

УДК 004.832.24

ДОСЛІДЖЕННЯ ТА РОЗРОБКА ГЕНЕТИЧНИХ АЛГОРИТМІВ РОЗВ'ЯЗАННЯ ІГРОВИХ ЗАДАЧ

Каверін А.М.

Науковий керівник – д.т.н., проф. Петров К.Е.

Харківський національний університет радіоелектроніки, кафедра ШІ
м. Харків, Україна

тел.: +38(099) 447-29-50, email: anton.kaverin@nure.ua

The purpose of this work is to research and analyze methods of solving game problems using genetic algorithms, as well as to develop an own approach based on them. It is necessary to develop your own algorithm or a part of the algorithm based on the genetic algorithm for solving game problems. The main goal is to improve qualitative or quantitative indicators in comparison with other considered algorithms when using the same time indicators.

Починаючи з кінця двадцятого сторіччя, набули популярності і важливості розробки, що пов'язані з використанням технологій штучного інтелекту при розв'язанні різноманітних практичних завдань.

Звичайно, що штучний інтелект торкнувся і ігрової сфери, де він використовується як для генерації ігрових мап, оптимізації комп'ютерного обладнання, так і для навчання штучного інтелекту проходженню ігор чи вирішенню певних ігрових задач з метою подальшого їх покращення, тобто навчання неігрових персонажів з метою їх перетворення в більш реалістичних та схожих на реальних гравців.

Одним із достатньо ефективних підходів до навчання неігрових персонажів в іграх є використання генетичних алгоритмів [1], які мають на меті створення якомога більш ефективного бота [2].

Проте важливо зазначити, що як і будь-який інший алгоритм, генетичний алгоритм має ряд серйозних недоліків [3].

Аналізуючи більшість сучасних популярних видів мутації можна помітити, що всі вони базуються на ймовірнісних показниках. Це означає, що рішення може бути знайдене з певною ймовірністю. Це значно впливає на інші показники роботи алгоритму, а саме на час його роботи та точність.

Так як найважливішою причиною проблеми збіжності генетичного алгоритму є повна залежність від ймовірнісних показників, то було вирішено удосконалити алгоритм для більш детермінованої його роботи. Головною ідеєю є збір статистики під час виконання навчання поколінь та, базуючись на цих даних, контролювати виконання операції мутації окремих генів чи окремих реальних об'єктів.

Основою алгоритму є збір та аналіз даних, щодо вирішення задачі попередніми поколіннями та визначення наступного стану гену на основі проаналізованих даних.

Для того, щоб досягти такої поведінки необхідно визначити дві характеристики, що будуть описувати стан системи у попередні моменти часу, а саме поняття: моди та тенденції. Їх значення можуть бути визначені на основі аналізу даних та виділення найбільш популярних значень попередніх поколінь з різницею в довжині оброблюваної кількості попередників.

Мода в загальному розумінні – це значення випадкової величини, що трапляється найчастіше зі всієї сукупності спостережень. Тобто це значення, що зустрічається в масиві даних частіше ніж інші. Також, в сукупності даних іноді можна виділити більше ніж одну моду. Транслюючи поняття моди на генетичній алгоритм можна визначити, що це значення бітів в геномі або реальне значення в поколінні, що зустрічається частіше за інше.

Тенденція або тренд в загальному розумінні – це значення випадкової величини, що трапляється найчастіше в дуже малій сукупності спостережень. На відміну від моди, яка по суті своїй є вираженням популярного значення протягом усього або досить великого періоду навчання, тенденція означає яскраву різку і ймовірно якісну відмінність, яка виникла декілька поколінь тому і має популярність серед всіх цих поколінь.

Окрім цього, необхідно визначити поняття мінімально допустимої кількості значень, яке буде використовуватися для знаходження найбільш самодостатнього значення гену певної позиції. Цей показник дорівнює мінімальній кількості елементів у відсотках чи в умовних одиницях, які мають бути присутні для того, щоб вважати це значення найпопулярнішим.

Зважаючи на все викладене вище, мета роботи полягає у розробці власного методу розв'язання ігрових задач на основі використання генетичних алгоритмів, який би мав кращі якісні чи кількісні показники у порівнянні з існуючими підходами.

Список використаних джерел:

1. A review on genetic algorithm: past, present, and future. [Електронний ресурс] – Режим доступу URL: <https://link.springer.com/article/10.1007/s11042-020-10139-6>
2. Генетический алгоритм на примере бота Robocode. [Електронний ресурс] – Режим доступу URL: <https://habr.com/ru/post/111811/>
3. Wirsansky E. Hands-On Genetic Algorithms with Python: Applying genetic algorithms to solve real-world deep learning and artificial intelligence problems. Birmingham: Packt Publishing, 2020. – 346 p.