

# **Використання генетичних алгоритмів для дослідження ігрових середовищ та знаходження оптимальних рішень на прикладі гри «Змійка»**

## **Зміст**

<b>Вступ.....</b>	<b>3</b>
<b>Генетичні алгоритми для дослідженні ігрових середовищ .....</b>	<b>4</b>
Генетичні алгоритми.....	4
Середовище для пошуку оптимальних рішень.....	5
Створення початкової популяції та оцінка пристосованості .....	7
Відбір кандидатів для схрещування.....	9
Генетичні оператори схрещування та мутації .....	11
Дослідницька робота.....	13
<b>Висновки.....</b>	<b>15</b>
<b>Список використаних джерел .....</b>	<b>16</b>
<b>Анотація.....</b>	<b>17</b>

## Вступ

Ігрові середовища завжди були важливим об'єктом дослідження в галузі штучного інтелекту. Завдяки ним ми маємо можливість розуміти та аналізувати різноманітні аспекти прийняття рішень, зокрема в умовах невизначеності та складних ігрових умовах. Класична гра «Змійка» є однією з найбільш впізнаваних грою у світі, та має досить просте середовище, що дозволяє шукати різноманітні рішення та тестувати нові або покращувати вже існуючі алгоритми.

Основною метою цієї наукової роботи є застосування генетичних алгоритмів для дослідження ігрового середовища гри «Змійка» та знаходження оптимальних рішень. Генетичні алгоритми, що використовуються для еволюції популяції особин-агентів, показали свою ефективність у вирішенні задач оптимізації та використовуються у різних сферах, включно з ігровими середовищами.

В даній роботі розглядається покрокове застосування генетичних алгоритмів. Буде продемонстровано всі етапи генетичних алгоритмів: створення початкової популяції та оцінку пристосованості, процес відбору кандидатів для формування наступного покоління та застосування генетичних операторів схрещування та мутації.

Ця робота може бути корисною для дослідників та розробників у галузі штучного інтелекту, які цікавляться застосуванням генетичних алгоритмів до ігрових середовищ та оптимізації поведінки агентів в цих середовищах. Результати дослідження можуть сприяти розвитку нових підходів до вирішення ігрових завдань та поліпшенню віртуальних агентів в ігрових програмах.

## Генетичні алгоритми для дослідженні ігрових середовищ

### Генетичні алгоритми

Генетичні алгоритми — це еволюційні алгоритми пошуку, що використовуються для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, подібних до механізмів біологічної еволюції.

Генетичні алгоритми використовуються для вирішення задач, в яких агентів можна представити у вигляді масиву чисел, який часто в термінології називають хромосомами. Для імплементування генетичного алгоритму у якості агентів використовувалися нейронні мережі.

На початку роботи алгоритму, створюється деяка кількість випадкових екземплярів, яку називають популяцією. Особини поміщаються у досліджуване середовище, де перевіряється їх адаптованість до нього. Рівень пристосованості особини оцінюється за допомогою спеціальної функції адаптованості(fitness function).

На основі оцінених значень проводиться селекція(відбір особин для схрещування). Після до вибірки застосовуються генетичні оператори, такі як схрещування та мутація, за допомогою чого створюється нова популяція нащадків.

Процедура, описана вище, продовжується доки не будуть виконанні одна з умов зупинки:

- знаходження глобального або оптимального вирішення;
- вичерпання кількості поколінь, які були виділенні на еволюцію;
- вичерпання часу, виділеного на пошук вирішення проблеми.

Генетичні алгоритми можна використовувати для пошуку рішень в дуже важких і великих просторах пошуку.

## Середовище для пошуку оптимальних рішень

Для демонстрації могутності генетичних алгоритмів, та дослідження їх можливостей було обрано середовище – популярна гра «Змійка».

Середовище для дослідження являє собою поле 30\*40 умовних квадратних одиниць, в якому знаходиться агент – змійка та ціль – фрукт (рис 1.).

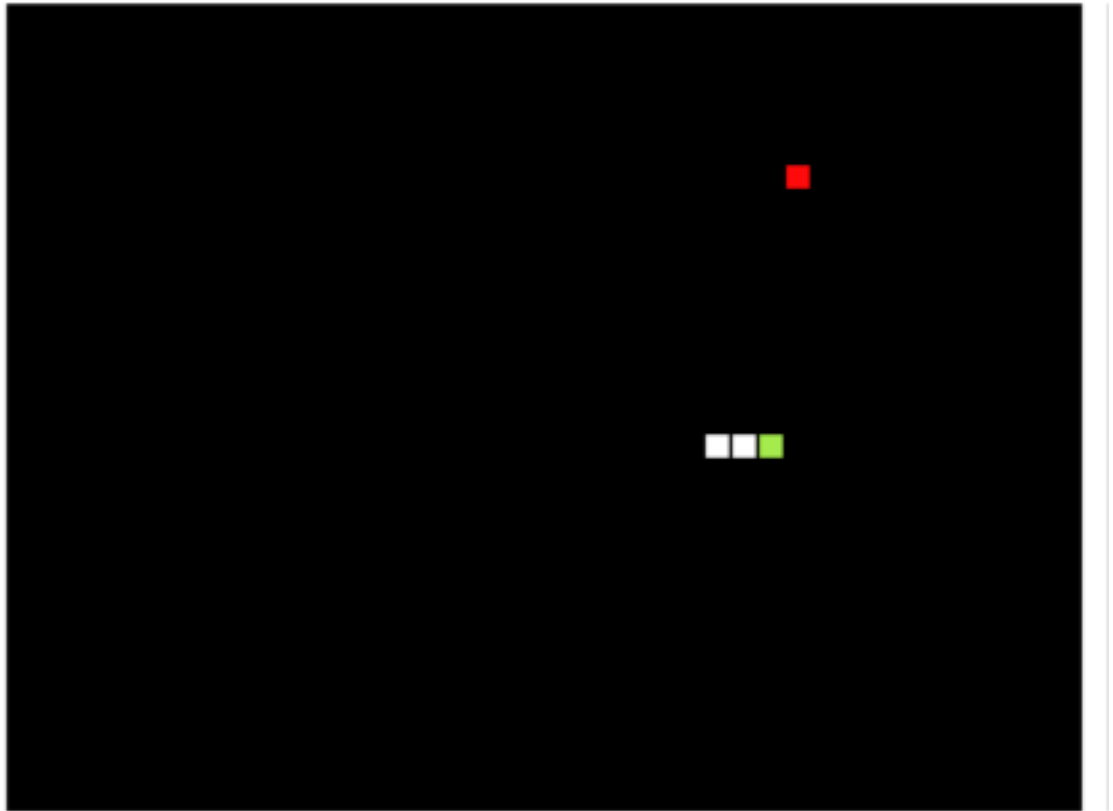


рис. 1, демонстрація середовища для дослідження

Змійка – агент, суб'єкт популяції, основною ціллю якого є дослідження ігрового середовища. Змійка здатна рухатися на всі 4 сторони. Має початкову довжину 3, та збільшує її на 1 при кожному «поїданні» фрукту. Діями змійки керує неймережа, яка на вхід приймає вектор з 11 чисел, які характеризують поточний стан змійки в середовищі, а на вихід видає 3 значення, ймовірності виконати якусь дію, що в сумі дають 1.

Вхід для неймережі задається вектором з 11 чисел, які несуть в собі інформацію про поточний стан особини в середовищі, а саме:

- наявність елемента тіла змійки або стіни перед, зліва та справа голови змійки;
- інформацію про рух змійки та напрямок;
- інформацію про положення голови змійки відносно фрукта.

Простір дій змійки, або вихід нейромережі – це вектор з 3-х чисел, кожне з яких показує ймовірність повороту наліво/направо або продовжувати рух прямо. Задача нейромережі, це фактично класифікація дій у даному середовищі в залежності від його стану.

Для пошуку оптимального рішення виконується пошук відповідних параметрів нейромережі агенту-змійки.

Необхідно зазначити, що в даній реалізації використовувалася доволі проста версія нейромережі, і середовище представлялося простими характеристиками. Для інших досліджень можна використати більш складні моделі, які наприклад прийматимуть весь вигляд середовища, поточний та попередній стан, для того, щоб розуміти напрямок руху.

## Створення початкової популяції та оцінка пристосованості

Першим етапом для імплементації генетичного алгоритму є створення початкової популяції. Переважно в перших поколіннях особини будуть зовсім неконкурентноспроможними та малоефективними в новому середовищі, проте з часом алгоритм знайде оптимальні рішення.

Початкова вибірка – це випадково ініціалізовані нейронні мережі (рис. 2). Для пришвидшення пошуку рішення можна використати агентів, які були навчені за допомогою інших методів, наприклад q-навчання[1] або ж policy gradient[2] методів, попередньо піддавши їх генетичному оператору мутацій для створення більшого різноманіття рішень.

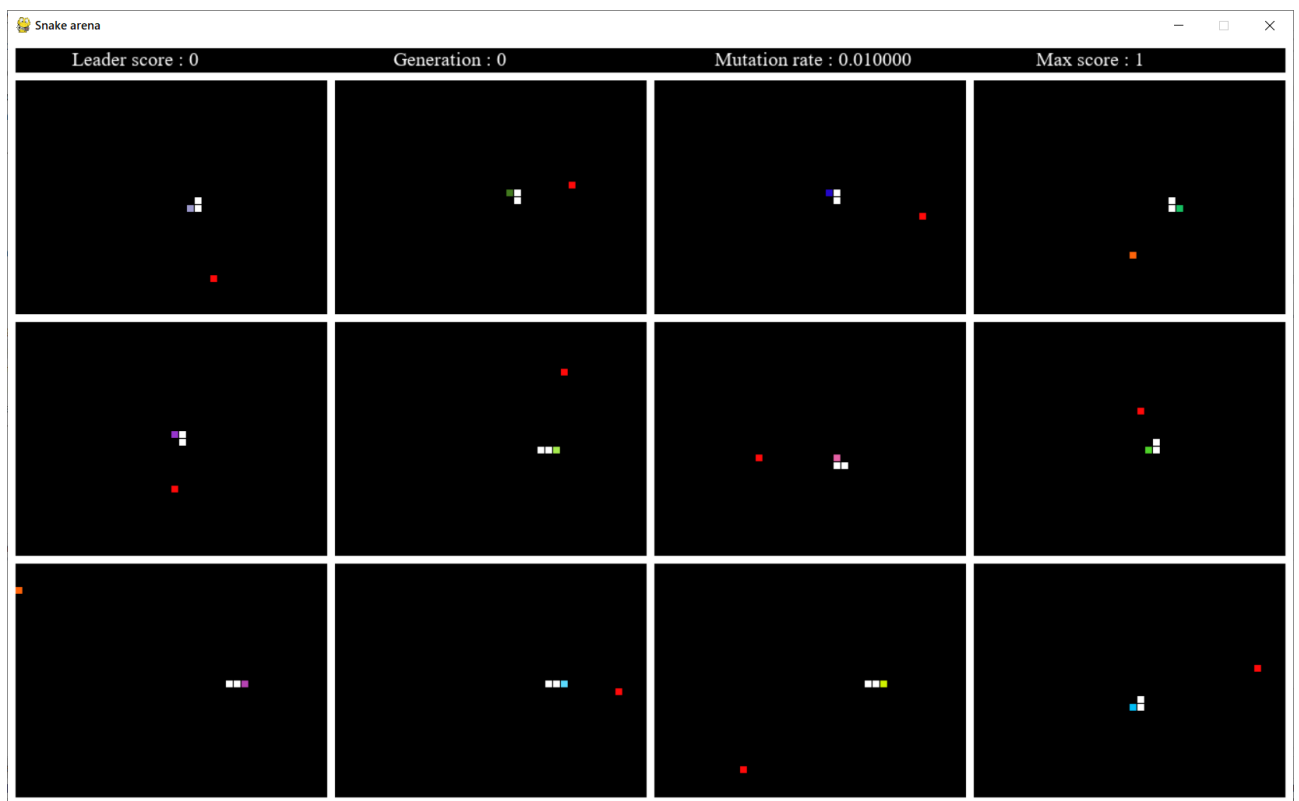


рис. 2, демонстрація непристосованої початкової вибірки

Обрання функції пристосованості є одним з найголовніших етапів при проектуванні систем, що оптимізуватимуться генетичними алгоритмами. Саме завдяки коректно обраній функції пристосованості можна оцінити успіхи особини в середовищі, врахувавши всі фактори, завдяки чому ми зможемо знайти швидше оптимальне рішення.

В даній роботі використовувалася наступна функція оцінки пристосованості:

$$\varepsilon_{fitness} = \sum_{i=0}^{l_{snake}-3} \frac{(S_{field} - (c_{steps} - (|x_{head} - x_{food}| + |y_{head} - y_{food}|)))}{(\frac{S_{field}}{l_{snake}})}$$

де:

$l_{snake}$  – довжина змійки на момент поразки,

$c_{steps}$  – кількість кроків, які здійснила змійка поки не дісталася до n-го фрукта,

$S_{field}$  – площа ігрового середовища, в умовних одиницях,

$x_{head}, y_{head}$  – координати голови змійки під час ініціалізації фрукта, необхідна для вимірювання відстані, яку треба пройти змійці до цілі,

$x_{food}, y_{food}$  – координати фрукта.



## **Відбір кандидатів для схрещування**

Відбір кандидатів, що будуть використані для схрещування один з найважливіших етапів даного алгоритму. На цьому етапі відбирається певна частина особин з популяції, на основі значення їх пристосованості.

Існують різні способи для проведення відбору. Зазвичай всі з них ґрунтуються на обрані найкращих особин, які проявили себе найкраще у тестовому середовищі.

Існують наступні методи відбору:

- відбір деякої кількості найкращих особин, що виражається якимсь числом від загальної кількості популяції, дане число буде гіперпараметром,
- відбір лише тих особин, значення пристосованості яких додало встановлений поріг, значення якого також буде гіперпараметром, який може бути адаптивним, та наприклад збільшуватися з кожним поколінням, адже під час еволюції і рівень навичок особин має рости,
- відбір особин для розмноження з усієї популяції, проте ймовірність вибору має бути пропорційна значенню функції пристосованості. Тобто агенти, які продемонстрували себе краще у середовищі матимуть більші шанси «дати потомство», ніж погано пристосовані.

Якщо користуватися двома першими методами, описаними вище, популяція з часом стане дуже монотонною та однотипною (рис. 3, колір голови зміїнок вказує на те, що в нас популяція дуже однотипна), що не є бажаним результатом, адже для пошуку оптимального рішення найкращим варіантом буде велика різноманітність особин. Дана ідея імплементована в останньому методі, та допомагає урізноманітнити майбутні популяції.

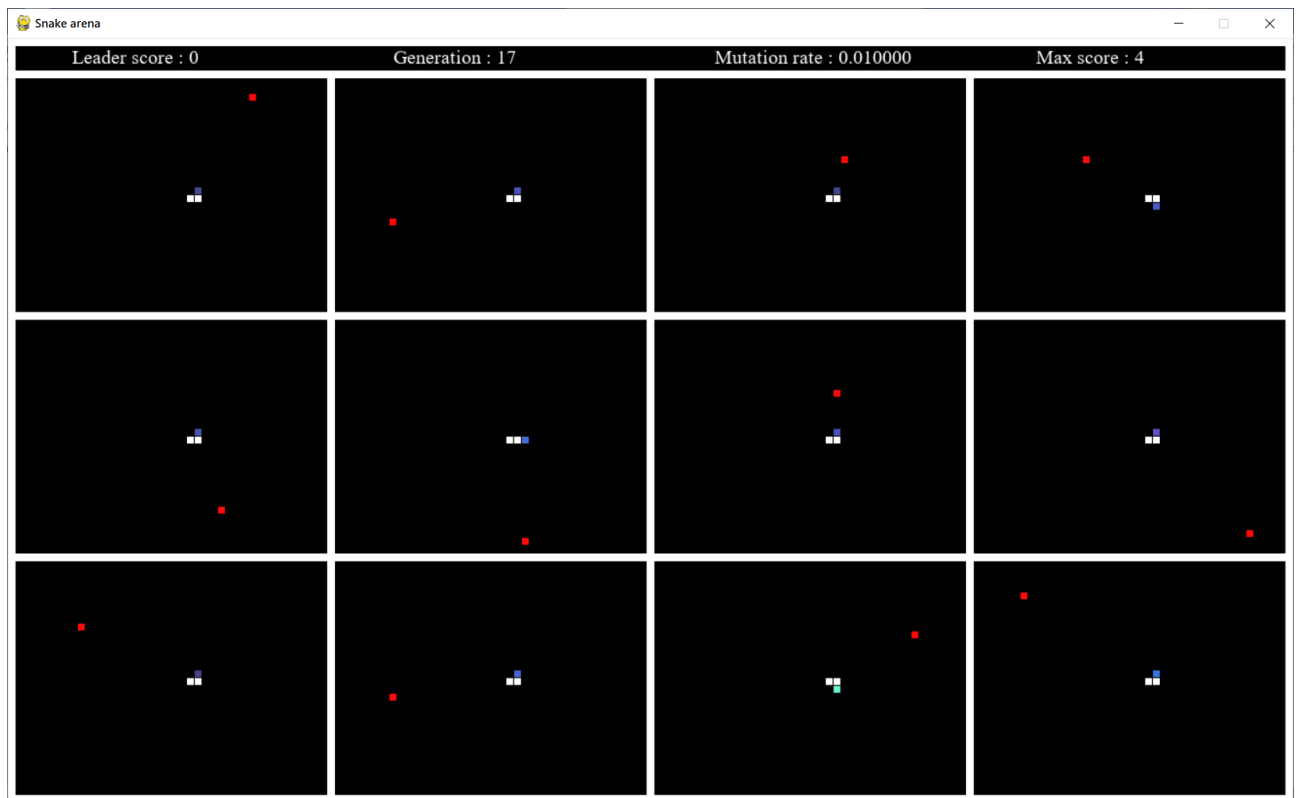


рис. 3, демонстрація однотипності змійок

Також варто зазначити, що можливий варіант, коли нащадки будуть менш пристосовані до середовища, ніж батьківські особини, тому рекомендовано зберігати батьківське покоління, і формувати вибірку для схрещування відразу з двох популяцій, таким чином ми зможемо запобігти деградуванню популяції.

## Генетичні оператори схрещування та мутації

Розмноження в генетичних алгоритмах статеве, для його проведення необхідно дві батьківські особини. Для розмноження зі створеної підвибірki  $N$  раз обираються дві батьківські особини, між якими застосовуються оператор схрещування, де  $N$  – розмір популяції.

Існують різні види операцій схрещування:

- дискретна рекомбінація;
- кросинговер.

Дискретна рекомбінація – це спосіб обміну генами між особинами. Виконується вона наступним чином: для кожного гену(в даному випадку ген – це один параметр нейромережі) батьківських особин ми з однаковою ймовірністю обираємо один, та записуємо в генотип дочірньої особини.

Кросинговер – це один із операторів рекомбінації в генетичних алгоритмах. В даній роботі було реалізовано одноточковий кросинговер. Він реалізується наступним чином: обирається випадковим чином точка серед параметрів нейромережі, і всі елементи, що йдуть до цієї точки беруться від однієї батьківської особини, а всі після від іншої.

Важливим генетичним оператором, який допомагає урізноманітнити популяцію, є мутацію (рис. 4, мутація кольору голови змійки). Подібно до природи, мутація змінює деякі гени агентів на нові значення.

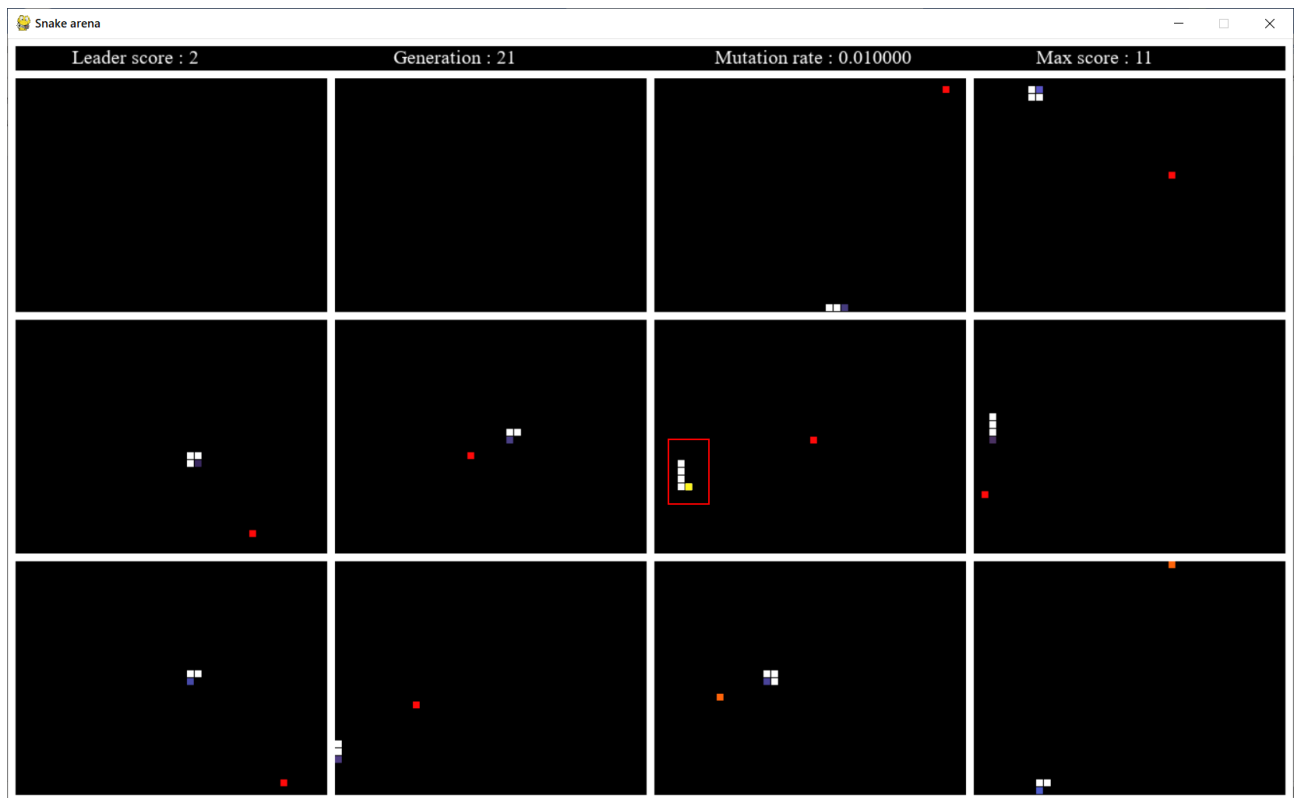


рис. 4, приклад мутації кольору голови змійки

Ймовірність мутації є гіперпараметром, який також можна зробити адаптивним, наприклад, на самому початку, коли особини ще зовсім не пристосовані до середовища, для того, щоб вони швидше шукали оптимальне рішення коефіцієнт мутації можна встановити високим, приблизно 0.1, і поступово, з поколіннями зменшувати його до якогось фіксованого мінімального значення, адже не бажано встановлювати в 0, бо так різноманітність може стати мінімальною. Або навпаки, якщо спостерігається стагнація популяцій в нових поколіннях, можна збільшити коефіцієнт мутації, стимулюючи формування нових рішень проблеми, яка постала.

## Дослідницька робота

Окрім уже наявних методів схрещування, в даній роботі розглядався інший підхід. Оскільки параметри нейромереж – це числа, і нам необхідно створити дочірню особину, було запропоновано наступне рішення – вагові коефіцієнти нової мережі визначити як середнє арифметичне між двома батьківськими відповідними генами. Дана ідея була імплементована в науковій роботі. Нижче на рис 5. наведені результати, в порівнянні з іншими операторами рекомбінації.

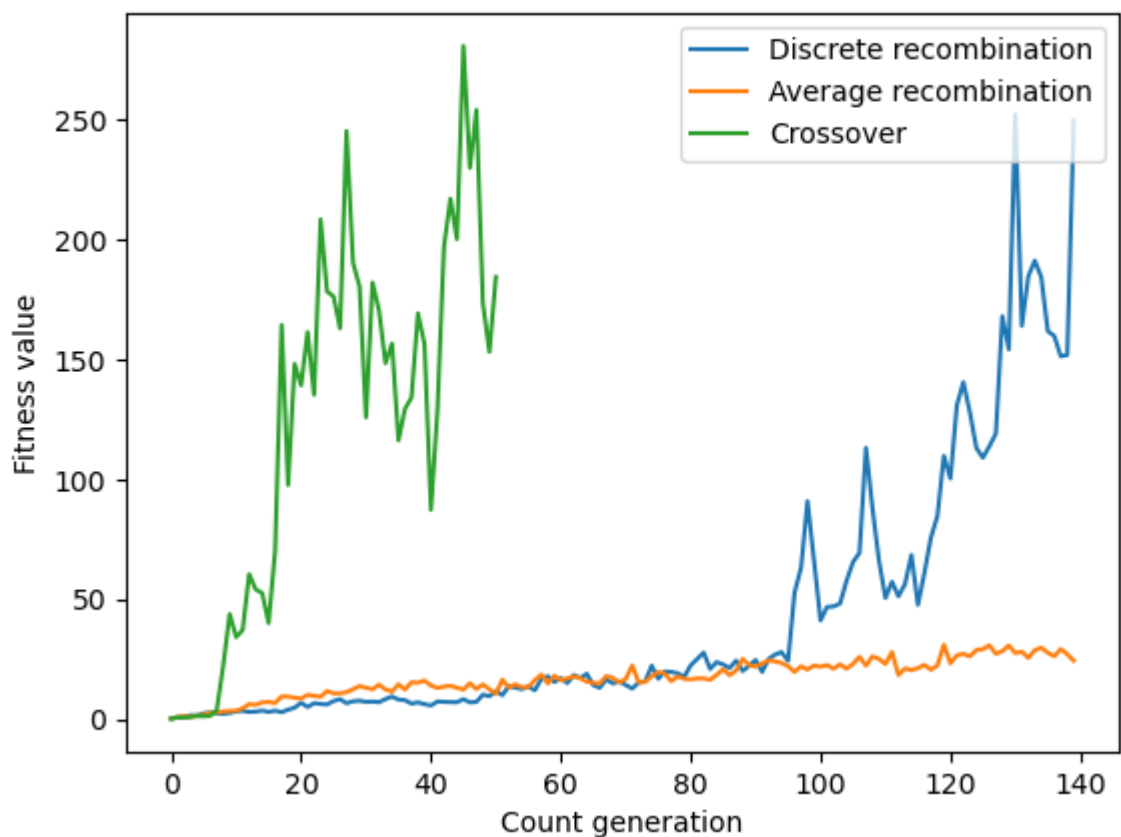


рис . 5, порівняння методів реплікації

Окрім цього було порівняно дві різні функції адаптивності, одна враховувала лише кількість фруктів, які змійка «з'їла», а інша приведена вище враховувала також і час, затрачений на пошук. Під час використання першої функції поведінка агентів була досить примітивною (див. рис. 2), вони просто обрали собі прямокутну траєкторію, і постійно за нею рухалися, в той час, як із спеціалізованою функцією поведінка стала більш складною, агенти намагалися

поєднувати час та ефективність ходів. Окрім цього було встановлено обмеження на кількість ходів, і якщо агент перевищував цю кількість, не діставшись цілі, він автоматично програвав. Максимальна початкова кількість ходів була встановлена в 70 (максимальний шлях, який треба пройти змійці до цілі, 30 – висота поля і 40 - ширина), і при кожному успішному «поїданні» фрукту збільшувала на 1, як і довжина змійки.

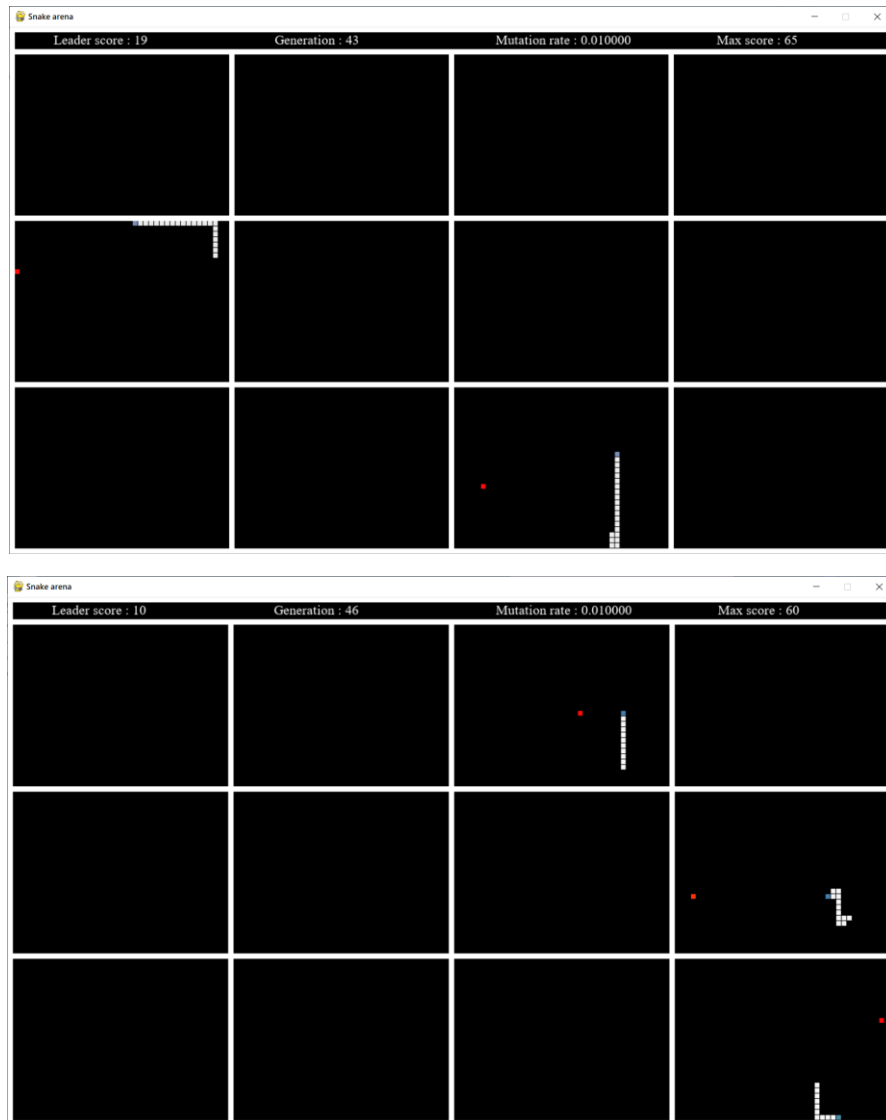


рис. 6, демонстрація примітивної логіки пошуку змійками

Однією з причин не достатньо складної поведінки є обрання достатньо примітивної нейромережі, і якщо проєкспериментувати з заміною моделі на інші, то можна добитися кращих результатів. Проте варто пам'ятати, що зі збільшенням складності моделі виросте і час, потрібний на її навчання.

## **Висновки**

У цій науковій роботі було проведено дослідження використання генетичних алгоритмів у ігровому середовищі «Змійка». Використання генетичних алгоритмів дозволило ефективно дослідити та оптимізувати стратегії поведінки агентів у складних ігрових умовах.

Результати дослідження показали, що генетичні алгоритми можуть бути успішно використані для оптимізації поведінки агентів у іграх, на прикладі гри «Змійка».

Отримані результати мають практичне значення для розробників ігор та дослідників у галузі штучного інтелекту. Цей досвід може бути використаний для створення більш інтелектуальних та адаптивних віртуальних агентів у різних ігрових середовищах. Крім того, отримані результати можуть служити основою для подальших досліджень у сфері оптимізації поведінки агентів у ігрових програмах та розширення можливостей застосування генетичних алгоритмів у галузі ігрового дизайну та розвитку штучного інтелекту.

### Список використаних джерел

1. <https://arxiv.org/abs/1312.5602>
2. <https://spinningup.openai.com/en/latest/algorithms/vpg.html>
3. <https://arxiv.org/ftp/arxiv/papers/2007/2007.12673.pdf>
4. <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>
5. <https://www.geeksforgeeks.org/genetic-algorithms/>
6. [https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%B0\\_%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B1%D1%96%D0%BD%D0%B0%D1%86%D1%96%D1%8F](https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%B0_%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B1%D1%96%D0%BD%D0%B0%D1%86%D1%96%D1%8F)
7. [https://uk.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D0%B9\\_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC](https://uk.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC)
8. [https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D0%B8%D0%BD%D0%B3%D0%BE%D0%B2%D0%B5%D1%80\\_\(%D0%B3%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D0%B9\\_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\)](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D0%B8%D0%BD%D0%B3%D0%BE%D0%B2%D0%B5%D1%80_(%D0%B3%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC))



## **Анотація**

У науковій роботі досліджено використання генетичних алгоритмів для аналізу ігрових середовищ та пошуку оптимальних рішень на прикладі популярної гри "Змійка". Завданням даного дослідження було вивчення можливостей застосування генетичних алгоритмів у ігрових середовищах та визначення їх ефективності в пошуку оптимальних рішень.

У роботі було змодельоване ігрове середовище і проведено його дослідження агентами-змійками, в результаті чого було отримано результати, і проведено їх аналіз. Здійснено оцінку ефективності генетичних алгоритмів та їх застосування для дослідження ігрових середовищ.

Результати дослідження вказують на успішну застосовність генетичних алгоритмів для оптимізації стратегій в грі "Змійка" і підтверджують їхню ефективність в аналізі ігрових середовищ. Отримані висновки можуть бути корисними для подальшого розвитку і вдосконалення алгоритмів в галузі ігрового моделювання та штучного інтелекту.

Ключові слова: генетичні алгоритми, ігрове середовище, оптимальне рішення, гра "Змійка", нейронні мережі, реплікація.