

О. В. БОНДАРЕНКО, О. В. УСТИНЕНКО, Р. В. ПРОТАСОВ, О. В. АРХІПОВ

ОПЕРАТОРИ СХРЕЩУВАННЯ ТА МУТАЦІЇ В СТОХАСТИЧНИХ АЛГОРИТМАХ

Описано актуальність освітлення сучасних стохастичних алгоритмів, вказано, що у останні два десятиліття спостерігається стрімкий розвиток стохастичних алгоритмів, який пояснюється збільшенням дослідницьких можливостей та зростанням обсягу даних, які обробляються. Ці алгоритми виявляються ефективними у розв'язанні складних оптимізаційних задач, що привертає увагу до них наукової спільноти та практиків у всьому світі. Освітлення ролі та огляд ключових операторів схрещування та мутації у стохастичних алгоритмах є актуальною науково-практичною задачею, яка сприятиме глибшому розумінню та популяризації цього напрямку. Проведено аналіз категорії геному та надано детальний огляд основних способів кодування генів для їх застосування. На конкретному прикладі геному, що відповідає задачі оптимального проєктування двоступінчастого співвісного редуктора, розглянуті типові та адаптовані оператори схрещування та мутації, що можуть бути використані для ефективного пошуку розв'язків. До кожного оператора надано текстовий опис та графічно-ілюстроване зображення, що дає змогу чіткіше, якісніше та швидше зрозуміти сутність та послідовності роботи оператора. Розглянуто роль та значення операторів схрещування та мутації в стохастичних алгоритмах. Підкреслюється, що оператори схрещування сприяють комбінації корисних генетичних властивостей, що покращує пристосованість нащадків. Збалансоване використання цих операторів разом з іншими етапами алгоритму, такими як мутація та відбір, важливе для досягнення оптимального балансу між експлуатацією алгоритму та інтенсифікацією пошуку. Вказано на важливість та функціональність операторів мутації в стохастичних алгоритмах оптимізації. Зазначено, що мутація дає змогу уникнути «застрягання» алгоритмів у локальних екстремумах, зберігаючи генетичне різноманіття та стимулюючи пошук нових оптимальних розв'язків. Особливу вагу має використання мутацій в умовах складної структури задачі або великого простору пошуку. Таким чином, оператори схрещування та мутації є ключовими елементами для підвищення ефективності пошуку оптимальних розв'язків.

Ключові слова: стохастичний алгоритм, оператор, схрещування, мутація.

O. BONDARENKO, O. USTYNYENKO, R. PROTASOV, O. ARKHIPOV

CROSSOVER AND MUTATION OPERATORS IN STOCHASTIC ALGORITHMS

The relevance of illuminating contemporary stochastic algorithms is highlighted, with the past two decades witnessing a rapid development in stochastic algorithms, attributed to increased research capabilities and growing data volumes. These algorithms prove effective in solving complex optimization problems, garnering attention from the global scientific community and practitioners worldwide. An exploration of the role and an overview of key crossover and mutation operators in stochastic algorithms represent a pertinent scientific and practical endeavor, fostering deeper understanding and popularization of this field. A genome category analysis is conducted, accompanied by a detailed review of primary gene encoding methods for practical application. Through a specific example of a genome corresponding to the optimal design of a two-stage helical gearbox, typical and adapted crossover and mutation operators are examined for efficient solution search. Each operator is provided with a textual description and a graphically illustrated representation, enhancing clarity, quality, and expediency in understanding the essence and operational sequences of the operator. The role and significance of crossover and mutation operators in stochastic algorithms are discussed, emphasizing that crossover operators facilitate the combination of beneficial genetic traits, enhancing offspring adaptability. Balanced utilization of these operators alongside other algorithmic stages, such as mutation and selection, is crucial for achieving an optimal balance between algorithm exploitation and search intensification. The importance and functionality of mutation operators in optimization stochastic algorithms are highlighted, indicating that mutation enables the avoidance of algorithmic stagnation in local extrema, preserving genetic diversity and stimulating the search for new optimal solutions. The particular significance of mutation usage in conditions of complex problem structures or large search spaces is noted. Thus, crossover and mutation operators are deemed key elements for enhancing the effectiveness of optimal solution search.

Keywords: stochastic algorithm, operator, crossover, mutation.

Вступ. Актуальність задачі. Протягом останніх двох десятиліть відбувся виражений вибух у розвитку стохастичних алгоритмів як методів раціонального проєктування технічних об'єктів та суттєво поширилося їх використання в інших галузях людської діяльності. І все це сталося на фоні відносно невеликих початкових зусиль, зроблених у середині минулого століття. На сьогоднішній день нараховується майже сотня широко відомих алгоритмів та їх модифікацій, а також декілька сотень інших, що знаходяться на етапі активного формування, розвитку та впровадження. Спостерігається інтенсивна активізація зацікавленістю та використанням стохастичних алгоритмів у країнах Східної та Західної Європи, США, країнах Близького Сходу та Китаї. Це пояснюється домінуючим застосуванням цих алгоритмів у науковому світі протягом вказаного періоду. З іншого боку, це стало результатом поступового розширення дослідницьких можливостей, які пропонують вказані алгоритми в різних галузях науки та практики. Спричинено це, зокрема, постійним збільшенням обсягу даних, що оброблюються, складністю задач та швидкого розвитку потужностей обчислювальної технології. Стохастичні алгоритми відрізняються від традиційних детермінованих методів своєю здатністю враховувати випадковість та ймовірнісний характер вхідних та поточних даних. Ця особливість

робить їх ефективними для розв'язання складних оптимізаційних задач у різноманітних їх варіаціях та передбачуваних зв'язках між параметрами. Наявність великої кількості варіантів стохастичних алгоритмів дає змогу науковій спільноті вибирати оптимальний підхід для конкретної задачі в залежності від її особливостей та вимог. Ця гнучкість робить їх популярними серед дослідників та практиків у різних областях, від інженерії та біоінформатики до фінансів та штучного інтелекту. Разом з тим, широке використання стохастичних алгоритмів вимагає від наукової спільноти постійного вдосконалення та стандартизації методів їхньої реалізації та оцінки ефективності.

Оператори схрещування та мутації є ключовими складовими стохастичних алгоритмів. Вони відіграють важливу роль у здатності алгоритмів до знаходження найкращих розв'язків у складних багатовимірних просторах пошуку. Оператори схрещування дають змогу комбінувати корисні характеристики батьківських особин, створюючи нові варіанти з потенційними перевагами. Цей процес сприяє ефективному пошуку оптимальних розв'язків шляхом поєднання корисних елементів різних батьківських особин. Мутація, з іншого боку, вносить випадкові зміни в інди-

відуальні хромосоми, що сприяє дослідженню нових варіантів і можливостей. Цей процес відіграє важливу роль у збереженні різноманітності та уникненні передчасної збіжності до локально оптимальних розв'язків. Використання операторів схрещування та мутації є важливим етапом у процесі реалізації еволюційних стохастичних алгоритмів, що дає змогу підтримувати високий рівень різноманітності у популяції та постійно вдосконалювати алгоритм для досягнення кращих результатів. Разом вказані оператори утворюють потужний інструмент для пошуку оптимальних рішень у різних сферах застосування.

Освітлення основних сучасних операторів схрещування та мутації, що використовуються в стохастичних алгоритмах, дасть змогу глибше зрозуміти сутність питання та провести актуалізацію цього наукового напрямку, а також глибше зрозуміти можливості використання та акцентувати їх ролі у ході формування пошукових популяцій, а також провести їх популяризацію.

Таким чином, освітлення основних сучасних операторів схрещування та мутації, що використовуються в стохастичних алгоритмах, є актуальною науково-практичною задачею.

Основна частина. Стохастичні алгоритми.

Щоб зрозуміти, що саме є стохастичним алгоритмом, потрібно розглянути ряд визначень деяких категорій та взаємозв'язки між ними.

Алгоритм – чіткий опис послідовності дій, які необхідно виконати при розв'язуванні задачі.

Загалом алгоритми поділяються на детерміновані та недетерміновані. *Детермінований* алгоритм передбачає єдиний шлях від вхідних даних до виходу. *Недетермінований* алгоритм відрізняється можливістю отримання результату декількома різними шляхами, тобто деякі шляхи виконання алгоритму можуть привести до однакового результату, а деякі – до результатів, що суттєво відрізняються від інших.

Саме недетерміновані алгоритми цікавлять нас у межах питання, що досліджується. Також доцільно вказати, що недетерміновані алгоритми розділяються на наступні: *апроксимаційні*, *ймовірнісні* та *стохастичні*.

Стохастичні алгоритми оптимізації [1–7] використовують випадковість у процесі пошуку оптимуму. Зазвичай їх застосовують, коли цільова функція складна, багатоекстремальна, з розривами та перешкодами, а простір пошуку відносно великий.

Стохастичні алгоритми, у свою чергу, умовно можна поділити на *евристичні* та *метаевристичні*.

Класифікаційні зв'язки та положення в них стохастичних алгоритмів зображено на рис. 1.

Геном. У відповідності до наведеної класифікації, кістяком стохастичних алгоритмів є еволюційні алгоритми, еволюційне програмування, еволюційні стратегії, генетичні алгоритми та строката родина метаевристичних алгоритмів. Всі вони використовують у тому чи іншому варіанті певні стратегії пошуку, що базуються на дослідженні простору параметрів задачі оптимізації. Основною базовою категорією при цьому є *геном*.

Геном – набір певних координат, що характеризує пробні точки багатовимірного простору параметрів задачі оптимізації. Кожен з параметрів проектування є геном. У межах інженерії та процесів проек-

тування у якості генома може виступати набір різноманітних параметрів, наприклад, коефіцієнти, геометричні параметри, константи тощо.

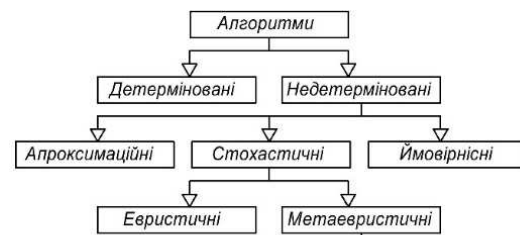


Рисунок 1 – Зв'язки та положення стохастичних алгоритмів у класифікаційній структурі

Слід зазначити, що існує широкий спектр способів кодування генів [8–12], які можуть бути використані в стохастичних алгоритмах для розв'язання різних задач оптимізації та пошуку розв'язків. Надаємо перелік основних способів кодування генів в стохастичних алгоритмах:

1. Бінарне кодування – гени представлені рядком бітів, де кожен біт відповідає певному гену.

2. Цільове кодування – гени кодуються у вигляді цілочисельних значень, які відповідають певним можливостям або варіантам.

3. Порядкове кодування – гени кодуються як порядкові числа, які вказують на позицію елемента у впорядкованому списку або послідовності.

4. Дійсні числа – гени кодуються як дійсні числа, які відповідають певним параметрам чи характеристикам.

5. Перестановка – гени представлені у вигляді перестановки, де кожен ген вказує на позицію об'єкту в послідовності.

6. Матричне кодування – гени представлені у вигляді матриці або двовимірного масиву, де кожен елемент відповідає певному гену.

7. Набір символів – гени представлені у вигляді символів, де кожен символ відповідає певному гену або параметру.

8. Функційне кодування – гени представлені у вигляді функцій чи програмного коду, що визначають поведінку або структуру об'єкта.

9. Мультимодальне кодування – гени представлені у вигляді комбінації різних типів кодування для кожного гена, що дозволяє використовувати різні характеристики об'єктів.

10. Фрагментарне кодування – гени представлені у вигляді фрагментів або частин об'єкта, які можуть бути комбіновані для створення нових рішень.

11. Комбіноване кодування – гени представлені у вигляді комбінації різних методів кодування, що дозволяє використовувати переваги кожного з них для конкретної задачі.

12. Символьне кодування – гени представлені у вигляді символів або об'єктів з визначеними властивостями, які можуть бути модифіковані у процесі оптимізації.

13. Часове кодування – гени представлені у вигляді послідовностей часових інтервалів або інтервалів активності.

Слід зауважити, що значна кількість задач інженерної оптимізації при розв'язанні стохастичними ал-

горитмами мають геном, що має бінарне або дійсне кодування. Але все ж таки переважаючим є дійсне кодування. Деякі оператори схрещування та мутації, що були розроблені для бінарного кодування, можуть бути адаптовані та результативно використані для дійсного кодування. Це дасть змогу значно збільшити кількість номінальних операторів схрещування та мутації, що можуть бути використані для дійсного кодування.

У даній роботі автори пропонують розглянути типові та найбільш розповсюджені, а також адаптовані для дійсного кодування оператори схрещування та мутації на прикладі реального генома, що відповідає задачі оптимального проектування двоступінчастого співвісного редуктора у вигляді, наданому на рис. 2. Такий геном має всі можливі характеристики та обмеження, що можуть бути до нього застосовані, тобто є гарним об'єктом для демонстрації.

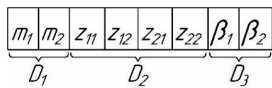


Рисунок 2 – Приклад генома пробної точки двоступінчастого співвісного редуктора

На рис. 2 прийняті наступні позначення: m_{μ} – відповідні модулі пар зубчастих коліс ($\mu = 1, 2$); $z_{\mu,k}$ – відповідні числа зубців коліс, k – номер колеса у зачепленні ($k = 1$ – ведуче колесо, $k = 2$ – ведене колесо); β_{μ} – кути нахилу зубців у зачепленнях.

Як бачимо з рис. 2, структура генома складається з 3-х ділянок D_1, D_2, D_3 , у локусах яких згруповано споріднені гени, тобто конструктивні параметри зубчастого привода, що мають однаковий змістовний сенс. Ділянки D_1 відповідають k модулів зубчастих зачеплень, ділянки D_2 відповідають r чисел зубців зубчастих коліс, ділянки D_3 відповідають t кутів нахилу зубців у зубчастих зачепленнях, які є серед параметрів проектування при формулюванні постановки задачі. Тобто, особина (пробна точка) в даній задачі має гени, які є дійсними значеннями.

Схрещування та мутації. Розглянемо на прикладі наведеного на рис. 2 генома широко вживані оператори схрещування та мутації, які можна використати при розв'язанні вказаної задачі.

Первинно розглянемо оператори схрещування. При одноточковому схрещенні [13] обирається будь-яким способом точка схрещування, і після цього проводиться обмін фрагментами хромосом двох батьків (Б1 та Б2), з формуванням двох нащадків (Н1 та Н2), як показано на рис. 3.

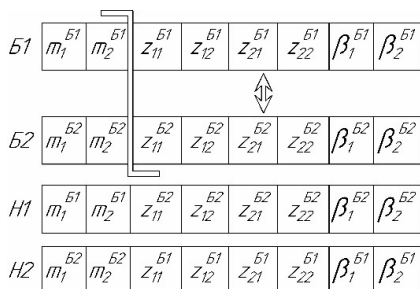


Рисунок 3 – Ілюстрація одноточкового схрещування

При багатоточковому схрещуванні [14] обирають дві та більше точок, потім батьки обмінюються відповідними фрагментами, як показано на рис. 4.

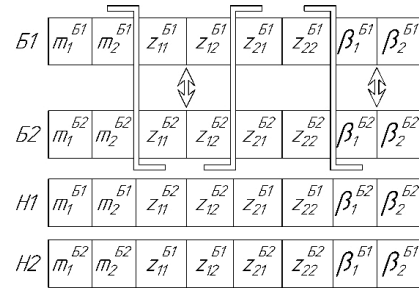


Рисунок 4 – Ілюстрація багатоточкового схрещування

Зважаючи на специфіку описаних операторів схрещування, можна рекомендувати метод багатоточкового схрещування з можливістю примусового обрання точок схрещування. В залежності від обмежень на змінні проектування, це дасть змогу проектувальнику вправно оперувати ділянками хромосоми, при необхідності залишаючи їх цілими або розривати їх у певних місцях.

При однорідному дискретному схрещуванні [15] геном нащадка створюється шляхом копіювання відповідного гена від першого або другого з батьків. Для цього будь-яким способом, зазвичай випадковим, генерується маска нащадка, довжина якої відповідає довжині хромосоми батьків. Наповнення маски складається з цифр 1 та 2, які відповідають номеру одного з двох батьків, ген яких входить до геному нащадка, як показано на рис. 5.

Б1	$m_1^{Б1}$	$m_2^{Б1}$	$z_{11}^{Б1}$	$z_{12}^{Б1}$	$z_{21}^{Б1}$	$z_{22}^{Б1}$	$\beta_1^{Б1}$	$\beta_2^{Б1}$
Б2	$m_1^{Б2}$	$m_2^{Б2}$	$z_{11}^{Б2}$	$z_{12}^{Б2}$	$z_{21}^{Б2}$	$z_{22}^{Б2}$	$\beta_1^{Б2}$	$\beta_2^{Б2}$
маска Н1	1	1	2	1	2	1	1	2
маска Н2	1	2	1	1	1	2	1	2
Н1	$m_1^{Б1}$	$m_2^{Б1}$	$z_{11}^{Б2}$	$z_{12}^{Б1}$	$z_{21}^{Б2}$	$z_{22}^{Б1}$	$\beta_1^{Б1}$	$\beta_2^{Б2}$
Н2	$m_1^{Б2}$	$m_2^{Б2}$	$z_{11}^{Б1}$	$z_{12}^{Б2}$	$z_{21}^{Б1}$	$z_{22}^{Б2}$	$\beta_1^{Б2}$	$\beta_2^{Б1}$

Рисунок 5 – Ілюстрація однорідного дискретного схрещування

При проміжному схрещуванні [16] значення гену нащадка визначається за залежністю $A' = A + \alpha(B - A)$, де A' – значення гену нащадка відповідно до значень генів батьків (A, B), а $\alpha \in [0, 1]$ – випадковий множник. Таким чином, хромосома одного (або двох) нащадка формується за окремими компонентами генома, як показано на рис. 6.

Схрещування перемішуванням [17] використовує техніку поєднання одноточкового схрещування з випадковим перемішуванням генів. Спочатку вибираються двоє батьків для схрещування. Далі застосовується техніка одноточкового схрещування, випадково або ні вибираючи точку для схрещування, з створенням двох прото-нащадків. Після виконання одноточкового схрещування гени в прото-нащадках перемішуються випадковим чином, як показано на

рис. 7. Зазначимо, що для випадку, коли генотип розбитий на логічні ділянки, змішування можна проводити лише у межах ділянок, а одноточкове схрещування проводити на межах ділянок.

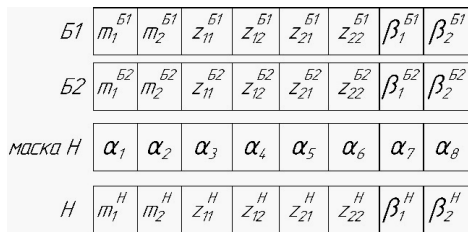


Рисунок 6 – Ілюстрація проміжного схрещування

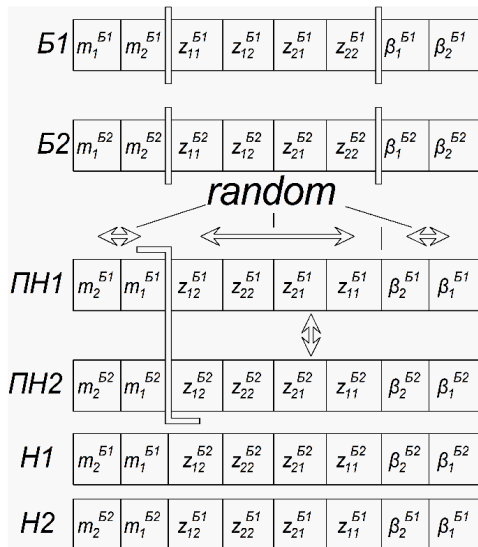


Рисунок 7 – Ілюстрація схрещування перемішуванням

Дискретне схрещування [17] використовує випадкове дійсне число для створення одного нащадка з двох батьків. На відміну від однорідного схрещування, при дискретному схрещуванні генерується лише один нащадковий елемент таким чином, який випадково вибирає ген з двох генів обох батьків у відповідних локусах. Ілюстрацію даного схрещування показано на рис. 8.

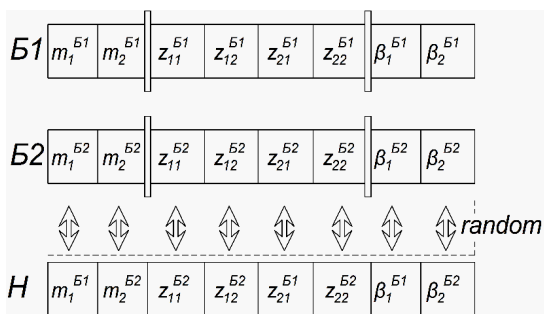


Рисунок 8 – Ілюстрація дискретного схрещування

Середнє схрещування – це техніка схрещування, що базується на значеннях генів батьків. Воно створює одного нащадка від середнього значення геномів двох батьків. Підхід вибирає двох батьків і генерує нащадковий елемент наступним чином: кожен ген нащадка береться шляхом усереднення генів обох батьків, як показано на рис. 9.

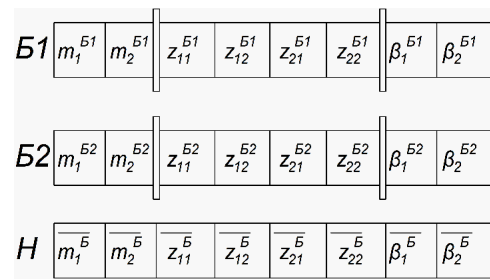


Рисунок 9 – Ілюстрація середнього схрещування

Плоске схрещування [18] – підхід для створення нових індивідів (нащадків) шляхом комбінування генетичної інформації двох батьків. У випадку плоского схрещування процес відбувається на основі випадкового дійсного числа та лінійних комбінацій генів батьків. Спочатку обираються двоє батьків з популяції. Потім вибирається вектор випадкових дійсних чисел r_i , де кожне r_i є значенням з одиничного інтервалу. Після цього створюються нащадки. Для кожного гена x'_i нащадка здійснюється обчислення його значення за допомогою лінійної комбінації генів батьків та випадкового дійсного числа: $x'_i = r_i x_{1,i} + (1 - r_i)x_{2,i}$.

Аналогічно обчислюється другий нащадок за допомогою відповідних батьківських генів та випадкового дійсного числа.

Таким чином, плоский кросовер дозволяє комбінувати генетичну інформацію батьків, зберігаючи лінійну структуру генів та використовуючи випадкові числа для регулювання співвідношення між генами батьків у нащадках. Ілюстрацію даного схрещування показано на рис. 10.

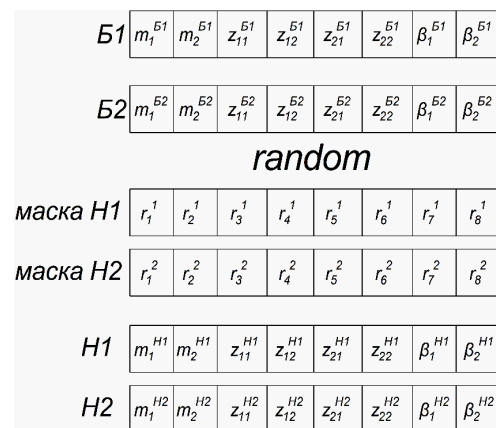


Рисунок 10 – Ілюстрація плоского схрещування

Поважливо-випадкове схрещування [16, 18] – підхід, що вибирає двох батьків для схрещування, а нащадки генеруються на основі вектора схожості батьків. Спочатку створюється вектор подібності таким чином, якщо обидва гени батьків мають однакові значення, тоді вектор подібності містить значення батьківського гену, в іншому випадку він містить нульове значення для цього гена.

Після створення вектора подібності створюються два нащадкових елемента відповідно до його значень. Якщо вектор подібності містить певне значення, тоді ген обох нащадків приймає це значення,

а якщо він містить нульове значення для цього гена, тоді нащадковий ген вибирається за допомогою рівномірного випадкового дійсного числа. Цей алгоритм дублює гени батьків у нащадків в кожній позиції, де вони ідентичні, як показано на рис. 11.

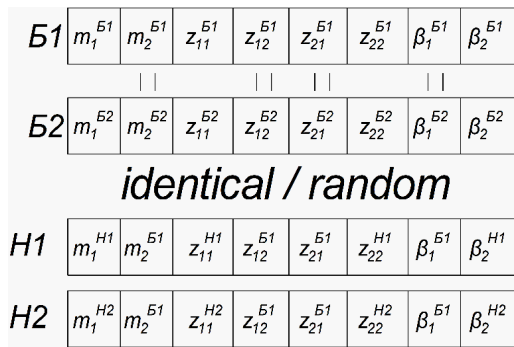


Рисунок 11 – Ілюстрація поважливо-випадкового схрещування

Оператори схрещування в стохастичних алгоритмах відіграють визначальну роль у процесі пошуку оптимальних рішень. Основною метою цих операторів є генерація нових індивідів на основі генетичного матеріалу батьківських особин. Це дає змогу алгоритму використовувати елементи успадкованості та рекомбінації, схожі на природний відбір, для досягнення кращих розв'язків у просторі пошуку.

Оператори схрещування можуть приймати різні форми, але їх загальна мета залишається однаковою: покращення якості рішень та забезпечення більшої різноманітності в популяції.

У генетичних, еволюційних та метаевристичних алгоритмах, які моделюють природний відбір та процеси, оператори схрещування відіграють ключову роль у виборі батьківських хромосом та обміні генетичним матеріалом між ними. Цей обмін сприяє комбінуванню корисних генетичних властивостей та покращенню пристосованості потомства. Такий механізм дає змогу алгоритму перейти від менш ефективних рішень до оптимальних шляхом об'єднання корисних аспектів генетичного матеріалу.

У еволюційних стратегіях оператори схрещування використовуються для еволюції стратегій агентів у просторі оптимізації. Тут, вони можуть впливати на параметри стратегій, такі як інтервали зміни або комбінації параметрів. Це відкриває можливості для ефективного дослідження простору параметрів та пошуку оптимальних стратегій у задачах оптимізації.

Ефективність операторів схрещування визначається їхнім внеском у збільшення різноманітності в популяції. Чим більше різноманітність, тим більше можливостей для алгоритму виявити нові та перспективні розв'язки. В той час як мутації можуть вводити нові елементи у популяцію, схрещування сприяє створенню комбінацій існуючих елементів, що дає змогу алгоритму ефективно пересуватися по простору пошуку.

У великому контексті, використання операторів схрещування в генетичних та еволюційних алгоритмах визначається їхньою спроможністю ефективно

поєднувати генетичний матеріал для створення нових, більш пристосованих індивідів, що веде до швидшого та ефективнішого пошуку оптимальних рішень.

Важливо збалансувати використання операторів схрещування з іншими етапами алгоритму, такими як мутація та відбір. Збереження різноманітності в популяції повинно поєднуватися з рекомбінацією, щоб забезпечити оптимальний баланс між експлуатацією алгоритму та інтенсифікацією пошуку.

Розглянемо далі більш живі оператори мутації. Зазвичай мутацію використовують для того, щоб відвести процес пошуку від локальних сплесків цільової функції. Також мутація дає можливість збільшити кількість популяції. Для задачі, що розглядається, пропонується використати наступні оператори.

Широко вживають оператор **інверсію** [19], принцип якого показано на рис. 12. В даному випадку змінюють місця сусідні гени. Але на відміну від класичного підходу, де точки розриву обираються випадково, пропонуємо використовувати варіант оперування лише у межах певних ділянок, щоб не сплутати смислове наповнення локусів хромосоми.

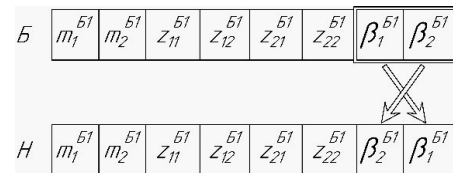


Рисунок 12 – Ілюстрація мутації інверсії

Інший варіант мутації – **крокова** [20, 21], принцип проілюстровано на рис. 13. У даному випадку значення випадкового гену випадково змінюється (зменшується або збільшується) на деяке значення – крок (δ), яке може бути як постійним, так і змінним, що може залежати від порядкового номеру покоління: $A' = A \pm \delta$.

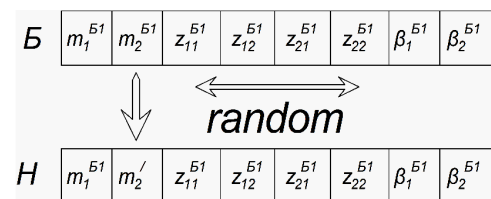


Рисунок 13 – Ілюстрація крокової мутації

Проста мутація [22, 23] використовує принцип внутрігенної мутації. При цьому в хромосомі випадковим чином обирається ген, а потім відбувається його випадкова зміна на заданому інтервалі допустимих значень, що не дорівнює поточному, як показано на рис. 14.

Відомий також модифікований оператор простої мутації, що називається **точковою мутацією**, який відрізняється тим, що в хромосомі мутує не один, а кілька генів із заданою ймовірністю.

Мутація інвертування (переворот) [19] може проходити в кількох варіантах реалізації. Наприклад, з деякою ймовірністю хромосома піддається мутації, і далі випадковий ген переводиться в інверсний стан, тобто приймає дзеркальне відносно середнього значення. Або спочатку вибираємо з деякою ймовірністю хромосому для операції мутації, а потім прохо-

димо по всіх генах і з деякою іншою ймовірністю виконуємо інвертування. Ілюстрація процесу мутації інвертуванням показана на рис. 15.

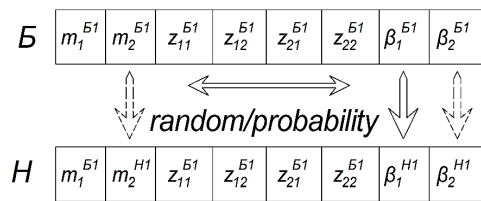


Рисунок 14 – Ілюстрація простої та точкової мутації

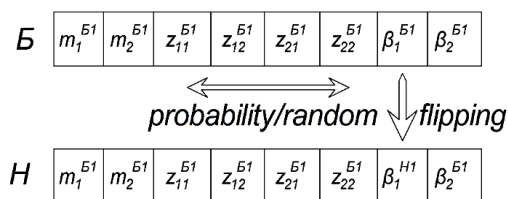


Рисунок 15 – Ілюстрація мутації інвертування (переворот)

Мутація обміну [24] є однією з ключових операцій у генетичних та еволюційних алгоритмах, що спрямована на зміну генетичного коду індивіда. У цьому процесі вибираються дві випадкові алелі на хромосомі, і їхні значення обмінюються. Це призводить до створення нового варіанту генетичного матеріалу індивіда, як показано на рис. 16. Для цього оператора у межах постановки задачі також пропонується використовувати варіант обміну лише у межах певних ділянок, щоб не сплутати смислове наповнення локусів хромосоми.

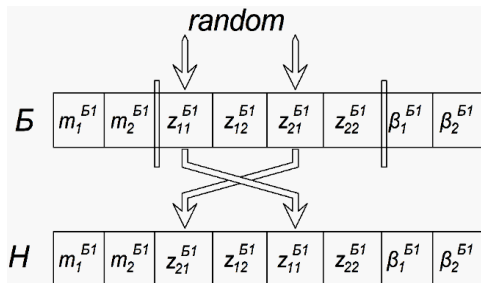


Рисунок 16 – Ілюстрація мутації обміну

Наступний варіант мутації – **скрамбл** [25]. При проведенні скрамбл мутації важливо визначити підмножину генів, яку буде змінено. Це робиться шляхом випадкового вибору сегменту ділянки (або всієї ділянки) з усього генетичного коду. Після вибору сегменту гени переміщуються випадковим чином, що призводить до створення нового варіанту хромосоми, як показано на рис. 17. Цей процес сприяє створенню різноманітності в генетичній популяції та розвиває потенціал виявлення оптимальних комбінацій генетичного матеріалу у нових поколіннях.

Мутація **віддзеркаленням** [26] є іншим методом, що може внести випадкові зміни в генетичний код індивіда в рамках стохастичного процесу. При її проведенні випадковим чином вибирається безперервна послідовність генів на хромосомі, знов, за наявності ділянок, необхідно притримуватися їх меж та не виходити за них. Далі обрані гени записуються в зворотному порядку, створюючи новий варіант генетичного матеріалу індивіда. Ілюстрація процесу мутації віддзеркаленням показана на рис. 18. Такі випадкові модифікації дають змогу алгоритму ефективно досліджувати простір пошуку для знаходження оптимальних рішень.

Мутація **переміщенням** [27] є також одним з цікавих методів генерування нової особини. Первинно вибирається випадковим чином ген або підмножина генів. Також випадковим чином обирається точка розриву генома, у яку буде переміщена обрана підмножина. Далі безпосередньо здійснюється операція переміщення обраної підмножини в точку розриву. Після чого гени зсуваються до початкової кількості локусів. Ілюстрація оператора процесу мутації переміщенням показана на рис. 19.

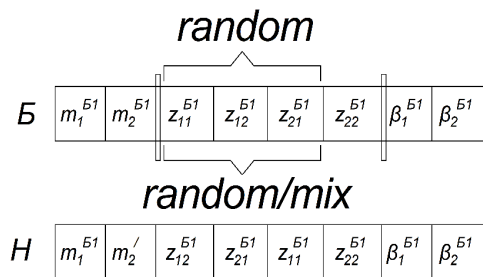


Рисунок 17 – Ілюстрація мутації скрамбл

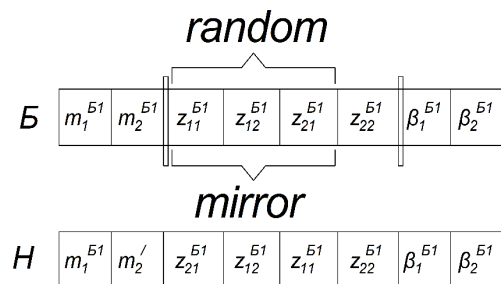


Рисунок 18 – Ілюстрація мутації віддзеркаленням

Мутація **переміщенням** [27] є також одним з цікавих методів генерування нової особини. Первинно вибирається випадковим чином ген або підмножина генів. Також випадковим чином обирається точка розриву генома, у яку буде переміщена обрана підмножина. Далі безпосередньо здійснюється операція переміщення обраної підмножини в точку розриву. Після чого гени зсуваються до початкової кількості локусів. Ілюстрація оператора процесу мутації переміщенням показана на рис. 19.

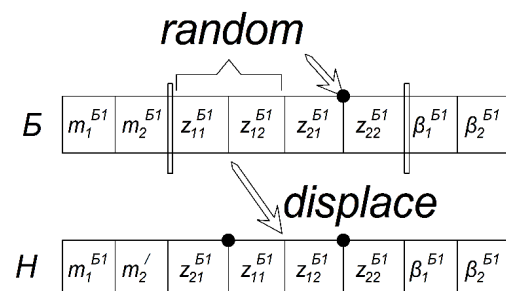


Рисунок 19 – Ілюстрація мутації переміщенням

Оператори мутації в стохастичних алгоритмах є важливим елементом, який відіграє ключову роль у забезпеченні різноманітності та ефективності процесу оптимізації. Вони сприяють збереженню та відтворенню генетичного різноманіття, даючи змогу уникнути «застрягнення» алгоритмів у локальних максимумах чи мінімумах. Оператори мутації допомагають зберігати потенційно корисні комбінації генетичної інформації та вносять випадкові зміни у популяцію, що забезпечує пошук нових оптимальних рішень. Їх використання особливо важливе в умовах, коли розв'язок задачі має складну структуру або простір пошуку є великим. Для алгоритмів, що динамічно змінюються, оператори мутації допомагають адаптуватися до змін у середовищі та ефективно реагувати на нові умови, забезпечуючи

постійний процес оптимізації та покращення рішень. Вони є ключовим елементом для підвищення ймовірності знаходження оптимальних розв'язків.

Висновки:

1. Доведено актуальність сучасних стохастичних алгоритмів, вказано, що у останні два десятиліття спостерігається стрімкий розвиток останніх, який пояснюється збільшенням дослідницьких можливостей та зростанням обсягу даних, які обробляються. Ці алгоритми є ефективними у розв'язанні складних оптимізаційних задач, що дає змогу їх активно використовувати науковою спільнотою та практиками у всьому світі.

2. Освітлено ролі та проведено огляд ключових операторів схрещування та мутації у стохастичних алгоритмах, що дає змогу глибше зрозуміти та популяризувати цей напрям.

3. Проведено аналіз категорії геному та надано детальний огляд основних способів кодування генів для їх застосування, що дає змогу зрозуміти основні принципи, підходи до цього процесу та відповідне ситуативне використання.

4. На конкретному прикладі геному, що відповідає задачі оптимального проектування двоступінчастого співвісного редуктора, розглянуті типові та адаптовані оператори схрещування та мутації, що можуть бути використані для ефективного пошуку розв'язків. До кожного оператора надано текстовий опис та графічно-ілюстроване зображення, що дає змогу чіткіше, якісніше та швидше зрозуміти сутність та послідовності роботи операторів, які є ключовими елементами для підвищення ефективності пошуку оптимальних розв'язків.

Список літератури / References (transliterated)

- Reddy Y., Jithendranath J., Chakraborty A.K., Guerrero J.M. (2022). Stochastic optimal power flow in islanded DC microgrids with correlated load and solar PV uncertainties. *Appl. Energy*, vol. 307, no. 118090.
- Ding H., Hu Z., Song Y. (2012). Stochastic optimization of the daily operation of wind farm and pumped-hydro-storage plant. *Renew. Energy*, no. 48, pp. 571–578.
- Jain P., Shashaani S., Byon E. (2023). Wake effect parameter calibration with large-scale field operational data using stochastic optimization. *Appl. Energy*, vol. 347, no. 121426.
- Ві'аїн'а О., Гхіонеа І.Г. (2023). On Solving Stochastic Optimization Problems. *Mathematics*, vol. 11, no. 4451.
- Scopelliti, D. On a Class of Multistage Stochastic Hierarchical Problems. *Mathematics*, vol. 10, no. 4044.
- Gassmann H.L., Schwartz S.L., Wallace S.W., Ziemba W.T. (2005) Introduction to stochastic programming applications. *Applications of*

Stochastic Programming; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, pp. 179–184.

- Saju S., K. Vijayakumar, Thirugnanam G. (2019) Stochastic Optimization Technique for Digital Image Watermarking. *Journal of Information and Computational Science*, vol. 9, Is.10, pp. 865–876.
- Anit Kumar. (2013). Encoding schemes in genetic algorithm. *International Journal of Advanced Research in IT and Engineering*, vol. 2, no. 3, pp. 1–7.
- B. Fox, M. McMahon. (1991). Genetic operators for sequencing problems, in *Foundations of Genetic Algorithms. G. Rawlins, Ed. Morgan Kaufmann Publishers*, San Mateo, CA, pp. 284–300.
- D. Fogel. (2000) *Evolutionary Computation. IEEE Press*, New York, pp.631–637.
- Serkan Bekiroglu, Tayfun Dede, Yusuf Ayyaz. (2009) Implementation of different encoding type son structural optimization based on adaptive genetic algorithm. *Finite Elements in Analysis and Design*, no. 45, pp. 826–835.
- Jyotishree R.K. (2012). Novel Encoding Scheme in Genetic Algorithms for Better Fitness. *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 1, Is. 6, pp. 214–218.
- John H. Holland. (1992). *Adaptation in Natural and Artificial Systems An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. 232 p.
- L.J. Eshelman. (1997) Crossover operator biases: Exploiting the population distribution. *Proc. the Seventh International Conference on Genetic Algorithms*, pp. 354–361.
- F.G. Lobo, D.E. Goldberg. (2004). The parameter-less genetic algorithm in practice. *Information Sciences*, vol. 167, no. 1, pp. 217–232.
- Tomasz Dominik Gwiżdza. (2006). *Genetic Algorithms Reference*. 47p.
- A.J. Umbarkar, P.D. (2015). Sheth Crossover operators in genetic algorithms: a review. *Ictact journal on soft computing*, vol. 6, is. 1, pp. 1083–1092.
- N.J. Radcliffe. (1991). Equivalence class analysis of genetic algorithms. *Complex Systems*, vol. 5, no. 2, pp. 183–205.
- Kusum Deep, Hadush Mebrahtu. (2011). Combined Mutation Operators of Genetic Algorithm for the Travelling Salesman problem. *IJCOP* Vol. 2, No. 3, Sep-Dec 2011, pp. 1–23.
- S.N. Sivanandam, S. N. Deepa. (2007). *Introduction to Genetic Algorithms*. Springer. 442 p.
- Nitasha Soni, Tapas Kumar, (2014), “Study of Various Mutation Operators in Genetic Algorithms”, *International Journal of Computer Science and Information Technologies*, vol. 5 (3), pp. 4519–4521.
- P. Tang, M. Tseng. (2012). Adaptive directed mutation for real-coded genetic algorithms. *Applied Soft Computing*, vol. 13(1), pp. 600–614.
- S.M. Lim. (2016). *Crossover and mutation operators of real coded genetic algorithms for global optimization problems*. University Putra Malaysia, Malaysia. 45p.
- P. Lamanaga, C.M.H. Kuijpers, R.H. Murga, I. Inza and S. Dizdarevic. (1999). Genetic Algorithms for the Travelling Salesman Problem: A Review. *Representations and Operators Artificial Intelligence Review*, no. 13, pp. 129–170.
- Syswerda, G. (1991). Schedule Optimization Using Genetic Algorithms. *Handbook of Genetic Algorithms*, New York, pp. 332–349.
- L. Davis. (1991). *Handbook of Genetic Algorithms Van Nostrand Reinhold*. New York. 385 p.
- Michalewicz Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin Heidelberg: Springer Verlag. 388p.

Надійшла (received) 30.01.2024

Відомості про авторів / About the Authors

Бондаренко Олексій Вікторович / Bondarenko Oleksiy – кандидат технічних наук (PhD in Eng. S.), Національний технічний університет «Харківський політехнічний інститут», докторант кафедри теорії і систем автоматизованого проектування механізмів і машин; м. Харків, Україна; тел.: (067) 189-97-00; ORCID: <https://orcid.org/0000-0002-2693-5301>; e-mail: avbondko@gmail.com

Устиненко Олександр Віталійович / Ustynenko Oleksandr – кандидат технічних наук (PhD in Eng. S.), доцент, старший науковий співробітник, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри теорії і систем автоматизованого проектування механізмів і машин; м. Харків, Україна; тел.: (093) 398-33-83; ORCID: <https://orcid.org/0000-0002-6714-6122>; e-mail: ustin1964@tmm-saprg.org

Протасов Роман Васильович / Protasov Roman – Словацький технічний університет в Братиславі, старший викладач кафедри транспортних технологій та машинобудування; м. Братислава, Словаччина; ORCID: <https://orcid.org/0000-0003-1611-0610>; e-mail: roman.protasov@stuba.sk

Архіпов Олександр Володимирович / Arkhipov Oleksandr – кандидат технічних наук (PhD in Eng. S.), доцент, Харківський національний автомобільно-дорожній університет, доцент кафедри комп'ютерної графіки; м. Харків, Україна; тел.: (099) 245-61-45; ORCID: <https://orcid.org/0000-0002-2287-1451>; e-mail: alex.khadi.kharkov@gmail.com