The Comparison of Steepest Ascent Hill Climbing and A-star for Classic Game

Wildan Budiawan Zulfikar
Department of Informatics
UIN Sunan Gunung Djati
Bandung, Indonesia
wildan.b@uinsgd.ac.id

Mohamad Irfan
Department of Informatics
UIN Sunan Gunung Djati
Bandung, Indonesia
irfan.bahaf@uinsgd.ac.id

Rijal Muharom Yahya
Department of Informatics
UIN Sunan Gunung Djati
Bandung, Indonesia
rijal.myahya@gmail.com

Diena Rauda Ramdania
Department of Informatics
UIN Sunan Gunung Djati
Bandung, Indonesia
diena.rauda@uinsgd.ac.id

Jumadi
Department of Informatics
UIN Sunan Gunung Djati
Bandung, Indonesia
jumadi@uinsgd.ac.id

***Abstract*—Tic-tac-toe is a classic game that remains popular today. This game requires two players. When applied to software, of course this game requires an AI-integrated bot as an opponent to play. Several algorithms can be used as artificial intelligence in tic-tac-toe games such as Steepest Ascent Hill Climbing and A-star. The purpose of this research is to find a more effective and efficient algorithm for this game. The components used to compare the two algorithms are the elapsed time and also the performance of the two algorithms. SAHC recorded a shorter elapsed time with 0.0448 ms compared to the A-star which recorded an average of 1.192 ms. However, based on other tests case, A-star performs better than SAHC.**

Keywords—tic tac toe, steepest ascent, hill climb, a-star

I. INTRODUCTION

Games are usually played by one or more players against other players, in the game the opponents of the players are not only humans, but players can also play against machines commonly referred to as bots or AI. Each bot or AI can have a different difficulty level. With AI, players can play alone against machines that are made to play like humans. In order for players to feel like playing against human players, bots or AI in their manufacture need an algorithm that can make the best decisions to beat or dispel so that human players don't win the game [1]–[3].

One of the classic games that can apply an AI is a simple 3x3 square board game that uses the X or O player symbol called the tic-tac-toe game. In the tic-tac-toe game, it is very easy to determine the size of wins and losses, because in this game players only have to fill the available game board with player symbols X or O by forming a situation in the form of a horizontal, vertical, or diagonal line to get it victory, so it is very possible that AI can compare its abilities with human abilities [4]–[6].

This game has become an object that is often used for research, tic-tac-toe itself has been implemented with various algorithms to serve as opponents or AI in the game. Of the many algorithms that have been applied to the tic-tac-toe game itself, many studies say that the algorithm used is more efficient, but does not provide clear data about the algorithm's efficiency with the tic-tac-toe game. The game is generally limited to a 3x3 square board, there are also larger sizes such as 5x5, 6x6, to NxN [7]. In addition, the game of tic-tac-toe has been limited to one game board. Tic-tac-toe is a simple game that is very popular but the game cannot be played by a single player, so the most likely thing is to create a bot that can replace the human role as the opponent of the player [2], [4], [6].

One of the algorithms that can be used in making AI is the A-star algorithm. By applying a heuristic, the A-star algorithm will remove steps that will not achieve the best-expected decision. In solving the problem, the A-star algorithm applies a heuristic technique, which is an assessment that will give a value or weight to each node that will direct the A-star to get the desired solution [8]–[13]. A-star will get the solution you are looking for if the heuristic is done correctly [14]–[17]. The heuristic itself is still an estimate or an ordinary guess, so it doesn't use any special formulas in it at all. So that in each case that has a heuristic, it must be different [11], [15], [18]–[21].

The algorithm that applies other heuristic functions is the SAHC (Steepest Ascent Hill Climbing) algorithm. It is a development of the Simple Hill-Climbing algorithm, Simple Hill Climbing determines the next condition by comparing the current condition with one of its successors and the better first successor will be the next, while SAHC will compare the current state with all the successors that are nearby, where the best successor or closest to the best solution will be the next condition [22]–[25]. The purpose of this study was to compare SAHC and A-star in the tic-tac-toe game. The two will be compared, several things that become components of the assessment, namely performance and speed of the process.

978-1-6654-4402-6/21/$31.00 ©2021 IEEE

II. METHODOLOGY

Inspired by a traditional game that was quite popular in its time, however, at this time the game has increased towards the digitalization era. Where every activity can be done with the help of technology, and it cannot be denied that games have also increased into digital games or PC games. This game is played on a board in the shape of a box or square, which is represented in a matrix (cell) with one cell indicating the coordinates or where the player will place the symbols. The tic-tac-toe game board in this application using a board measuring 3x3 squares by 5x5 squares. The boxes will be filled with symbols or images that represent each player, the maximum player is two players who play in turns. The player who is faster to form a line (horizontal, diagonal, or vertical) with a symbol or image representing it will win.

In the design of the game system, artificial intelligence or AI will be designed which will be the opponent to play in this game. The CPU that is made will have a turn after the player selects the box on the game board. When the CPU's turn arrives at the terminal starting the system process to work, the CPU will look for the best path to win the game or make the situation a draw. In finding the best path, the CPU will create a search tree where each tree will be calculated the best value that will be the final goal, namely victory or draw. After getting the best flow, the CPU will select the box which is the beginning of the best flow. After the CPU's turn is over, the player's turn to select the box on the game board again, then the CPU's turn arrives again and will start the search for the best path with the condition of the game board when the CPU's turn arrives. Therefore, bots need a search system that uses the appropriate algorithm to become an artificial intelligence or AI. This study uses Steepest Ascent Hill Climbing (SAHC) with A-Star and then compares which one is better between the two.

A. SAHC

This algorithm is a continuation of the Simple Hill-Climbing algorithm, in which the search path will be centered on the left side of the new search tree. It will find the best groove by using the result of the value of each box on the game board that forms the winning state.

Testing will begin by providing a situation on the board where the player will select a box in the upper left corner of the board. The player will use the symbol X while the algorithm will use the symbol O as well as the initial on the quest will be given a value of 99, which can be seen in fig 1.

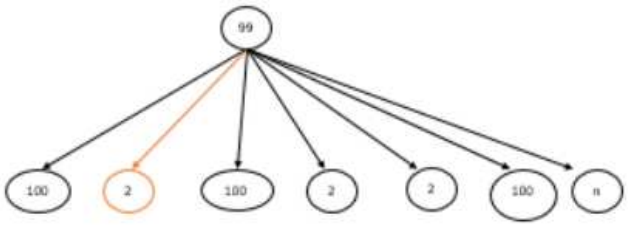


Fig. 1. First turn by SAHC

In fig 1, the search is carried out based on the path (win condition) that exists in the selected empty box. On the turn of the computer, it can be seen that the first selected value is the path that has a value of 2, but on the first turn of the computer several paths have the same value and the steepest ascent hill-climbing algorithm will choose the first best value then continue the search first until there are no boxes blank on the board.

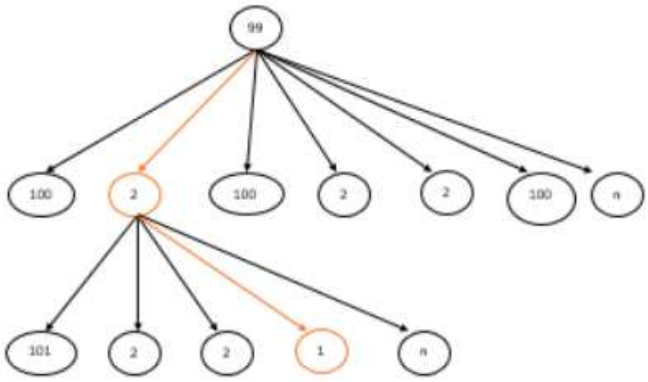


Fig. 2. Second turn by SAHC

In the second turn made by playing the selected path, the value is 1, this is obtained because at each turn the value given will be according to the existing rules where if there is an opponent's symbol on the path it will be given a value of 99, but in fig 2 the player gets a value of 1 because on the path that is given this value there are 2 player symbols, therefore the value obtained is 1 from the reduction of 3 boxes on the path minus the number of player symbols on that path.

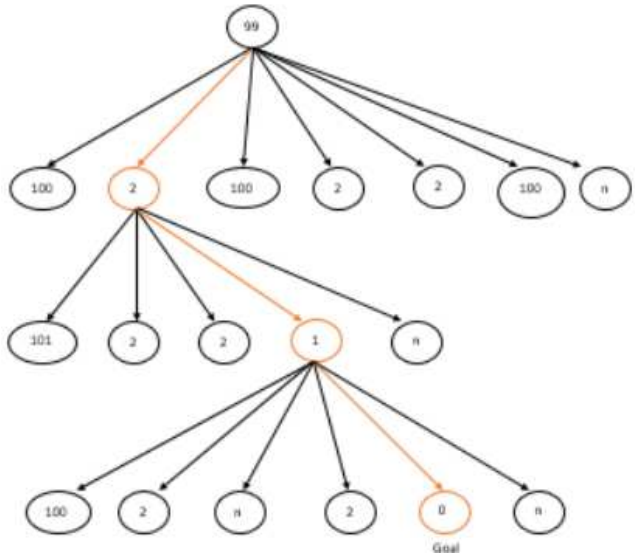


Fig. 3. Third turn by SAHC

In the third step, fig 3 explains that the value chosen is a value of 0. This value is obtained if the path with that value gets a winning state or if the path contains a symbol of the opponent who will win and will automatically be given a value of 0 to avoid the opponent winning or if the player who gets the search turn wins as in Fig 3.

B. A-star (A-Star)

Same as the previous test, the algorithm will accept the game board with the condition that the player selects the box in the upper left corner of the board and gives an initial value

of 99. In fig 4, the total depth of the search tree is added to the difference between the player's possible wins minus the computer's possible winnings, and a value of 3 is obtained as the first smallest value.

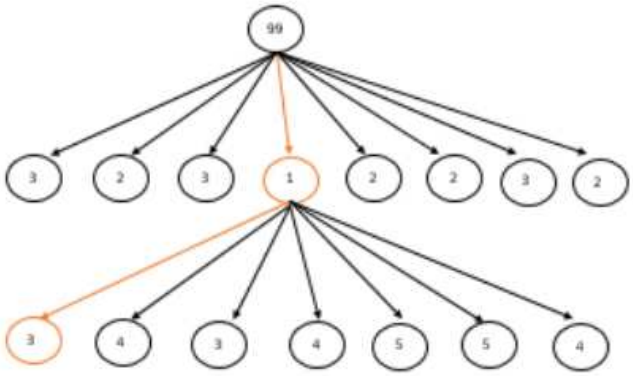


Fig. 4. Second turn by A-star

Fig 5 explains that the value is chosen is the same as the previously selected value and in this turn, not only one value is the same, but the A-Star algorithm will choose the smallest value first because in that turn the value sought is the smallest value in the series of values in the series. that turn. Next, the value chosen is 5, even though the value is greater than the previous value, the value 5 is the smallest value in the series of values that exist in this fourth turn.

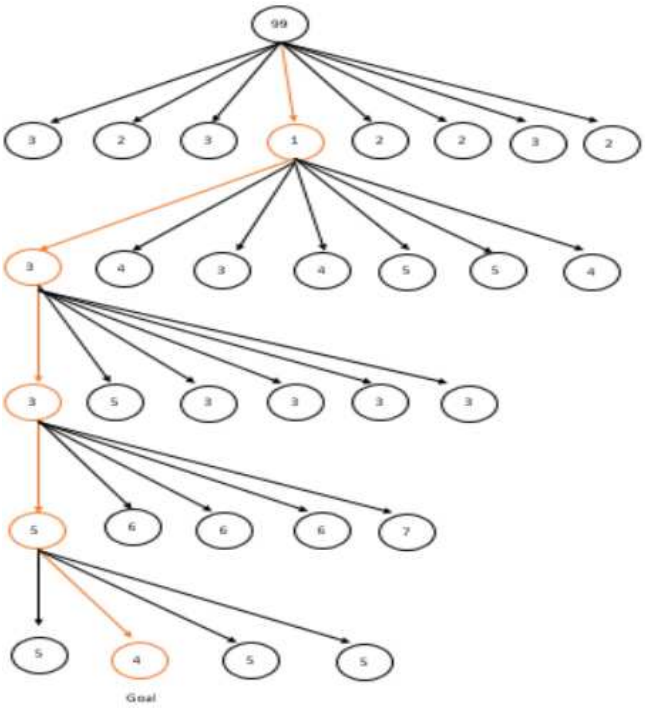


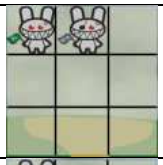
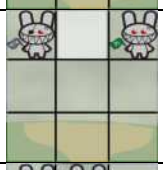
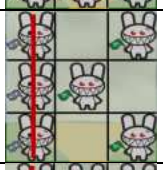
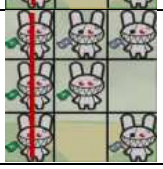
Fig. 5. Full turn by A-star

Fig 5 also explains that after that the value chosen is 4 which is the goal of the search. It will give a small value if the opponent will win and a smaller value if he will win or if the value obtained is the smallest value from a row of search turns then that value will be chosen.

III. Result and Discussion

In this study, five experiments were carried out on both algorithms with the same prefix pattern or first choice. From these five game trials, the time taken by the algorithm will also be calculated in making decisions to select empty boxes in turn. The time taken will be compared to see which of the two algorithms is more efficient in terms of the algorithm's travel time and to see which empty box option is selected as the final decision.

TABLE I. SAHC Test Result

Initial State	Final State	Result	Average Time (ms)
		Draw	0.107
		Draw	0.031
		Draw	0.031
		Lose	0.029
		Win	0.026

Tables 1 and 2 show the results of experiments with almost the same prefix, the two algorithms select different empty boxes in the initial conditions and have different final states. This happens because the two algorithms have differences in determining the value to look for, the SAHC algorithm uses the path (winning state) as a reference for determining the value, while the A-star algorithm uses the entire game board to determine the value.

TABLE II. A-star Test Result

Initial State	Final State	Result	Average Time
		Draw	2,757

		Draw	0,817
		Draw	0,634
		Draw	0,956
		Lose	0,797

This can also result in the player winning against the computer where the SAHC algorithm the player can win one game, while the A-star algorithm is a little difficult for the player to win the game. At the average time, the computer in the game proves that the SAHC algorithm with an average time of 0.0448 ms faster than the A-star algorithm which has an average time of 1.192 ms. In this case, it can be made clear that although the SAHC algorithm is more efficient than the A-star algorithm in terms of the algorithm's travel time, it is possible that the SAHC algorithm can be defeated by more players than the probability that the A-star algorithm can be defeated even with the least chance of defeat.

From the results that have been obtained, there are various kinds of certain factors that can affect the speed and choice of the algorithm selected, including how to determine the value to be used in the search algorithm, taking the path used in the search. and the specifications of the device used to run the game application.

Based on the results of tests carried out on the system that has been carried out, it can be concluded that the SAHC algorithm and the A-star algorithm that are applied to the tic-tac-toe game application both algorithms can be applied well to the game so that the game can be played. The black box testing that has been done also gets results where all the functions in the game run properly so that the system functions according to the previously designed design. At the beta testing stage, the responses from users who have played the tic-tac-toe game using the SAHC algorithm and the A-star algorithm are very diverse, the results of the survey that have been conducted also get a percentage value of 78% of the user's understanding of the application and 82% usability of the application.

TABLE III. SAHC AND A-star SUMMARY RESULT

No.	Object	SAHC	A-star
1	Elapsed Time	0.045	1.192
2	Win Possibility	High	Low

Table 3 explains that the SAHC algorithm is faster than the A-star algorithm, but the probability of the player winning against the computer that is implemented with the SAHC algorithm is more than the computer that is implemented with the A-star algorithm as in the table above.

IV. CONCLUSION

Based on the test results that have been done, it has been proven that the SAHC algorithm is faster than the A-star algorithm with an average time of about 0.045 ms for the SAHC algorithm and 1.192 ms for the A-star algorithm, but the A-star algorithm is less likely to be won by the player than on the SAHC algorithm. The reason is, there are several causal factors that can affect the performance of the algorithm, namely in the form of techniques in determining the value to be used for search by the algorithm, because each algorithm has its own formula or way to determine the value to be searched to determine the search flow, other factors that can be used for searching. affect the performance of the algorithm is the specification of the software as well as the hardware used to run both in this case.

The research that has been carried out has the novelty of several previous studies that have been carried out. Where the addition of game options in the form of a player against player in this study that has been made, is a game option that does not exist in some previous studies.

REFERENCES

[1] A. Matallaoui, N. Hanner, and R. Zarnekow, "Introduction to Gamification: Foundation and Underlying Theories," 2017, pp. 3–18.

[2] J. Beck, "Arithmetic progressions and tic-tac-toe games," *Dev. Math.*, vol. 16, pp. 41–93, 2008.

[3] D. Curran and C. O'Riordan, "Cultural evolution for sequential decision tasks: Evolving tic-tac-toe players in multi-agent systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3102, pp. 72–80, 2004.

[4] B. Sandlund, K. Staley, M. Dixon, and S. Butler, "Numerical Tic-Tac-Toe on the 4×4 board," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8591 LNCS, pp. 537–546.

[5] S. Isard and C. Longuet-Higgins, "Modal Tic-Tac-Toe," in *Logic, Language, and Probability*, Springer Netherlands, 1973, pp. 189–195.

[6] M. Pontier and G. F. Siddiqui, "An affective agent playing Tic-Tac-Toe as part of a healing environment," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5925 LNAI, pp. 33–47.

[7] J. Hunt, "TicTacToe Game," Springer, Cham, 2019, pp. 423–433.

[8] C. Ju, Q. Luo, and X. Yan, "Path Planning Using an Improved A-star Algorithm," in *Proceedings - 11th International Conference on Prognostics and System Health Management, PHM-Jinan 2020*, 2020, pp. 23–26.

[9] R. Yang and L. Cheng, "Path Planning of Restaurant Service Robot Based on A-star Algorithms with Updated Weights," in *Proceedings - 2019 12th International Symposium on Computational Intelligence and Design, ISCID 2019*, 2019, vol. 1, pp. 292–295.

[10] P. O. N. Saian, Suyoto, and Pranowo, "Optimized A-Star algorithm in hexagon-based environment using parallel bidirectional search," in *Proceedings of 2016 8th International Conference on Information Technology and Electrical Engineering: Empowering Technology for Better Future, ICITEE 2016*, 2017.

[11] Y. Li, D. Dong, and X. Guo, "Mobile robot path planning based on improved genetic algorithm with A-star heuristic method," in *ITAIC 2020 - IEEE 9th Joint International Information Technology and Artificial Intelligence Conference*, 2020, pp. 1306–1311.

[12] I. Wojnicki, S. Ernst, and W. Turek, "A robust heuristic for the multidimensional A-star/wavefront hybrid planning algorithm," in

Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), 2015, vol. 9120, pp. 282–291.

[13] K. Rana and M. Zaveri, "A-star algorithm for energy efficient routing in wireless sensor network," in *Communications in Computer and Information Science*, 2011, vol. 197 CCIS, pp. 232–241.

[14] Y. Syukriah, F. Falahah, and H. Solihin, "Penerapan Algoritma a* (star) untuk Mencari Rute Tercepat dengan Hambatan," in *Seminar Nasional Telekomunikasi dan Informatika (SELISIK)*, 2016, no. 1, pp. 219–224.

[15] X. Liu and D. Gong, "A comparative study of A-star algorithms for search and rescue in perfect maze," in *2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings*, 2011, pp. 24–27.

[16] J. A. Aslam, E. Pelekhov, and D. Rus, "The star clustering algorithm for information organization," in *Grouping Multidimensional Data: Recent Advances in Clustering*, Springer Berlin Heidelberg, 2006, pp. 1–23.

[17] Y. Zhang, L. ling Li, H. C. Lin, Z. Ma, and J. Zhao, "Development of path planning approach based on improved A-star algorithm in AGV system," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2018, vol. 246, pp. 276–279.

[18] R. Septiana, I. Soesanti, and N. A. Setiawan, "Evaluation function effectiveness in Wireless Sensor Network routing using A-star algorithm," in *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*, 2016.

[19] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-star Algorithm: An improved A-star Algorithm for AGV Path Planning in a Port Environment," *IEEE Access*, pp. 1–1, 2021.

[20] N. K. Kang, H. J. Son, and S. H. Lee, "Modified A-star algorithm for modular plant land transportation," *J. Mech. Sci. Technol.*, vol. 32, no. 12, pp. 5563–5571, Dec. 2018.

[21] Y. Cai and X. Ji, "ASA-routing: A-Star adaptive routing algorithm for network-on-chips," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11335 LNCS, pp. 187–198.

[22] Y. Sazaki, H. Satria, A. Primanita, and R. Apriliansyah, "Application of the Steepest Ascent Hill Climbing Algorithm in the Preparation of the Crossword Puzzle Board," in *Proceeding of 2018 4th International Conference on Wireless and Telematics, ICWT 2018*, 2018.

[23] Y. Sazaki, A. Primanita, H. Satria, and R. Apriliansyah, "A comparison application of the genetic and steepest ascent hill climbing algorithm in the preparation of the crossword puzzle board," in *Proceeding of 2018 12th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2018*, 2018.

[24] J. Arriaga and M. Valenzuela-Rendón, "Steepest ascent hill climbing for portfolio selection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7248 LNCS, pp. 145–154.

[25] H. F. Mustika, L. P. Manik, A. F. Syafiandini, Z. Akbar, and Y. Rianto, "Searching Region of Interest from News Website using Steepest Ascent Hill Climbing Algorithm," in *2019 International Conference on Computer, Control, Informatics and its Applications: Emerging Trends in Big Data and Artificial Intelligence, IC3INA 2019*, 2019, pp. 120–124.