

## ГЕНЕРАЦІЯ СКІНЧЕНИХ АВТОМАТІВ ДЛЯ РОЗВ'ЯЗАННЯ ІГРОВИХ ЗАДАЧ ЗА ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

*Запорізька державна інженерна академія, кафедра ПЗАС*

Останнім часом все частіше застосовуються генетичні алгоритми для вирішення різних типів задач, таких як: моделювання еволюційних процесів, прогнозування, ігрові стратегії, задачі оптимального управління, оптимізація функцій, створення розкладів, тощо. Широкий спектр застосування та швидкий темп розвитку таких алгоритмів говорять про те, що на даний момент це один із найперспективніших напрямків у штучному інтелекті і програмуванні.

Генетичні алгоритми – це адаптивні методи пошуку оптимальних рішень, що засновані на генетичних процесах біологічних організмів: біологічні популяції розвиваються на протязі кількох поколінь, підкоряючись законам природного відбору і за принципом «виживає найбільш пристосований» відкритого Чарльзом Дарвіном[2]. Вони працюють з сукупністю "особин" - популяцією, кожна з яких представляє можливе рішення даної проблеми. Кожна особина оцінюється мірою її "пристосованості" згідно з тим, наскільки "добре" відповідає їй рішення задачі. Основні оператори, які при цьому використовуються - це селекція, при якій відбираються найбільш пристосовані особини, схрещування, при якому з генотипу двох батьків утворюється генотип нащадку та мутація, коли деякі гени нащадка змінюються випадковим чином.

Крім вищезазначених задач, що може вирішувати еволюційні алгоритми є ще одна - навчання нейронних мереж. Нейроеволюція - це форма машинного навчання, що використовує еволюційні алгоритми для тренування нейронних мереж[3]. При використанні генетичних алгоритмів для навчання багатослойного перцептронного нейронного мережі можна діяти двома способами. Перший – шукати безпосередньо комбінацію ваг синаптичних зв'язків нейронної мережі, а другий - шукати нейронну мережу з потрібними параметрами (кількість шарів і нейронів у них, функція активації, входи – виходи мережі і т.д.) , тобто вирішується завдання структурної оптимізації. В даній роботі було застосовано перший метод.

В той же час завжди актуальним залишається автоматне програмування, в рамках якого поведінка програм описується за допомогою скінчених детермінованих автоматів. Скінчений автомат - це абстрактна машина, яка може перебувати в одному з декількох різних, наперед визначених станів [1]. Скінчений автомат також може визначити набір умов, які вказують, коли необхідно перейти до іншого стану. Поточний стан визначає, як автомат поводить.

Скінчені автомати використовували навіть на ранніх етапах розвитку програмування комп'ютерних ігор. Наприклад, примари у відомій грі Pac-Man були реалізовані як скінчені автомати. Вони могли вільно переміщатися та переслідувати або уникати гравця. У кожному стані вони вели себе по-різному, і їх переходи визначали дії гравця. Наприклад, якщо персонаж їв таблетку харчування, стан примар міг змінитися із переслідування на уникання. Хоча скінчені автомати були винайдені достатньо давно і використовуються вже протягом досить довгого часу, вони досі залишаються актуальними і корисними у сфері комп'ютерних ігор. Той факт, що вони є відносно легкі для розуміння та впровадження й сприяє їх частому використанні при розробці ігор.

Проте існують деякі складності – є такі типи задач, для яких побудова таких автоматів є досить трудомістким та затратним завданням або навіть неможливою. Тому в рамках даної роботи цей процес було автоматизовано – реалізована генерація автоматів за допомогою генетичних алгоритмів.

Незважаючи на те, що і генетичні алгоритми, і скінчені автомати є досить популярними методами для вирішення певних задач, залишається недостатньо дослідженими об'єднання цих двох стратегій для досягнення потрібних результатів. В даній роботі їх було об'єднано, що дозволило використовувати сильні сторони кожної із них. Для дослідження можливості об'єднання генетичних алгоритмів та скінчених автоматів була обрана ігрова задача. В даній задачі на полі знаходиться персонаж та інші предмети (перешкоди, їжа, тощо). Задача вважається вирішеною при знаходженні виходу або при отриманні висновків що виходу не має. При чому вирішення задачі вважається тим кращим, чим менше кроків буде зроблено персонажем для досягнення цілі.

Поле гри поділено на умовні клітинки, кожна з яких пов'язана з окремим станом автомату. Хід ігрового персонажа – це перехід у певний стан. На вхід подається напрямок руху персонажу - вперед, назад, вліво, вправо або по діагоналях (усього 8 можливих напрямків). На виході отримується 1 (вихід знайдено) або 0 (вихід не знайдено через якусь причину - наприклад він заблокований перешкодами).

За пошук оптимального шляху відповідає генетичний алгоритм. За допомогою нього знаходяться синаптичні ваги для нейронної мережі. На кожному етапі мережа аналізує стани автомату, в які можна перейти із поточного та обирає найкращий, після чого виконує перехід у нього. Використана нейрона мережа має наперед задану структуру - 2 шари, кількість нейронів у шарі прив'язана до кількості вхідних параметрів. Використання генетичного алгоритму для навчання мережі дозволило уникнути деяких труднощів, які б були присутні при навчанні методом з учителем-важко передбачити приклади, на яких можна було б навчити дану мережу. Одна особина в генетичному алгоритмі є закодованим набором значень синаптичних вагів мережі. Фітнес-функція генетичного алгоритму оцінює якість досягнення поставлених цілей - чи прийшов персонаж до виходу під керівництвом даної мережі, за скільки ходів він її досяг та чи зібрав при цьому всю їжу. Таким чином, із покоління в покоління ігровий персонаж все ближче досягає поставленої мети, забезпечуючи оптимальні переходи скінченого автомату.

У процесі дослідження виявлені проблеми, що виникали при реалізації поставленого завдання - по-перше це складність визначення архітектури мережі та потрібних вхідних параметрів для оптимальної роботи алгоритму, по-друге - небезпека попадання у локальний мінімум при знаходженні рішення генетичним алгоритмом. Перша проблема вирішена експериментальним чином – задавались мережі різної архітектури і виявлені найбільш відповідні даній задачі. Для вирішення другої задачі запропоновані прийоми виведення алгоритму з локального мінімуму.

## Висновки:

1. Було вивчено особливості генетичних алгоритмів та скінчених автоматів, область їх застосувань.
2. Реалізоване об'єднання стратегій автоматного програмування та еволюційних алгоритмів.
3. На практиці реалізована ігрова задача, яка використовує сильні сторони обох стратегій
4. Зроблені висновки про можливість та доцільність об'єднання генетичних алгоритмів та скінчених автоматів для вирішення поставленої задачі.

## Література

1. David M Bourg with Glenn Seemann. AI for Game Developers//O'Reilly Media, Inc, 2004.
2. Исаев Сергей. Популярно о генетических алгоритмах./[Електронний ресурс] [www.algolist.manual.ru](http://www.algolist.manual.ru)
3. «ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ» Научный журнал КубГАУ, №91(07), 2013 года