УДК 518.14

ВИКОРИСТАННЯ АЛГОРИТМІВ КЛІТИННИХ АВТОМАТІВ В ЗАДАЧАХ ПОШУКУ МІНІМАЛЬНОГО ШЛЯХУ

Ванін В.В., д.т.н.,
vaninvladimir30@gmail.com, ORCID: 0000-0001-7008-7269
Залевська О.В., к.т.н.,
o.zalevska@kpi.ua, ORCID: 0000-0002-3163-1695
Захаркін М.С.,
zaharmisha70@gmail.com, ORCID: 0000-0003-4609-8130
Куйбіда П.К.,
pavel.kuybida@gmail.com, ORCID: 0000-0002-4767-3937
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (Україна)
Zhu Shiwei, ORCID: 0000-0002-2875-0706
zhusw@sdas.org,
Information Research Institute, Qilu University of Technology (Shandong Academy of Sciences), (Jinan, China)

У роботі розглядається використання клітинних автоматів для удосконалення пошуку найкоротшого шляху між об'єктами, на шляху до яких є перепони. В запропонованому алгоритмі, за основу взято клітинний автомат, а саме гра «Життя). Правила побудови структури клітинного автомату вимагають змін, в залежності від вимог, що стоять перед користувачем. В статті наведений можливий варіант пошуку найкоротшого шляху за допомогою клітинного автомату на прикладі гри. Метою гри є дістатись певної наперед заданої клітинки ігрового поля за мінімальну кількість кроків. Задача має багато аналогів та використовується в різних сферах науки від звичайних ігр до закритих військових стратегій.

Розглянуті існуючі аналоги алгоритмів пошуку найкоротшого шляху володіють рядом недоліків. До них можна віднести роботу лише з структурою графів, знаходження найкоротшого шляху тільки з однієї вершини та інші. Також існує недолік, що притаманний всім методам - вони завжди намагаються просуватися в напрямку мети, навіть якщо це неправильний шлях. Ці недоліки приводять до збільшення часу реалізації алгоритму, застрягання персонажів в кутках та між перепонами.

В наведеному алгоритмі поле розглядається як елемент поля клітинного автомату. На першому кроці випадковим чином генеруються перешкоди, що не зникають протягом всієї гри. Визначаємо клітину, як об'єкт який рухається по певному набору правил, що залежать від заповнення сусідніх клітин. Якщо на даній позиції є перешкода руху, то в залежності від того який об'єм займає фігура присвоюємо значення 1 або

0 в характеристичній матриці відповідної розмірності. Правила руху визначаються наявністю перешкоди, на скільки сильно ця перешкода заповнює простір (чи є можливість рухатись далі по клітинці, чи необхідно здійснювати обхід перешкод).

При наведеному алгоритмі зникають наведені недолі алгоритму, а швидкість виконання значно зменшується.

Ключові слова: мінімальний шлях, клітинний автомат, правила побудови клітинних автоматів, алгоритми пошуку мінімальної відстані.

Постановка проблеми. Визначення оптимального маршруту між об’єктами може здійснюватися в статичному, динамічному режимах і режимах реального часу. Це питання розглядається в багатьох важливих програмах у сфері GPS, відеоігор, робототехніки, логістики та симуляції натовпу - часові середовища. Проблема пошуку шляху може мати багато різних форм, включаючи ті, що стосуються одного агенту, групи агентів, конкурентного пошуку, динамічних змін навколишнього середовища, неоднорідної місцевості, мобільних пристроїв і неповна інформація. Алгоритм пошуку шляху та створення графіка для його реалізації є двома основними компонентами пошуку рушення проблеми.

Аналіз останніх досліджень та публікацій. Розглянемо існуючі алгоритми знаходження мінімального шляху між об’єктами. Алгоритм Дейкстри — це метод визначення найкоротшого маршруту між вершинами графа[11]. З початкової точки об'єкта починається робота. кожного разу перевіряє найближчу недосліджену вершину, додаючи найближчі вершини до списку вершин, які слід враховувати. Якщо на графі немає ребер із від’ємними значеннями, цей алгоритм завжди визначає найкоротший маршрут між початковою точкою та бажаним пунктом призначення [18].

Порівняльна простота реалізації алгоритму Дейкстри, низька поліноміальна обчислювальна складність і потенціал для машинної реалізації є його перевагами.

До недоліків алгоритму можна віднести наступне:

- працює тільки зі структурами графів, які мають невід'ємні довжини дуг; - шукає найкоротші шляхи та їх значення лише з однієї вершини;

- щоб обчислити ТК між усіма парами вершин, необхідно повторити процес для кожної пари вершин у вихідному графі N разів;

- не шукає рефлексивних замикань;

- має досить складне програмне представлення

Алгоритм Беллмана-Форда має часову складність O (V cdot E), де V – кількість вершин у графі, а E – кількість ребер [19]. Виконується V кроків, що є причиною цієї складності. Дальше проходимо всі ребра графа на кожній фазі. Ключовою перевагою алгоритму Беллмана-Форда є його здатність працювати з негативними вагами. Алгоритм Беллмана-Форда набагато складніше зрозуміти, ніж метод Дейкстри. Через те, що графіки з від’ємними вагами зустрічаються відносно рідко, підхід Дейкстри має

більше застосувань.

Алгоритм Беллмана-Форда може обробляти орієнтовані та неорієнтовані графи з невід'ємними вагами, як було зазначено раніше. Однак, якщо негативних циклів немає, він може обробляти лише орієнтовані графи з негативними вагами [19].

Формулювання цілей статті. Проаналізувати специфіку роботи алгоритмів пошуку найкоротшого шляху, розглянути засоби реалізації програмної системи, описати та розробити основні функції програмної реалізації в якій буде застосовуватися обраний алгоритм.

Основна частина. Визначення маршруту може здійснюватися в статичному, динамічному режимах і режимах реального часу, і воно є основною частиною багатьох важливих програм у сферах GPS, відеоігор, робототехніки, логістики та симуляції натовпу - часові середовища. За останні 20 років численні інновації підвищили точність і ефективність методів пошуку шляху, але це питання продовжує залишатися в центрі інтенсивних досліджень.

Проблема пошуку шляху може мати багато різних форм, включаючи ті, що стосуються одного агенту, групи агентів, конкурентного пошуку, динамічних змін навколишнього середовища, неоднорідної місцевості, мобільних пристроїв і неповна інформація. Кожне з цих питань має унікальне застосування в багатьох сферах. Алгоритм пошуку оптимального шляху та створення графічної інтерпретації є двома основними компонентами пошуку шляху.

Цей метод визначає клітину автомату як рухомий об'єкт, який відповідає набору правил, які залежать від заповнення сусідніх клітин. Якщо в цьому місці є перешкода для руху, то значенню клітини присвоюється значення об'єму який займає перешкода. Дані заносяться до характеристичної матриці. Характеристична матриця повністю визначає рух персонажа.

Наведемо діаграму прецедентів та класів застосунку, що реалізують наведений алгоритм (рис.1 та рис.2).

Рис. 1. Діаграма прецедентів

Перш за все користувач має можливість згенерувати поле, перейти до наступного кроку і знайти найкоротший шлях.

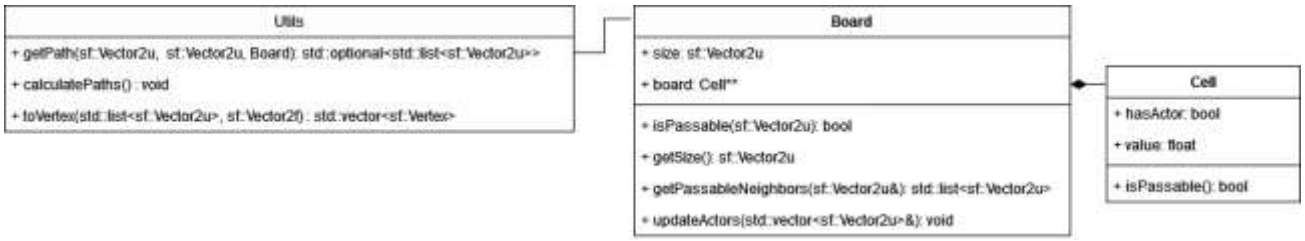

Рис. 2. Діаграма класів

Отже, застосунок буде складатись з таких модулів як:

- клас Board — відповідає за сітку клітинного автомату;
- клас Cell — відповідно за клітинку;
- глобальна функція getPath (на діаграмі - в блоці Utils) – реалізація алгоритму A* для сітки клітин, використовує вкладену структуру node, що використовується в алгоритмі для відвіданих та не відвіданих комірок, для оцінки евристики використовується Евклідова відстань.

За допомогою запропонованого методу можна розв’язувати задачі, використовуючи апарат клітинних автоматів для знаходження найкращого рішення, а не лише мінімального значення.

Розглянемо приклад використання пошуку оптимального алгоритму у клітинних автоматах та опис різних ситуацій, в яких знаходиться початкова координата. На рис. 3 показано ітераційні дані, які були зроблені для першого випадку, а у рис. 4 – другі та на рис. 5 вказано знайдений шлях.

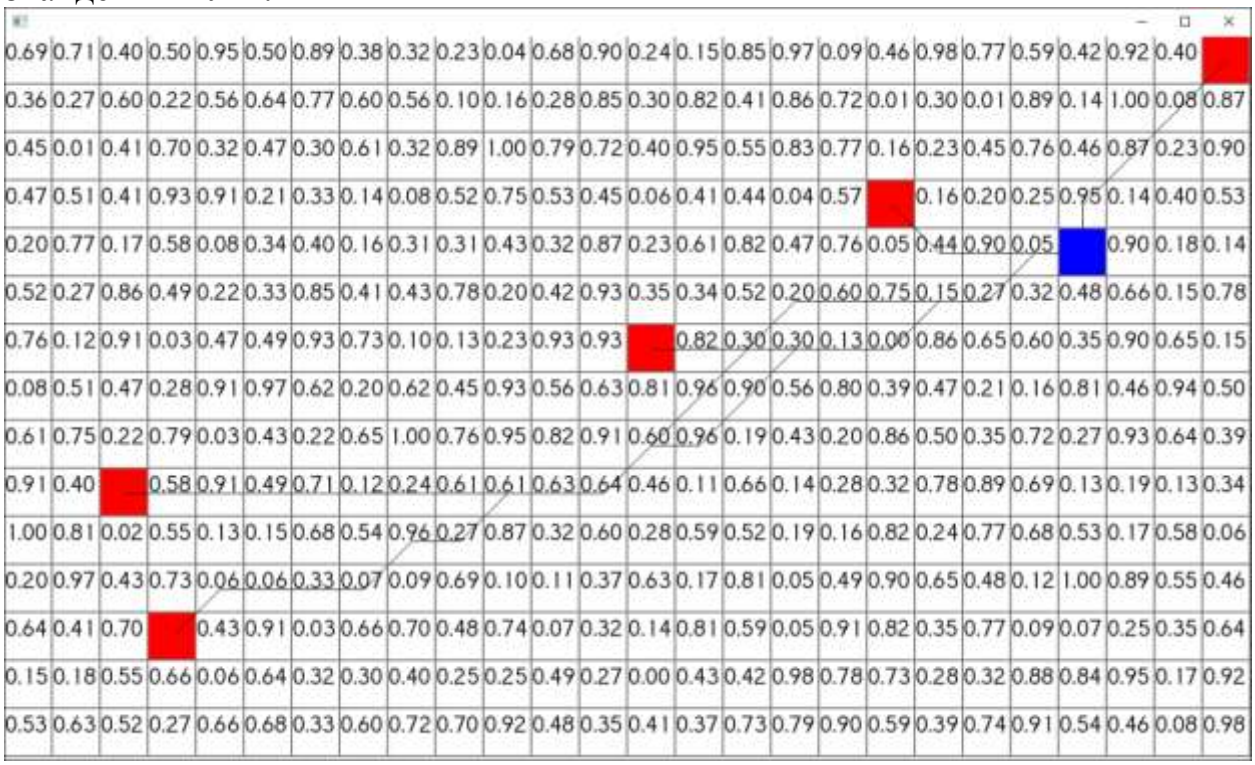

Рис. 3. Початок роботи програми. Випадок 1.

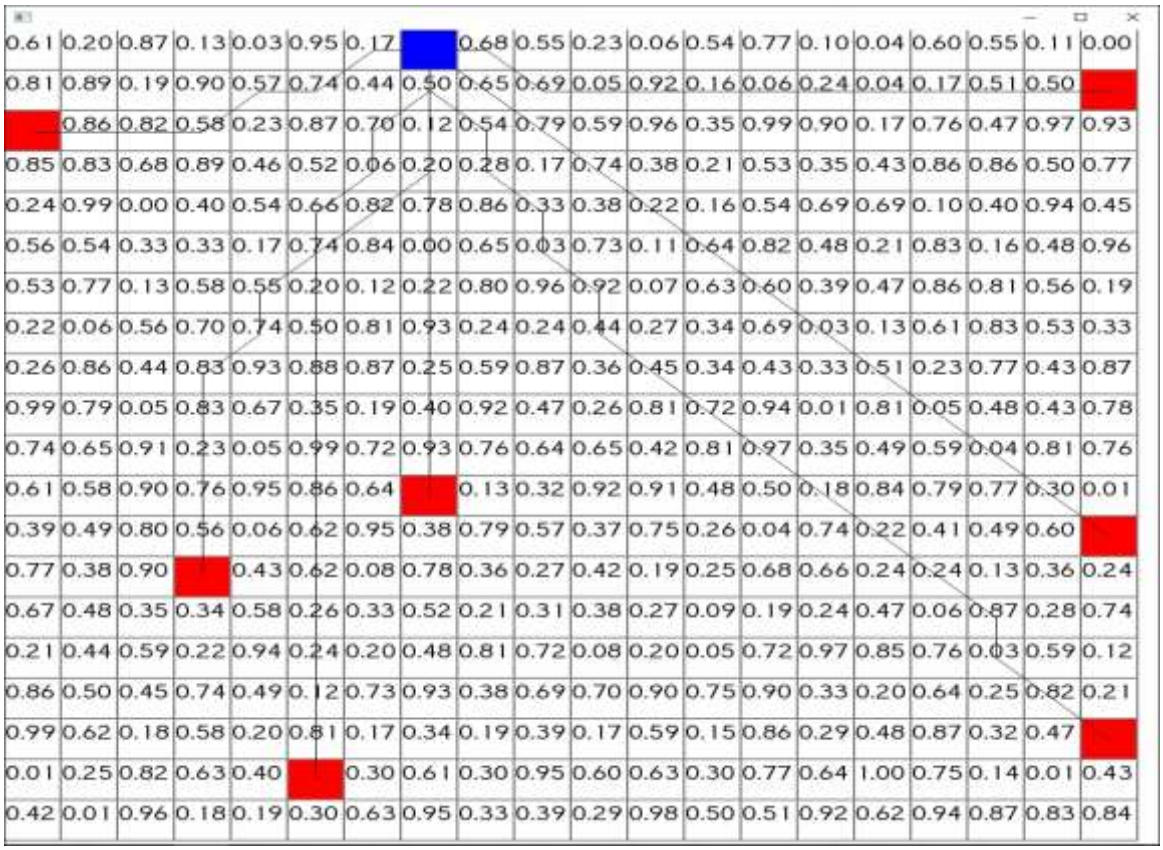

Рис. 4. Перша ітерація реалізації другого випадку побудованого алгоритму

Рис. 5. Результат роботи програми. Випадок 2.

Висновки. Запропонований спосіб знаходження оптимального мінімального шляху з перешкодами між двох об'єктів є швидший за існуючі аналоги та позбавлений таких недоліків як робота тільки зі структурами графів, які мають невід'ємні довжини дуг, пошук найкоротшого шляху та значення лише з однієї його вершини. Встановлені правила побудови клітинного автомату дають можливість не лише знаходження шляху між вершинами, а й дослідити автомат на оптимальність знайденого шляху.

Література

1. Shimbel A. Structure in communication nets. Proceedings of the Symposium on Information Networks. 1955.
2. O. Zalevska *et al*., "Construction and Study of the Mathematical Model for the System Using Three-Dimensional Cellular Automata,"(2021) *2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, 49-52, doi: 10.1109/CADSM52681.2021.9385235.
3. Ванін, В. В., Залевська, О. В., & Чередніченко, В. О. Структура тривимірного клітинного автомату для побудови зображення динамічних систем. *Сучасні проблеми моделювання*, 2019. Вип. 15, С.43-50. https://doi.org/10.33842/2313-125X/2019/15/43/150.
4. Залевська О., Фіногенов О., Ібнухсейн І., Суворова В. Використання засобів онлайн технологій для побудови тривимірних клітинних автоматів. *Сучасні проблеми моделювання*, 2021. Вип. 22, с. 39-47. https://doi.org/10.33842/22195203/2021/22/39/47
5. "Experiments with the Graph Traverser program" by Doran, J. E.; Michie, D. 1966-09-20. ISSN 0080-4630. S2CID 21698093.
6. The Quest for Artificial Intelligence by Nilsson, Nils J., 2009-10-30, ISBN 9780521122931.
7. A* Search Algorithm. Режим доступу: https://www.geeksforgeeks.org/a-search-algorithm/.
8. Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction. Режим доступу: https://www.freecodecamp.org/news/ dijkstras-shortest-path-algorithm-visual-introduction/.
9. M. N. S. Swamy, K. Thulasiraman Graphs, *Networks, and Algorithms.* Wiley, 1981p. ISBN-13 978-0471035039.
10. Labeling Algorithm for Shortest Paths on Road Networks. / [Abraham I., Delling D., Goldberg A., Werneck R.]. Philadelphia. *Symposium on Experimental Algorithms*, 2011. pp. 230-241.
11. Dijkstra E. A note on two problems in connexion with graphs. *In.: Numerische Mathematik*. 1959. V. 1. pp. 269-271.
12. Ford L, Fulkerson D. Flows in Networks. Princeton: Princeton University Press, 1962. 253 p.
13. Bellman-Ford Algorithm Visually Explained. Режим доступу:

https://blog.devgenius.io/bellman-ford-algorithm-visually-explained-e940b6edb00a.
14. A. Makarenko, B. Goldengorin and D. Krushinsky, "Game ‘Life’ with Anticipation Property", *Cellular Automata. ACRI 2008. Lecture Notes in Computer Science*, vol. 5191, 2008, [online] Available: https://doi.org/10.1007/978–3-540-79992-4_10

USE OF CELLULAR AUTOMATA ALGORITHMS IN MINIMUM PATH SEARCH PROBLEMS

Volodymyr Vanin, Olga Zalevska, Mikhailo Zakharkin, Pavlo Kuibida
Zhu Shiwei

The paper considers the use of cellular automata to improve the search for the shortest path between objects with obstacles on the way. The proposed algorithm is based on a cellular automaton, namely the game "Life). The rules for constructing the structure of the cellular automaton require changes, depending on the requirements of the user. The article presents a possible variant of finding the shortest path with the help of a cellular automaton using the example of a game. The aim of the game is to reach a certain predetermined cell of the playing field for a minimum number of steps.The problem has many analogues and is used in various fields of science from ordinary games to fixed military strategies.

The considered existing analogues of shortest path search agglomerations have a number of drawbacks. These include working only with the structure of graphs, finding the shortest path from only one vertex and others. There is also a drawback that is inherent in all methods - they always try to move in the direction of the goal, even if it is the wrong path. These disadvantages lead to an increase in the time of implementation of the algorithm, stuck characters in the corners and between obstacles.

In the above algorithm, the field is considered as a field element of a cellular automaton. In the first step, obstacles are randomly generated, which do not appear throughout the game. We define a cell as an object that moves according to a certain set of rules that depend on the filling of neighboring cells. If there is an obstacle at a given position, then depending on the volume occupied by the figure, we assign a value of 1 or 0 in the characteristic matrix of the corresponding dimension. The rules of movement are determined by the presence of an obstacle, how much this obstacle fills the space (is it possible to move further along the cell, or is it necessary to bypass the obstacles).

With this algorithm, the above disadvantages of the algorithm disappear, and the speed of execution is significantly reduced.

Keywords: minimum path, cellular automaton, rules for constructing cellular automata, algorithms for finding the minimum distance.

References

1. Structure in communication nets. Proceedings of the Symposium on Information Networks (1955), Shimbel A.
2. O. Zalevska et al., "Construction and Study of the Mathematical Model for the System Using Three-Dimensional Cellular Automata,"(2021) IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), pp. 49-52, doi: 10.1109/CADSM52681.2021.9385235.
3. Vanin, V. V., Zalevska, O. V., & Cherednichenko, V. O. (2019). The structure of a three-dimensional cellular automaton for image construction of dynamical systems. *Modern problems of modeling*, (15), 43-50. https://doi.org/10.33842/2313-125X/2019/15/43/150 [in Ukrainian]
4. Zalevska, O., Finogenov, O., Ibnukhsein, I., & Suvorova, V. (2021). The use of online technologies for the construction of three-dimensional cellular automata. *Modern problems of modeling*, (22), 39-47. https://doi.org/10.33842/22195203/2021/22/39/47 [in Ukrainian]
5. "Experiments with the Graph Traverser program" by Doran, J. E.; Michie, D. 1966-09-20. ISSN 0080-4630. S2CID 21698093.
6. The Quest for Artificial Intelligence by Nilsson, Nils J., 2009-10-30, ISBN 9780521122931.
7. A* Search Algorithm [Electronic resource] - Access mode to the resource: https://www.geeksforgeeks.org/a-search-algorithm/.
8. Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction [Electronic resource] - Access mode to the resource: https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/ (accessed 28.11.2022).
9. M. N. S. Swamy, K. Thulasiraman Graphs, Networks, and Algorithms - Wiley, 1981. ISBN-13 978-0471035039.
10. Labeling Algorithm for Shortest Paths on Road Networks (2011) / [Abraham I., Delling D., Goldberg A., Werneck R.]. Philadelphia. *Symposium on Experimental Algorithms*, 230-241.
11. Dijkstra E. A note on two problems in connection with graphs: Numerische Mathematik (1959). V. 1. pp. 269-271.
12. Ford L. Flows in Networks / Ford L., Fulkerson D. - *Princeton: Princeton University Press*, 1962. 253 p.
13. Bellman-Ford Algorithm Visually Explained. Date of update: 08.07.2020 - URL: https://blog.devgenius.io/bellman-ford-algorithm-visually-explained-e940b6edb00a (accessed 28.11.2022).
14. A. Makarenko, B. Goldengorin and D. Krushinsky, "Game ‘Life’ with Anticipation Property", *Cellular Automata. ACRI 2008. Lecture Notes in Computer Science*, vol. 5191, 2008, [online] Available: https://doi.org/10.1007/978–3-540-79992-4_10