

DOI: 10.31319/2519-8106.1(40)2019.166070

УДК 519.7

Л.О. Олійник, к.ф.-м.н, доц. каф. прикладної математики, ola19@i.ua

С.М. Бажан, здобувач ступеня доктора філософії 1-го року навчання, stasbazhan@gmail.com
Дніпровський державний технічний університет, м. Кам'янське

АЛГОРИТМ ПОШУКУ ЕКСТРЕМУМІВ ФУНКЦІЇ ОДНІЄЇ ЗМІННОЇ

Генетичні алгоритми досить широко сьогодні використовуються при розв'язанні великого кола задач оптимізації. При застосуванні генетичних алгоритмів процедура рекомбінації елементів області пошуку наближеного розв'язку виконується так званими операторами схрещування (кросоверу) та мутації. В цій роботі запропоновано алгоритм пошуку глобального екстремуму функції однієї змінної, де операторами кросоверу є стохастичні матриці другого порядку.

Ключові слова: генетичні алгоритми; оператори схрещування; кросовер.

Today, Genetic algorithms are widely used in solving a large range of optimization tasks. In the application of genetic algorithms, the procedure for recombining the elements of the search area of the approximate solution, performed by the operators of the crossing (crossover) and mutations. In this work, an algorithm for searching the global extremum of a function of one variable is proposed, where the crossover operators are stochastic matrices of the second order.

Keywords: genetic algorithms; cross-breeding operators; crossover.

Постановка проблеми

Генетичний алгоритм ([англ. genetic algorithm](#)) — це еволюційний алгоритм пошуку, що використовується для розв'язання задач оптимізації шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію [1]. В цій роботі використовується існуюча сьогодні в теорії генетичних алгоритмів термінологія. Вектори розв'язки — хромосоми, схрещування (кросовер) — обмін координатами (генами) між векторами розв'язками, Функція для якої визначається екстремум носить назву фітнес-функції (функції пристосованості). У цьому підході змінні, що характеризують рішення, представлені у вигляді генів в хромосом.

На даний момент однією з важливих питань в теорії генетичних алгоритмів і їх модифікованих версій є пошук найкращого балансу між продуктивністю і точністю.

Ефективність генетичного алгоритму залежить від вдалого створення первинної популяції. Для цього існує безліч підходів. Ефективність генетичних алгоритмів багато в чому залежить від досвіду програміста в цьому питанні. Це є свідченням недостатності теоретичної бази у застосування генетичних алгоритмів.

Аналіз останніх досліджень та публікацій

Виникнення елементів ГА датується 50-60-ми роками ХХ століття. Перші математичні описи та спроби симуляції еволюції були проведені у 1954 році Нільсом Баричеллі. Активізація застосування генетичних алгоритмів в Україні припадає на кінець минулого століття. Вже після 2000 років, з'являлись практичні та фундаментальні роботи, які були зареєстровані в електронному каталозі дисертацій України [2].

Сьогодні генетичні алгоритми займають чільне місце серед наближених методів розв'язання багатьох NP-складних задач і досить активно розвиваються, про що свідчить зростаюче число публікацій і конференцій, а також поява спеціалізованого журналу «The Genetic Algorithm Digest».

Формулювання мети дослідження

Метою даної роботи є розробка алгоритму наближеного розв'язку задачі пошуку глобального екстремуму функції однієї змінної на основі алгебраїчної модифікації формування популяції векторів-хромосом. На основі матричного кросоверу побудувати і обґрунтувати збі-

жність послідовності наближених розв'язків задачі про глобальний екстремум функції в заданій області пошуку.

Опис генетичного алгоритму для функції однієї змінної

Нехай $F(x)$ функція визначена на множині $[a, b]$. Необхідно знайти її глобальний мінімум. На відміну від звичайного генетичного алгоритму початкова популяція розв'язків-хромосом складається з двох точок ξ, η відрізка $[a, b]$, які обираються випадково.

Випадково обрані точки ξ, η представляються як координати двовимірного вектору $x = \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} \in R^2$. Цей вектор приймаємо за початкову батьківську хромосому. Також до множини батьківських хромосом відносимо вектори $x_a = \begin{Bmatrix} a \\ \xi \end{Bmatrix} \in R^2$, $x_b = \begin{Bmatrix} \eta \\ b \end{Bmatrix} \in R^2$. Отже, початкова популяція векторів-хромосом складається з трьох векторів, або чотирьох точок відрізка $[a, b]$. Процедuru рекомбінації, тобто отримання нової популяції векторів-хромосом потомків, отримуємо в результаті застосування до цих векторів лінійних операторів, що породжуються стохастичними двовимірними матрицями вигляду $A = \begin{pmatrix} \alpha & 1 - \alpha \\ 1 - \alpha & \alpha \end{pmatrix}$, де α — випадкові числа з відрізка $[0, 1]$. Матриці даного вигляду утворюють клас стохастичних інволютивних операторів, що виконують обмін координатами між векторами початкової популяції батьківських хромосом. Ця операція є аналогом кросоверу у класичному розумінні.

У просторі R^2 окрім евклідової будемо користуватись так званою «кубічною нормою» $\|x\| = \max(|\xi|, |\eta|)$ ([3]). Тоді норма матриці A дорівнює одиниці. Отже, кожний вектор після перетворень дасть по парі точок в кожній з трьох частин відрізка $[a, b]$. В результаті отримаємо вісім точок (з урахуванням ξ, η), які є точками нової популяції розв'язків. Для отриманих восьми точок обчислюються значення фітнес-функції $F(x)$. На кожному з трьох відрізків знаходимо найменше значення фітнес функції. Позначимо ξ_1 , точку в якій фітнес-функція має найменше з отриманих трьох значень, а η_1 — відповідає найменшому значенню фітнес-функції на інших двох відрізках. Таким чином утворюється новий відрізок $(\xi_1 - \eta_1, \xi_1 + \eta_1)$, тобто нова область пошуку. Якщо, $\xi_1 - \eta_1 < a$, або $\xi_1 + \eta_1 > b$, то граничними точками нового відрізка обираються граничні точки початкового відрізка. Зрозуміло, що $[\xi_1 - \eta_1, \xi_1 + \eta_1] \subseteq [a, b]$. Таким чином точка ξ_1 є першим наближенням розв'язку задачі. Перший ітераційний крок застосування алгоритму завершено.

На наступному ітераційному кроці процедура рекомбінації нових значень наближень шуканого розв'язку повторюється і визначається точка ξ_2 , де фітнес-функція має най менше значення у відрізку $[\xi_1 - \eta_1, \xi_1 + \eta_1]$. В результаті другого ітераційного кроку отримуємо нову область пошуку $[\xi_2 - \eta_2, \xi_2 + \eta_2] \subseteq [\xi_1 - \eta_1, \xi_1 + \eta_1] \subseteq [a, b]$.

Продовжуючи таким чином процес, отримуємо послідовність вкладених відрізків, яка гарантує збіжність послідовності точок ξ_k до розв'язку. Зупинка ітераційного процесу відбувається коли величин $|\xi_k - \xi_{k-1}|$ не перевищує задану точність обчислень.

Реалізацію наведеного алгоритму створено у вигляді програмного модуля на алгоритмічній мові C#, за допомогою комп'ютерного додатку *Microsoft Visual Studio*.

Аналіз отриманих результатів роботи програмного продукту

Програмний модуль призначений для знаходження екстремумів функцій однієї змінної і дозволяє дослідити етапи роботи алгоритму з певним рівнем ілюстрації ітераційних кроків.

В даній програмі можна обрати одну з запропонованих функцій для знаходження глобального екстремуму функцій. На рис. 1 наведено елементи інтерфейсу програми.

Для проведення тестування програми знаходження екстремуму обираємо функцію №1 яка має вигляд $F(x) = 10 + x^2 - 10 \cos(2\pi x)$ графік функції можна переглянути на вкладці графіків рис. 2, це відома функція яка носить назву функція Растригина ([4]). Функція Растригина — функція, яка використовується для тестування ефективності алгоритмів оптимізації, типовий приклад нелінійної мультимодальної функції. Запропонована в 1974 році Леонардом Растригиним. Знаходження мінімуму цієї функції є досить важким завданням через велику кількість локальних мінімумів.

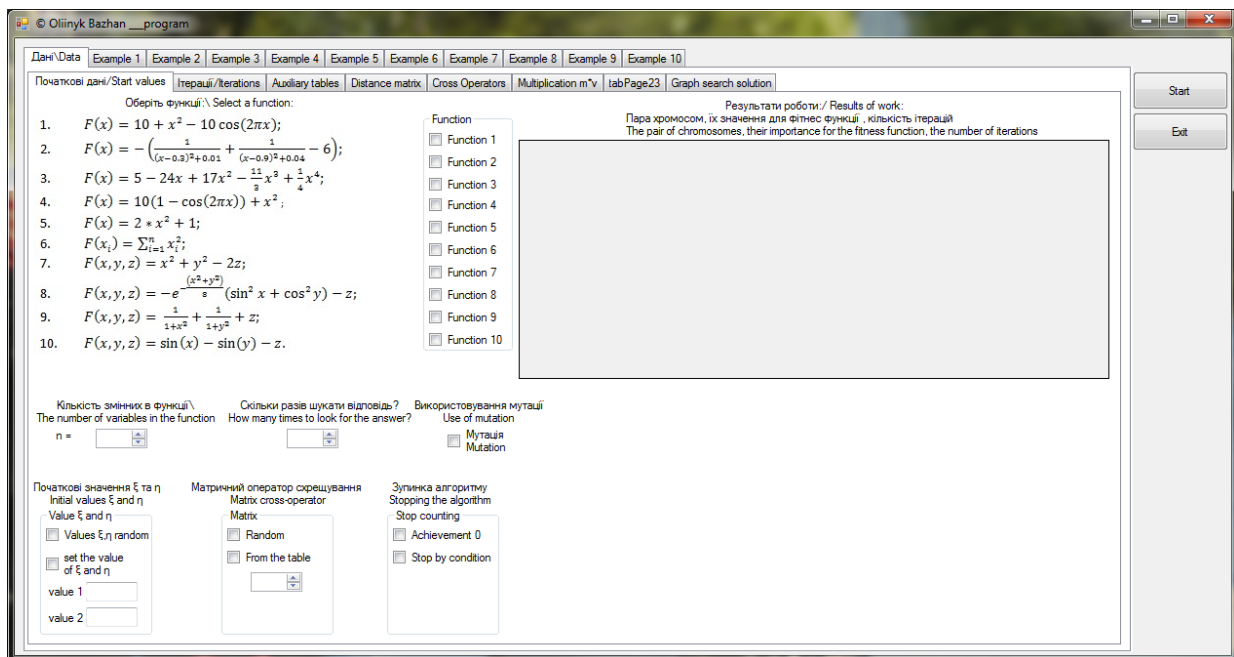
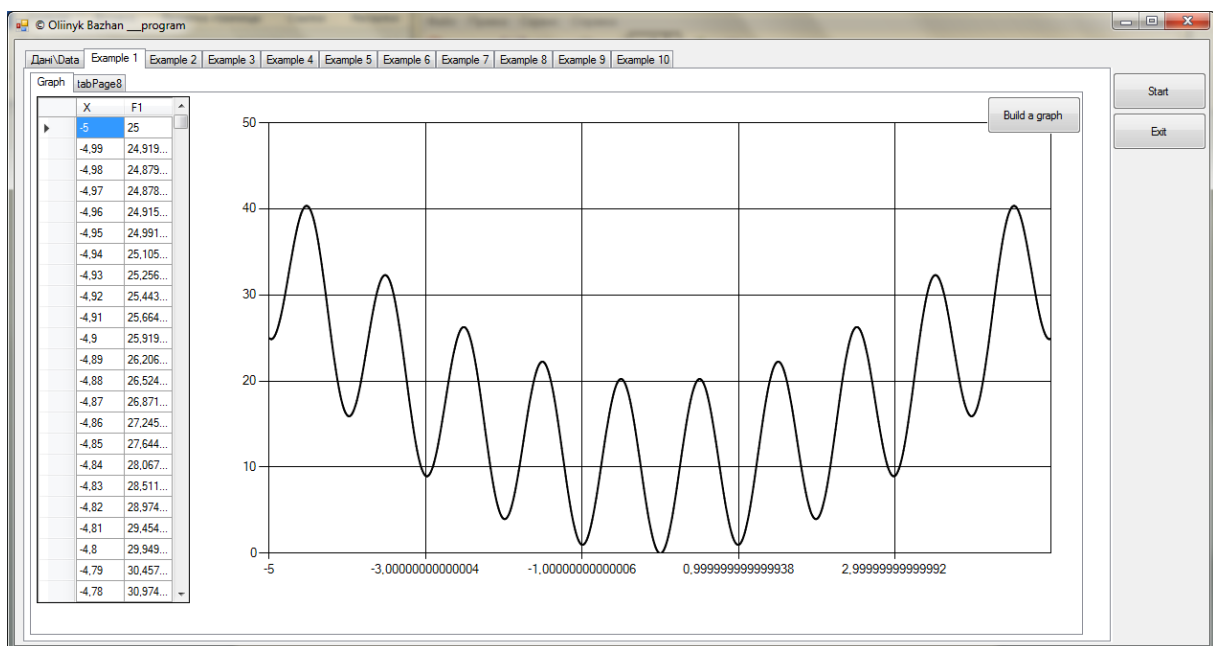


Рис. 1. Початкове вікно програми

Рис. 2. Графік функції Растригина на відрізку $[-5,12; 5,12]$

На початковій сторінці задаються початкові дані, надалі інші параметри задаються автоматично, для роботи генетичного алгоритму. Значення змінної n , яка відображає кількість змінних приймає функція. Обрати яку кількість разів алгоритм буде шукати мінімум від 1 до 10, наприклад 5 разів. Також користувач може обрати, як будуть задаватись початкові дані для першої батьківської хромосоми, випадково чи за значенням введеним користувачем. Також необхідно обрати спосіб зупинки алгоритму, наприклад якщо відомо, що функція має екстремум в точці нуль, це перший варіант, і другий — в будь-якій іншій точці. Не менш важливим є спосіб визначення матриці схрещування. Значення елементів матриці обираються випадково. Після введення всіх необхідних початкових даних рис. 3, програма виконує пошук екстремальної точки. Результат відображається на початковій сторінці, для кращої наочності.

Кількість змінних в функції \ The number of variables in the function
 n =

Скільки разів шукати відповідь? How many times to look for the answer?

Використовування мутації Use of mutation
 Мутація Mutation

Початкові значення ξ та η Initial values ξ and η
 Value ξ and η
 Values ξ, η random
 set the value of ξ and η
 value 1
 value 2

Матричний оператор схрещування Matrix cross-operator
 Matrix
 Random
 From the table

Зупинка алгоритму Stopping the algorithm
 Stop counting
 Achievement 0
 Stop by condition

Рис. 3. Вікно початкових даних для функції Растригина

Очевидно, що обрана функція має багато екстремумів, глобальний екстремум знаходиться в точці $x = 0, F(x) = 0$. Отримуємо результат роботи програмного модуля пошуку мінімуму функції за допомогою генетичного алгоритму.

Результати роботи / Results of work:
 Пара хромосом, їх значення для фітнес функції, кількість ітерацій
 The pair of chromosomes, their importance for the fitness function, the number of iterations

-1,49802514656571E-05	1,64714211138215E-05	4,45207586352581E-08	5,38253033255387E-08	274
-8,70323729968849E-06	5,75411813072734E-06	1,50274725996269E-08	6,568736665713E-09	68
-1,09152421147647E-05	8,92011210493278E-06	2,3636930635007E-08	1,57857407145912E-08	122
-9,19690842839131E-06	6,40822017139381E-06	1,67806231132772E-08	8,14702794116329E-09	74
-8,66507284343285E-06	8,90609879823847E-06	1,48959706791629E-08	1,57361821351287E-08	108

Рис. 4. Вікно результатів роботи алгоритму пошуку екстремуму

Екстремум шукався 5 разів, в перших двох стовпчиках відображаються точки в яких функція набуває глобального мінімуму, в наступних значення фітнес функції в цих точках та в останньому стовпчику — кількість ітерацій, які були витрачені на пошук відповіді. Кожний з п'яти варіантів пошуку розв'язка можна продивитись на спеціальній вкладці, де відображається видозміна роботи алгоритму рис. 5.

Видно, як змінюється різниця між попередніми значеннями фітнес функції та на останньому кроці. Отже генетичний алгоритм знайшов необхідний екстремум функції.

Аналіз результатів

Як було зазначено, для пошуку глобального мінімуму функції виконувалося п'ять спроб. Кількість ітерацій складала від 68 до 274. Як відомо, функція Растригина набуває екстремуму в точці $x = 0, F(x) = 0$, найкращий результат роботи алгоритму складає $x = 0,00000575, F(x) = 0,00000000657$, при чому жодного разу алгоритм не обирав хибний локальний екстремум. Отриманий результат наближений до відомого результату з похибкою $eps = 0,00000001$. На рис. 6, наведено динаміку зміни «найкращих» значень функції, при послідовній зміні поколінь в даному алгоритмі. Очевидно, що є спільна тенденція, спочатку графік поводить дещо хаотично, а потім починає стабілізуватись.

© Oliinyk Bazhan __program

Дані/Data Example 1 Example 2 Example 3 Example 4 Example 5 Example 6 Example 7 Example 8 Example 9 Example 10

Початкові дані/Start values Ітерації/Iterations Auxiliary tables Distance matrix Cross Operators Multiplication m^v TabPage23 Graph search solution

1 search 2 search 3 search 4 search 5 search 6 search 7 search 8 search 9 search 10 search

0	-5.12	-3.2797708...	-2.6192676...	-0.7790385...	0.7790385...	0.1185353...	-0.1185353...	2.6192676...	3.2797708...	5.12		2.6616782...	-0.1185353...
	28.924713...	22.616567...	24.181670...	8.7924599...	2.6616782...	2.6616782...	8.7924599...	24.181670...	22.616567...	28.924713...		3.1612355...	3.3983062...
1	-5.12	-3.6104782...	-3.0686739...	-1.5591522...	-0.2810769...	-0.8228811...	-1.0173479...	1.2284448...	1.7702491...	3.2797708...		1.0943434...	-1.0173479...
	28.924713...	30.721497...	10.333329...	21.748197...	1.0943434...	6.2562474...	12.019239...	10.158864...	11.864921...	22.616567...		2.5931302...	2.7875970...
2	-5.12	-3.8817541...	-3.4373175...	-2.1990717...	-1.1506791...	-1.5951158...	-1.7546350...	0.0875666...	0.5320033...	1.7702491...		1.4834546...	0.0875666...
	28.924713...	17.703323...	31.049553...	11.690328...	12.787557...	20.811106...	5.4807867...	1.4834546...	20.081536...	11.864921...		3.9693208...	0.4444366...
3	-3.8817541...	-2.8660333...	-2.5014663...	-1.4857455...	-0.6257595...	-0.5903265...	-1.1211785...	0.3899612...	0.7545283...	1.7702491...		0.9992121...	-0.9903265...
	17.703323...	11.552463...	26.256909...	22.167358...	4.0182453...	0.9992121...	17.428818...	17.855651...	10.284828...	11.864921...		1.8757068...	1.7448548...
4	-2.8660333...	-2.0328476...	-1.7337969...	-0.9006111...	-0.1951731...	-0.4942238...	-0.6015604...	0.6380126...	0.9370633...	1.7702491...		1.6498257...	0.9370633...
	11.552463...	4.3446943...	14.022363...	2.6984186...	18.394027...	20.237672...	6.6609485...	16.877013...	1.6498257...	11.864921...		2.9699110...	0
35	-0.0050766...	-0.0032830...	-0.0026392...	-0.0008455...	0.0006730...	2.9258571...	-0.002018...	0.002466...	0.0031104...	0.0049040...		1.6983632...	2.9258571...
	0.0051125...	0.0021382...	0.0013818...	0.0001418...	8.0799263...	1.6983632...	8.9867452...	0.0012070...	0.0019193...	0.0047709...		0.0033122...	0.0030811...
36	-0.0032830...	-0.0018117...	-0.0012836...	0.0001876...	0.0014333...	0.0009053...	0.0007157...	0.0029046...	0.0034327...	0.0049040...		6.9884547...	0.0001876...
	0.0021382...	0.0006511...	0.0003268...	6.9884547...	0.0001016...	0.0001626...	0.0004076...	0.0016738...	0.0023377...	0.0047709...		0.0019993...	0.0032450...
37	-0.0032830...	-0.0020761...	-0.0016429...	-0.0004360...	0.0005858...	0.0001526...	-2.8486399...	0.0017527...	0.0022258...	0.0034327...		1.6099015...	-2.8486399...
	0.0021382...	0.0008551...	0.0005354...	3.7719066...	1.6099015...	4.6218147...	6.8083941...	0.0006375...	0.0009829...	0.0023377...		0.0020732...	0.0022287...

Рис. 5. Процес пошуку мінімуму на різних ітераціях

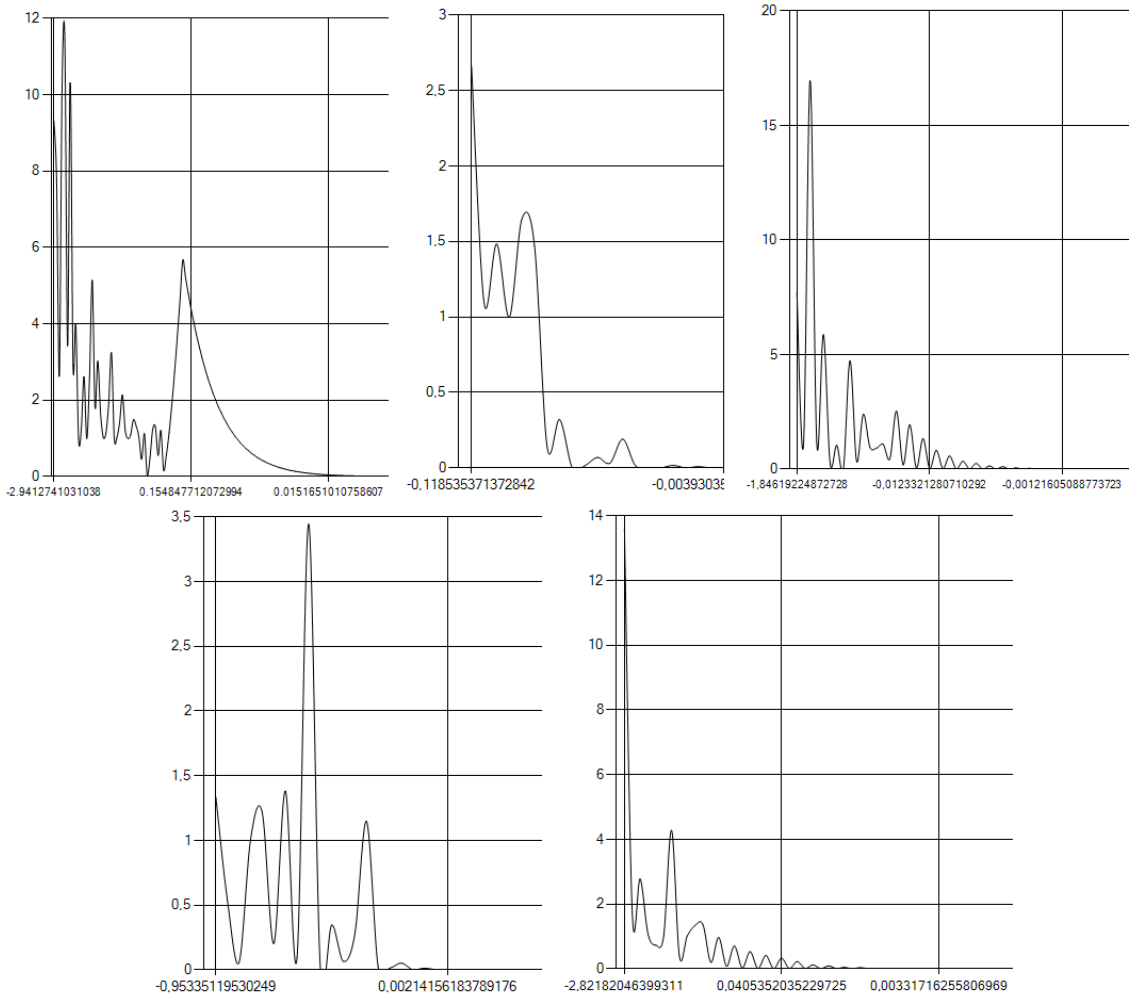


Рис. 6. Графіки пошуку глобального мінімуму функції Растрігіна

Висновки та перспективи подальших досліджень

В даній роботі був запропонований модифікований генетичний алгоритм та розроблено програмний модуль, що реалізує цей алгоритм. Отримані результати свідчать про те, що даний алгоритм є ефективним інструментом для розв'язання задач визначення глобального екстремуму функцій однієї змінної.

Список використаної літератури

1. Генетичні алгоритми [Електронний ресурс] // Вікіпедія — вільна енциклопедія – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Генетичний_алгоритм.
2. Електронний каталог дисертацій України – [Електронний ресурс] – Режим доступу до ресурсу: <http://www.disslib.org/catalogue.html>
3. Гантмахер Ф. Р. Теория матриц / Ф. Р. Гантмахер. – Москва: Наука, 1967. – 567 с.
4. Кононюк А. Е. Дискретно-непрерывная математика в 12-и кн./ Кононюк А. Е./ Алгоритмы – Киев: «Освіта України», 2017. – Кн. 10, Ч.3 – 444 с.

ALGORITHM FOR THE SEARCHING OF EXTREMA OF THE FUNCTION WITH ONE VARIABLE

Oliynyk L.O., Bazhan S.M.

Abstract

Today, genetic algorithms are widely used in solving a large range of optimization tasks. That is why it is relevant to study the theoretical foundations of the algorithms of this class.

In the article the authors propose an algebraic model of the algorithm for finding an approximate solution to the problems of finding the global minimum of functions with one variable.

Unlike the standard genetic algorithm, the initial population of chromosomes lies at two points. These points are considered as coordinates of a two-dimensional vector, that is altered in stochastic matrices. As a result, a set of approximate values of the solution of the problem is obtained. The convergence of the sequence of approximations is ensured by the fact that at each step of the algorithm, the search area (segment) is constructed in the search field of the previous step.

The results of this algorithm were checked on test functions with a rather complicated structure and gave a positive result as compared with the current data.

The algorithm is written in an algorithmic language *c#* with the help of the Microsoft Visual Studio application.

The obtained results testify that this algorithm is effective for solving tasks of determining the global extremum of functions with one variable.

References

- [1] Генетичні алгоритми [Genetic Algorithms]. (n.d.). [uk.wikipedia.org](https://uk.wikipedia.org/wiki/Genetic_Algorithm) Retrieved from: https://uk.wikipedia.org/wiki/Genetic_Algorithm [in Ukrainian]
- [2] Elektronnyy kataloh dysertatsiy Ukrayiny [Electronic catalog of theses of Ukraine]. (n.d.). [disslib.org](http://www.disslib.org/catalogue.html) Retrieved from: <http://www.disslib.org/catalogue.html>. [in Ukrainian]
- [3] Gantmacher F.R., (1967) *Teoriya matrits [The Theory of Matrices]*. Moscow: Science [in Russian].
- [4] Kononyuk A., *Dyskretno-neperervna matematyka v 12-ti kn. [Discrete-continuous mathematic in 12th book.] Algorithms, (Book 10, Part 3)*, Kiev: «Osvita of Ukraine», [in Russian].