новий генотип з кращою оцінкою швидко витіснить не такі хороші комбінації генів, виключаючи тим самим можливість отримати кращого розв'язку на їх основі. Можна запропонувати три основних шляхи усунення передчасної збіжності: збільшення розміру популяції, використання самоадаптованих генетичних операторів і створення „банку” особин для заміни.

В першому випадку, збільшуючи розмір популяції, можна сподіватись на досягнення різноманітності генотипу в популяції. Але з іншого боку, збільшення числа особин призведе до збільшення пам'яті і часу роботи алгоритму. Цей підхід може бути ефективним або при паралельних обчисленнях, або при наявності досить простої цільової функції.

Другий і найбільш розповсюджений спосіб, – використання самоадаптованих алгоритмів є і найбільш ефективним. Самоадаптація полягає в використанні динамічних мутацій. Динамічні мутації в залежності від особин, що схрещуються змінюють значення ймовірності мутації, тим самим стає можливим самокерування алгоритму. В таких випадках вибирається малий розмір популяції.

В третьому підході створюється масив для збереження особин, генотип яких був втрачений при формуванні нових поколінь і інколи ці особини додаються в популяцію [11].

1.11. Налаштування ГА

Генетичний алгоритм проводить пошук розв'язків за допомогою:

- відбору гіперплощин (hyperplane sampling) шляхом кросинговера
- методу hill-climbing шляхом мутації

Дослідження показали, що на простих задачах з малим розміром популяції ГА з мутацією (і без кросинговера) розв'язок отримують швидше, а на складних багатоекстремальних функціях краще використовувати ГА з кросинговером, оскільки цей метод надійніший.

З точки зору теореми шаблонів, мутація лише шкодить зростанню кількості представників хороших шаблонів, зайвий раз руйнуючи їх. Але мутація необхідна для ГА з малим розміром популяції, бо для них властива передчасна збіжність (premature convergence) – ситуація, коли в деяких позиціях всі особини мають один і той же біт, не відповідний глобальному екстремуму.

Тиск відбору (selection pressure) — міра того, наскільки відрізняються шанси кращої і гіршої особин популяції потрапити в проміжну популяцію. Для пропорційного відбору ця величина із збільшенням середньої пристосованості популяції зменшується, прямуючи до 1. Для ефективної роботи генетичного алгоритму необхідно підтримувати рівновагу між дослідженням і використанням.

- При збільшенні ймовірності схрещування або мутації і зменшенні тиску відбору (за рахунок використання інших стратегій відбору) розмноження представників пристосованих шаблонів сповільнюється, та зате відбувається інтенсивний пошук інших шаблонів.
- Зменшення ймовірності схрещування або мутації і збільшення тиску відбору веде до інтенсивного використання знайдених хороших шаблонів, але менше уваги приділяється пошуку нових.

Необхідність збалансованої збіжності ГА:

- швидка збіжність може привести до сходження до неоптимального розв'язку

- повільна збіжність часто приводить до втрати знайденої якнайкращої особини.

1.12. Модифікації ГА

▪ Алфавіт

Аргументи на користь кодування бінарним алфавітом: забезпечує кращий пошук за допомогою гіперплощин, оскільки надає максимальну їх кількість. Наприклад, при кодуванні 2^L значень для бінарного алфавіту кількість гіперплощин буде 3^L , а при використанні, чотиризначного алфавіту – $5^{L/2}$. Для того, щоб кожен символ зустрічався в кожній позиції потрібен менший розмір популяції. Навіть для двох рядків, існує ймовірність, що на кожній позиції в популяції є і 0, і 1. Якщо ж алфавіт більшої потужності, то до використання мутації велика частина простору пошуку буде недоступна з точки зору кросинговера, після використання мутації стане недоступна інша частина. Проте небінарні алфавіти часто забезпечують наочніше представлення розв'язків задачі.

▪ Кодування параметрів

Для більшості функцій ГА працюватимуть краще при кодуванні параметрів кодом Грея, а не прямим бінарним кодом. Це пов'язано з тим, що відстань Хеммінга не завжди є критерієм близькості – наприклад, числа 7 і 8 розрізняються на 4 біта. Бінарне кодування додає додаткові розриви, що ускладнює пошук. Приклад: нехай потрібно мінімізувати функцію $f(x) = x^2$. Якщо в початковій популяції переважали хороші від'ємні рішення, то швидше за все ми прийдемо до розв'язку $-1 = 11\dots1$. Але досягти глобального мінімуму $00\dots0$ буде практично неможливо, оскільки зміна будь-якого біта приводитиме до погіршення розв'язку. При кодуванні кодом Грея такої проблеми не виникає. Інколи застосовується кодування з плаваючою точкою, яке теж є більш вдалим, ніж пряме бінарне.

▪ Стратегії відбору

Ранговий відбір: для кожної особини її ймовірність потрапити в проміжну популяцію пропорційна її порядковому номеру у відсортованій за зростанням

пристосованості популяції. Такий вид відбору не залежить від середньої пристосованості популяції.

Турнірний відбір (tournament selection): з популяції випадковим чином вибирається t особин, і краща з них поміщається в проміжну популяцію. Цей процес повторюється N разів, доки проміжна популяція не буде заповнена. Найбільш поширений варіант при $t = 2$. Турнірний відбір агресивнішим, ніж пропорційний.

Відбір усіченням (truncation selection): популяція сортується по пристосованості, потім береться задана частка кращих, і з них випадковим чином N разів вибирається особина для подальшого розвитку.

▪ Кроссовер

Двоточковий кроссовер: вибираються 2 точки розділу, і батьки обмінюються проміжками між ними: При цьому визначальна довжина вимірюється в кільці – для шаблону 1*****1 при двоточковому кроссовері вона дорівнюватиме 1, хоча при одноточковому була 6.

Однорідний кроссовер: один з дітей успадковує кожен біт з ймовірністю p_0 у першого батька і з $(1 - p_0)$ в другого, друге дитя отримує не успадковані першим біти. Зазвичай $p_0 = 0.5$. Однорідний кроссовер в більшості випадків руйнує шаблон, тому його краще не використовувати для відбору гіперплощин, проте при малому розмірі популяції він перешкоджає передчасному сходженню.

▪ Стратегії формування нового покоління

Два основні типи формування нового покоління після кроссовера і мутації:

- діти заміщають батьків
- нове покоління складається з сукупності і дітей, і їх батьків

Також застосовується принцип елітизму: у нове покоління включається задана кількість кращих особин попереднього покоління (часто одна краща особина). Використання другої стратегії і елітизму не допускає втрати кращих розв'язків. Наприклад, якщо популяція зійшлася в локальному максимумі, а мутація вивела один з рядків в область глобального, то при заміщенні батьків мабуть, що ця особина в результаті схрещування буде втрачена, і рішення

задачі не буде отримано. Якщо ж використовується елітизм, то отриманий хороший розв'язок залишатиметься в популяції до тих пір, доки не буде знайдено кращий [12].

1.13. Переваги та недоліки ГА

Генетичні алгоритми мають такі переваги:

- ГА не потрібно жодної інформації про поведінку функції (наприклад диференційованості і неперервності)
- Розриви, які існують на поверхні відповіді, мають незначний ефект на повну ефективність оптимізації.
- ГА доволі стійкі до потрапляння в локальний оптимум.
- ГА підходять для розв'язку багатомасштабних проблем оптимізації.
- ГА можна використовувати для широкого класу задач
- ГА прості в реалізації
- ГА можна використовувати для задач зі змінним середовищем.

Фактори, що створюють складності для генетичних алгоритмів можна розбити на такі групи:

Властивості функцій пристосованості, що створюють складність для ГА:

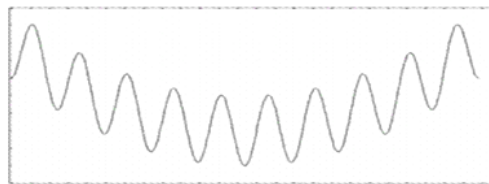


Рисунок 2.12 – Функція Растрінга

Багатоекстремальність: створюється множина помилкових аттракторів.

Приклад — функція Растрігіна:

Оманливість (desertion): функція побудована так, що шаблони низького порядку відводять популяцію до локального екстремуму [13].

Приклад: нехай рядок складається з 10-ти чотирибітних підрядки. Нехай u_i дорівнює кількості одиниць в i -тому підрядку. Задамо функцію $g(u)$ такою таблицею:

u 0 1 2 3 4

$g(u)$ 3 2 1 0 4

і нехай функція пристосованості дорівнює сумі $g(u_i)$ по всіх $i = 1..10$:

$$f = \sum_{i=1}^{10} g(u_i)$$

(2.9)

Ізольованість («пошук голки в копні сіна»): функція не надає жодної інформації, що підказує, в якій області шукати максимум. Лише випадкове попадання особини в глобальний екстремум може вирішити задачу [14].

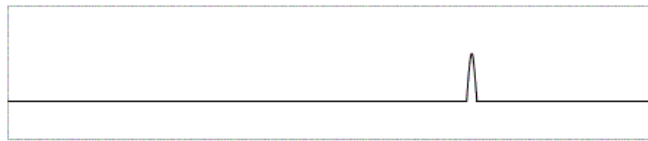


Рисунок 2.14 – Ізольована функція

Додатковий шум (noise): значення пристосованості шаблонів дуже розкидані, тому часто навіть хороші гіперплощини низького порядку не проходять відбір, що сповільнює пошук розв'язку.

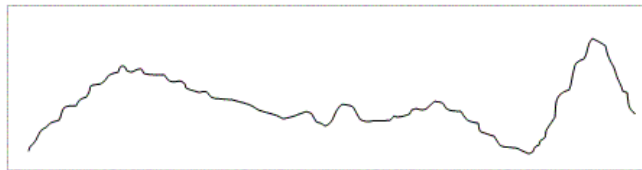


Рисунок 2.15 – „Зашумлена” функція

Таким чином можна виділити такі особливості використання генетичних алгоритмів:

Генетичні алгоритми – це універсальний метод оптимізації багатопараметричних функцій, а це дозволяє розв'язувати широкий спектр задач.

Генетичні алгоритми мають безліч модифікацій і дуже сильно залежать від параметрів. Часто невелика зміна одного з них може привести до неочікуваного покращення результату

Необхідно пам'ятати, що використання ГА слід лише в тих випадках, коли для даної задачі немає відповідного спеціального алгоритму розв'язку.

Сьогодні ведуться пошуки генетичних стратегій, які б усунули всі недоліки практичного використання генетичних алгоритмів. Першими кроками тут є

різноманітні види мутацій та алгоритми вибору батьків для формування нащадка. На сьогодні існує принаймні три класи задач, які можуть бути розв'язані за допомогою ГА:

- Задача швидкої локалізації одного оптимального розв'язку
- Задача визначення кількох (чи всіх) глобальних екстремумів
- Задача опису ландшафту функції, яка може супроводжуватись виділенням не лише глобальних але і локальних оптимумів.

Простір пошуку розв'язку другої задачі досягається за рахунок певного поєднання параметрів і великої чисельності популяції. При цьому ГА може виділити кілька екстремумів. Для виділення кількох глобальних максимумів найбільше підходить дослідження таких параметрів як аутбридінгу в поєднанні з інбридингом для вибору особин в нову популяцію. Максимальна ефективність досягається в поєднанні аутбридінгу на початку пошуку для отримання максимально широкого «дослідження» та інбридінгу в кінці з метою уточнення розв'язку в локальних групах [15].