DOI 10.36074/grail-of-science.10.11.2023.40

ВИКОРИСТАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ ІЗ ЗАСТОСУВАННЯМ ВИПАДКОВИХ ПРОЦЕСІВ ПРИ РОЗВ'ЯЗУВАННІ ЗАДАЧ ОПТИМІЗАЦІЇ

Бажан Станіслав Миколайович
здобувач ступеня доктор філософії
кафедри прикладної та вищої математики
Дніпровський державний технічний університет, Україна

Анотація. У дослідженні аналізуються різні способи кодування елементів у генетичному алгоритмі. Нарівні зі звичайними методами, такими як бінарне кодування та код Грея, в роботі вивчаються: гамма- та дельта-коди Еліаса, гамма-коди Левенштейна, кодування Голомба, коди Райса та інші. Для тестування вибраних підходів застосовувався адаптований генетичний алгоритм із можливістю зміни довжини рядків. Перевірка відбувалась на наборі тестових функцій. Запропоновано алгоритм векторного управління параметрами. Під час проведення дослідження застосовувалися: методи багатокритеріальної оптимізації, методи стохастичного оптимізації, метод ітерацій. Представлено пошук оптимальних розв'язків з урахуванням застосування модифікованого генетичного алгоритму за різних популяцій і поколінь. Показано ефективність використання в генетичному алгоритмі для вирішення задач оптимізації не тільки добре відомих методів бінарного кодування, а й альтернативних способів. Особливий інтерес застосування альтернативних способів кодування становлять для гібридних генетичних алгоритмів.

Ключові слова: генетичний алгоритм, бінарне кодування, задача оптимізації, хромосома, цільова функція, тестові функції, код Райса, «колесо рулетки»

Вступ. Людство на протязі своєї еволюції завжди розглядало завдання оптимізації: від пошуку найкоротших доріг до дизайну автомобільного кузова для мінімізації опору повітря. Теорія оптимізації має застосування у різних сферах – не лише в наукових дослідженнях або технічних інноваціях, але й у різноманітних галузях людської діяльності. Серед сучасних методів оптимізації виділяються генетичні алгоритми. Завдання оптимізації стають все більш актуальними в сучасній науці та техніці, завдяки чому було розроблено численні методи й алгоритми для пошуку оптимальних значень. Серед них можна виділити такі: пошук широти, метод дихотомії, градієнтний спуск, метод золотого перерізу та інші. Ці методи вважаються класичними та здатні розв'язувати багато завдань з високою точністю. Втім, у них є певні обмеження. Наприклад, при роботі з нелінійними функціями може виникнути складність з локальними екстремумами, де глобальний оптимум може бути пропущений на користь локального [14]. Також, із збільшенням кількості параметрів, які потрібно оптимізувати, росте обчислювальна вартість, що призводить до збільшення часу обчислень. Генетичні алгоритми (ГА) є частиною більш широкої категорії методів під назвою еволюційні обчислення. Еволюційні обчислення, у свою чергу, належать до групи наближених методів розв'язання, які часто класифікують як «м'які обчислення» [5].

В області генетичних алгоритмів виділяється видатна праця Корте Б., Виген Дж. [15], де розкрито основні принципи ГА та еволюційного моделювання, а також представлено ключові стратегії та концепції генетичного пошуку. У дослідженні планується надати стислий огляд основ генетичної оптимізації, доповнивши теорію актуальними прикладами застосування генетичних алгоритмів. Незважаючи на велику кількість науково-літературних джерел, що присвячені генетичним алгоритмам, досі існує потреба в детальному дослідженні теорії генетичних алгоритмів, враховуючи їхню важливість і сучасну актуальність при розв'язуванні задач оптимізації. [4]

Мета дослідження полягає в аналізі еволюції та сучасного стану генетичних алгоритмів. Особлива увага приділена оцінці ефективності ГА з бінарним кодуванням у пошуку оптимальних значень багатоекстремальних функцій з великою кількістю параметрів в контексті різних оптимізаційних завдань.

Завдання дослідження:

— аналіз та вивчення основних принципів роботи генетичних алгоритмів і методів їх застосування при розв'язанні оптимізаційних завдань;

— розробити алгоритм, який комбінує генетичний підхід із використанням випадкових процесів для покращення результатів оптимізації;

— провести серію експериментів для визначення ефективності запропонованого методу на різних типах задач оптимізації;

— зробити порівняльний аналіз отриманих результатів з традиційними підходами розв'язування оптимізаційних задач;

— висунути рекомендації щодо оптимальних умов застосування розробленого методу в практичних задачах оптимізації.

Методи дослідження. Генетичні алгоритми ґрунтуються на основах синтетичної теорії еволюції та впроваджують ключові принципи дарвінівської теорії еволюції, такі як спадковість, мінливість та природний відбір. У відмінності від багатьох традиційних алгоритмів оптимізації, які фокусуються на постійному вдосконаленні одного розв'язку, ГА працюють з популяцією розв'язків, представлених у вигляді хромосом, кожна з яких кодує потенційний варіант розв'язку. Ефективність кожної особини вимірюється за допомогою її «пристосованості», яка вказує на якість відповідного розв'язка задачі. Для цього використовується функція пристосованості (або цільова функція чи фітнес-функція). Вона допомагає виділити найбільш пристосовані елементи для їх подальшого використання, а менш ефективні виключаються з популяції, що пришвидшує знаходження оптимального розв'язку [16]. Основна ідея полягає в тому, щоб поступово підвищувати загальну пристосованість популяції, наближаючись до бажаного розв'язку.

Принцип ітерацій. Генетичний алгоритм працює за принципом ітерацій. На кожному етапі до існуючих особин застосовуються генетичні оператори, породжуючи нові. Потім всі особини проходять оцінку за цільовою функцією, і найбільш пристосовані з них вибираються для наступної ітерації. Цей цикл триває доти, поки не будуть отримані заздалегідь визначені результати або число ітерацій не досягне встановленого максимуму. Сигналом до завершення ітерацій може бути також стабільність популяції – коли її особини протягом декількох поколінь залишаються майже незмінними. Це зазвичай вказує на те, що розв'язок, близький до оптимального, вже знайдено. Зазвичай кінцевим розв'язком є найкращий елемент популяції з останнього покоління.

Виклад основного матеріалу. Генетичні оператори служать інструментом перетворення одного набору даних на інший. Ці оператори допомагають реалізовувати ідеї спадковості та варіативності в контексті віртуальних популяцій. Оскільки генетичні оператори працюють на основі ймовірності, це вносить елемент випадковості та гнучкості у функціонування ГА. Серед основних генетичних операторів варто відзначити оператор схрещування та мутації [2, с. 83].

Параметри, які підлягають оптимізації, кодуються у вигляді двійкових послідовностей визначеної довжини, яка забезпечує потрібний рівень точності. У цьому контексті вводиться поняття «хромосома», що представляє собою кожен елемент вектора параметрів який оптимізується:

$$X = (X_1, X_2, ..., X_n) \in \Omega_{add} \tag{1}$$

Ці хромосоми проходять процес еволюції протягом численних ітерацій, які також називаються поколіннями чи генераціями. Для кожного сформованого покоління хромосоми підвергаються оцінці за допомогою певного критерію, який відомий як функція пристосованості. Хромосоми з вищим показником отримують перевагу під час відбору. Заслуговує уваги метод відбору, який називається «колесом рулетки» [13]. За цим методом, відбір відбувається відповідно до функції розподілу, яка формується на базі розрахованих значень пристосованості хромосом. Механіка «колеса рулетки» наступна:

1. Розраховуємо функцію пристосованості $F_k(X_k)$ для кожної хромосоми X_k, $k \in (1 \dots N_{pop})$, N_{pop} – розмір популяції;
2. Визначаємо загальну функцію пристосованості для популяції:

$$f \sum_{K=1}^{N_{pop}} F_k(X_k) - \min_{j=1,Npop} (F_j(X_j)) \tag{2}$$

3. Розраховуємо імовірність відбору p_k для кожної хромосоми X_k та встановлюємо кумулятивну ймовірність для кожної з них:

$$p_k = \frac{F_k(X_k) - \min\limits_{j=1,N_{pop}} \left(F_j(X_j)\right)}{f}, k = 1 \dots N_{pop} \tag{3}$$

4. Вираховуємо сукупну ймовірність q_k для кожної хромосоми X_k.

$$q_k = \sum_{j=1}^{k} p_j, k = 1 \dots N_{pop.} \tag{4}$$

Процес відбору розпочинається із визначення значення вектора, який оптимізується N_{pop} разів. Під час цього кроку на кожній ітерації вибирається певна хромосома наступним способом: генеруємо випадкове число r_k, і якщо це число менше кумулятивної ймовірності q_k хромосоми X_k , то обираємо *k*-ту хромосому для наступного покоління.

У цьому дослідженні використовується метод одноточкового схрещування. Після відбору та схрещування до нової генерації застосовується мутація, яка допомагає «вивести» популяцію з локальних значень і забезпечує захист від занадто швидкої конвергенції. Деякий біт обраної особини може змінюватися з низькою ймовірністю ($P_{myt} \leq 1\%$) [3]. Сам алгоритм управління виробничими процесами, інтегрований із генетичним алгоритмом і має містити наступні етапи:

1. Створення початкової популяції. Ініціалізація популяції особинами, де компоненти масиву (біти) ініціалізуються випадковим чином в межах встановлених користувачем обмежень.
2. Налаштування параметрів генетичного алгоритму. Встановлення розміру популяції особин (N_{pop}); встановлення максимальної кількості поколінь (N_{pok}); Визначення ймовірності схрещування ($P_{skresch}$); визначення ймовірності мутації (P_{myt}), що вказує на кількість хромосом у популяції, які підлягають мутації.
3. Створення початкової популяції. Початкова популяція формується за допомогою випадкової ініціалізації.
4. Визначення пари для схрещування. Для вибору пари особин для схрещування використовується метод рулетки, або пропорційний відбір. Цей метод сприяє ефективному розподілу генетичної різноманітності в популяції та підвищує ймовірність вибору більш пристосованих особин для подальшого еволюційного вдосконалення популяції.
5. Процедура схрещування. Обираємо випадкову точку розриву $r \in [0,1]$, а потім обмінюємо відповідні ділянки між двома хромосомами. Цей метод називається одноточковим кросинговером.
6. Мутація особин. Застосовуємо метод мутації за допомогою випадкового вибору точки на хромосомі.
7. Перевірка критеріїв завершення еволюційного процесу. Якщо критерії завершення не досягнуті, повертаємося до кроку 4. Інакше завершуємо процес. Критеріями можуть бути: досягнення заданої кількості поколінь або отримання визначеного числа схожих особин.
8. Побудова Парето-оптимального варіанту. На основі конкретних вимог задачі управління формується генетичний алгоритм із визначеними параметрами, які встановлюються керівництвом.

Цей алгоритм інтегрує генетичний алгоритм для оптимізації виробничих процесів, дозволяючи знаходити оптимальні конфігурації і покращувати якість виробничих операцій.

У дослідженні представлена адаптація генетичного алгоритму з наступними характеристиками. Хромосомою слугує набір оптимізованих параметрів $X = (X_1, X_2, ..., X_n) \in \Omega_{add}$. При цьому кожен представник популяції повинен відповідати діапазону допустимих значень Ω_{add}. В рамках цього методу особину X_k, $K=1...N$, N_{pop} розглядають у контексті поколінь N_{pok} [7].

Функція адекватності для кожної хромосоми розраховується індивідуально і відображає значення цільової функції, обчислене через векторний аналіз залежно від визначених критеріїв. Відібрані гени (елементи особини, такі як $(X_{i1}, X_{i2}, ..., X_{in})$ для комбінування повинні бути унікальними, а їхні значення мають входити в рамки дозволених обмежень. Якщо умови порушуються, гени обираються знову. Для проведення мутації слід визначити специфічний ген, що підлягає змінам, та визначити його місце в популяції; також потрібно вказати конкретний біт у гені для зміни [12].

Дослідження генетичних алгоритмів вже отримало увагу у достатньо великій кількості наукових робіт, де зазвичай вчені аналізують такі параметри як ймовірність мутації, розмір популяції, методи селекції тощо. Елементи популяції зазвичай представляються у вигляді бінарних кодів, використовуючи або традиційне бінарне кодування або код Грея. Робота спрямована на вивчення альтернативних методів бінарного кодування дійсних чисел у контексті генетичних алгоритмів. При розробці такого алгоритму ключовим є вибір способу представлення елементів.

Класичний генетичний алгоритм за замовчуванням використовує бінарний вектор як основну хромосому. Хромосомна репрезентація залежить від методу кодування. Зазвичай у генетичних алгоритмах використовується двійкове представлення хромосом, де кожен біт відповідає певному ступеню числа «2». Наприклад, двійковий рядок [10011] представляє число «19», так як $1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 19.$ Цей метод кодування можна назвати стандартним двійковим кодом. Існує ряд інших підходів до двійкового кодування, зокрема:

Код Грея – це специфічне двійкове представлення цілих чисел, де два послідовних числа мають лише одну відмінну бінарну цифру, що гарантує в метриці Хеммінга відстань між їхніми кодами «1». Стандартне грей-кодування цілого числа ґрунтується на застосуванні оператора *XOR* до звичайного двійкового коду числа та цього ж коду, зсунутого на один біт вправо. При цьому, останній біт відкидається. Для численних завдань код Грея часто виявляється більш вдалим ніж стандартне двійкове кодування. Основна причина в тому, що код Грея зберігає наближену структуру сусідства у двійковому просторі. Таким чином, він не вносить нових оптимальних розв'язків, які відсутні в оригінальній цільовій функції. Додатково, через більшу кількість сусідніх точок у коді Грея порівняно зі звичайною цільовою функцією, кількість оптимальних розв'язків у просторі пошуку коду Грея зазвичай зменшується [8].

Досить рідко можна зустріти застосування генетичного алгоритму до бінарних кодів із змінюваною довжиною. Важливі цифри у такому коді розпочинаються від старших значущих позицій. Наприклад, у двійковому числі «0000011012», що дорівнює «13» (розглядаючи останні 4 цифри), головна цифра розташована на четвертій позиції від кінця. Подібно до звичайного запису в десятковій системі, порядок числа визначається місцем основної значущої цифри, не враховуючи переддекількох незначущих нулів. В даному випадку, порядок становить чотири. Методи цього класу є перетворювальними та адаптивними, що дозволяє їх використання незалежно від фіксованої довжини даних. Бінарний код представляє число *N* як послідовність із *N* нулів, доповнену однією одиницею на кінці.

Гамма-коди Еліаса базуються на діапазоні числа, що кодується. При кодуванні числа *n*, його гамма-код (γ-код) відображає стандартне двійкове представлення числа, з вилученням найстаршого біту, та доповнюється бінарним записом цього числа зліва. δ-коди Еліаса є модифікацією γ-кодів, де бінарна частина також представлена в γ-коді. Для формування δ-коду числа спочатку кодуємо *L* (довжина двійкового запису числа *n*) за допомогою γ-коду, а потім додаємо двійкове представлення числа, з виключенням найстаршого біту.

Основна різниця між γ-кодами та δ-кодами полягає в способі представлення експонент. У γ-кодах експоненти представлені у бінарному форматі, в той час як у δ-кодах експоненти додатково кодуються з використанням γ-кодування. При порівнянні перших чисел γ-коди зазвичай коротші за δ-коди. Це означає, що γ-коди виявляються ефективнішими для чисел, розподіл яких наближається до нуля.

Щодо Гамма-кодів Левенштейна, вони формуються шляхом інвертування послідовності бітів двійкового запису числа та додавання прапорцевого біта перед кожним бітом, окрім найостаннішого. Останній прапорцевий біт, який дорівнює «1», відповідає найстаршому біту первинного двійкового запису числа *n*.

Коди Голомба використовуються для представлення числа n за допомогою двох компонентів: бінарного коду *q*, який дорівнює результату цілочисельного ділення *n* на *m*, та залишку *r* після ділення *n* на *m*.

Якщо *m* є потенцією двійки, тобто може бути виражена у формі *2xb*, залишок *r* представляється як звичайний двійковий код, що має довжину *b* біт.

У випадку, коли r менший за величину, отриману за формулою *2xb-m*, код залишку формується з двійкового запису *r*, який займає *b-1* біт. У протилежному випадку, коли *r* більший або дорівнює цій величині, залишок кодується двійковим записом числа *r+2xb-m,* який займає *b* біт. Таким чином, коди Голомба дозволяють ефективно представляти числа в залежності від конкретних характеристик діапазону чисел та вибраного параметра *m*.

Коди Райсу дозволяють представляти число *n* за допомогою двох компонентів: бінарного коду числа q, отриманого в результаті цілочисельного ділення n на *2xk,* та двійкового запису залишку r після такого ділення. Залишок *r* кодується як звичайний двійковий код, що займає *k* біт.

Часто при роботі з генетичним алгоритмом, застосовують ту чи іншу модифікацію як наприклад [9,10]. Традиційний підхід працює з хромосомами фіксованої довжини, тоді як у цьому дослідженні була реалізована можливість використання хромосом зі змінною довжиною. Додатково, були інтегровані оператори схрещування, оптимізовані для таких змінних довжин.

Дослідження продуктивності генетичного алгоритму виконувалося на основі оптимізації таких тестових функцій: Розенброка, Шекеля, Растригіна, Катнікова, Гриванка та Де Йонга. Кількість параметрів варіювалася в межах від 2 до 4. У генетичному алгоритмі із стандартним бінарним та грей-кодуванням довжина кожного параметру встановлювалася як 10 біт. А у генетичному алгоритмі, що використовував коди Еліаса (гамма- та дельта-коди), коди Левенштейна, а також коди Голомба та Райса, довжина кожного параметра коливалася від 1 до 15 біт [11]. Аналіз проводився на основі 100 ітерацій алгоритму. Розмір популяції складав 50 особин, та було 50 поколінь. Для вимірювання ефективності методик була застосована метрика надійності.

Надійність вимірюється як відсоток успішних виконань алгоритму з урахуванням загальної кількості його запусків. Отримані результати представлені в таблиці 1. З результатів видно, що найкращу продуктивність демонструє алгоритм, який використовує кодування розв'язків за методом Райса.

Таблиця 1

Результати тестування генетичного алгоритму на основі оптимізації тестових функцій

Функція Растригіна						
Спосіб кодування	Прямий бінарний код	Код Грея	Гамма-коди Еліаса	Дельта-коди Еліаса	Гамма-коди Левенштейна	Коди Райсу
Надійність	25	67	50	17	17	84
Функція Griванка						
Спосіб кодування	Прямий бінарний код	Код Грея	Гамма-коди Еліаса	Дельта-коди Еліаса	Гамма-коди Левенштейна	Коди Райсу
Надійність	25	75	50	17	17	84
Функція Катнікова						
Спосіб кодування	Прямий бінарний код	Код Грея	Гамма-коди Еліаса	Дельта-коди Еліаса	Гамма-коди Левенштейна	Коди Райсу
Надійність	75	84	67	17	84	100
Функція Шекеля						
Спосіб кодування	Прямий бінарний код	Код Грея	Гамма-коди Еліаса	Дельта-коди Еліаса	Гамма-коди Левенштейна	Коди Райсу
Надійність	50	75	67	20	84	100
Функція Де Йонга						
Спосіб кодування	Прямий бінарний код	Код Грея	Гамма-коди Еліаса	Дельта-коди Еліаса	Гамма-коди Левенштейна	Коди Райсу
Надійність	75	75	100	100	67	100
Функція Розенброку						
Спосіб кодування	Прямий бінарний код	Код Грея	Гамма-коди Еліаса	Дельта-коди Еліаса	Гамма-коди Левенштейна	Коди Райсу
Надійність	59	84	50	34	50	84

Генетичний алгоритм із стандартним бінарним кодуванням найбільш ефективний для простих завдань, де допустима область характеризується сприятливими для оптимізації характеристиками (велика, опукла). Однак, на основі отриманих даних, можна зробити висновок, що для більш складних оптимізаційних завдань кодування розв'язків методом Райса демонструє кращу ефективність [19]. Для визначення статистичної важливості отриманих даних було використано критерій *ANOVA*. Цей критерій є варіантом непараметричної оцінки Манна-Уітні за рівнем значущості α = 0,05. Кожний алгоритм було попарно порівняно з іншими на основі даних з таблиці 1. Завдяки методу *ANOVA* було встановлено, що різниця в результативності між алгоритмами є статистично вагомою [19].

Отже, було продемонстровано важливість застосування в генетичних алгоритмах, спрямованих на розв'язання оптимізаційних задач, не лише знайомих методів бінарного кодування (як прямий та код Грея), а й новітніх підходів. Альтернативні методи кодування виявляються особливо привабливими для гібридних генетичних алгоритмів, комбінованих із локальною оптимізацією.

Висновки. Визначено, що на сьогоднішній день генетичні алгоритми втілюються в практику в численних наукових областях. Вони активно використовуються для вирішення проблем, таких як оптимізація функцій [9], задачі теорії Графів (такі як маршрут Комівояжера, розфарбування графа), налаштування штучних нейронних мереж, задачі компонування, формування розкладів [1], апроксимація функцій та певні завдання в біоінформатиці. Важливо розуміти, що ГА не завжди підходять для знаходження абсолютного глобального оптимуму. Вони не забезпечують гарантії виявлення ідеального роз'вязку у визначений проміжок часу. Однак це не означає, що ГА не корисні. Вони дійсно ефективні для пошуку розв'язка, яке близьке до оптимального, в короткий часовий інтервал.

Було досліджено ключові принципи роботи генетичних пошукових алгоритмів. Основну увагу приділено важливим генетичним операторам, стратегіям та моделям, що використовуються в рамках ГА. Акцент зроблено на вдалому досвіді використання ГА для розв'язання оптимізаційних завдань, що мають багато екстремумів. Специфіка цього дослідження полягає у всебічному аналізі ефективності генетичних алгоритмів у контексті багатопараметричних оптимізаційних проблем. Основна увага приділена вивченню операторів, методик та стратегій, які використовуються в рамках ГА. Окрім того, підкреслено, що традиційний генетичний алгоритм із бінарним кодуванням може бути ефективно адаптований для роботи на графічних процесорах, сприяючи високій продуктивності паралельних обчислень.

Список використаних джерел:

[1] Бажан С., Олійник Л. (2022). Про алгоритм пошуку оптимального плану для транспортної задачі спеціального вигляду. Міжнародний науковий журнал «GRAIL OF SCIENCE» № 11. (https://doi.org/10.36074/grail-of-science.24.12.2021.039) : за матеріалами II Міжнародної науково-практичної конференції «Modern science: concepts, theories and methods of basic and applied research» (Index Copernicus)

[2] Безклубенко І.С., Гетун Г.В., Баліна О.І., Буценко Ю.П. (2022) Дослідження властивостей множини ефективних значень критеріїв в задачі оптимізації інженерної мережі. Управління розвитком складних систем. № 51, С. 81-86.

[3] Горда О. В., Цюцюра С. В., Лященко Т. О. (2022) Cognitive elements of information environments. *Управління розвитком складних систем*. № 51. С. 49–57, dx.doi.org\10.32347/2412-9933.2022.51.49-57

[4] Гулаєва, Н. М., Шило, В. П., М.М. Глибовець (2021). Генетичні алгоритми як обчислювальні методи скінченновимірної оптимізації. Кібернетика та комп'ютерні технології.

[5] Гуляницький Л. Ф., Мулеса О. Ю. (2016) Прикладні методи комбінаторної оптимізації. Київ: Київський національний університет імені Т. Г. Шевченка, 133 с.

[6] Димова Г. О. (2022) Розробка моделі складання розкладу занять методом еволюційного пошуку. Таврійський науковий вісник. Серія: Технічні науки. № (2). С. 3-9. https://doi.org/10.32851/tnv-tech.2022.2.1

[7] Канаєв Є. Ю., Арсенюк І. Р., Месюра В. І. (2021) Обґрунтування доцільності застосування генетичного алгоритму для задачі проходження лабіринтів. Тези доповідей L науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії. Вінниця: ВНТУ.

[8] Македон В.В., Чабаненко А.В.(2022) Факторні складові цифровізації глобальної економіки та макроекономічних систем країн світу. Ефективна економіка. № 1.DOI: 10.32702/2307-2105-2022.1.11

[9] Олійник Л.О., Бажан С.М. (2019) «Застосування алгебраїчної модифікації генетичного алгоритму в задачах визначення глобального екстремуму функцій однієї змінної», 52-53с., матеріали Міжнародної наукової конференції «Математичні проблеми технічної механіки та прикладної математики-2019», Дніпро, Кам'янське, Україна.

[10] Олійник Д., Олійник Л. (2022) Про ефективність операторної модифікації генетичного алгоритму в задачах двовимірної оптимізації. Грааль науки. № (11). С. 221-229. https://doi.org/10.36074/grail-of-science.24.12.2021.038

[11] Abu-Arqub Omar, Abo-Hammour Zaer, Momani Shaher (2014) Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems. Applied Mathematics and Information Sciences. Vol. 8, N 1. P. 235- 248.

[12] Gen M. Cheng R. (1997) Genetic Algorithms and Engineering Design. John Wiley & Sons, New York,. 352 p.

[13] Gorbatyuk I., Terentyev O., Sviderskyi A., Dolya E. (2022) Purpose and requirements for deep disintegration of local action. In collective monograph: Theoretical and scientific foundations in research in Engineering. International Science Group. Boston: Primedia eLaunch. 2022. Pp. 437-443. Available at: DOI-10.46299/ISG.2022.MONO.TECH.1. ISBN 978-1-68564-501-4.

[14] Herrera F., Lozano M., Verdegay J.L. (1998) Tackling real-coded genetic algorithms: operators and tools for the behaviour analysis Artificial Intelligence Review. Vol. 12, N 4. P. 265-319.

[15] Korte B., Vygen J. (2018) Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics), Springer. New York, 455 p.

[16] Makedon V., Mykhailenko O., Dzyad O. (2023) Modification of Value Management of International Corporate Structures in the Digital Economy. European Journal of Management Issues. №31(1). pp. 50-62. https://doi.org/10.15421/192305

[17] Whitley L. D. (1999) A Free Lunch Proof for Gray versus Binary Encoding. Proc. Genetic and Evolutionary Computation Conference, 194 p.

