

УДК 621.372

Доц. В.М. Теслюк, канд. техн. наук; аспір. Р.З. Кривий;
аспір. Тарік (Мох'д Тайсір) Алі АльОмарі;
асист. Т.М. Теслюк – НУ "Львівська політехніка"

РОЗРОБЛЕННЯ ПІДСИСТЕМИ ДЛЯ РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ ЕВОЛЮЦІЙНИМИ МЕТОДАМИ

Описано особливості розроблення підсистеми для розв'язання оптимізаційних задач за допомогою генетичних алгоритмів. Розроблено структуру, лінгвістичне, програмне та математичне забезпечення підсистеми та результати тестування.

*Assoc. prof. V.M. Teslyuk; post-graduate R.Z. Kryvyj;
post-graduate Tarik (Mokh'd Taysir) Ali Alemari;
assist. T.M. Teslyuk – NU "Lviv'ska Politekhnik"*

Development of subsystem for decision of optimization tasks by evolutionary methods

The features of development of subsystem are described for the decision of optimization tasks by genetic algorithms. A structure, linguistic, programmatic and mathematical providing of subsystem and testing results, is developed.

Вступ. Задача пошуку оптимальних рішень – одна з найпоширеніших і актуальних. Тому виникає потреба в методах, алгоритмах і комп'ютерних підсистемах, які дають змогу користувачу знаходити оптимальні варіанти рішень серед множини альтернатив.

Практично всі системи, які використовують генетичний алгоритм (ГА) для розв'язання оптимізаційних задач (OGA, Awakener, EVO) [1], розроблялись без досвіду попередніх розробок. У цих системах відсутні засоби модифікації структури і операцій ГА. Цей чинник може негативно вплинути на результат роботи системи. Тому для розроблення таких підсистем найкраще використовувати об'єктно-орієнтований підхід, який на цей час вважають найкращим для опису об'єктів, їхніх властивостей та методів. Саме його було використано під час розроблення підсистеми Genet, що дають змогу продовжити удосконалення системи враховуючи її історію та результати роботи.

1. Розроблення структурної схеми підсистеми

Для створення підсистеми розв'язання оптимізаційних задач було побудовано структурну схему, яку зображено на рис. 1.

У підсистемі передбачено режими роботи: оброблення даних, виконання ГА і виводу результатів. Підсистема оброблення даних виконує такі функції, а саме: перевірку коректності введених даних та перетворення цільової функції в зворотний польський запис. Після цього дані передаються наступній підсистемі, в якій виконуються основні операції генетичних алгоритмів над переробленою цільовою функцією. Підсистема виводу результатів дає змогу переглянути результати у текстовому і графічному форматах.

Для опису цільової функції та обмежень використано числовий вираз, який є записом, складеним за певними правилами зі сталих, імен, знаків операцій та дужок. Сталі та імена у виразі позначають операнди, а знаки опера-

цій з дужками задають послідовність операцій, виконання яких породжує значення виразу.

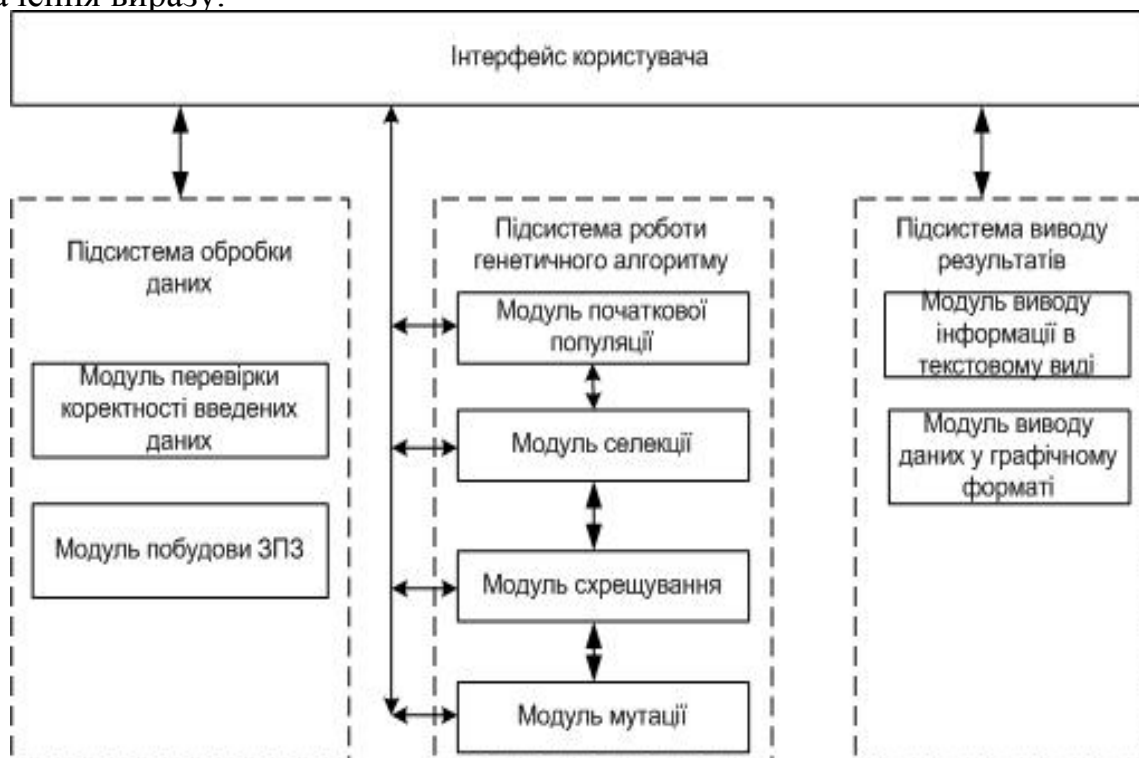


Рис. 1. Структурна схема підсистеми

2. Розроблення лінгвістичного забезпечення підсистеми

Вирази містять: цілі й дійсні сталі та імена змінних; символи "+", "-", "*", "/" двомісних арифметичних операцій; імена (ідентифікатори) одномісних функцій $\sin$ та $\cos$; дужки "(" та ")"

Задачу обчислення значення виразу розбито на дві підзадачі:

- прочитати вираз і побудувати його внутрішнє подання;
- за внутрішнім поданням виразу обчислити його значення.

Існує кілька способів представити послідовність операцій, задану виразом. Один із них – це подання виразу його зворотним польським записом (ЗПЗ) [1].

2.1. Алгоритм побудови виразів

Звичною формою представлення виразів є інфіксна, коли знак бінарної операції записується між позначеннями операндів цієї операції, наприклад, $a+b$. А запис знаків операцій після позначень операндів, тобто постфіксний запис, виглядає так $ab+$. Такий запис має також назву зворотного польського, оскільки його запропонував польський логік Ян Лукасевич [2]. Вираз є послідовністю символів-цифр, букв та інших знаків, що утворюють лексичні одиниці, або лексеми, чотирьох різновидів: сталі, імена (позначення функцій), знаки операцій та дужки. Вираз розглядають саме як послідовність лексем, які ми одержуємо послідовним читанням виразу зліва направо.

Процес побудови ЗПЗ виразу можна подати послідовністю вигляду – (вихідна послідовність лексем; магазин; непрочитана частина виразу). Ім'я функції записується в магазин і видається безпосередньо за ЗПЗ виразу її аргументу. Ім'я функції виштовхується з вершини з появою у вхідному виразі

знака операції або закриваючої дужки. Після того, як вираз прочитано, в магазині ще можуть залишитися знаки операцій; їх треба записати у вихідну послідовність.

Описане оброблення лексем подано фрагментом блок-схеми, яку зображено на рис. 2:

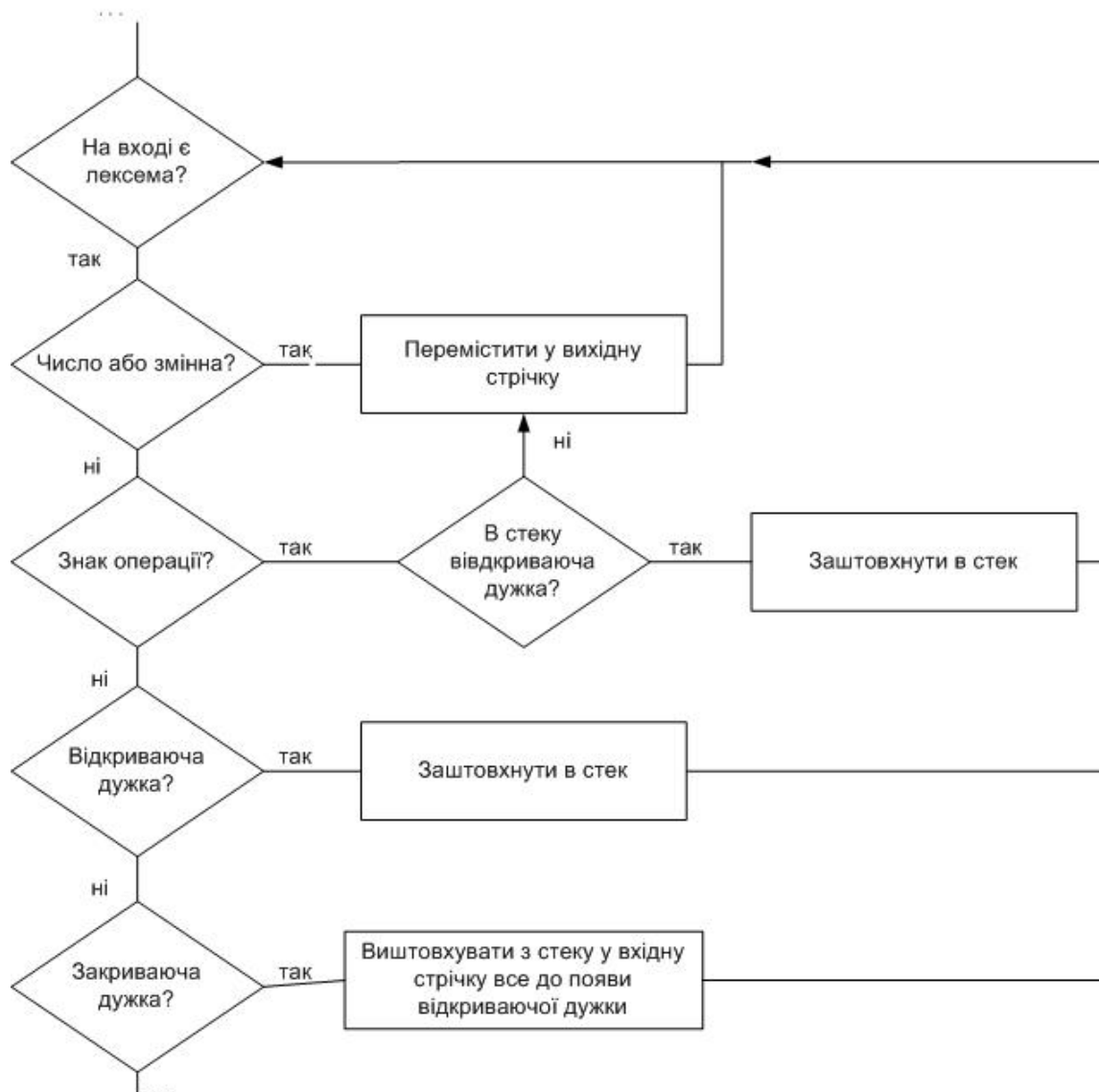


Рис. 2. Фрагмент блок-схеми алгоритму побудови ЗПЗ

2.2. Алгоритм обчислення виразу за його ЗПЗ

Позначення операндів у ЗПЗ передують знакам операцій, які до них застосовуються, тому при читанні ЗПЗ спочатку обчислюються та запам'ятовуються операнди, а потім до них застосовується операція.

Потім ЗПЗ виразу читається, а для обчислень застосовується стек. На цьому це вже стек операндів, а не знаків операцій. Числа, що є значеннями сталих чи змінних, переносяться в стек. Якщо черговою лексемою є знак двомісної операції, то зі стеку вилучаються два верхні елементи, і результат застосування операції до цих значень записується в стек. За знаку одномісної операції з стеку вилучається лише один елемент. Ім'я функції на вході задає її застосування до елемента з вершини стеку та вміщення результату в стек.

Після закінчення вхідного списку лексем у магазині зберігається лише одне число – значення всього виразу.

Процес обчислення можна подати послідовністю пар такого вигляду: стек операндів; непрочитана частина ЗПЗ. Спочатку стек порожній, а в кінці в ньому єдине значення. Оброблення ЗПЗ можна здійснити за алгоритмом фрагмент блок схеми якого зображено на рис. 3:

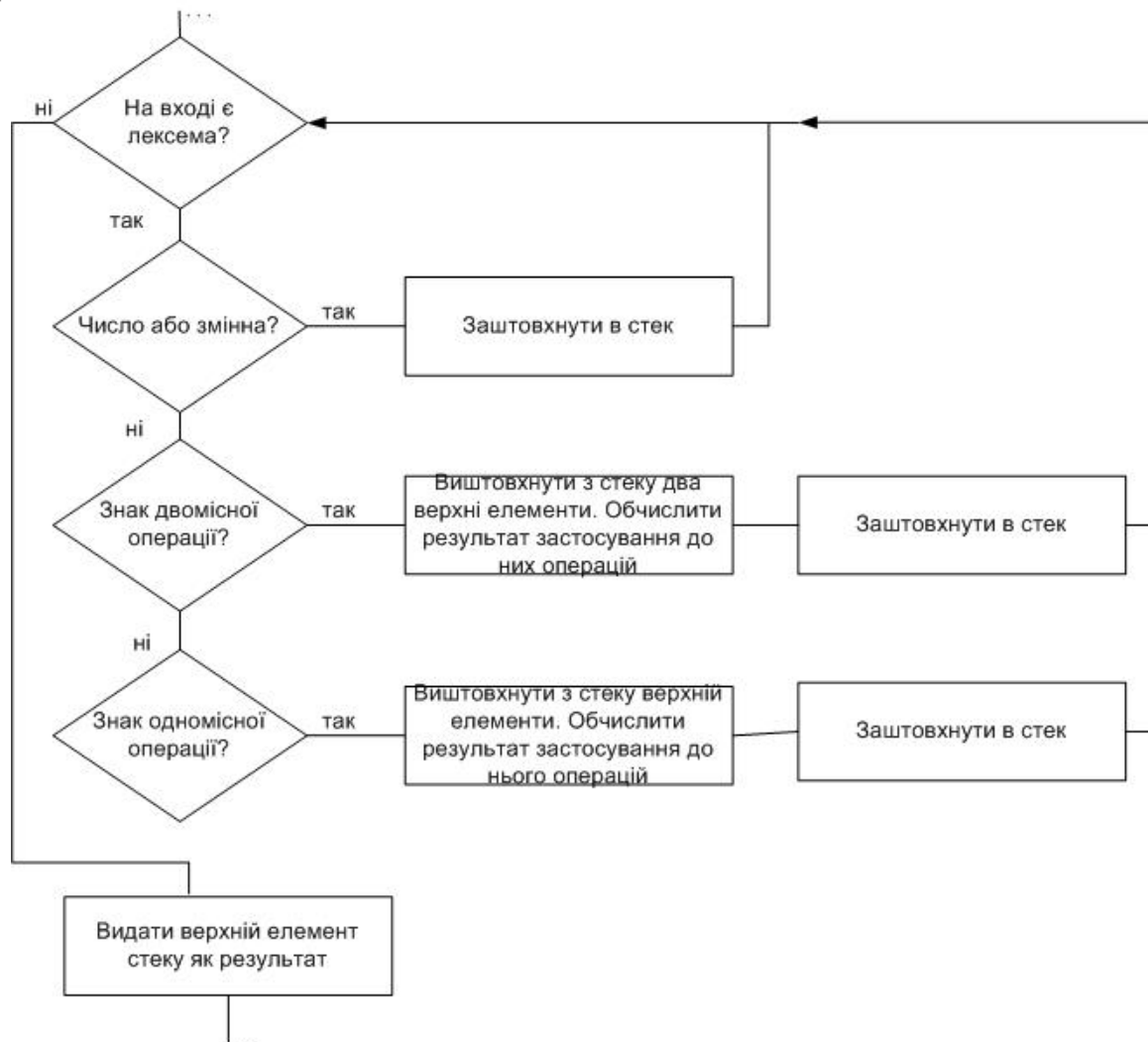


Рис. 3. Фрагмент блок-схеми алгоритму обчислення виразу за його ЗПЗ

2.3. Побудова основних алгоритмів роботи підсистеми

Розроблена підсистема функціонує за визначеним наперед алгоритмом, який використовується практично усіма генетичними алгоритмами. Робота цього алгоритму розпочинається з вводу вхідних даних, а саме: цільової функції, значень обмежень, кількості популяцій та хромосом та ін. [3].

Другий крок передбачає генерацію значень для хромосом, де використано генератор рівномірного закону розподілу випадкової величини. Наступний крок передбачає оцінку "здоров'я" хромосом та відбір найкращих результатів (найоптимальніших). Крок номер 5 призначений для виконання операції схрещування. Згідно до якої користувач на власний розсуд може встановити виконання цієї операції в автоматичному режимі. Разом з тим, в цьому блоці можуть відбуватися мутації, ймовірність яких, у вхідних даних, задає користувач підсистеми.

3. Основні меню підсистеми

Приклад основного меню розробленої підсистеми наведено на рис. 4. Основне меню містить три закладки: Вхідні дані, Протокол роботи, Графіки.

Генетичні алгоритми

Вхідні дані | Протокол роботи | Графіки

Кількість змінних: 9 [Крок1]

Обмеження для змінних

	x1	x2	x3	x4	x5	x6	x7	x8	x9
min	0	0	0	0	0	0	0	0	0
max	1	1	1	1	1	1	1	1	1

[Крок2]

Цільова функція: $1/(1+\text{sq}(x1)+\text{sq}(x2))$ ☒ мінімізація ☐ максимізація [Крок3]

Параметри запуску

Кількість особин: 2

Мутація: ☒ Використовувати мутація
Імовірність мутації: 1
Значення зміни: 0.01

Параметри виходу: ☒ Кількість поколінь: 50
☐ Відсутні зміни ЦФ
Межі відхилення: 0.01
N поколінь: 0.01 [Крок4]

[Старт]

Рис. 4. Закладка "Вхідні дані" (зміна кількості вхідних змінних)

Закладка "Вхідні дані" призначена для вводу вхідних даних підсистеми, зокрема: кількості змінних, значення меж для обмежень вхідних параметрів, вид цільової функції, вибору пошуку мінімуму чи максимуму, кількості особин, кількості поколінь та параметрів мутації. Після введення у відповідні поля всіх вищезазначених змінних необхідно натиснути кнопку "Початок" і підсистема розпочне виконувати розрахунки.

Отримані результати можна переглянути з допомогою закладок "Протокол роботи" та "Графіки" (див. рис. 5 і рис. 6).

Закладка "Протокол роботи" дає змогу користувачу переглянути результати розрахунку в текстовому виді, де наведено інформацію про покоління, значення змінних та значення цільової функції, відсоткове відношення отриманого значення цільової функції сили значень ЦФ усієї популяції.

Отримані результати відображаються за допомогою графіків. Зокрема ця інформація відображається з допомогою закладки "Графіки", де показана картина зміни середнього значення ЦФ по популяції та максимальне і мінімальне значення (див. рис. 5).

4. Результати дослідження та тестування розробленої підсистеми

Для перевірки правильності та коректності роботи підсистеми і розв'язання оптимізаційних задач використано тестові приклади, які наведені в літературних джерелах разом з розв'язанням [4]. Зокрема, використано тестову безумовну оптимізаційну задачу, де цільова функція має такий вигляд: $f(x, y) = 1/(1 + x^2 + y^2)$. Зрозуміло, що оптимум отримуємо при $x = y = 0$. Отримані

результати наведено на рис. 5 та 6, які збігаються з розрахунковим і є підтвердженням коректності та правильності роботи реалізованої підсистеми.

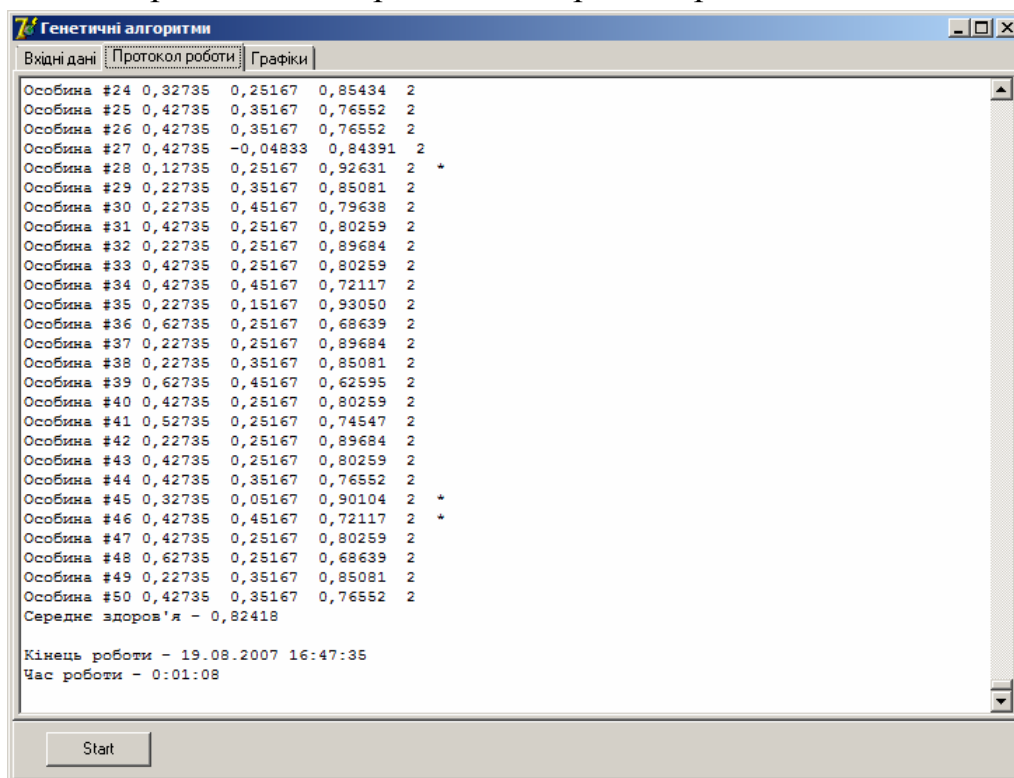


Рис. 5. Закладка "Протокол роботи" (зірочками позначено мутації)

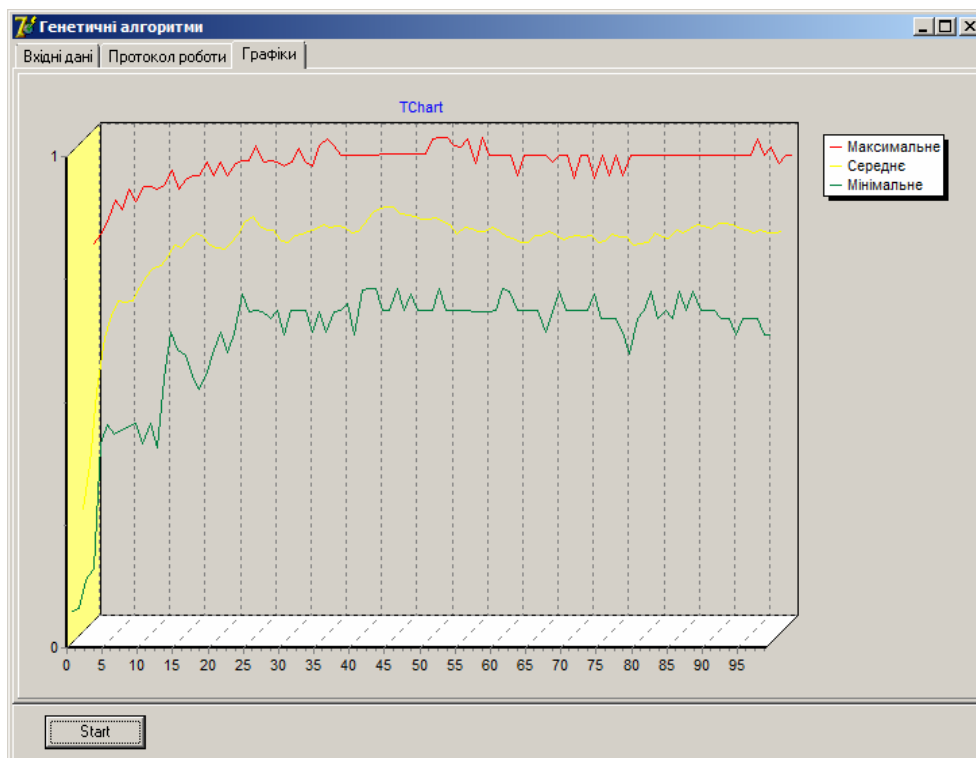


Рис. 6. Закладка "Графіки"

Висновки

Таким чином, розроблено підсистему Genet, зокрема побудовано її структуру, програмне та лінгвістичне забезпечення підсистеми, використано зворотний польський вираз для опису параметрів цільової функції. Наведено

результати тестування роботи основних складових підсистеми, що підтверджує коректність та правильність роботи підсистеми.

Література

1. [Електрон. ресурс]. – Доступний з: http://sourceforge.net/softwaremap/trove_list.php?form_cat=621.
2. **Lukasiewicz, Jan** (1957). Aristotle's Syllogistic from the Standpoint of Modern Formal Logic: Oxford University Press.
3. **Курейчик В.В., Нужнов Е.В.** Возможности организации интегрированной инструментальной среды поддержки процедур генетического поиска и оптимизации решений // Новости искусственного интеллекта, 2003. – № 5. – 236 с.
4. **Michalewicz, Zbigniew**. Genetic algorithms + data structures = evolution programs. – 3rd rev. and extended ad, Springer – Verlag Berlin Heidelberg New York, 1996. – P. 141-149.

УДК 681.3+519.6

Доц. О.А. Пастух, канд. техн. наук –
Європейський університет, Тернопільська філія

МАТЕМАТИЧНИЙ ФОРМАЛІЗМ ФУНКЦІОНУВАННЯ КВАНТОВОГО ПРОЦЕСОРА

Розглянуто математичний формалізм функціонування квантового процесора, охоплює математичний опис станів квантових бітів регістра квантового процесора у вигляді комплексного векторного простору хвильових функцій. Наведено математичний опис квантової статистичної моделі, яка інтегрує у собі основні стадії (запис, обчислення, вивід) роботи квантового процесора і показано неспроможність використання класичної статистичної моделі для цих потреб. Математичний опис однокубітних, двокубітного та багатокубітних логічних елементів (унітарних операторів).

Assoc. prof. O.A. Pastukh – European university, Ternopol department

Mathematical formalism of functioning quantum processor

Mathematical formalism of functioning quantum processor has been analyzed. It include mathematical description of states of quantum bits of register of quantum processor as of complex vector space of wave functions. Mathematical description of quantum statistical model if integrate bases phases (enter, computation, output) of operation quantum processor and show impossibility of classical statistical model for needs. Mathematical description of logical gates for one quantum bit, two quantum bit and many quantum bits (unitary operators).

Вступ. Все більше науковців, які працюють у сфері інформаційних технологій, у своїх роботах проявляють інтерес до квантових інформаційних технологій, зокрема до їх основного обчислювального елемента – квантового процесора. Це зумовлено тим, що квантовий процесор має ефект квантового паралелізму та квантової інтерференції, які розширюють його обчислювальні можливості порівняно з класичним процесором. Основою роботи квантового процесора, крім фізичної реалізації, є математичне забезпечення – математичний формалізм його функціонування.

Аналіз досліджень і публікацій. Математичний апарат, який використовується для математичної формалізації функціонування квантового процесора використовує надбання квантової теорії, теорії алгоритмів та ін. і висвітлений у багатьох роботах, як іноземних [1, 2], так і українських [3, 4] вчених.