Лабораторна робота №3

Виконав: Гаврилюк Максим Васильович, К28

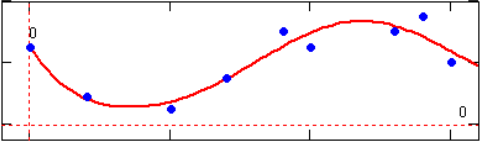
Обрана тема: “Побудова символьної регресії

за допомогою генетичного алгоритму”

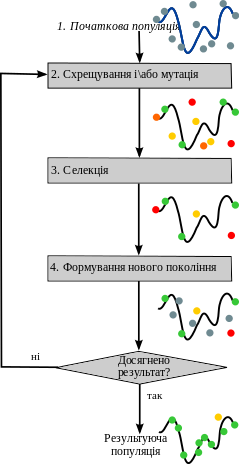
Використана бібліотека: **GAlib**

**Про тему:**

*Символьна регресія* — це тип регресійного аналізу, який використовується для пошуку найкращої моделі, що відповідає заданому набору даних. Тобто задача така, на вхід подається набір Х та F(X), а на вихід повертається формула, яка найкращим чином описує цей набір даних.



Функція на виході Вхідні дані

*Генетичний алгоритм* – це алгоритм пошуку, що використовується для вирішення задач оптимізації шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Задача кодується так, щоб вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Випадковим чином в масиві створюється деяка початкова популяція. Особи оцінюються з використанням функції допасованості, в результаті якої кожній особі присвоюється певне значення допасованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень допасованості вибираються особи, допущені до схрещення (селекція). До осіб застосовується «генетичні оператори» схрещення і оператор мутації, створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (поколінь), поки не буде виконано критерій зупинки алгоритму (кількість поколінь).

*В своїй лабораторній я зберігаю математичні вирази у вигляді польської нотації (спочатку знак операції, потім ідуть операнди), тому хромосомою є масив, де кожен елемент це або символ операції, або \*х\*, або константа.*

**Про використану бібліотеку:**

GAlib містить набір об’єктів генетичного алгоритму C++. Бібліотека містить інструменти для використання генетичних алгоритмів для оптимізації в будь-якій програмі C++ з використанням будь-якого представлення та генетичних операторів. Документація містить розширений огляд того, як реалізувати генетичний алгоритм, а також приклади, що ілюструють налаштування класів GAlib.

*GAlib розроблена Метью Волом (Matthew Wall), потім розробка та підтримка передані Массачусетському технологічному інституту (MIT). З тих пір бібліотека стала вільною з можливістю використання в комерційних цілях.*

Посилання: <http://lancet.mit.edu/ga/>

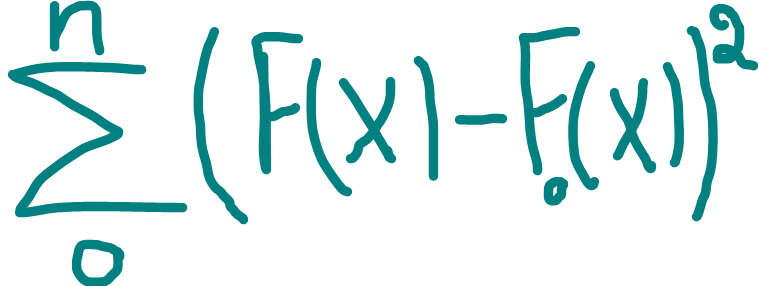
**Про алгоритм:**

Для початку потрібно визначити власні функції ініціалізіції хромосоми, мутації, кросинговеру.

*Функція ініціалізації:*

Створює випадковий вираз у виді польської нотації.

*Функція пристосованості:*

Сума квадратів різниць між фактичним значенням F(X) та F(X), що повертає задана хромосома, для кожго Х із вхідних даних.

Оскільки найкраще, коли ця сума = 0, то ми повертаємо 1/sum, в ідеалі sum = 0, функція пристосованості -> нескінченність.

*Функція мутації:*

Я додав декілька видів мутації:

1)Заміна операнда, операції на випадкову іншу.

2)Додавання або видалення випадкового коефіцієнта чи доданка.

3)Повна заміна виразу на випадковий інший.

4)Додавання, видалення тригонометричної функції попереду.

**5)Оптимізація коефіцієнтів виразу: хромосома проходить власний генетичний алгоритм, метою якого є підібрати максимально підходящі коефіцієнти.**

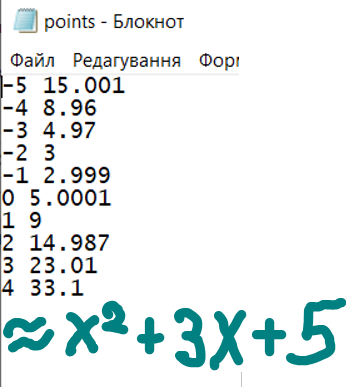
*Функція кросинговеру:*

На вхід подаються дві хромосоми, дві дочірні хромосоми отримують випадковим чином відповідний ген від першої або від другої хромосоми.

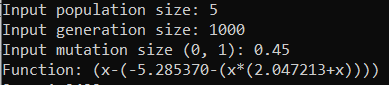
Після чого ці функції додаються до об’єкту нашої хромосоми.

Далі створюється об’єкт GASimpleAlg, передається наша хромосома, задаються: розмір кожної популяції, кількість популяцій, та вірогідність мутації.

Далі викликається метод evolve(), що проводить еволюцію, та повертається особина з найкращим значенням пристосованості (найменшою сумою квадратів), що і є шуканою функцією.

**Приклад роботи:**

**Вхідні данні**



**Вихідні данні**

Якщо побудувати цю функцію, то вона практично накладається та теоретичну:

