ЛЕКЦІЯ 9.
ЕВОЛЮЦІЙНИЙ ПІДХІД ТА ОСНОВИ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Питання лекції

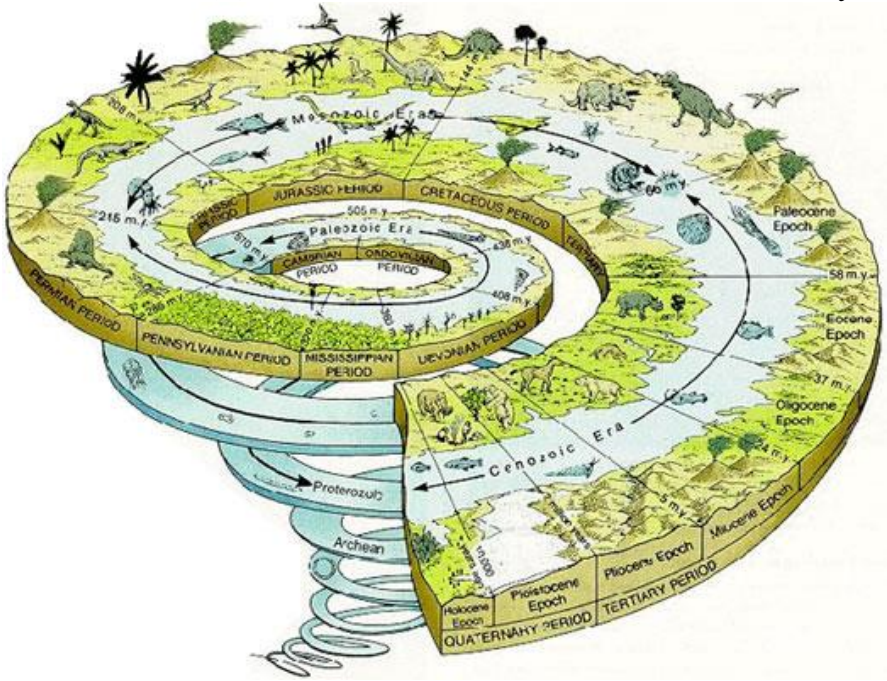
Вступ
1. Природний відбір у природі
2. Основні поняття генетичних алгоритмів
3. Особливості генетичних алгоритмів
4. Задачі оптимізації і застосування алгоритмів
5. Опис типового генетичного алгоритму
6. Класичний генетичний алгоритм
Висновки

ВСТУП

Для створення інтелектуальних систем часто застосовують **еволюційний підхід**, що умовно можна поділити на:

• **Еволюційні алгоритми** (моделювання загальних закономірностей еволюції). Це системи, які використовують ***еволюційні принципи розвитку популяції***. Вони успішно використовуються для завдань функціональної оптимізації і можуть легко бути описані математичною мовою.

• **Еволюційні моделі**. Це системи, які відтворюють біологічні популяції чи системи і поки що не є корисними в прикладному сенсі. Еволюційні моделі більше схожі на біологічні системи, мають складну поведінку, мало спрямовані на вирішення технічних завдань. До цих систем відносять так зване «штучне життя».



Еволюційні алгоритми

До них відносяться:
• Генетичні алгоритми

• Мурашині алгоритми
• Еволюційні стратегії
• Еволюційне програмування
• Генетичне програмування

Еволюційні алгоритми - термін, що часто використовується для загального опису алгоритмів пошуку, оптимізації або навчання, що засновані на формалізованих принципах природного еволюційного процесу.

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

Основною перевагою еволюційних методів оптимізації **є можливості вирішення багатомодальних (з кількома локальними екстремумами) завдань з великою розмірністю** за рахунок поєднання елементів випадковості і детермінованості точно так само, як це відбувається у природному середовищі. Детермінованість цих методів полягає в моделюванні природних процесів відбору, що відбуваються за строго визначеними правилами, основним з яких є закон еволюції: «перемагає найсильніший».

Іншим важливим чинником ефективності еволюційних алгоритмів є відтворення процесів розвитку. Розглянуті варіанти рішень можуть за певним правилом породжувати нові рішення, які будуть на - слідувати кращі риси своїх попередників. В якості випадкового елемента в методах еволюційних обчислень використовується моделювання процесу мутації. З її допомогою характеристики того чи іншого рішення можуть бути випадково змінені, що призведе до нового напрямку в процесі еволюції рішень і може прискорити процес вироблення кращого рішення.

Переваги еволюційних алгоритмів

• Широка область застосування.
• Можливість проблемно - орієнтованого кодування рішень, підбору початкових умов, комбінування еволюційних обчислень з не еволюційними алгоритмами, продовження процесу еволюції до тих пір, поки є необхідні ресурси.
• Придатність для пошуку в складному просторі рішень великої розмірності.
• Відсутність обмежень на вид цільової функції.
• Ясність схеми і базових принципів еволюційних обчислень.
• Інтегрованість еволюційних обчислень з іншими некласичними парадигмами штучного інтелекту, такими як штучні нейромережі та нечітка логіка.

Недоліки еволюційних алгоритмів

• Евристичний характер еволюційних обчислень не гарантує оптимальності отриманого рішення (правда, на практиці, найчастіше, важливо за заданий час

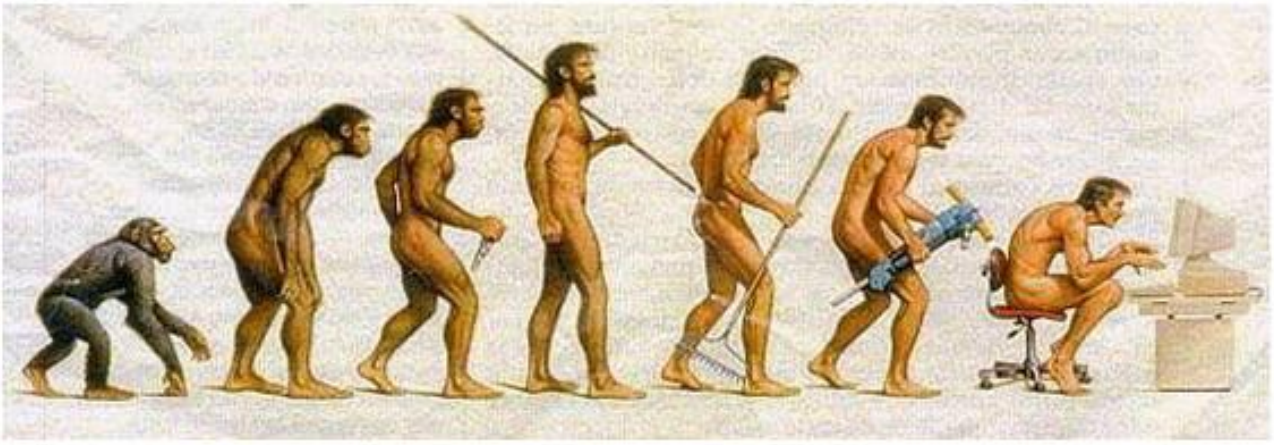
отримати одне або кілька субоптимальних альтернативних рішень, тим більше, що вихідні дані в задачі можуть динамічно змінюватися, бути неточними або неповними).

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

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

• Невирішеність питань самоадаптації.

Еволюційна теорія стверджує, що кожен біологічний вид цілеспрямовано розвивається і змінюється для того, щоб щонайкраще пристосуватися до навколишнього середовища. У процесі еволюції багато видів комах і риб придбали захисне фарбування, їжак став невразливим завдяки голкам, людина стала власником дуже складної нервової системи. *Можна сказати, що еволюція - це процес оптимізації всіх живих організмів.* Розглянемо, якими ж засобами природа вирішує цю задачу оптимізації.

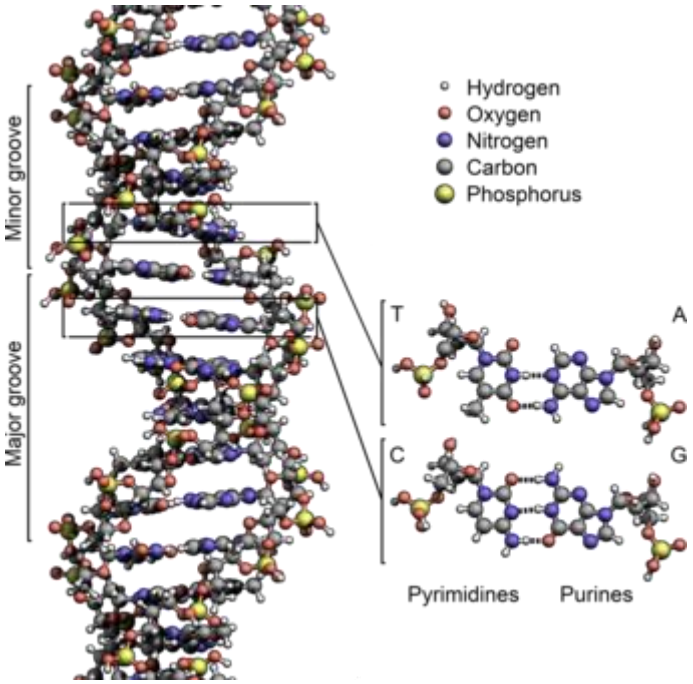


Основний механізм еволюції - це природний відбір. *Його суть полягає в тому, що більш пристосовані особи мають більше можливостей для виживання і розмноження і, отже, приносять більше нащадків, ніж погано пристосовані особи.* При цьому завдяки передачі генетичної інформації (генетичному спадкуванню) нащадки успадковують від батьків основні їхні якості. Таким чином, нащадки сильних індивідуумів також будуть відносно добре пристосованими, а їхня частка в загальній масі особин буде зростати. Після зміни декількох десятків або сотень поколінь середня пристосованість особин даного виду помітно зростає.

1. Природний відбір у природі

Щоб зробити зрозумілими принципи роботи генетичних алгоритмів, пояснимо також, як улаштовані механізми генетичного спадкування в природі. У кожній клітині будь-якої тварини утримується вся генетична інформація цієї особи. Ця інформація записана у виді набору дуже довгих молекул ДНК (Дезоксирибо Нуклеїнова Кислота). Кожна молекула ДНК - це ланцюжок, що складається з молекул нуклеотидів чотирьох типів, що позначаються A, T, C і G.

Власне, інформацію несе порядок проходження нуклеотидів у ДНК. Таким чином, генетичний код індивідума - це просто дуже довгий рядок символів, де використовуються всього 4 букви. У тваринній клітині кожна молекула ДНК оточена оболонкою - таке утворення називається *хромосомою*.



Кожна уроджена якість особи (колір ока, спадкоємні хвороби, тип волосся і т.д.) кодується визначеною частиною хромосоми, що називається *геном* цієї властивості. Наприклад, ген кольору ока містить інформацію, що кодує визначений колір очей. Різні значення гена називаються його *алелями*.

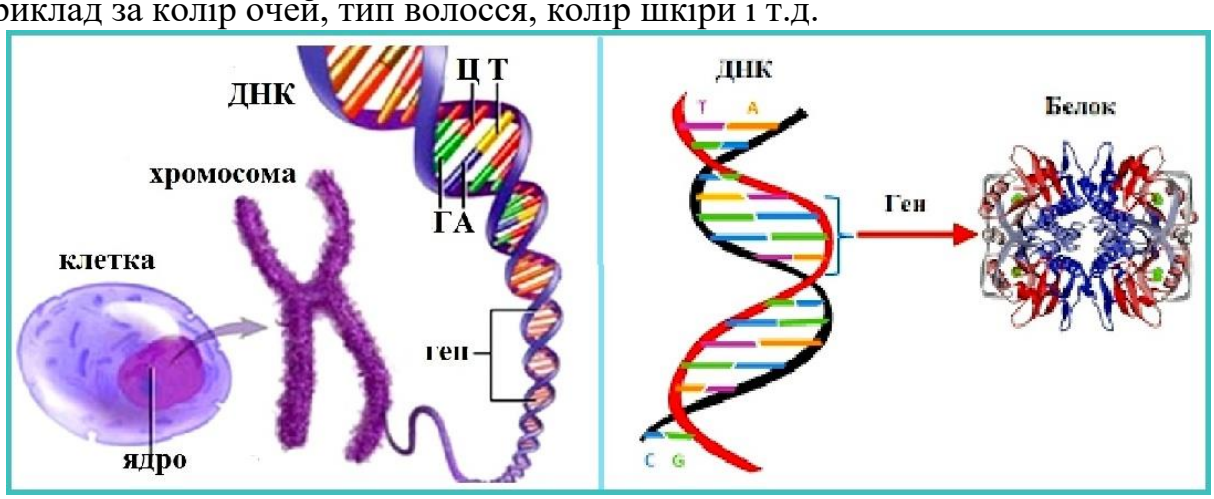
При розмноженні тварин відбувається злиття двох батьківських статевих клітин і їхні ДНК взаємодіють, утворюючи ДНК нащадка. Основний спосіб взаємодії - **кроссовер** (cross-over, схрещування). При кроссовері ДНК предків розділяються на дві частини, а потім обмінюються своїми половинками.

При спадкуванні можливі **мутації** через радіоактивність або інші впливи, у результаті яких можуть змінитися деякі гени в статевих клітинах одного з батьків. Змінені гени передаються нащадкові і додають йому нові властивості. Якщо ці нові властивості корисні, вони, швидше за все, збережуться в даному виді - при цьому відбудеться стрибкоподібне підвищення пристосованості виду. Ключову роль в еволюційній теорії грає **природний відбір**. Його суть полягає в тому, що найбільш пристосовані особи краще виживають і приносять більше нащадків, чим менш пристосовані. Помітимо, що сам по собі природний відбір ще не забезпечує розвиток біологічного виду.

Тому дуже важливо зрозуміти, яким чином відбувається спадкування, тобто як властивості нащадка залежать від властивостей батьків. **Основний закон спадкування інтуїтивно зрозумілий кожному - він полягає в тому, що нащадки схожі на батьків.** Зокрема, нащадки більш пристосованих батьків будуть, швидше за все, одними з найбільш пристосованих у своєму поколінні. Щоб зрозуміти, на чому заснована ця подібність, потрібно трохи поглибитися в будову природної клітини - у світ генів і хромосом. Майже в кожній клітині будь-якої особи є набір

хромосом, що несуть інформацію про цю особу. Основна частина хромосоми - нитка ДНК, що визначає, які хімічні реакції будуть відбуватися в даній клітині, як вона буде розвиватися і які функції виконувати.

Ген - це відрізок ланцюга ДНК, відповідальний за визначену властивість особи, наприклад за колір очей, тип волосся, колір шкіри і т.д.



Уся сукупність генетичних ознак людини кодується за допомогою приблизно 60 тис. генів, довжина яких складає більш 90 млн. нуклеотидів.

Розрізняють два види клітин: статеві (такі, як сперматозоїд і яйцеклітина) і соматичні. У кожній соматичній клітині людини утримується 46 хромосом. Ці 46 хромосом - насправді 23 пари, причому в кожній парі одна з хромосом отримана від батька, а друга - від матері. Парні хромосоми відповідають за однакові ознаки - наприклад, батьківська хромосома може містити ген чорного кольору ока, а парна їй материнська - ген блакитного кольору. **Існують визначені закони, що керують участю тих або інших генів у розвитку особи.** Зокрема, у нашому прикладі нащадок буде чорнооким, оскільки ген блакитних очей є "слабким" і придушується геном будь-якого іншого кольору.

У статевих клітинах хромосом тільки 23, і вони непарні. При заплідненні відбувається злиття чоловічої і жіночої статевих клітин і виходить клітина зародка, що містить 46 хромосом. Які властивості нащадок одержить від батька, а які - від матері? Це залежить від того, які саме статеві клітини брали участь у заплідненні. Процес вироблення статевих клітин (так називаний мейоз) в організмі піддається ймовірностям, завдяки яким нащадки багато в чому відрізняються від своїх батьків. При мейозі, відбуваються наступне: парні хромосоми соматичної клітини зближаються впритул, потім їхні нитки ДНК розриваються в декількох випадкових місцях і хромосоми обмінюються своїми частинами (рис. 1).

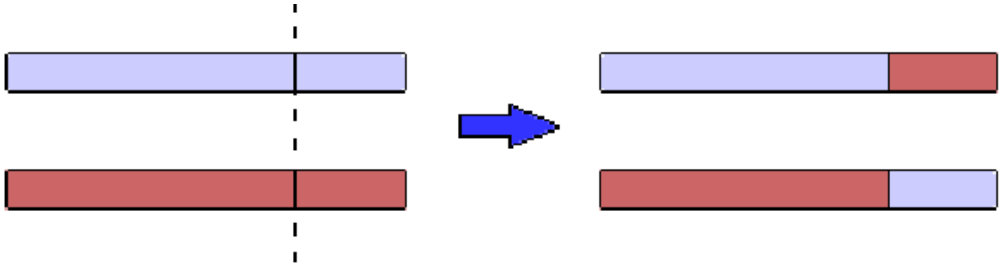


Рис. 1. Умовна схема кросінговера

Цей процес забезпечує появу нових варіантів хромосом і називається "кросінговер". Кожна із знову з'явившихся хромосом виявиться потім усередині однієї з статевих клітин, і її генетична інформація може реалізуватися в нащадках даної особи.

Другий важливий фактор, що впливає на спадковість, - це *мутації,* що виражаються в зміні деяких ділянок ДНК. Мутації також випадкові і можуть бути викликані різними зовнішніми факторами, такими, як радіоактивне опромінення. Якщо мутація відбулася в статевій клітині, то змінений ген може передатися нащадкові і проявитися у виді спадкоємної хвороби або в інших нових властивостях нащадка. Вважається, що саме мутації є причиною появи нових біологічних видів, а кросінговер визначає вже мінливість усередині виду (наприклад, генетичні розходження між людьми).

2. Основні поняття генетичних алгоритмів

Генетичний алгоритм (англ. Genetic algorithm) — це евристичний алгоритм пошуку, використовуваний для рішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Є різновидом еволюційних обчислень (англ. evolutionary computation). Відмінною рисою генетичного алгоритму є акцент на використання оператора «схрещування», що робить операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещування в живій природі. «Батьком-засновником» генетичних алгоритмів, вважається Джон Холланд (англ. John Holland), книга якого «Адаптація в природних і штучних системах» (англ. Adaptationin Naturaland Artificial Systems) є основною працею в цій області досліджень.

При описі генетичних алгоритмів використовуються означення, запозичені з генетики. Наприклад, мова йде про популяції особин, а як базові поняття застосовуються ген, хромосома, генотип, фенотип, алель. Також використовуються відповідним цим термінам означення з технічного лексикона, зокрема, ланцюг, двоїчна послідовність, структура.

Популяція - це кінцева множина особин.

Особи, що входять у популяцію, у генетичних алгоритмах представляються *хромосомами* з закодованим у них множинами параметрів задачі, тобто рішень, які інакше називаються точками в просторі пошуку (search points). У деяких роботах особи називаються організмами.

Хромосоми (інші назви - **ланцюжки або кодові послідовності**) - це упорядковані послідовності генів.

Ген (також називаний властивістю, знаком або детектором) - це атомарний елемент генотипу, зокрема, хромосоми.

Генотип або структура - це набір хромосом даної особи. Отже, особами популяції можуть бути генотипи або одиничні хромосоми (у досить розповсюдженому випадку, коли генотип складається з однієї хромосоми).

Фенотип - це набір значень, що відповідають даному генотипові, тобто декодована структура або множина параметрів задачі (рішення, точка простору пошуку).

Алель - це значення конкретного гена, також обумовлене як значення властивості або варіант властивості.

Локус або позиція вказує місце розміщення даного гена в хромосомі (ланцюжку). Множина позицій генів - це **локи**.

Дуже важливим поняттям у генетичних алгоритмах вважається **функція пристосованості (fitness function)**, інакше називана **функцією оцінки**. Вона представляє міру пристосованості даної особи в популяції. Ця функція відіграє найважливішу роль, оскільки дозволяє оцінити ступінь пристосованості конкретних особин у популяції і вибрати з них найбільш пристосовані (тобто найбільші значення функції, що мають пристосованості) відповідно до еволюційного принципу виживання «найсильніших» (найкраще пристосувалися). Функція пристосованості також одержала свою назву безпосередньо з генетики. Вона впливає на функціонування генетичних алгоритмів і повинна мати точне і коректне означення. У задачах оптимізації функція пристосованості, як правило, оптимізується (точніше кажучи, максимізується) і називається **цільовою функцією.** У задачах мінімізації цільова функція перетворюється, і проблема зводиться до максимізації. У теорії керування функція пристосованості може приймати вид *функції похибки*, а в теорії ігор - *вартісної функції*. На кожній ітерації генетичного алгоритму пристосованість кожної особи даної популяції оцінюється за допомогою *функції пристосованості*, і на цій основі створюється наступна популяція особин, що складають множину потенційних рішень проблеми, наприклад, задачі оптимізації.

Чергова популяція в генетичному алгоритмі називається **поколінням**, а до знову створюваної популяції особин застосовується термін «нове покоління» або «покоління нащадків».

Приклад

Нехай дана деяка складна функція (цільова функція), що залежить від декількох змінних, і потрібно знайти такі значення змінних, при яких значення функції максимальне. Задачі такого роду називаються задачами оптимізації і вони зустрічаються на практиці дуже часто.

Один з найбільш наочних прикладів - *задача розподілу інвестицій*. У цій задачі змінними є обсяги інвестицій у кожен проект (10 змінних), а функцією, яку потрібно максимізувати - сумарний дохід інвестора. Також приведені значення мінімального і максимального об’єму вкладення в кожний із проектів, що задають область зміни кожної із змінних.

Спробуємо розв’язати цю задачу, застосовуючи відомі нам природні способи оптимізації. *Будемо розглядати кожен варіант інвестування (набір значень змінних) як індивідуум, а прибутковість цього варіанта - як пристосованість цього індивідуума*. Тоді в процесі еволюції (якщо ми зуміємо його організувати) пристосованість індивідуумів буде зростати, а виходить, будуть з'являтися усе більш і більш дохідні варіанти інвестування. Зупинивши еволюцію в деякий момент і вибравши найкращого індивідуума, ми одержимо досить гарне рішення задачі.

Тут **генетичний алгоритм** - це проста модель еволюції в природі, яка реалізована у вигляді комп'ютерної програми, що здійснює оптимізацію.

У ньому використовуються як аналог механізму генетичного спадкування, так і аналог природного відбору. При цьому зберігається біологічна термінологія в спрощеному виді. От як моделюється генетичне спадкування

Хромосома - Вектор (послідовність) з нулів і одиниць.

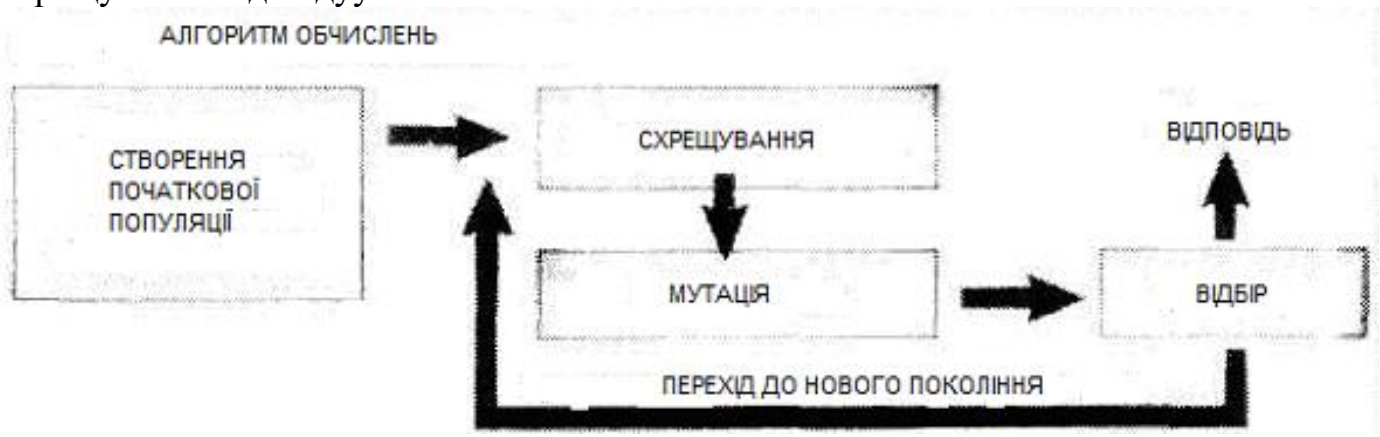
Кожна позиція (біт) називається **геном**.

Ідивідум= генетичний код. - Набір хромосом= варіант рішення задачі

Кросовер - Операція, при якій дві хромосоми обмінюються своїми частинами

Мутація - Випадкова зміна однієї або декількох позицій в хромосомі.

Щоб змоделювати еволюційний процес, згенеруємо спочатку випадкову популяцію - кілька індивідумів з випадковим набором хромосом (числових векторів). Генетичний алгоритм імітує еволюцію цієї популяції як циклічний процес схрещування індивідуумів і зміни поколінь.



Життєвий цикл популяції - це кілька випадкових схрещувань (за допомогою кроссовера) і мутацій, у результаті яких до популяції додається якась кількість нових індивідуумів. **Відбір у генетичному алгоритмі** - це процес формування нової популяції зі старої, після чого стара популяція гине. Після відбору до нової популяції знову застосовуються операції кросовера і мутації, потім знову відбувається відбір, і так далі.

Відбір у генетичному алгоритмі тісно зв'язаний із принципами природного відбору в природі в такий спосіб:

Пристосування індивідуума	Значення цільової функції на цьому індивідуумі
Виживання найбільш пристосованих	Популяція наступного покоління формується у відповідності з цільовою функцією. Чим пристосованіший індивідуум, тим більша ймовірність його участі у кросовері, тобто розмноженні

Таким чином, модель відбору визначає, яким чином варто будувати популяцію наступного покоління. *Як правило, ймовірність участі індивідуума в схрещуванні береться пропорційно його пристосуванню*. Часто використовується так звана стратегія елітизма, при якій кілька кращих індивідуумів переходять у наступне

покоління без змін, не беручи участі у кросовері і доборі. У будь-якому випадку кожне наступне покоління буде в середньому краще попереднього. Коли пристосованість індивідуумів перестає помітно збільшуватися, процес зупиняють і як розв'язання задачі оптимізації беруть найкращого зі знайдених індивідуумів.

Повертаючись до задачі оптимального розподілу інвестицій, пояснимо особливості реалізації генетичного алгоритму в цьому випадку.

1. Індивідум = варіант рішення задачі = набір з 10 хромосом X_j

2. Хромосома X_j = обсяг вкладення в проект, j = 16-розрядний запис цього числа

3. Так як обсяги вкладень обмежені, не всі значення хромосом є припустимими. Це враховується при генерації популяцій.

4. Так як сумарний обсяг інвестицій фіксований, то реально варіюються тільки 9 хромосом, а значення 10-ої визначається по них однозначно.

Нижче приведені результати роботи генетичного алгоритму для трьох різних значень сумарного обсягу інвестицій *K* (рис. 2). Квадратами на графіках прибутків відзначено, який обсяг вкладення в даний проект рекомендовано генетичним алгоритмом. Видно, що при малому значенні *K* інвестуються тільки ті проекти, які прибуткові при мінімальних вкладеннях.

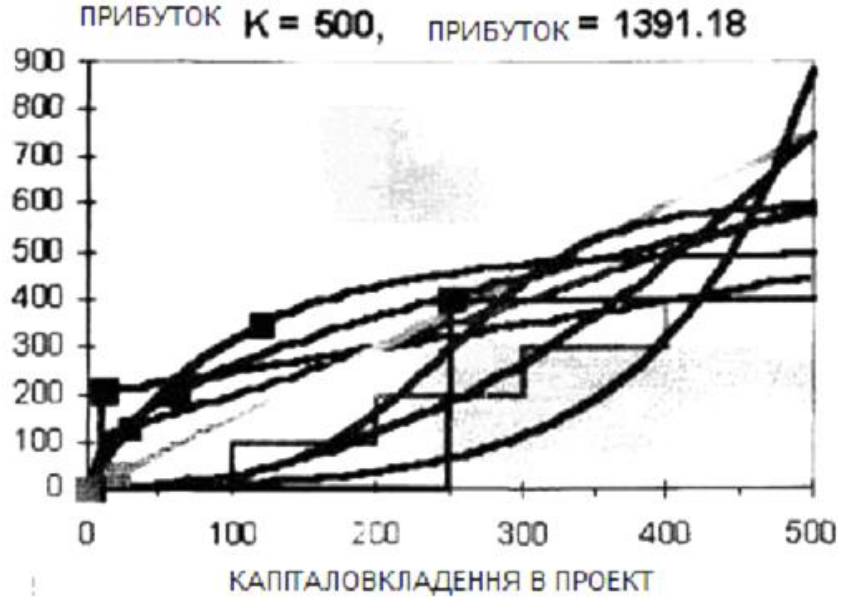


Рис. 2.

Якщо збільшити сумарний обсяг інвестицій, стає прибутковим вкладати гроші й у більш дорогі проекти (рис. 3).

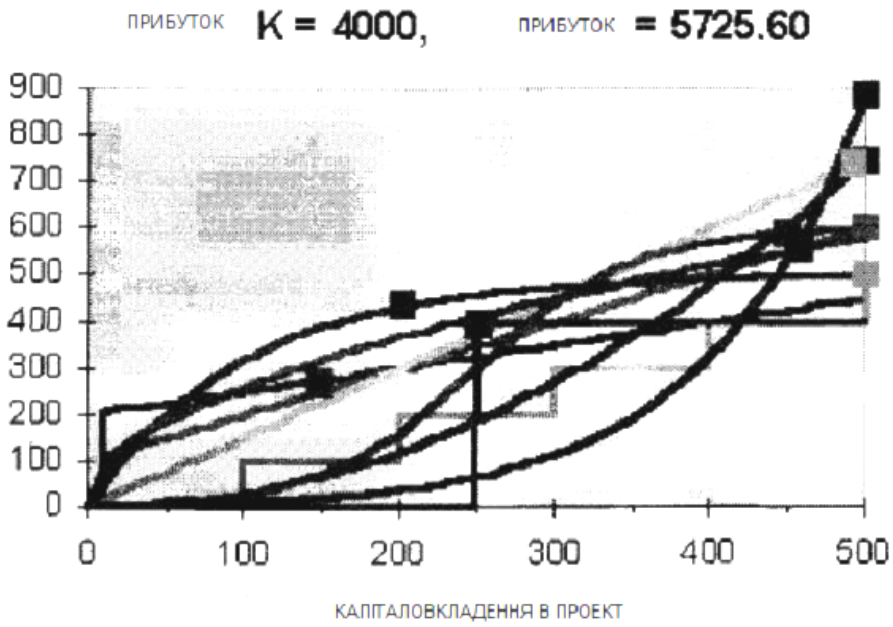


Рис. 3.

При подальшому збільшенні *K* досягається поріг максимального вкладення в прибуткові проекти, і інвестування в малоприбуткові проекти знову набуває сенсу.

3. Особливості генетичних алгоритмів

Генетичний алгоритм - новітній, але не єдино можливий спосіб розв’язання задач оптимізації. Віддавна відомі два основних шляхи розв’язання таких задач - **переборний** і **локально-градієнтний**. У цих методів свої достоїнства і недоліки, і в кожному конкретному випадку варто подумати, який з них вибрати. Розглянемо достоїнства і недоліки стандартних і генетичних методів на прикладі класичної *задачі комівояжера* (TSP - travelling salesman problem). Суть задачі полягає в тому, щоб знайти найкоротший замкнений шлях обходу декількох міст, заданих своїми координатами. Виявляється, що вже для 30 міст пошук оптимального шляху являє собою складну задачу, що спонукала розвиток різних нових методів (у тому числі нейромереж і генетичних алгоритмів).

Кожен варіант розв’язання (для 30 міст) - це числовий ряд, де на j-ому місці стоїть номер j-ого по порядку обходу міста. Таким чином, у цій задачі 30 параметрів, причому не всі комбінації значень припустимі. Природно, першою ідеєю є повний перебір усіх варіантів обходу.

Переборний метод найбільш простий по своїй суті і тривіальний у програмуванні. Для пошуку оптимального розв’язання (точки максимуму цільової функції) потрібно послідовно обчислити значення цільової функції у всіх можливих точках, запам'ятовуючи максимальне з них (рис.7.4).

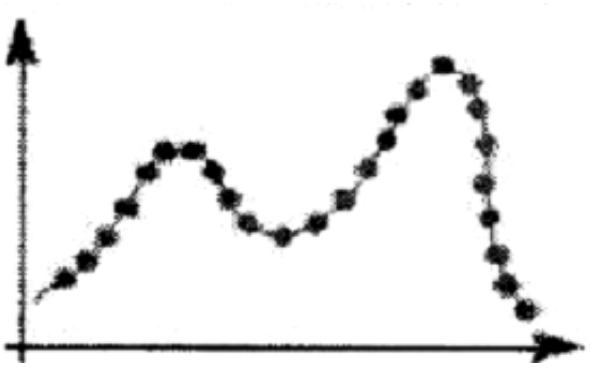


Рис. 4.

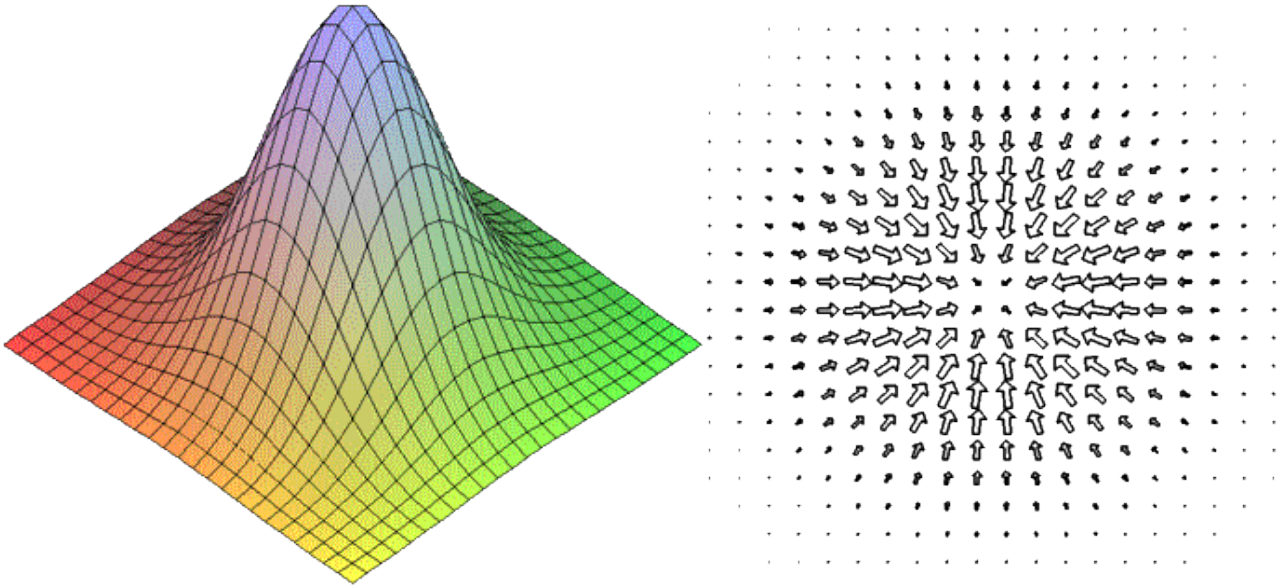
Недоліком цього методу є велика обчислювальна вартість (складність). Зокрема, у задачі комівояжера буде потрібно прорахувати довжини більш 10^{30} варіантів шляхів, що зовсім нереально.

Однак, якщо перебір усіх варіантів за розумний час можливий, то можна бути абсолютно упевненим у тому, що знайдене розв'язання дійсно оптимальне.

Другий популярний спосіб заснований на *методі градієнтного спуска*.

Градіє́нт — міра зростання або спадання в просторі якоїсь фізичної величини на одиницю довжини.

Для позначення градієнта використовується оператор Гамільтона ∇, або оператор {**grad**}



Операція градієнта перетворює пагорб (ліворуч), якщо дивитися на нього зверху, на поле векторів (праворуч). Видно, що вектори спрямовані «вгору», і чим крутіший нахил, тим вони довші.

При *методі градієнтного спуска* спочатку вибираються деякі випадкові значення параметрів, а потім ці значення поступово змінюють, домагаючись найбільшої швидкості росту цільової функції (рис.5).

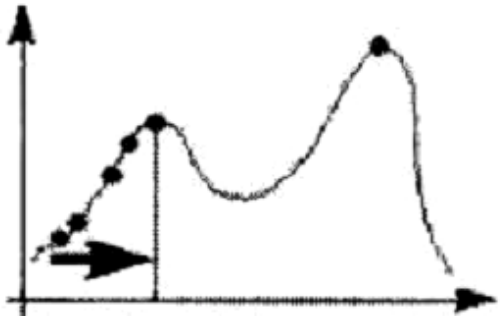


Рис. 5

Досягши локального максимуму, такий алгоритм зупиняється, тому для пошуку глобального оптимуму будуть потрібні додаткові зусилля.

Градієнтні методи працюють дуже швидко, але не гарантують оптимальності знайденого розв’язання. Вони ідеальні для застосування в так званих унімодальних задачах (рис. 6), де цільова функція має єдиний локальний максимум (він же - глобальний). Легко бачити, що задача комівояжера не є унімодальною.

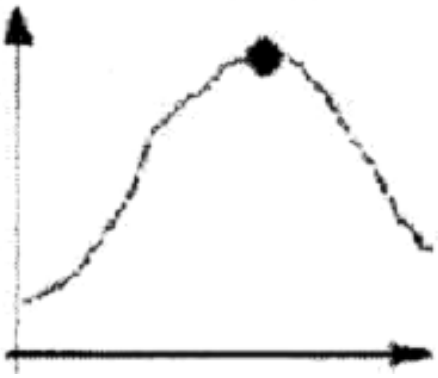
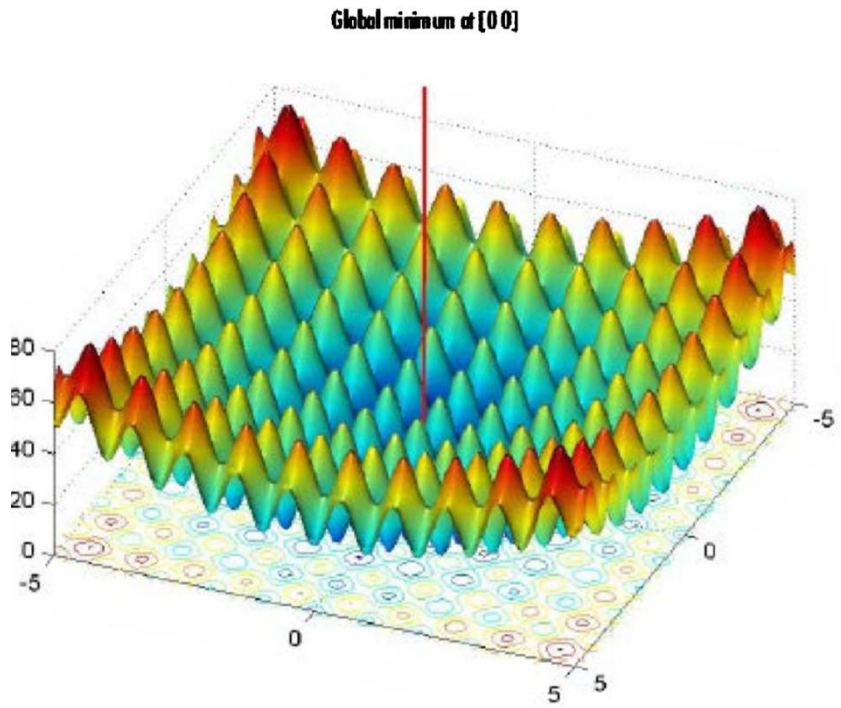


Рис. 6.

Типова практична задача, як правило, *мультимодальна і багатомірна*, тобто містить багато параметрів. Для таких задач не існує жодного універсального методу, що дозволяв би досить швидко знайти абсолютно точне розв’язання (рис. 7).

Приклад складної функції для двох незалежних змінних (функція Растригина)

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$$



Однак, комбінуючи переборний і градієнтний методи, можна сподіватися отримати хоча б наближене розв’язання, точність якого буде зростати при збільшенні часу розрахунку (рис. 8).

Генетичний алгоритм являє собою саме такий комбінований метод. Механізми схрещування і мутації в якомусь змісті реалізують переборну частину методу, а добір кращих розв’язань - градієнтний спуск. На рисунку показано, що така комбінація дозволяє забезпечити стійко гарну ефективність генетичного пошуку для будь-яких типів задач (рис. 9).

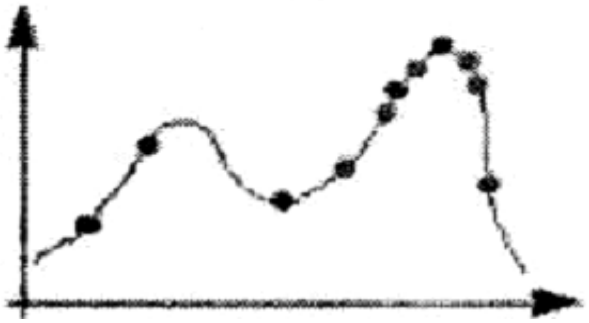


Рис. 8.

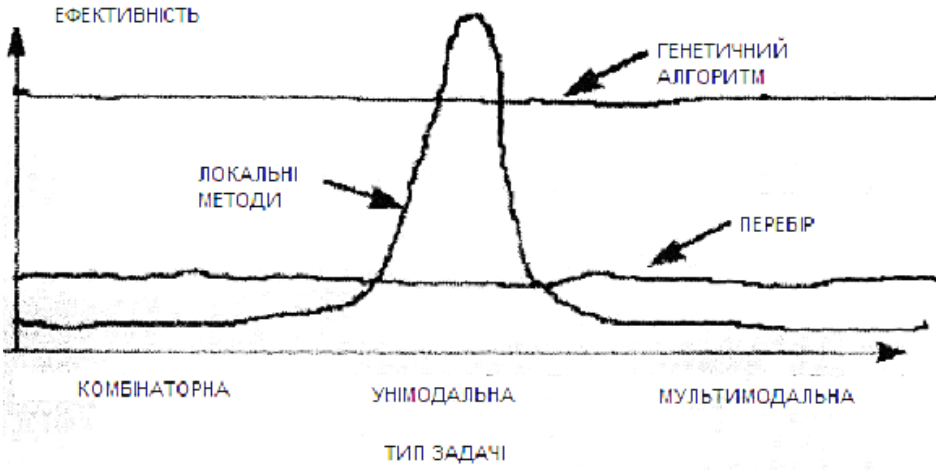


Рис. 9.

Отже, якщо на деякій множині задана складна функція від декількох змінних, то *генетичний алгоритм - це програма, що за розумний час знаходить точку, де значення функції досить близьке до максимально можливого*. Вибираючи прийнятний час розрахунку, ми одержимо одне з кращих розв'язань, що узагалі можливо одержати за цей час.

4. Задачі оптимізації і застосування алгоритмів

Генетичні алгоритми в основному застосовуються для розв'язання задач оптимізації, тобто задач, у яких є деяка функція декількох змінних $F(x_1, x_2, ... ,x_n)$ і необхідно знайти або її максимум, або її мінімум. Функція F називається *цільовою функцією*, а змінні - *параметрами функції*. Генетичні алгоритми "пришиваються" до даної задачі в такий спосіб.

Параметри задачі є генетичним матеріалом - *генами*. Сукупність генів складає *хромосому*. Кожна особа має свою хромосому, а, отже, свої набори параметрів. Підставивши параметри в цільову функцію, можна одержати якесь значення. Те, наскільки це значення задовольняє поставленим умовам, визначає характеристику особи, що називається **пристосованістю** (**fitness**). Функція, що визначає пристосованість повинна задовольняти наступній умові: *чим "краща" особа, тим вище пристосованість*. Генетичні алгоритми працюють з популяцією як правило фіксованого розміру, що складається з особин, заданих за допомогою способу, описаного вище. Особи "схрещуються" між собою за допомогою *генетичних операторів* (про те, як відбувається цей процес – буде описано окремо), і в такий спосіб виходять *нащадки*, причому частина нащадків замінюють представників більш старого покоління у відповідності *зі стратегією формування нового покоління*. Вибір особин для схрещування проводиться відповідно *до селективної стратегії (selection strategy)*. Заново сформована популяція знову оцінюється, потім вибираються найбільш гідні для схрещування особи, що схрещуються між собою, виходять «діти» і займають місце «старих» індивідумів і т.д. Усі це продовжується доти поки не знайдеться особа, гени якої представляють оптимальний набір параметрів, при яких значення цільової функції близьке до максимуму або мінімуму, або дорівнює йому. Зупинка роботи ГА може відбутися також у випадку, якщо популяція вироджується, тобто якщо практично немає розмаїтості в генах особин популяції, або якщо просто вийшов ліміт часу. Виродження популяції називають *передчасною збіжністю (premature convergence)*.

Може створитися враження, що ГА є просто перекрученим варіантом випадкового пошуку. Але пристосованість була введена зовсім не даремно. Справа в тім, що вона безпосередньо впливає на шанс особи взяти участь у схрещуванні з наступним «народженням дітей». Вибираючи щораз для схрещування найбільш пристосованих особин, можна з визначеним степенем упевненості стверджувати, що нащадки будуть або не набагато гірші, ніж батьки, або кращі їх. Приблизно цю величину впевненості можна оцінити за допомогою *теореми шаблонів (теореми шим)*.

Теоретичні аспекти, що випливають на ГА:

- Представлення даних в генах
- Генетичні оператори
- Моделі ГА
- Функції
- Стратегії відбору і формування нового покоління

Приклади задач де застосовуються ГА

• Екстремальні задачі (пошук точок мінімуму і мінімуму);
• Задачі про найкоротший шлях;
• Задачі компонування;
• Складання розкладів;
• Апроксимація функцій;
• Добір (фільтрація) вхідних даних;
• Настроювання штучної нейронної мережі;
• Моделювання штучного життя (Artificial life systems) ;
• Біоінформатика (згортання білків і РНК);
• Ігрові стратегії;
• Нелінійна фільтрація;
• Агенти, що розвиваються/машини (Evolvable agents/machines);
• Оптимізація запитів в базах даних
• Різноманітні задачі на графах (задача комівояжера, розфарбування, знаходження паросполученнь)
• Навчання штучної нейронної мережі
• Штучне життя

Деякі розділи можуть містити підпункти. Так, наприклад, екстремальні задачі включають у себе цілий клас задач лінійного і нелінійного програмування.

Усього застосувань дуже багато, тому приведений список не є вичерпним.

Приклад

Розглянемо простий приклад - задачу знаходження максимуму функції, заданої виразом (1) для цілочислової змінної x, що приймає значення від 0 до 31.

$$f(x) = 2x^2+1 \quad (1)$$

Для застосування генетичного алгоритму необхідно насамперед закодувати значення змінної **x** у виді двійкових послідовностей. Очевидно, що цілі числа з інтервалу [0, 31] можна представити послідовностями нулів і одиниць, використовуючи їхнє представлення в двійковій системі числення.

Число 0 при цьому записується як 00000, а число 31 - як 11111. У даному випадку хромосоми здобувають вид двійкових послідовностей, що складаються з 5 бітів, тобто ланцюжками довжиною 5.

Також очевидно, що в ролі функції пристосованості буде виступати цільова функція *f(x)*, задана виразом (1). Тоді пристосованість хромосоми ch_i, i = 1,2.....N, буде визначатися значенням функції *f(x)* для **x**, рівного фенотипові, що відповідає генотипові ch_i . Позначимо ці фенотипи ch_i*. У такому випадку значення функції

пристосованості хромосоми ch_i (тобто $F(ch_i)$) буде дорівнювати $f(ch_i^*)$. Виберемо випадковим чином вихідну популяцію, що складається з 6 кодових послідовностей (наприклад, можна 30 разів підкинути монету); при цьому N=6.

Припустимо, що обрано хромосоми

ch_1 = [10011]
ch_2 = [00011]
ch_3 = [00111]
ch_4 = [10101]
ch_5 = [01000]
ch_6 = [11101]

Відповідні ним фенотипи - це представлені нижче числа з інтервалу від 0 до 31:

ch_1^* = 19
ch_2^* =3
ch_3^*= 7
ch_4^* = 21
ch_5^* = 8
ch_6^* = 29

По формулі $f(x) = 2x^2+1$ (1) розраховуємо значення функції пристосованості для кожної хромосоми в популяції й одержуємо

$F(ch_1)$ = 723
$F(ch_2)$= 19
$F(ch_3)$ = 99
$F(ch_4)$ = 883
$F(ch_5)$ = 129
$F(ch_6)$=1683

Селекція хромосом. *Методом рулетки*, вибираємо 6 хромосом для репродукції. Колесо рулетки представлено на рис. 10.

$$v(ch_i) = p_s(ch_i)100\% , \quad (2)$$

$$p_s(ch_i)=\frac{F(ch_i)}{\sum_{i=1}^{N} F(ch_i)} \quad (3)$$

де

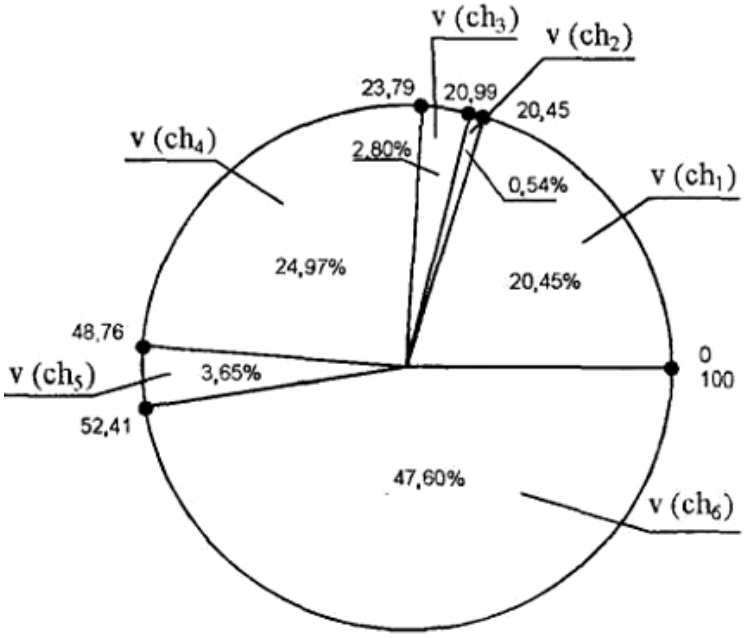


Рис. 10. Колесо рулетки для селекції в прикладі 2.

Припустимо, що обрано числа

97 26 54 13 31 88.

Це означає вибір хромосом

ch_6 ch_4 ch_6 ch_1 ch_4 ch_6

Нехай схрещування виконується з ймовірністю p_c=1. Припустимо, що для схрещування сформовані пари

ch_1 і ch_4 , ch_4 і ch_6 , ch_6 і ch_6 .

Крім того, припустимо, що випадковим чином обрана точка схрещування, рівна 3 для хромосом ch_1, і ch_4, а також точка схрещування, рівна 2 для хромосом ch_4 і ch_6 (рис.11).

$$\left.\begin{array}{l} ch_1 = [10011] \\ ch_4 = [10101] \end{array}\right\} \xrightarrow{\text{СХРЕЩУВАННЯ}} \begin{array}{l} [10001] \\ [10111] \end{array}$$

$$l_k = 3$$

$$\left.\begin{array}{l} ch_4 = [10101] \\ ch_6 = [11101] \end{array}\right\} \xrightarrow{\text{СХРЕЩУВАННЯ}} \begin{array}{l} [10101] \\ [11101] \end{array}$$

$$l_k = 2$$

Рис. 11. Процес схрещування хромосом у прикладі 2.

За умови, що ймовірність мутації p_m=0, у нову популяцію включаються хромосоми

Ch_1 = [10001]
Ch_2 = [10111]
Ch_3 = [10101]
Ch_4 = [11101]
Ch_5 = [11101]
Ch_6 = [11101]

Для розрахунку значень функції пристосованості цих хромосом необхідно декодувати їхні двійкові послідовності, що представляють, і одержати відповідні їм фенотипи. Позначимо їх Ch_i*. У результаті декодування одержуємо числа (з інтервалу від 0 до 31)

Ch_1*= 17
Ch_2* = 23
Ch_3* = 21
Ch_4* = 29
Ch_5* = 29
Ch_6* = 29

Відповідно, значення функції пристосованості хромосом нової популяції, розраховані по формулі (1), складуть

$F(Ch_1)$= 579
$F(Ch_2)$=1059
$F(Ch_3)$= 883
$F(Ch_4)$=1683
$F(Ch_5)$ = 1683
$F(Ch_6)$=1683

Легко помітити, що в цьому випадку середнє значення пристосованості зросло з 589 до 1262.

Звертаємо увагу, що якщо на наступній ітерації будуть сформовані для схрещування пари хромосом, наприклад, Ch_4 і Ch_2, Ch_5 і Ch_2 або Ch_6 і Ch_2 із точкою схрещування 2 або 3, то серед інших буде отримана хромосома [11111] з фенотипом, рівним числу 31, при якому оптимізуєма функція досягає свого максимуму. Значення функції пристосованості для цієї хромосоми виявляється найбільшим і складає 1923. Якщо таке сполучення пар у даній ітерації не відбудеться, то можна буде очікувати утворення хромосоми з найбільшим значенням функції пристосованості на наступних ітераціях. Хромосома [11111] могла бути отримана і на поточній ітерації у випадку формування для схрещування пари Ch_1 і Ch_6 із точкою схрещування 3.

Відзначимо, що при довжині хромосом, рівній 5 бітам, простір пошуку дуже малий і нараховує всього 2^5=32 точки.

Представлений приклад має виняткова демонстраційний характер. Застосування генетичного алгоритму для такого простого прикладу практично недоцільне, оскільки його оптимальне рішення може бути отримане миттєво. Однак цей приклад придатний для вивчення функціонування генетичного алгоритму.

Також варто згадати, що в малих популяціях часто зустрічаються ситуації, коли на початковому етапі кілька особин мають значно більші значення функції приналежності, чим інші особи даної популяції. Застосування методу селекції на основі «колеса рулетки» дозволяє в цьому випадку дуже швидко вибрати «найкращі» особи, іноді - протягом «життя» одного покоління. Однак такий розвиток подій вважається небажаним, оскільки він стає головною причиною передчасної збіжності алгоритму, що називається збіжністю до неоптимального рішення. З цієї причини використовуються й інші методи селекції, що відрізняються від колеса рулетки, або застосовується масштабування функції пристосованості.

5. Опис типового генетичного алгоритму

Задача кодується таким чином, щоб її рішення могло бути представлене у виді вектора («хромосома»). Випадковим чином створюється деяка кількість початкових векторів («початкова популяція»). Вони оцінюються з використанням «функції пристосованості», у результаті чого кожному векторові привласнюється визначене значення («пристосованість»), що визначає ймовірність виживання організму, представленого даним вектором. Після цього з використанням отриманих значень пристосованості вибираються вектори (селекції), допущені до «схрещування». До цих векторів застосовуються «генетичні оператори» (у більшості випадків «схрещування»-crossoveri «мутація»-mutation), створюючи в такий спосіб наступне «покоління». Особи наступного покоління також оцінюються, потім виробляється селекція, застосовуються генетичні оператори і т.д. Так моделюється «еволюційний процес», що продовжується кілька життєвих циклів (поколінь), поки не буде виконаний **критерій зупинки алгоритму**. Таким критерієм може бути:

- отримання глобального, або субоптимального рішення;
- вичерпання числа поколінь, відпущених на еволюцію;
- вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми служать, головним чином, для пошуку рішень у дуже великих, складних просторах пошуку.

Можна виділити наступні етапи генетичного алгоритму:

1. Створення початкової популяції
2. Обчислення функцій пристосованості для особин популяції (оцінювання)

• (Початок циклу)

1. Вибір індивідів з поточної популяції (селекція)
2. Схрещування і\або мутація
3. Обчислення функцій пристосованості для всіх особин
4. Формування нового покоління
5. Якщо виконуються умови іншого, то (кінець циклу), інакше (початок циклу).

Створення початкової популяції

Перед першим кроком потрібно випадковим чином створити якусь початкову популяцію; навіть якщо вона виявиться зовсім неконкурентоспроможною, генетичний алгоритм усе рівно досить швидко переведе її в життєздатну популяцію. Таким чином, на першому кроці можна особливо не намагатися зробити занадто дуже пристосованих особин, досить, щоб вони відповідали форматові особин популяції, і на них можна було підрахувати функцію пристосованості (Fitness). Підсумком першого кроку є популяція H, що складається з N особин.

Відбір

На етапі відбору потрібно з усієї популяції вибрати визначену її частку, що залишиться "у живих" на цьому етапі еволюції. Є різні способи проводити відбір. Ймовірність виживання особи h повинна залежати від значення функції пристосованості Fitness (h). Сама частка що вижила s зазвичай є параметром генетичного алгоритму, і її просто задають заздалегідь. За підсумками відбору з *N* особин популяції *H* повинні залишитися s особин, що ввійдуть у підсумкову популяцію *H'*. Інші особи гинуть.

Розмноження

Розмноження в генетичних алгоритмах зазвичай полове - щоб зробити нащадка, потрібно декілька батьків; звичайно, потрібно рівно два. Розмноження в різних алгоритмах визначається по-різному - воно, зазвичай, залежить від представлення даних. *Головна вимога до розмноження - щоб нащадок або нащадки мали можливість успадкувати риси обох батьків, "змішавши" їх яким-небудь досить розумним способом.*

Взагалі говорячи, для того щоб провести операцію розмноження, потрібно вибрати *(1-s)p/2* пар гіпотез з H і провести з ними розмноження, одержавши по два нащадка від кожної пари (якщо розмноження визначене так, щоб давати одного нащадка, потрібно вибрати *(1-s)p* пар), і додати цих нащадків у *H'*. У результаті *H'* буде складатися з N особин. Чому особи для розмноження звичайно вибираються з усієї популяції H, а не з елементів, що вижили на першому кроці, H_0 (хоча останній варіант теж має право на існування)? Справа в тім, що головна критика багатьох генетичних алгоритмів - *недолік розмаїтості (diversity) в особинах*. Досить швидко виділяється один-єдиний генотип, що являє собою локальний максимум, а потім всі елементи популяції програють йому вибір, і вся популяція "забивається" копіями цієї особи. Є різні способи боротьби з таким небажаним ефектом; один з них - вибір для розмноження не самих пристосованих, але узагалі всіх особин.

Мутації

До мутацій відноситься все те ж саме, що і до розмноження: є деяка частка мутантів m, що є параметром генетичного алгоритму, і на кроці мутацій потрібно вибрати m особин, а потім змінити їх відповідно до заздалегідь визначених операцій мутації.

6. Класичний генетичний алгоритм

Основний (класичний) генетичний алгоритм (також називаний елементарним або простим генетичним алгоритмом) складається з наступних кроків:

1) ініціалізація, або вибір вихідної популяції хромосом;
2) оцінка пристосованості хромосом у популяції;
3) перевірка умови зупинки алгоритму;
4) селекція хромосом;
5) застосування генетичних операторів;
6) формування нової популяції;
7) вибір «найкращої» хромосоми.

Блок-схема основного генетичного алгоритму зображена на рис. 12.

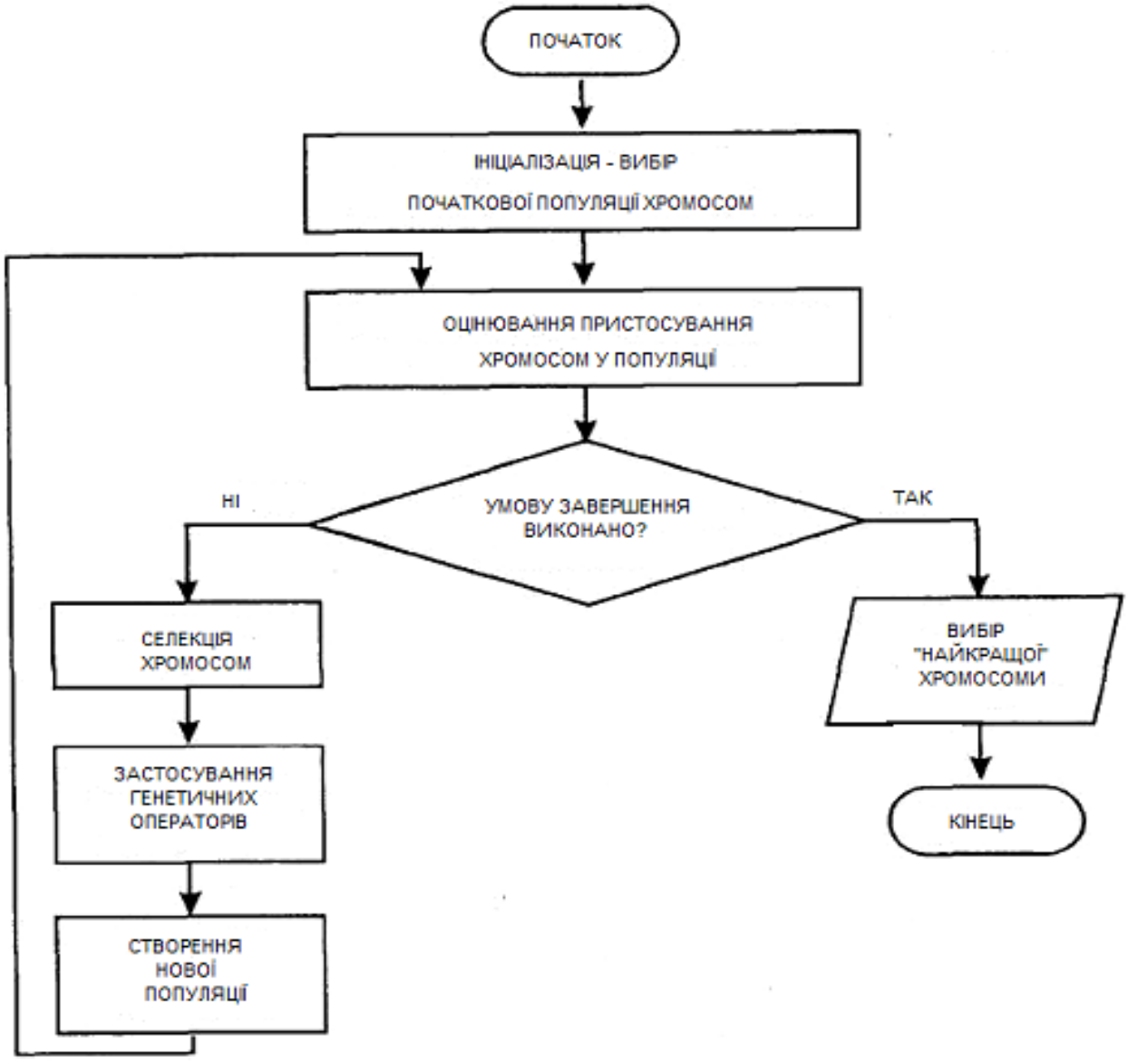


Рис. 12. Блок-схема генетичного алгоритму.

Розглянемо конкретні етапи цього алгоритму більш докладно з використанням додаткових подробиць, представлених на рис. 13.

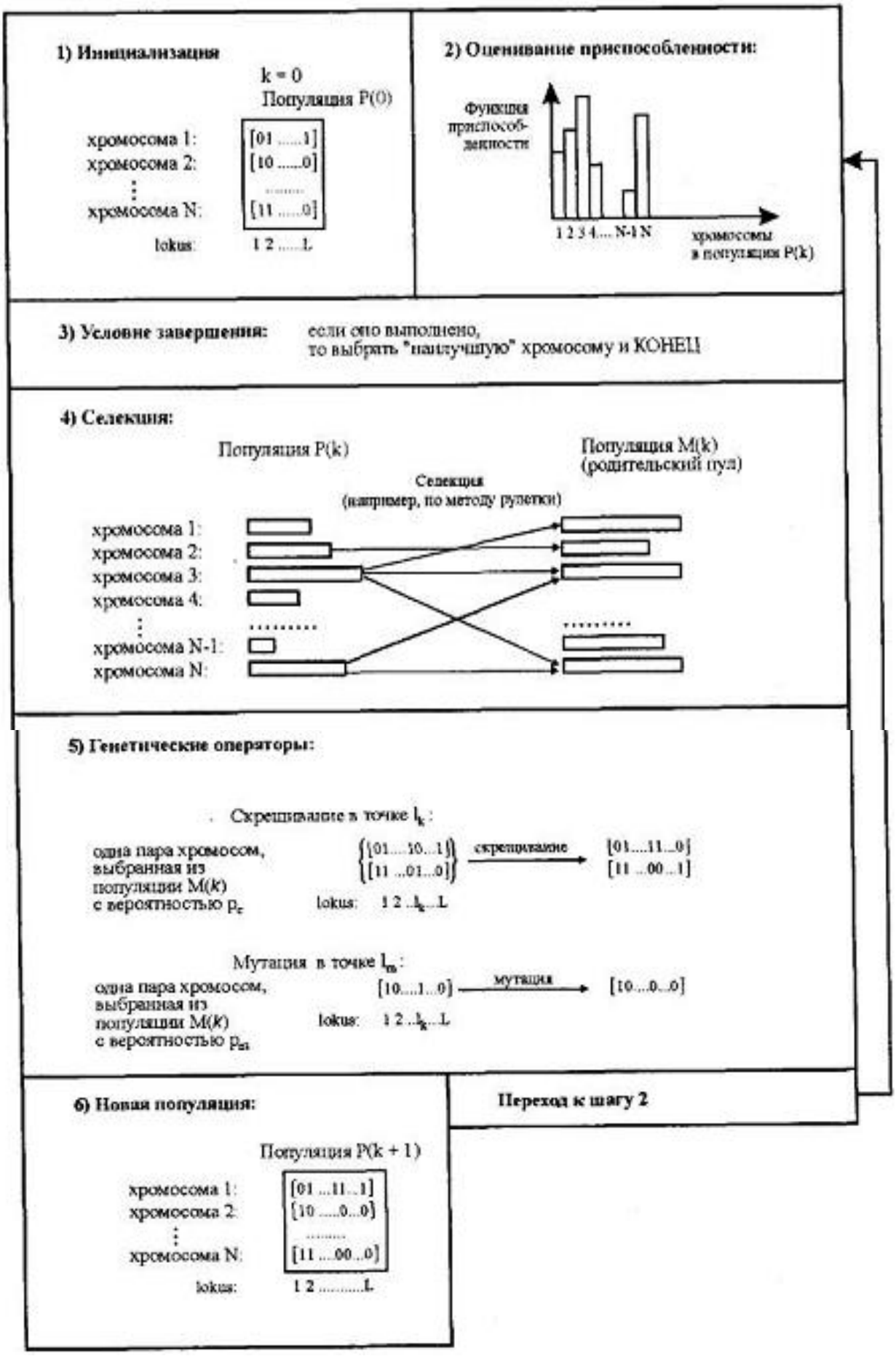


Рис. 13. Схема виконання генетичного алгоритму.

Ініціалізація, тобто формування вихідної популяції, полягає у випадковому виборі заданої кількості хромосом (особин), що представляються двоїчними послідовностями фіксованої довжини.

Оцінювання пристосованості хромосом у популяції складається в розрахунку функції пристосованості для кожної хромосоми цієї популяції. Чим більше значення цієї функції, тим вище «якість» хромосоми. Форма функції пристосованості залежить від характеру розв'язуваної задачі. Передбачається, що функція пристосованості завжди приймає ненегативні значення і, крім того, що для рішення оптимізаційної задачі потрібно максимізувати цю функцію. Якщо вихідна форма функції пристосованості не задовольняє цим умовам, то виконується відповідне перетворення (наприклад, задачу мінімізації функції можна легко звести до задачі максимізації).

Перевірка умови зупинки алгоритму. Визначення умови зупинки генетичного алгоритму залежить від його конкретного застосування. В оптимізаційних задачах, якщо відомо максимальне (або мінімальне) значення функції пристосованості, то зупинка алгоритму може відбутися після досягнення очікуваного оптимального значення, можливо - із заданою точністю. Зупинка алгоритму також може відбутися у випадку, коли його виконання не приводить до поліпшення вже досягнутого значення. Алгоритм може бути зупинений після закінчення визначеного часу виконання або після виконання заданої кількості ітерацій. Якщо умова зупинки виконана, то виконується перехід до завершального етапу вибору «найкращої» хромосоми. У противному випадку на наступному кроці виконується селекція.

Селекція хромосом полягає у відборі (по розрахованим на другому етапі значенням функції пристосованості) тих хромосом, що будуть брати участь у створенні нащадків для наступної популяції, тобто для чергового покоління. Такий відбір виконується відповідно до принципу природного відбору, по якому найбільші шанси на участь у створенні нових особин мають хромосоми з найбільшими значеннями функції пристосованості. Існують різні методи селекції. Найбільш популярним вважається так називаний метод рулетки (roulette wheel selection), що свою назву одержав за аналогією з відомою азартною грою. Кожній хромосомі може бути зіставлений сектор колеса рулетки, величина якого встановлюється пропорційно значенню функції пристосованості даної хромосоми. Тому чим більше значення функції пристосованості, тим більше сектор на колесі рулетки. Усе колесо рулетки відповідає сумі значень функції пристосованості всіх хромосом розглянутої популяції.

Кожній хромосомі, що позначається ch_i для i=1,2,..., N (де N позначає чисельність популяції) відповідає сектор колеса v(ch_i), виражений у відсотках відповідно до формули

$$v(ch_i) = p_s(ch_i)100\% , \tag{7.2}$$

де

$$p_s(ch_i)=\frac{F(ch_i)}{\sum_{i=1}^{N} F(ch_i)} \tag{7.3}$$

причому $F(ch_i)$ - значення функції пристосованості хромосоми ch_i , а $p_s(ch_i)$ - ймовірність селекції хромосоми ch_i . Селекція хромосоми може бути представлена як результат повороту колеса рулетки, оскільки «вигравша» (тобто обрана) хромосома відноситься до того сектора колеса, що випав. Очевидно, що чим більше сектор, тим більше ймовірність «перемоги» відповідної хромосоми. Тому ймовірність вибору даної хромосоми виявляється пропорційною значенню її функції пристосованості.

Якщо все коло колеса рулетки представити у виді цифрового інтервалу [0, 100], то вибір хромосоми можна ототожнити з вибором числа з інтервалу [a, b], де a і b позначають відповідно початок і закінчення фрагмента кола, що відповідає цьому секторові колеса; очевидно, що $0 \le a < b \le 100$. У цьому випадку вибір за допомогою колеса рулетки зводиться до вибору числа з інтервалу [0, 100], що відповідає конкретній точці на колі колеса. Інші методи селекції будуть розглядатися далі.

У результаті процесу селекції створюється батьківська популяція, яка також називана батьківським пулом (mating pool) з чисельністю N, рівною чисельності поточної популяції.

Застосування генетичних операторів до хромосом, відібраним за допомогою селекції, приводить до формування нової популяції нащадків від створеної на попередньому кроці батьківської популяції.

У класичному генетичному алгоритмі застосовуються два основних генетичних оператори: оператор схрещування (crossover) і оператор мутації (mutation). Однак слід зазначити, що оператор мутації грає явно другорядну роль у порівнянні з оператором схрещування. Це означає, що схрещування в класичному генетичному алгоритмі виконується практично завжди, тоді як мутація - досить рідко. Ймовірність схрещування, як правило, досить велика (зазвичай $0{,}5 \le p_c \le 1$), тоді як ймовірність мутації встановлюється досить малою (найчастіше $0 \le p_т \le 0{,}1$). Це випливає з аналогії зі світом живих організмів, де мутації відбуваються надзвичайно рідко. У генетичному алгоритмі мутація хромосом може виконуватися на популяції батьків перед схрещуванням або на популяції нащадків, утворених у результаті схрещування.

Оператор схрещування. На першому етапі схрещування вибираються пари хромосом з батьківської популяції (батьківського пула). Це тимчасова популяція, що складається з хромосом, відібраних у результаті селекції і призначених для подальших перетворень операторами схрещування і мутації з метою формування нової популяції нащадків. На даному етапі хромосоми з батьківської популяції поєднуються в пари. Це виконується випадковим способом відповідно до ймовірності схрещування pc. Далі для кожної пари відібраних у такий спосіб батьків розігрується позиція гена (локус) у хромосомі, що визначає так називану точку схрещування. Якщо хромосома кожного з батьків складається з L генів, то очевидно, що точка схрещування k являє собою натуральне число, менше L. Тому фіксація точки схрещування зводиться до випадкового вибору числа з інтервалу [1, -1]. У результаті схрещування пари батьківських хромосом виходить наступна пара нащадків:

1) нащадок, хромосома якого на позиціях від 1 до k складається з генів першого батька, а на позиціях від k+1 до L - з генів другого батька;

2) нащадок, хромосома якого на позиціях від 1 до k складається з генів другого батька, а на позиціях від k+1 до L - з генів першого батька.

Оператор мутації з ймовірністю $p_т$ змінює значення гена в хромосомі на протилежне (тобто з 0 на 1 або назад). Наприклад, якщо в хромосомі [100110101010] мутації піддається ген на позиції 7, то його значення, рівне 1, змінюється на 0, що призводить до утворення хромосоми [100110001010]. Як уже згадувалося вище, ймовірність мутації зазвичай дуже мала, і саме від неї залежить, буде даний ген мутировати чи ні. Ймовірність $p_т$ мутації може емулюватися, наприклад, випадковим вибором числа з інтервалу [0, 1] для кожного гена і добором для виконання цієї операції тих генів, для яких розігране число виявляється меншим або рівним значенню $p_т$.

Формування нової популяції. Хромосоми, отримані в результаті застосування генетичних операторів до хромосом тимчасової батьківської популяції, включаються до складу нової популяції. Вона стає так називаною поточною популяцією для даної ітерації генетичного алгоритму. На кожній черговій ітерації розраховуються значення функції пристосованості для всіх хромосом цієї популяції, після чого перевіряється умова зупинки алгоритму й або фіксується результат у виді хромосоми з найбільшим значенням функції пристосованості, або здійснюється перехід до наступного кроку генетичного алгоритму, тобто до селекції. У класичному генетичному алгоритмі вся попередня популяція хромосом заміщується новою популяцією нащадків, що має ту ж чисельність.

Вибір «найкращої» хромосоми. Якщо умова зупинки алгоритму виконана, то варто вивести результат роботи, тобто представити шукане рішення задачі. Кращим рішенням вважається хромосома з найбільшим значенням функції пристосованості.

На завершення варто визнати, що генетичні алгоритми успадкували властивості природного еволюційного процесу, що складаються в генетичних змінах популяцій організмів з часом. Головний фактор еволюції - це природний відбір (тобто природна селекція), що приводить до того, що серед генетично розрізняючихся особин однієї і тієї ж популяції виживають і залишають у спадщину тільки найбільш пристосовані до навколишнього середовища. У генетичних алгоритмах також виділяється етап селекції, на якому з поточної популяції відбираються і включаються в батьківську популяцію особи, що мають найбільші значення функції пристосованості. На наступному етапі, що іноді називається еволюцією, застосовуються генетичні оператори схрещування і мутації, що виконують рекомбінацію генів у хромосомах.

Операція схрещування полягає в обміні фрагментами

ланцюжків між двома батьківськими хромосомами. Пари батьків для схрещування вибираються з батьківського пула випадковим чином так, щоб ймовірність вибору конкретної хромосоми для схрещування дорівнювала б ймовірності pc. Наприклад, якщо в якості батьків випадковим чином вибираються дві хромосоми з батьківської популяції чисельністю N, то $pc = 2N$. Аналогічно, якщо з батьківської популяції чисельністю N вибирається 2z хромосом ($z \le N/2$), які утворять z пар батьків, то $pc=2z/N$. Звертаємо увагу, що якщо всі хромосоми поточної популяції об'єднані в пари до схрещування, то $pc=1$. Після операції схрещування батьки в батьківській популяції заміщаються їхніми нащадками.

Операція мутації змінює значення генів у хромосомах із заданою ймовірністю рт способом, представленим при описі відповідного оператора. Це приводить до інвертування значень відібраних генів з 0 на 1 і зворотно. Значення $p_т$, як правило, дуже мале, тому мутації піддається лише невелика кількість генів. Схрещування - це ключовий оператор генетичних алгоритмів, що визначає їхню можливості й ефективність.

Мутація грає більш обмежену роль. Вона вводить у популяцію деяку розмаїтість і попереджає втрати, що могли б відбутися унаслідок виключення якого-небудь значного гена в результаті схрещування.

Основний (класичний) генетичний алгоритм, як ми вже відмічали, відомий у літературі як інструмент, у якому виділяються три види операцій: репродукції, схрещування і мутації. Терміни селекція і репродукція в даному контексті використовуються як синоніми. При цьому репродукція в даному випадку зв'язується скоріше зі створенням копій хромосом батьківського пула, тоді як більш розповсюджений зміст цього поняття означає процес формування нових особин, що відбуваються від конкретних батьків . Якщо ми приймаємо таке тлумачення, то оператори схрещування і мутації можуть вважатися операторами репродукції, а селекція - добором особин (хромосом) для репродукції.

Генетичні оператори

Частково про генетичні оператори ми вже говорили. Зупинимося на їхньому описі більш докладно. Генетичні оператори необхідні, щоб застосувати принципи спадковості і мінливості до віртуальної популяції. Крім відмінних рис, про які буде розказано нижче, у них є така властивість як ймовірність. Тобто описувані оператори не обов'язково застосовуються до всіх особин, які схрещуються, що вносить додатковий елемент невизначеності в процес пошуку рішення. У даному випадку, невизначеність не має на увазі негативний фактор, а є таким собі "степенем волі" роботи генетичного алгоритму. Тут описуються тільки два найпоширеніші і необхідних оператори. Існують і інші генетичні оператори (наприклад, інверсія), але вони застосовуються дуже рідко і тому ми про них говорити не будемо.

Оператор кросінговера

Оператор кросінговера (crossover operator), також називаний кросовером, є основним генетичним оператором, за рахунок якого виконується обмін генетичним матеріалом між особами. Оператор моделює процес схрещування особин. Нехай є дві батьківські особи з хромосомами $X=\{xi , i=1,L\}$ і $Y=\{yi , i=1,L\}$. Випадковим чином визначається точка усередині хромосоми, у якій обидві хромосоми діляться на дві частини і обмінюються ними. Назвемо цю точку точкою розриву. Узагалі говорячи, в англомовній літературі вона називається точкою кросінговера (crossover point), просто, точка розриву більш образна назва і до того ж дозволяє в деяких випадках уникнути тавтології. Описаний процес зображено на рис.7.14.

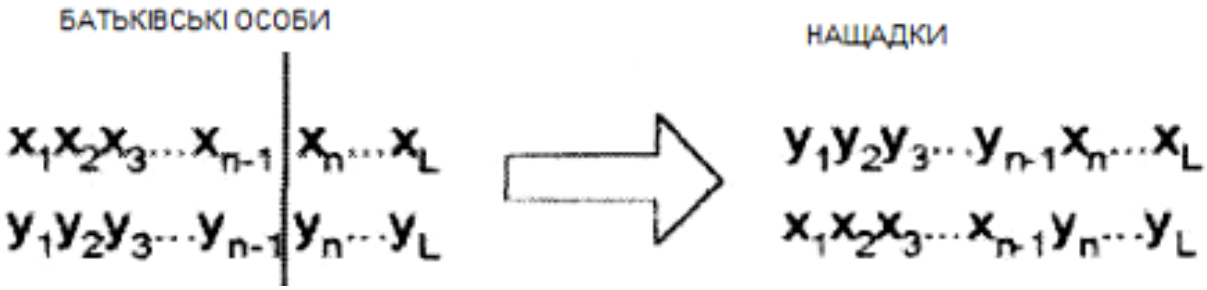


Рис.14. Кросінговер

Даний тип кросінговера називається *одноточковим*, так як при ньому батьківські хромосоми розрізаються тільки в одній випадковій точці. Також існують 2-х і n-точковий оператори кросінговера. У 2-х точковому кросінговері точок розриву 2, а n-точковий кросінговер є своєрідним узагальненням 1- і 2-точкового кросінговерів для n > 2.

Крім описаних типів кросінговерів є ще однорідний кросінговер. Його особливість полягає в тім, що значення кожного біта в хромосомі нащадка визначається випадковим чином з відповідних бітів батьків. Для цього вводиться деяка величина $0<p_0<1$, і якщо випадкове число більше p_0, то на n-у позицію першого нащадка потрапляє n-й біт першого батька, а на n-у позицію другого - n-й біт другого батька. У протилежному випадку до першого нащадка потрапляє біт другого батька, а до другого - першого. Така операція проводиться для всіх бітів хромосоми.

Ймовірність кросінговера найвища серед генетичних операторів і дорівнює зазвичай 60% і більше.

Оператор мутації

Оператор мутації (mutation operator) необхідний для "вибивання" популяції з локального екстремуму і сприяє захистові від передчасної збіжності. Досягаються це за рахунок того, що інвертується випадково обраний біт у хромосомі, що і показано на рис. 15.

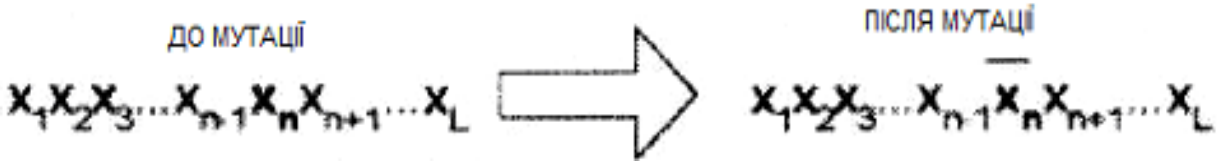


Рис. 15. Мутація

Так само як і кросінговер, мутація проводиться не тільки по одній випадковій точці. Можна вибирати деяку кількість точок у хромосомі для інверсії, причому їхнє число також може бути випадковим. Також можна інвертувати відразу деяку групу точок, що йдуть підряд. Ймовірність мутації значно менша ймовірності кросінговера і рідко перевищує 1%. Серед рекомендацій з вибору ймовірності мутації нерідко можна зустріти варіанти 1/L або 1/N, де L - довжина хромосоми, N - розмір популяції. Необхідно також відзначити, що деякі автори вважають, що оператор

мутації є основним пошуковим оператором і відомі алгоритми, що не використовують інших операторів (кросінговер, інверсія і т.д.) крім мутації.

Використовуються також кросінговери:

• Арифметичний кросінговер (Arithmetical crossover).
• Геометричний кросінговер (Geometrical crossover,).
• BLX-a кросінговер (BLX-a crossover).
• Імітований бінарний кросінговер (Simulated binary crossover).
• Нечітка рекомбінація (Fuzzy recombination).

Стратегії формування нового покоління

Після схрещування особин необхідно вирішити проблему про те, які з нових особин ввійдуть у наступне покоління, а які - ні, і що робити з їх предками. Є два способи:

1. Нові особи (нащадки) займають місця своїх батьків. Після чого настає наступний етап, у якому нащадки оцінюються, відбираються, дають потомство і поступаються місцем своїм "дітям".

2. Створюється проміжна популяція, що містить у собі як батьків, так і їхніх нащадків. Члени цієї популяції оцінюються, а потім з них вибираються N найкращих, котрі і ввійдуть у наступне покоління.

Ті хто знайомий з еволюційними стратегіями, то з цими способами ви вже зустрічалися. Який з цих двох варіантів краще – відповісти однозначно важко. Вочевидь другий варіант практичніший (так як не дозволяє заміняти пристосованих батьків на "невідомо кого"), але тут може бути більше проблем з передчасною збіжністю, чим у першому варіанті. До того ж він вимагає сортування масиву розміром (як мінімум) 2*N. Узагалі говорячи, можна комбінувати стратегії відбору і формування наступного покоління як завгодно - обмежень немає ніяких.

Принцип "елітизму"

Суть цього принципу полягає в тім, що в нове покоління включаються кращі батьківські особи. Їхнє число може бути від 1 і більше. Використання "елітизму" дозволяє не втратити гарне проміжне рішення, але, у той же час, через це алгоритм може "застрягти" у локальному екстремумі. Однак, досвід використання принципу "елітизму" , дозволяє зробити висновок, що в більшості випадків "елітизм" анітрохи не шкодить пошукові рішення, і головне - це надати алгоритмові можливість аналізувати різні рядки з простору пошуку.

У класичному описі генетичного алгоритму мається на увазі створення популяції нащадків і заміщення батьківських особин їх "дітьми". Такий підхід досить гарний, але не особливо ефективний, тому що нащадки, отримані в результаті генетичних перетворень, можуть бути гірші батьківських особин. У результаті з'явилося кілька підходів, "виправляючих" дане явище, які можна об'єднати, використовуючи, як ми казали, поняття "елітизм" (elitism) (іноді "стратегія елітизму" (elitist strategy). Нижче буде приведено короткий опис цих підходів.

Умовно елітизм можна розділити на два класи-підходи, назвемо їх конкурентним і неконкурентним. Коротко їхня суть у наступному:

1. *Конкурентний підхід*. Батьківські особи "змагаються" з нащадками і переможці (або переможець) переходять у наступне покоління.

2. *Неконкурентний підхід*. У цьому випадку частина батьківської підпопуляції (випадкова або визначена за заданим правилом) переходить у нове покоління без яких-небудь заперечень з боку електорату.

Іншими словами, у першому класі нащадки після створення мають, загалом, рівні права з батьками на те, щоб перейти в нове покоління і визначальну роль тут грає пристосованість особи, а не її положення на генеалогічному дереві. В другому класі старі індивіди мають визначений пріоритет і навіть якщо всі нащадки будуть краще будь-якого батька, якась частина батьківської підпопуляції неминуче буде присутня у новому поколінні.

Розглянемо кожен клас більш докладно. При цьому будемо по можливості брати до уваги напрацювання в області еволюційних стратегій (evolutionary strategies).

1. Конкурентний підхід

Виділимо в класі конкурентних підходів 2 підкласи (у дужках приведені жаргонні назви, що зустрічаються):

• глобальне змагання (масове побоїще, жорстоке і нещадне)

• локальне змагання (бої помісних князьків на кулачках)

При глобальному змаганні спочатку створюються усі нащадки (ключове слово "усі"), які потім змагаються на загальних підставах з усіма батьками (ключове слово "усіма") і ті, хто виявляться кращими, незалежно від віку, переходять у нове покоління. Тобто після створення нащадків визначається їхня пристосованість і, знаючи пристосованість у поточній популяції (тобто в популяції, з якої вибиралися батьківські особи), можна, відсортувавши особин з поточної популяції і нащадків, сформувати популяцію для наступного покоління. У даному випадку оптиміст скаже, що вибрали кращу частину, а песиміст скаже, що відкинули гіршу частину, і обидва будуть праві, точно також як і у випадку зі склянкою з водою. Нащадки не обов'язково змагаються з усіма особами з поточної популяції, можна улаштувати чемпіонат узявши тільки батьківських особин і їхніх нащадків. Головне в глобальних змаганнях, щоб особи змагалися разом.

Трохи інший підхід використовується в локальних змаганнях.

Розглянемо досить розповсюджений випадок, коли дві батьківські особи використовуються для створення двох нащадків. Тоді, безпосередньо після створення, виконується оцінка нащадків, а потім нащадки змагаються тільки зі своїми батьками. Тут проблеми родини вирішуються усередині родини. У такий спосіб популяція нового покоління формується з переможців численних локальних змагань.

З погляду еволюційних стратегій конкурентний підхід відповідає (mu + lambda) стратегії (яка називається "плюсстратегія" (plus strategy)), де mu -- кількість батьківських особин, а lambda -- кількість створених нащадків.

2. Неконкурентний підхід

У неконкурентному підході все значно простіше. Після оцінки поточної популяції вибираються кілька найкращих особин (тобто особин які моють максимальну пристосованість) і вони "автоматично" попадають у наступне покоління. При цьому елітні особи можуть брати участь у схрещуванні нарівні з іншими особами. У силу того, що, таким чином, батьківські особи не змагаються в тій або іншій формі з нащадками, такий підхід і одержав назву неконкурентний. В еволюційних стратегіях неконкурентний підхід відповідає (mu, lambda) стратегії (дослівний переклад виглядає як "кома-стратегія" (comma strategy)), зміст змінних mu і lambda залишається колишнім.

Тут ми проаналізуємо наслідки використання, або не використання того або іншого підходу до елітизму. Що ж тягне використання елітизму? Очевидно, що при елітизмі повільніше обновляється генетична інформація в популяції, так як частина особин, точніше їх геноми, залишаються незмінними при зміні поколінь. І хоча таке можливо і без елітизма (наприклад, схрещування двох "схожих" батьківських особин може привести до створення ідентичних з ними нащадків) у випадку з елітизмом ймовірність цієї події істотно збільшується, насамперед, у силу специфіки самого підходу до елітизму. Добре це чи погано?

Спробуємо проаналізувати.

Для цього необхідно зрозуміти, коли (не)вигідне інтенсивне відновлення геномів популяції. Очевидно, що якщо популяція "наближається" до глобального оптимуму (тобто знаходиться на порівняно невеликому від нього видаленні), то різкі зміни особин (більшість з яких мають високу пристосованість) небажані, тому що можуть погіршити відповідні цим особинам рішення. Тому, у даному випадку, елітизм, як спосіб збереження "гарних" особин, необхідний. Потрібно відзначити, що описана ситуація, як правило, не спостерігається на початку еволюції, за винятком випадків, коли вирішується дуже проста задача, але для них ГА, по великому рахунку, не потрібні. Якщо ж еволюційний пошук знаходиться на самому початку, то, мабуть, необхідно активно досліджувати простір пошуку, щоб виділити в ньому потенційні області, при влученні в які, популяція буде мати високу пристосованість. Однак, якщо в результаті генетичних перетворень виходить особа, що знаходиться в одній з таких областей (і допустимо має максимальну пристосованість у порівнянні з іншими особами в популяції) то необхідно, щоб характерні для цієї особи генотипічні (і, відповідно, фенотипічні) ознаки закріпилися в популяції, інакше дана область на невизначений час буде "загублена".

Висновки

Сьогодні продовжується впровадження генетичних алгоритмів у прикладні області та програми. Однак слід відмітити, що практичне застосування генетичні алгоритми знайшли лише для рішення дуже складних оптимізаційних задач, з якими інші алгоритми оптимізації не справляються.