18. CROSS RECURRENCE PLOT TOOLBOX. Retrieved from: http://tocsy.pik-potsdam.de/CRPtoolbox/index.html

SOLOVIEV Vladimir,
Doctor of Physical and Mathematical Sciences, Professor, Chair of the Department of Informatics and Applied Mathematics, Kryvyi Rih State Pedagogical University, Ukraine
SERDIUK Oleksandr,
Candidate of Economic Sciences, Department of Informatics and Applied Mathematics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

THE MODELS OF APPLICATION THE RECURRENCE ENTROPY AND RECURRENCE PERIOD DENSITY ENTROPY TO THE ANALYSIS OF COMPLEX SYSTEMS DYNAMICS

Summary. Introduction. *Complex systems are the systems with a large number of agents that interact with each other, and during such interactions new characteristics of the systems are generated. The using of quantitative methods in the modeling processes involves measurement procedures where the system complexity indicators are quite important. One such approach is to use the concept of entropy, which has been successfully used to study economic systems, which are also complex systems. Currently, among many ways of determinng the entropy of a complex system, there are several new ones that rely on the analysis of recurrence diagrams, a technique developed by Norbert Marwan. Consideration of two such methods was taken for the study conducted in this paper.*

The purpose *of the article is to describe and evaluate the possibility of applying to the economic series two concepts of entropy based on the analysis of recurrence diagrams: recurrence entropy and recurrence period density entropy.*

Results. *The paper deals with the method of calculation of two entropy types on the basis of Norbert Marwan recurrence diagrams: entropy of the recurrence period density and recurrence entropy. The behavior of entropies on synthetic time series with and without noise is demonstrated. The general neighborhood error that can be used in practical research are identified. Based on the application of the moving window procedure, the stability of the calculated values on synthetic time series is demonstrated. The results of application the entropies to time series DJIA and oil prices are demonstrated to determine the possibility of applying the entropy values considered.*

Conclusions. *The analysed indicators have shown a high sensitivity to critical and crisis phenomena in economic systems. There is a greater sensitivity of the recurrence period density entropy to changes in the system, and more time during which the recurrence entropy begins to respond to the preparation in the economic system of a critical phenomenon. Thus, the recurrence period density entropy and recurrence entropy can be successfully applied to the study of the state of economic systems and serve as pre-cursor of the arrival of uniques states in the economic system in the future.*

Keywords: *complex system, entropy, recurrence plot, critical event.*

Одержано редакцією 17.06.2019 р.
Прийнято до публікації 04.09.2019 р.

УДК 519.6: 004.023

DOI 10.31651/2076-5886-2019-2-34-43

PACS PACS 02.60.-x, 02.60.Pn, 02.70.-c

ЧЕРКАС Дар'я Валеріївна
магістрантка спеціальності «Прикладна математика» Черкаського національного університету імені Богдана Хмельницького
e-mail: cherkas.dasha1@gmail.com

КРАСНОШЛИК Наталія Олександрівна
кандидат технічних наук, доцент,
доцент кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького

e-mail: wlik007@ukr.net

ДОСЛІДЖЕННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ РОЗВ'ЯЗУВАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ

Генетичні алгоритми є одним з важливіших напрямів дослідження у еволюційному моделюванні. Їх часто використовують для знаходження розв'язку задач в різних областях, коли традиційні методи виявляються недостатньо ефективними.

У роботі розглянуто генетичні алгоритми Continuous genetic algorithm та Binary genetic algorithm для розв'язування задач глобальної оптимізації. Описано особливості розв'язування задачі комівояжера за допомогою генетичного алгоритму. Всі методи були реалізовані у середовищі GNU Octave. Генетичний алгоритм використано для розв'язання задачі комівояжера. Досліджено ефективність генетичних алгоритмів в залежності від значень його параметрів при знаходженні глобального мінімуму деяких тестових функцій. Проведено порівняння роботи генетичних алгоритмів з методом кажанів, методом імітації відпалу та класичними методами оптимізації.

Ключові слова: *генетичний алгоритм, задача оптимізації, задача комівояжера.*

Вступ

Дана робота присвячена одному з оптимізаційних методів. Як відомо, оптимізаційні задачі полягають у знаходженні мінімуму (максимуму) заданої функції. Таку функцію називають цільовою. Як правило, цільова функція – складна функція, що залежить від деяких вхідних параметрів. У оптимизаційних задачах потрібно знайти значення вхідних параметрів, при яких цільова функція досягає мінімального (максимального) значення. Існує цілий клас оптимізаційних методів. Умовно всі оптимізаційні методи можна розділити на методи, що використовують поняття похідної (градієнтні методи) і стохастичні методи (наприклад, методи групи Монте-Карло). За їх допомогою можна знайти екстремальне значення цільової функції, але не завжди можна бути впевненим, що отримано значення глобального екстремуму. Знаходження локального екстремуму замість глобального називається передчасною збіжністю. Крім проблеми передчасної збіжності існує інша проблема – час процесу обчислень. Найчастіше більш точні оптимізаційні методи працюють дуже довго. Для вирішення поставлених проблем і проводиться пошук нових оптимізаційних алгоритмів. Запропоновані у 1975 році Джоном Холландом генетичні алгоритми (ГА) засновані на принципах природного відбору Ч. Дарвіна. ГА відносяться до стохастичних методів. Ці алгоритми успішно застосовуються у різних областях діяльності (економіка, фізика, технічні науки і т.п.) [1].

Постановка проблеми

Генетичний алгоритм (genetic algorithm) – це еволюційний алгоритм пошуку, що використовується для розв'язання задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Задача кодується таким чином, щоб її розв'язання могло бути представлено у вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи допущені до схрещення (селекція). До осіб

застосовується «генетичні оператори» схрещення (crossover) і мутації (mutation), 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, поки не буде виконано критерій зупинки алгоритму:

Метою оптимізаційної задачі є вибір допустимого або оптимального розв'язку з множини альтернатив для досягнення поставленої мети. Оптимізаційна задача повинна відповідати двом основним вимогам: повинні існувати як мінімум два розв'язки, і треба знати, в якому сенсі шукане рішення має бути найкращим.

Математична модель оптимізаційної задачі складається з трьох складових: цільової функції, обмежень, граничних умов [2].

Пошук глобального мінімуму функції $f : R^n$ при наявності явних обмежень здійснюється на деякій власній підмножині Ω метричного простору R^n:

$$f(x) = f(x_1,...,x_n) \to \min,\ x \in \Omega,\ \Omega \subset R^n, \quad (1)$$

де підмножина Ω визначається обмеженнями типу рівностей:

$$q(x) = 0, \text{ де } q : R^n. \quad (2)$$

Метою даної статті є реалізація та дослідження генетичних алгоритмів розв'язування задач оптимізації.

Методи розв'язання

Класичний генетичний алгоритм оптимізації

В теорії ГА використовується біологічна термінологія в спрощеному вигляді [3]:

• хромосома – вектор чисел;

• ген – біт хромосоми;

• популяція – сукупність особин.

Як критерій життєздатності особини виступає функція пристосованості F – відображення сукупності хромосом особини на множину дійсних чисел. Точний вираз для функції пристосованості складається індивідуально для кожної задачі. В цьому випадку елементи x підмножини Ω в (1) можна розглядати як особини деякої популяції точок n-мірного простору, а значення функції $f(x)$ в цих точках можна розглядати як значення функції пристосованості $F = -f(x)$. Основні кроки роботи класичного ГА можна подати у наступному вигляді [3]:

1) генерація початкової популяції з k особин;

2) обчислення функції пристосованості кожної особини і відкидання найменш пристосованих особин;

3) вибір серед решти особин двох батьків;

4) застосування до батьківських хромосомах оператора схрещування з певною ймовірністю (при відсутності схрещування дочірні хромосоми рівні батьківським);

5) застосування до отриманих дочірніх хромосом оператора мутації з певною імовірністю (за відсутності мутації хромосоми не змінюються);

6) повторення кроків 3-5, поки популяція знов не буде містити k особин;

7) повторення кроків 1-6 до тих пір, поки не буде досягнуто критерію закінчення процесу.

Схрещування (кросовер) – це операція, при якій дві хромосоми обмінюються своїми частинами. У моделі, описуваної ГА, кросовер виконує оператор схрещування. Існують різні алгоритми оператора схрещування, найпростіші з них – це одно- і багатоточковий. В одноточковому варіанті відбувається розрив двох батьківських хромосом у випадковій позиції, після чого вони обмінюються отриманими ділянками. При багатоточковому кросовері хромосоми обмінюються ділянками, отриманими в результаті декількох розривів. Мутацією називається випадкова зміна одного або декількох генів хромосоми. Як критерій закінчення алгоритму можуть виступати дві умови:

- досягнення заданої максимальної кількості ітерацій (поколінь);
- збіжність популяції.

Збіжністю популяції називається такий її стан, коли всі особини популяції практично однакові і знаходяться в області деякого екстремуму. ГА контролюється наступними параметрами [3]:

- обсягом популяції k;
- ймовірністю мутації;
- ймовірністю схрещування;
- числом нових особин на кожному етапі розвитку популяції m;
- максимальним числом ітерацій алгоритму N;
- при використанні критерію збіжності популяції – точністю збіжності Q.

Особливості розв'язування задачі комівояжера за допомогою генетичного алгоритму

Постановка задачі комівояжера полягає у наступному. Нехай відстані між містами відомі та задані в матриці відстаней:

$$C = \begin{pmatrix} \infty & c_{12} & \dots & c_{1n} \\ c_{21} & \infty & \dots & c_{11} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & \infty \end{pmatrix}. \qquad (3)$$

Комівояжер повинен побувати у кожному місті по одному разу. Це означає, що він повинен увійти в кожне місто і вийти з нього по одному разу. Маршрут обходу міст може починатися з будь-якого міста. Але закінчитися він повинен у тому ж місті, з якого почався. Необхідно побудувати маршрут, що проходить через усі міста найменшої довжини. Знаки «∞» у матриці C, які розташовані по діагоналі, використовуються для заборони переходу з кожного міста в це ж місто, тобто для заборони повернення [4].

Задача полягає у тому, щоб мінімізувати функцію мети

$$F = \sum_{j=1}^{N} \sum_{i=1}^{N} C_{ij} \to \min . \qquad (4)$$

Для застосування генетичного алгоритму у якості особини приймають запис, а у кожної особини є набір хромосом. Задають його одновимірним масивом, який є послідовністю обходу заданих пунктів із поверненням у пункт відправлення. Так само є покоління із k особин, у яких є власний маршрут обходу. Наступним етапом є їх розмноження та мутація. Для кожної особини рахують функцію придатності, в даній задачі це довжина маршруту. Чим вона менша, тим краще. Найбільш пристосованих особин схрещують, перевіряючи, щоб не було повторень. Це необхідно для того, що в кожному пункті потрібно побувати тільки один раз. Після схрещення отримують нове покоління, у якому знов шукають найбільш пристосовані особини і т.д., поки не буде знайдене мінімальне значення функції пристосованості. Перевага генетичного

алгоритму в тому, що при розмноженні не потрібно обирати від батьків найкраще, тому що краще рішення все одно отримаємо через декілька поколінь, а іноді обираючи гірше можна отримати кращий результат [2].

Один з операторів, що відіграє вирішальну роль у розв'язанні задачі комівояжера, здійснюючи обмін інформацією між особинами під час генерації та допомагає отримати оптимальний розв'язок – це кросовер. В останні 20-30 років були запропоновані кілька кросоверів для задачі комівояжера. Один із них – PMX (partially mapped crossover) [5], принципи його роботи полягає у наступному:

1) вибір підрядків: вирізати два підрядки однакового розміру для кожного з батьків на одних і тих же позиціях;

2) обмін підрядками: обміняти два обрані підрядки, щоб створити прототип нащадка;

3) визначення списку відображень: визначити співвідношення відображення на основі вибраних підрядків;

4) прийняття потомства: прийняти прототипи нащадків за допомогою співвідношення відображення.

Результати дослідження

Continuous genetic algorithm та Binary genetic algorithm були реалізовані у середовищі розробки GNU Octave. Проведено дослідження Continuous genetic algorithm та Binary genetic algorithm для розв'язання задачі (1) з використанням різних значень параметрів npar – кількість змінних оптимізаційної функції (параметрів), maxit – максимальне число ітерацій, popsize – розмір популяції, mutrate – швидкість мутації, selection – частка популяції, що зберігається. Отримані результати наведено у табл. 1.

Було проведено порівняння Continuous genetic algorithm та Binary genetic algorithm. Отримані результати свідчать, що при збільшенні частоти мутацій погіршується ефективність роботи алгоритмів, отже, оптимальним значенням цього параметру є значення від 0.1 до 0.5. Якщо популяцію (popsize) зробити завеликою або замалою, то алгоритми теж показують гірші результати. При дуже великому значенні selection, наприклад, якщо ми залишимо 70% «населення», також отримаємо гірші результати, ніж якщо б залишили 50%. Порівняльний аналіз результатів роботи Continuous genetic algorithm та Binary genetic algorithm показав, що Continuous genetic algorithm демонструє кращі результати.

Таблиця 1

Результати обчислювальних експериментів

Вхідні параметри	Назва алгоритму	
	Continuous genetic algorithm	Binary genetic algorithm
	Тестова функція: $f(x_1, x_2) = x_1 \sin(4x_1) + 1.1x_2 \sin(2x_2)$ Точний розв'язок: $f(9.039, 8.668) = -18.5547$, $0 \le x_1 x_2 \le 10$	
npar = 2, maxit = 300 popsize = 12 mutrate = 0.15 selection = 0.5	$f(x_1 x_2) = -18.5547$ $x_1 = 9.0384$; $x_2 = 8.6693$	$f(x_1 x_2) = -18.5275$ $x_1 = 9.0196$; $x_2 = 8.6667$

Продовження таблиці 1

npar = 2 maxit = 200 popsize = 20 mutrate = 0.2 selection = 0.5	$f(x_1x_2)$ = -18.5351 x_1=9.039; x_2 =8.6361	$f(x_1x_2)$ = -18.5275 x_1=9.0196; x_2 =8.6667
npar = 2 maxit = 200 popsize = 12 mutrate = 0.8 selection = 0.5	$f(x_1x_2)$ = -18.552 x_1=9.0389; x_2 =8.6564	$f(x_1x_2)$ = -18.4945 x_1=9.0588; x_2 =8.6275
npar = 2 maxit = 200 popsize = 12 mutrate = 0.15 selection = 0.7	$f(x_1x_2)$ = -18.5518 x_1=9.0337; x_2 =8.6751	$f(x_1x_2)$ = -18.5249 x_1=9.0187; x_2 =8.6663
npar = 2 maxit = 200 popsize = 8 mutrate = 0.15 selection = 0.7	$f(x_1x_2)$ = -18.551 x_1=9.046; x_2 =8.6709	$f(x_1x_2)$ = -18.268 x_1=9.0196; x_2 =8.7843

Порівняємо роботу ГА з іншими евристичними методами оптимізації. Для порівняння обрали метод кажанів та метод імітації відпалу, які також були реалізовані у середовищі Octave. Отримані результати наведено у табл. 2.

Таблиця 2

Результати обчислювальних експериментів

Назва алгоритму			
Continuous genetic algorithm	Binary genetic algorithm	Метод кажанів	Метод імітації відпалу
Тестова функція: $f(x_1,x_2) = 0.5 + \frac{\sin^2\left(x_1^2+x_2^2\right)^2 - 0.5}{1+0.001\left(x_1^2+x_2^2\right)^2}$ Точний розв'язок: $f(0,0) = 0$, $-100 \le x_1, x_2 \le 100$			
$f(x_1x_2)$ = 1.26e-09 x_1=0.037726 x_2=0.012659	$f(x_1x_2)$ = 0 x_1=0 x_2=0	$f(x_1x_2)$ = 1.1218e-07 x_1=-0.080498 x_2= 0.08097	$f(x_1x_2)$ = 3.967854 x_1= 0.015863 x_2= 0.005481
Тестова функція: $f(x_1,x_2) = 0.5 + \frac{\sin^2\sqrt{x_1^2+x_2^2} - 0.5}{1+0.01\left(x_1^2+x_2^2\right)}$ Точний розв'язок: $f(1.897,1.006) = -0.5231$, $-\infty \le x_1, x_2 \le \infty$			
$f(x_1x_2)$ = -0.52311 x_1= 0.30007 x_2= 2.126	$f(x_1x_2)$ = -0.52311 x_1= 2.1176 x_2= 0.35294	$f(x_1x_2)$ = -0.52311 x_1= 1.4073 x_2= 1.6216	$f(x_1x_2)$ = -0.48758 x_1= 0.058699 x_2= -0.201340

Продовження таблиці 2

Тестова функція: $f(x_1,x_2)=J_0\left(x_1^2+x_2^2\right)+0.1\lvert 1-x_1\rvert+0.1\lvert 1-x_2\rvert$ Точний розв'язок: $f(1,1.6606)=-0.3356,\ -\infty\le x_1,x_2\le\infty$			
$f(x_1,x_2)$=-0.32108 x_1= 0.9074 x_2= 1.7111	$f(x_1,x_2)$=-0.33482 x_1= 1.6471 x_2= 1.0196	$f(x_1,x_2)$=-0.33211 x_1= 1.5932 x_2= 1.0964	$f(x_1,x_2)$=0.30187 x_1= -0.41483 x_2= 0.40925
Тестова функція: $f(x_1,x_2)=x_1\sin(4x_1)+1.1x_2\sin(2x_2)$ Точний розв'язок: $f(9.039,8.668)=-18.5547,\ 0\le x_1,x_2\le 10$			
$f(x_1,x_2)$=-18.5547 x_1= 9.0384; x_2= 8.6693	$f(x_1,x_2)$=-18.5275 x_1= 9.0196; x_2= 8.6667	$f(x_1,x_2)$=-18.5275 x_1= 9.0196; x_2= 8.6667	$f(x_1,x_2)$=-18.220 x_1= 8.5295; x_2= 8.5057

Було проведено порівняння Continuous genetic algorithm, Binary genetic algorithm з методом кажанів та методом імітації відпалу. Порівняльний аналіз результатів роботи показав, що алгоритми Continuous genetic algorithm, Binary genetic algorithm показали кращі результати для більшості функцій.

Порівняємо роботу ГА з класичними чисельними методами безумовної багатовимірної мінімізації, які були реалізовані у середовищі Octave. Отримані результати наведено у табл. 3.

Таблиця 3

Результати обчислювальних експериментів

Назва алгоритму							
Continuous genetic algorithm	Binary genetic algorithm	Метод координатного спуску	Метод конфігурацій (Хука-Дживса)	Метод найшвидшого градієнтного спуску	Метод спряжених градієнтів (Флетчера-Рівса)	Метод Ньютона	Метод Марквардта
Тестова функція: $f(x)=45x_1^2-88x_1x_2+45x_2^2+102x_1+268x_2-21$ Точний розв'язок: $f(-92.9613,-93.9173)=-1.7172e+04$							
-17170.4132 -92.9613 -93.9173	17170.4132 -92.9613 -93.9173-	-1.7172e+04 -92.0335 -92.9661	-1.7172e+04 -92.0337 -92.9663	-1.7172e+04 -92.0337 -92.9663	-1.7172e+04 -92.0334 -92.9660	-1.7172e+04 -92.0337 -92.9663	-1.7172e+04 -92.0337 -92.9663
Тестова функція: $f(x)=\sin^2 3\pi x_1+(x_1-1)^2(1+\sin^2 3\pi x_2)+(x_2-1)^2(1+\sin^2 3\pi x_2)$ Точний розв'язок: $f(1,1)=0,\ -10\le x_1,x_2\le 10$							
1.8415e-05 1.0001 1.0041	0.034559 1.0196 1.0196	3.5299 2.3183 2.3291	0.0051012 1.0075 1.0075	0.10987 1.3296 1.0000	0.10987 1.3296 1.0000	7.9491 -0.97702 1.02298	7.9491 -0.97702 1.02298
Тестова функція: $f(x)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$ Точний розв'язок: $f(0,0)=0,\ -5\le x_1,x_2\le 5$							
8.52e-07 0.00060753 -0.00075773	0 0 0	0.29864 -1.74755 0.87378	0.00062497 -0.012500 -0.012500	0.29864 -1.74755 0.87373	0.50234 -1.53909 0.58679	0.87736 -1.07054 0.92946	0.0000000001 4035 0.0000049927 2.0000049927

Було проведено порівняння Continuous genetic algorithm, Binary genetic algorithm з методом координатного спуску, методом найшвидшого градієнтного спуску, метод спряжених градієнтівм (Флетчера-Рівса), методом Ньютона і методом Марквардта. Порівняльний аналіз результатів роботи показав, що алгоритми Continuous genetic algorithm, Binary genetic algorithm показали кращі результати для більшості функцій.

Застосуємо генетичний алгоритм до розв'язання задачі комівояжера на тестовому прикладі з 20 міст. Отримані результати наведено на рис. 1.

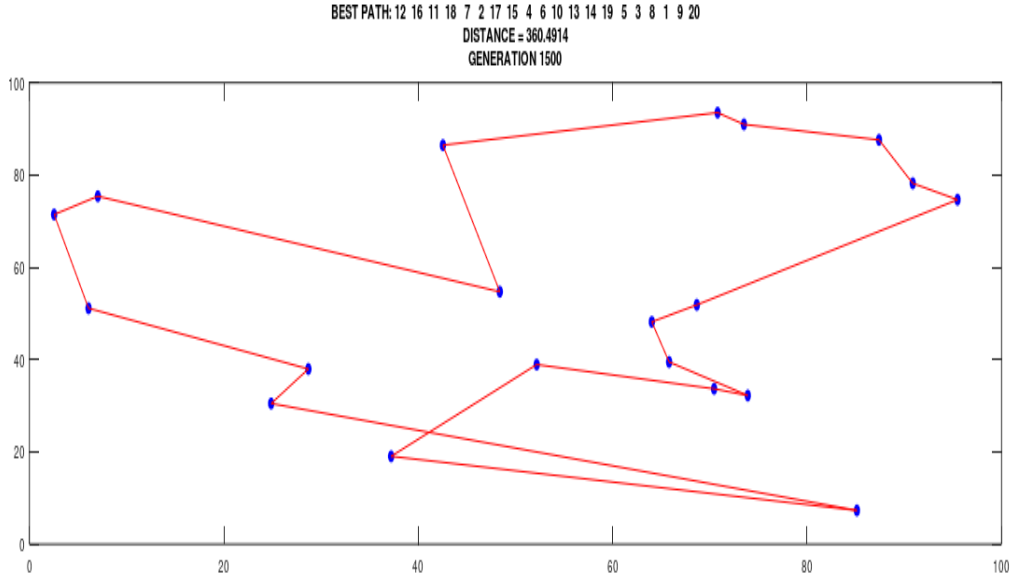

Рис. 1 Маршрут комівояжера на ітерації 1500

Мінімальна відстань, якої вдалося досягти в за допомогою генетичного алгоритму 360.4914. Це доводить ефективність ГА в таких важких проблемах, як задача комівояжера. На прикладі цієї задачі можна спостерігати, як ГА створює розв'язок, не маючи попередніх знань про маршрути подорожі. На відміну від інших евристичних методів, ГА використовує природні правила відбору, кросовера та мутації, щоб зробити обчислення легшими та швидшими. Ці речі роблять його більш цінним, більш ефективним ніж інші евристичні алгоритми.

Висновки

У роботі розглянуто основні поняття і принцип роботи генетичних алгоритмів. Розроблено програми у середовищі GNU Octave, які реалізують Continuous genetic algorithm, Binary genetic algorithm для знаходження мінімуму функцій та розв'язання задачі комівояжера.

Досліджено залежність ефективності даних алгоритмів від розміру популяції, швидкості мутації, частки популяції, що зберігається та максимального числа ітерацій. В результаті обчислювальних експериментів визначено оптимальні значення для наведених параметрів.

Проведено порівняння роботи генетичних алгоритмів з евристичними та класичними методами оптимізації. Серед класичних методів розглянуто метод координатного спуску, метод найшвидшого градієнтного спуску, метод спряжених градієнтів (Флетчера-Рівса), метод Ньютона і метод Марквардта, а серед евристичних – метод кажанів, метод імітації відпалу. Встановлено, що алгоритми Continuous genetic

algorithm та Binary genetic algorithm показали кращі результати для більшості тестових функцій.

Таким чином, доцільно застосовувати розглянуті генетичні алгоритми для задач мінімізації складних функцій.

Список використаної літератури:

1. Панченко, Т. В. Генетические алгоритмы: учеб.-метод. пособие / Т. В. Панченко. – Астрахань: Астрахан. ун-т., 2007. – 87 с.
2. Гладков, Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: ФИЗМАТЛИТ, 2010. – 368 с.
3. Савин, А. Н. Применение генетических алгоритмов для решения задач оптимизации на параллельных и распределенных вычислительных системах / А. Н. Савин, И. В. Дружинин, А. А. Ерофтиев // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. – 2013. – Т. 13, вып. 1. – С. 99-109.
4. Івченко, І.Ю. Математичне програмування: Навчальний посібник / І.Ю. Івченко. – К.: Центр учбової літератури, 2007. – 232 с.
5. Goldberg, D.E. Alleles, Loci and the Traveling Salesman Problem / D.E. Goldberg, & R. Lingle // Proceedings of the First International Conference on Genetic Algorithms. – 1985. – P. 154-159.

Bibliography:

1. Panchenko, T.V. (2007). Geneticheskie algoritmy: ucheb.-metod. posobie [Genetic algorithms: the methodical manual]. Astrakhan: Univer. Astrakhan [in Russian].
2. Gladkov, L.A., Kurejchik, V.V., & Kurejchik, V.M. (2010). Geneticheskie algoritmy [Genetic algorithms]. Moskva: FIZMATLIT [in Russian].
3. Savin, A. N., Druzhinin, I. V., Eroftiev, A. A. (2013). Primenenie geneticheskikh algoritmov dlya resheniya zadach optimizaczii na parallelnykh i raspredelennykh vychislitelnykh sistemakh [The use of genetic algorithms to solve optimization problems on parallel and distributed computing systems]. *Izv. Sarat. un-ta. Nov. ser. Ser. Matematika. Mekhanika. Informatika*, 13, 1, 99-109 [in Russian].
4. Ivchenko, I. Yu. (2007), Matematichne programuvannya: Navchalnij posibnik [Mathematical Programming: A Tutorial]. Kyiv: Czentr uchbovoyi literaturi [in Ukraine].
5. Goldberg, D.E., & Lingle, R. (1985). Alleles, Loci and the Traveling Salesman Problem. *Proceedings of the First International Conference on Genetic Algorithms*, 154-159.

CHERKAS Dariya,
student, The Bohdan Khmelnytsky National University of Cherkasy
KRASNOSHLYK Natalia
Candidate of Technical Sciences, Associate Professor, The Bohdan Khmelnytsky National University of Cherkasy

RESEARCH OF GENETIC ALGORITHMS OF SOLUTION OF OPTIMIZATION PROBLEMS

Summary. Introduction. *Genetic algorithms are one of the most important areas of research in evolutionary modeling. They are often used to solve problems in different areas when traditional methods are not effective enough.*

A genetic algorithm is an evolutionary search algorithm used to solve optimization and modeling problems by sequentially selecting, combining, and varying the parameters sought using mechanisms that resemble biological evolution.

The purpose *of this article is to implement and investigate genetic algorithms for solving optimization problems.*

Results*. The basic concepts and principle of operation of genetic algorithms are considered in the work. The formulation of the problem of unconditional optimization and the travelling salesman problem is given. The peculiarities of solving the travelling salesman problem using a genetic algorithm are described.*

GNU Octave applications have been developed that implement Continuous genetic algorithm and Binary genetic algorithm for minimization of functions and solve the travelling salesman problem. The dependence of the efficiency of these algorithms on the size of the population, the mutation rate, the proportion of the stored population and the maximum number of iterations are investigated. As a result of computational experiments, the optimal values for the given parameters are determined.

Comparison of genetic algorithms with heuristic and classical optimization methods is compared. The classic methods include coordinate descent method, gradient descent method, conjugate gradient method (Fletcher-Reeves method), Newton method and Marquardt method, and among the heuristic methods, bat algorithm and simulated annealing.

Conclusions. *Genetic algorithms have been used for minimization of functions and solve the traveling salesman problem. The efficiency of genetic algorithms is studied depending on the values of its parameters when finding the global minimum of some test functions. Comparison of genetic algorithms with bat algorithm, simulated annealing and classical optimization methods are compared. Continuous genetic algorithm and Binary genetic algorithm have been found to perform better for most test functions.*

Thus, it is advisable to apply the genetic algorithms under consideration to solve optimization problems.

Keywords: *genetic algorithm, optimization problem, traveling salesman problem.*

Одержано редакцією 18.07.2019 р.
Прийнято до публікації 09.10.2019 р.

УДК 519.6

DOI 10.31651/2076-5886-2019-2-43-58

ГОЛОВНЯ Борис Петрович
доктор технических наук, доцент кафедры прикладной математики и информатики Черкасского национального университета имени Богдана Хмельницкого
e-mail: bpgolovnya@gmail.com
ORCID 0000-0002-9242-3937

К ВОПРОСУ О ПРЕПОДАВАНИИ МЕТОДА СОПРЯЖЕННЫХ ГРАДИЕНТОВ

Метод сопряженных градиентов является лучшим из известных итерационных методов решения систем линейных уравнения с симметричной матрицей. На его основе разработано много высокоскоростных методов решения произвольных систем алгебраических уравнений. В то же время, традиционное объяснение принципов работы этого метода очень сложно. Как показывает практика, студенты плохо понимают его. В работе предложено интуитивно понятное объяснение принципов работы метода сопряженных градиентов.

***Ключевые слова*:** *системы линейных алгебраических уравнений, итерационные методы решения СЛАУ, метод сопряженных градиентов.*

Постановка задачи

Метод сопряженных градиентов является одним из самых быстросходящихся итерационных методов для решения систем линейных алгебраических уравнений с симметричной матрицей. Но, при всей простоте программной реализации метода, его математические основы достаточно сложны (см. например [1],[2],[3]). Очень часто при его пояснении используются такие понятия, как проекционные методы, ортогонализация по Арнольди и Ланцошу, скорейший спуск, пространства Крылова, сопряженные направления и многое другое. Все это делает метод непонятным на интуитивном уровне. По этой причине метод сопряженных градиентов очень непросто объяснять студентам. В работе приведено несколько отличающееся от традиционных обоснование метода. Здесь используются только понятие ортогональности, метод Грамма-Шмидта и понятие А-скалярного произведения. Понятие проекции используется только для улучшения уже построенного и понятного метода. Такой подход оказывается интуитивно понятным и гораздо более простым методически чем