

ДОСЛІДЖЕННЯ ОПТИМАЛЬНОСТІ ВИКОРИСТАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ПРОХОДЖЕННЯ ЛАБІРИНТУ

Канаєв Євгеній Юрійович,
студент

Арсенюк Ігор Ростиславович,
к. т. н., доцент

Вінницький національний технічний університет
м. Вінниця, Україна

exellman.99@gmail.com

igrosars@gmail.com

Вступ. Поняття алгоритму належить до підвалин математики. Обчислювальні процеси алгоритмічного характеру відомі людству з глибокої давнини, проте чітке поняття алгоритму сформувалося лише на початку ХХ ст. Для комп'ютерних програм алгоритм є списком деталізованих інструкцій, що реалізують процес обчислення, який, починаючи з початкового стану, відбувається через послідовність логічних станів та завершується кінцевим станом. Генетичні алгоритми виникли у результаті спостереження і спроб копіювання природних процесів, що відбуваються у світі живих організмів, зокрема, еволюції та пов'язаної з нею селекції популяцій живих організмів [1].

Генетичний алгоритм – це метод, що відображає природну еволюцію методів вирішення проблем, і в першу чергу задач оптимізації. Його процедури пошуку засновані на механізмах природного відбору та спадкоємства. В них використовується еволюційний принцип виживання найбільш пристосованих особин [2].

Мета роботи. Аналіз та дослідження переваг використання генетичного алгоритму для проходження лабіринту та покращення функціональності програмного забезпечення з його використанням.

Матеріали та методи. Наразі усе актуальнішими стають новітні алгоритми, відомі як нейронні мережі та генетичні алгоритми.

Генетичні алгоритми надають можливість швидкої генерації прийнятних розв'язків задач, які неможливо розв'язати іншими аналітичними методами,

уникаючи повного перебору і значно скорочуючи часові витрати.

Принцип генетичного алгоритму ґрунтується на основі еволюційного процесу [3]:

- створення первинного покоління;
- цикл перетворень (мутацій);
- цикл породження нових осіб (кросінговер);
- селекція (відбір).

Таким чином, через декілька поколінь залишатимуться лише найкращі екземпляри з усіх, що у контексті вирішуваних задач означає найкращі розв'язки.

Розглянемо генетичний алгоритм на прикладі вирішення проблеми проходження лабіринту. Для цього нам потрібно здійснити певні кроки (рис. 1).

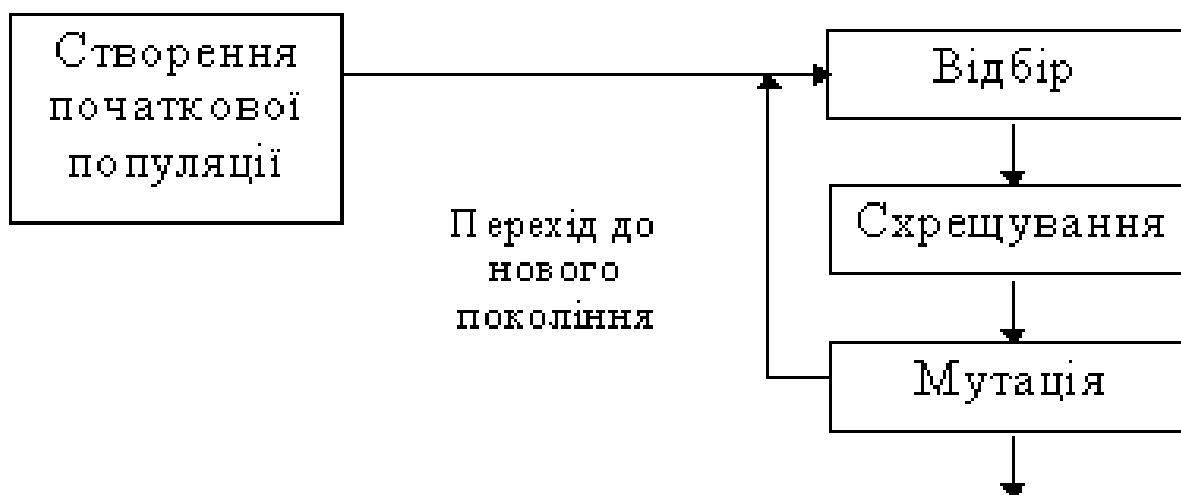


Рис. 1. Схема основних компонент генетичного алгоритму

До першого кроку слід випадковим чином створити деяку початкову популяцію. Навіть якщо популяція виявиться абсолютно неконкурентоздатною, генетичний алгоритм все одно достатньо швидко переведе її у придатну для життя та проходження лабіринту популяцію. Таким чином, на першому кроці можна навіть не намагатися зробити надто допасованих осіб, достатньо, щоби вони відповідали формату осіб популяції, які могли б обстежувати лабіринт, і на них можна було порахувати функцію допасованості. Наслідком першого кроку є популяція N , що налічує N осіб.

На етапі відбору слід із усієї популяції вибрати її певну долю, яка залишиться в «живих» на цьому етапі популяції. Є кілька способів провести

відбір. Ймовірність виживання особи h повинна залежати від значення її допасованості $Fitness(h)$.

Фітнес-функція даного алгоритму використовує шлях, заданий у хромосомі для досягнення Фінішною клітини.

$$\varphi = \left(\frac{\gamma - \alpha}{\mu - \alpha} \right) \times 100,$$

де: φ – значення пристосованості; γ – поточна клітина; α – стартова клітина; μ – фінішна клітина.

Сама ж доля відібраних s (осіб) зазвичай є параметром генетичного алгоритму, і її просто задають заздалегідь. Унаслідок відбору із N осіб популяції N повинні залишитись sN осіб, які увійдуть у наступну популяцію N' . Решта осіб «загине» [4].

Розмноження у генетичних алгоритмах зазвичай статеве: щоби «народити» нащадка, необхідні кілька батьків, зазвичай потрібна участь двох. Розмноження у різних алгоритмах описується по різному і, звісно, залежить від формату осіб. Головна вимога до розмноження – щоби нащадок, чи нащадки, мали можливість успадкувати риси усіх батьків, «змішавши» їх якимось достатньо розумним чином.

Розмноження або оператор рекомбінації застосовують відразу ж після оператора відбору батьків для отримання нових особин-нащадків. Сенса рекомбінації полягає у тому, що створені нащадки повинні наслідувати генну інформацію від обох батьків, у нашому випадку, цією інформацією буде пройдений шлях у лабіринті. Розрізняють дискретну рекомбінацію та кросинговер.

Приклад операції розмноження: Вибрати $(1-s)p/2$ пар гіпотез із N і провести з ними розмноження, отримавши по два нащадки від кожної пари (якщо розмноження описано так, щоб давати одного нащадка, слід вибрати $(1-s)p$ пар, і додати цих нащадків у N' . У результаті N' буде складатися з N особин [5].

Особи для розмноження зазвичай вибираються з усієї популяції N , а не з тих, що вижили на першому кроці (хоча останній варіант теж має право на

існування). Справа в тому, що головна проблема генетичних алгоритмів – недостача різноманітності (diversity) в особах. Достатньо швидко виділяється єдиний генотип, який являє собою локальний максимум і згодом усі елементи популяції програють йому у відборі, і вся популяція «забивається» копіями цієї особи. Існують різні способи боротьби з таким небажаним ефектом; один з них – вибір для розмноження не з самих «допасованих», а взагалі зі всіх осіб.

До мутацій відноситься все те ж, що і до розмноження: є деяка доля мутантів m , що є параметром генетичного алгоритму, і на кроці мутацій необхідно вибрати mN осіб, а згодом змінити їх згідно із заздалегідь заданими операціями мутації.

Алгоритм завершує роботу, коли популяція зійшлася, тобто не виробляє потомство, яке значно відрізняється від попереднього покоління. У нашому випадку це буде тоді, коли популяція дістанеться до виходу та буде знайдено оптимальний шлях. Саме тому, алгоритму не доведеться створювати нової популяції. Коли алгоритм зійшовся, на виході матимемо набір оптимальних рішень (шлях) проходження лабіринту [6].

Результати та обговорення.

Для перевірки оптимальності використання генетичного алгоритму для проходження лабіринту, було розроблено додаток [7], де і проводились порівняльні тестування з іншими алгоритмами. У порівнянні з такими додатками проходження лабіринту на базі генетичного алгоритму як A-Mazer, Maze Navigation було додано функціонал вибору популяції, час життя осіб та запис пройдених шляхів, що дало нам більше можливостей для точнішого тестування.

У результаті аналізу можна зробити висновок, що генетичний алгоритм у середньому на 10% швидше знаходить шлях у лабіринті за такі алгоритми як: правила правої руки, алгоритм застави, алгоритм Тремо, заповнення тупиків; та на 5% швидше, ніж рекурсивний алгоритм.

Доцільність застосування генетичного алгоритму полягає в тому, що даний алгоритм, як правило, “помиляється” (йде в невірному напрямку) не більше ніж на 5 – 10%, проте у результаті, перевіривши всі напрями, знаходить вихід, гарантуючи високу швидкість роботи. Особливістю генетичного алгоритму є акцент на використання оператора «схрещування», який виконує операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещування у живій природі.

Висновки. Обрано метод для розв'язання поставленої задачі та обґрунтована ефективність його використання.

Здійснено аналіз актуальності досліджуваної теми, відповідних проблем, запропоновано їх рішення та розглянуто інструментарії для реалізації інформаційної технології.

Запропонований генетичний алгоритм буде покладено в основу створення інформаційної технології для проходження лабіринту

Здійснено аналіз актуальності досліджуваної теми. Обрано метод для розв'язання задачі проходження лабіринту та обґрунтована ефективність його використання. Запропонований генетичний алгоритм буде покладено в основу створення інформаційної технології для проходження лабіринту

СПИСОК ЛІТЕРАТУРИ.

1. Канаєв Є. Ю., Арсенюк І. Р., Месюра В. І. Обґрунтування доцільності застосування генетичного алгоритму для задачі проходження лабіринтів. / Тези доповідей І науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії. – Вінниця: ВНТУ, 2021. Електронний ресурс (режим доступу <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12522/10611>)
2. Субботін, С. О. Неітеративні, еволюційні та мультиагентні методисинтезу нечіткологічних і нейромережних моделей [Текст] / С. О. Субботін. – Запоріжжя : ЗНТУ, 2009. – 375 с.
3. Малихіна М. П., Частікова В. А., Власов К. А. Дослідження ефективності роботи модифікованого генетичного алгоритму в задачах комбінаторики // Сучасні проблеми науки та освіти. – 2013. – No 3.
4. Van Vliet, D. Improved shortest path algorithms for transport networks [Електронний ресурс] режим доступу <http://www.sciencedirect.com/science/article/pii/0041164778901028>.
5. Barricelli, Nils Aall. Symbiogenetic evolution processes realized by artificial methods. *Methodos*: 143 – 182.
6. Introduction to Genetic Algorithms [Електронний ресурс] - <https://towardsdatascience.com/introduction-to-genetic-algorithms-including->

example-code-e396e98d8bf3.

7. Арсенюк І. Р. Канаєв Є. Ю., Комп'ютерна програма "Програмний модуль подолання лабіринту на базі генетичного алгоритму". Свідоцтво про реєстрацію авторського права на твір № 115466. Дата реєстрації 26.10.2022.