УДК 681.12

Астаф'єв В.В.
Белзецький Р.С.

Програмне забезпечення на основі генетичного алгоритму для пошуку виходу з лабіринту

Вінницький національний технічний університет

Анотація

Проведено огляд існуючих алгоритмів пошуку виходу з лабіринту. Розглянуто метод пошуку виходу з лабіринту за допомогою генетичного алгоритму та його перевагу над іншими алгоритмами. Доведено доцільність його використання для виконання даної задачі.

Ключові слова: лабіринт, пошук шляху, генетичний алгоритм, програмне забезпечення, відбір.

Abstract

A review of existent methods of looking for an exit from a maze. The method looking for an exit from a maze using a genetic algorithm and its advantage over other algorithms is considered. The expediency of its use for this task is proved.

Keywords: maze, path search, genetic algorithm, software, selection.

Вступ

Значна кількість проблем реального світу вирішуються завдяки покроковому просуванню, де кожен вибір дозволяє просунутися далі по шляху. Задача знаходження шляху в лабіринті відноситься до подібних проблем. Тому її вирішення буде не менш важливим.

Метою даної бакалаврської дипломної роботи є розширення функціональних можливостей програмного забезпечення для пошуку виходу з лабіринту.

Аналіз дослідження

Дослідження даної теми є безперечно актуальним, адже розробка, аналіз та застосування ефективних і універсальних методів розв'язку задач (алгоритмів) є наразі ключовою задачею комп'ютерних наук, а тема еволюційних алгоритмів ще не достатньо досліджена.

Для початку визначимо деякі поняття. Вхід у лабіринт – це місце, де ви починаєте свою подорож; зазвичай вхід знаходиться на краю лабіринту. Мета полягає в тому, щоб підняти точку, до якої ви хочете досягти. Ціль може бути в будь-якому місці лабіринту, включаючи вихід. Вузол вважатиметься входом, пунктом призначення, а також будь-якою точкою, де коридор розгалужується або закінчується в глухому куті. Відрізок шляху між сусідніми вузлами називається заглушкою. Маршрут являє собою ряд відгалужень. Стіна - це одна з двох сторін дороги. Що таке стіна в підземному лабіринті, зрозуміло і без пояснень. У садовому лабіринті, наприклад, стіною може слугувати живопліт або невисокий насип, що обмежує шлях з обох боків [1].

Існує ряд різних алгоритмів вирішення лабіринтів, тобто методів автоматичного пошуку виходу.

Дотримання уздовж стін - це простий алгоритм вирішення лабіринтів. Пріоритетним для нього є прохідний об'єкт лабіринту («ти»), він завжди дуже швидкий і не використовує додаткову пам'ять. Ми починаємо йти алеями і завжди повертаємо праворуч (або завжди ліворуч), коли доходимо до розвилки. Щоб застосувати це рішення до реального лабіринту, вам потрібно покласти руку на праву (або ліву) стіну і постійно притискати її до стіни, коли ви рухаєтеся лабіринтом. Якщо необхідно, ви можете вибрати клітинки, які вже відвідали, і клітинки, відвідані двічі. Зрештою, ви можете повернутися до рішення, лише переглянувши клітинки, відвідані один раз. Цей метод не обов'язково знаходить найкоротший розв'язок і взагалі не працює, якщо ціль знаходиться в центрі лабіринту і оточена замкнутим ланцюгом, оскільки ви обійдете середину і зрештою потрапите до

початку. Примикання вздовж стіни в тривимірному лабіринті можна виконати детермінованим шляхом, спроектувавши тривимірні переходи в 2D площині, тобто уявити, що проходи вгору насправді ведуть на північний захід, а проходи вниз на південний схід, а потім застосувати звичайні правила для прохід уздовж стін.

Алгоритм Пледжа. Це модифікована версія стінного проходу, що дозволяє вам стрибати між «острівцями», щоб вирішувати лабіринти, які не можуть проходити вздовж стін. Це гарантований спосіб досягти зовнішнього краю будь-якого 2D лабіринту з будь-якого місця всередині, але він не може зробити протилежне пошуку рішень всередині лабіринту. Це чудово реалізувати за допомогою відповідного робота-лабіринту, оскільки він зможе вибратися з будь-якого лабіринту, жодним чином не позначаючи та не запам'ятовуючи шлях. Ми починаємо з вибору напрямку і завжди йдемо в цьому напрямку, коли це можливо. Притулившись до стіни, починаємо йти за нею, поки не зможемо знову йти в обраному напрямку. Варто відзначити, що прохід уздовж стіни повинен починатися з протилежної стіни, де ми відпочивали. Якщо перехрест у цій точці повернеться, це може призвести до розвороту на середині переїзду і повернення назад тим же шляхом, яким ми прийшли. Під час руху вздовж стіни враховуйте кількість поворотів, наприклад, лівий поворот дорівнює -1, а правий – 1. Припиніть рух вздовж стін і почніть рух у вибраному напрямку, лише коли загальна кількість поворотів дорівнює, тобто якщо ви Повернувшись на 360 градусів або більше, продовжуйте рухатися вздовж стіни, поки вона не «розплутається».Розрахунок гарантує, що рано чи пізно ми досягнемо віддаленої частини «острова», де ми зараз знаходимося, і перескочимо на інший острів у обраному напрямку, а потім продовжимо стрибки між островами, поки не притулимося до прикордонної стіни, потім прогулянка по стінах приведе нас до виходу. Пам'ятайте, що алгоритм Plage може змусити нас відвідати прохід або вихідну точку кілька разів, але наступного разу у вас завжди буде інша кількість поворотів. Не простежуючи шлях, єдиний спосіб дізнатися, що нерозв'язний лабіринт — це постійне збільшення суми поворотів, хоча у розв'язуваних спіралеподібних лабіринтах сума поворотів також може досягати великих значень [2].

Генетичний алгоритм – це евристичний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. «Батьком-засновником» генетичних алгоритмів вважається Джон Голланд, книга якого «Адаптація в природних і штучних системах» є фундаментальною у цій сфері досліджень[3].

Доцільність застосування генетичного алгоритму полягає в тому, що даний алгоритм, як правило, "помиляється" не більше ніж на 5–10%, 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

Проведено аналіз існуючих алгоритмів пошуку виходу з лабіринту. Досліджено переваги використання генетичного алгоритму при пошуку виходу з лабіринту. Доведено ефективність його застосування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Riccardo Poli, William B. Langdon, Nicholas Freitag McPhee, A Field Guide to Genetic Programming Paperback – March 26, 2008.
2. Maze Classification [Електронний ресурс] – 2021. – Режим доступу до ресурсу: https://www.astrolog.org/labyrnth/algrithm.htm.
3. Генетичний алгоритм [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/ Генетичний алгоритм.

Астаф'єв Вадим Віталійович **–** студент групи 2КН-18б, факультету інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: stud.vadim.astafiev@vntu.edu.ua

Белзецький Руслан Станіславович – к. т. н., старший викладач кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

Vadym Astafiev V. **–** Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: stud.vadim.astafiev@vntu.edu.ua

Ruslan Belzetskii S. – Ph. D., Assistant Professor of Department of Computer Science, Vinnytsia National Technical University, Vinnytsia,