ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse42932020.pdf
https://doi.org/10.30534/ijatcse/2020/42932020

A Review on Informed Search Algorithms for Video Games Pathfinding

Azyan Yusra Kapi[1,2,3], Mohd Shahrizal Sunar[1,2], Muhamad Najib Zamri[1,2]
[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia
[2]Media and Game Innovation Centre of Excellence, Institute of Human Centred Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia
[3]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 81700 Pasir Gudang, Johor, Malaysia
Corresponding author: shahrizal@utm.my

ABSTRACT

Pathfinding is a broadly applied algorithm that involved the discovery of routes between two positions by avoiding obstacles at the same time. Recently, a significant number of researchers focusing on informed search algorithms for pathfinding concerning games. However, review regarding the latest optimization in the pathfinding algorithm and its advantages still lacks in the literature. To organize this heterogeneity, this paper presents a review that focused on numerous modifications to enhance the execution of the informed search algorithm through four classified perspectives: i) modification to the graph representation, ii) enhancement of heuristic function, iii) hybrid search algorithm, and iv) new data structure. This paper also aims to discuss common challenges faced by pathfinding in video games and providing future trends for optimization. While incorporating pathfinding optimization over the past decade, this paper also aims to assist new researchers by emphasizing the potential path for further exploration.

Key words : Games, heuristic, optimization, pathfinding.

1. INTRODUCTION

Pathfinding algorithm is said to be the core of numerous applications and has wide-ranging usage in everyday life. Over the years, pathfinding algorithms have received increasing attention in many applications such as video games, robotics, metabolic pathways [1], augmented reality [2], and global positioning system [3]. Pathfinding algorithm addressed the question in the discovery of the shortest route between two positions and avoided obstacles [4].

According to the study done by Newzoo [5] in 2019, video games generated USD 152.1 billion in the global games market. In video games, pathfinding represents a significant component to a character to find the shortest, fastest, and cheapest way possible to navigate from one place to another within the game environment. As the quest for practicality increases in video games, the problem of pathfinding is loaded with many added requirements and limitations aimed at developing more credible results for character navigation [6], thus causes navigation to become an important issue. Hence, the pathfinding algorithm plays a vital role in this billion-dollar industry. Besides, calculations in pathfinding can be very challenging, and this is where heuristics play important parts that provide supplementary data to the algorithm, which intends to accelerate performance. Pathfinding can be categorized to two major groups, namely informed and uninformed search [7]. Both search terms and heuristic function's definition are further described in Section 3.3 and Section 3.2, respectively.

An enormous volume of pathfinding algorithms with different characteristics and fundamental traits have been recommended and developed over the past years and observed to be a favourable navigation method for characters in games. Optimization can be defined as a procedure of deciding the greatest resolution to maximize or minimize the variables involved in the problems to achieve maximum performance and useful practicality [8]. Due to the broad area of research in search algorithms, this paper only focused on optimization for informed or heuristic search while an uninformed search or blind search is beyond the scope of this study. Although there are various review techniques to enhance existing pathfinding algorithms in previous literature, most of the review summarized algorithms in general view without emphasizing on its optimization perspectives. New researchers in this area ought to have a general understanding but it will remain insufficient in terms of reaching a new idea on how to further optimize the pathfinding techniques. A thorough search of the related literature yielded only one related article by Cazenave [9] in 2006. Cazenave explained related optimization done for pathfinding by three research perspectives: i) changed data structure, ii) heuristics, and iii) algorithm.

Besides, this paper organizes the optimization of informed search algorithms which classified in four research

perspectives by adding another research perspective from Cazenave as presented in Section 3. Furthermore, this study presents a review of common challenges for pathfinding in video games. Besides, this research complements an earlier study which reviews several optimization and variant of pathfinding algorithms such as [4], [10], [11], and [12]. The existence of periodic review on a regular basis becomes essential as a guideline to the new researchers in recognizing the improvement of pathfinding study and emphasize the possible research in the character's navigation using heuristic pathfinding techniques. The optimization techniques in pathfinding as a comparative and benchmarked is essential to seek a potential research trend. Therefore, this study also classifies the hidden research area of heuristic pathfinding optimization technique by presenting future trends of optimization for navigation in games.

2. CHALLENGES FOR PATHFINDING IN VIDEO GAMES

Pathfinding in the static environment poses fewer challenges than the dynamic environment, which the latter refers to the changeable position of obstacle [12]. In computer games, pathfinding is very important to allow non-player characters (NPCs) to move to specific target positions [13]. Although in many applications, the problem is restricted to locate the shortest path possible between points A and B, this is usually not the case in the video games viewpoint.

Over the years, the pathfinding algorithm provides navigation for characters in games, and a variety of optimization approaches have been performed in the literature. Barnouti, Al-Dabbagh, & Sahib Naser [14] pointed out that the path problem in industrial games needs to be resolved in real-time, usually with restricted memory and limited CPU time. Reference in [15] supported the statements which stated that most common pathfinding challenges in games were massive memory usage and time-consuming in execution.

Previously, Botea et al. [4] identified issues regarding the need to fulfill the aggressive CPU and current gaming memory limits. In addition, they discussed that there is always the need to consider the struggle to control a middle ground between memory and time constraints. They also added that another measure should be taken into account, which is the path quality. Therefore, they concluded that the goal in pathfinding is to improve performance on one or more of these parameters without the other criteria being compensated for.

One of the popular informed search algorithms is the A* algorithm [16]. This algorithm and several remarkable variants are undoubtedly successful in practical applications and will almost certainly remain a foundation of solutions for the future [17]. However, Rabin [17] said that many games make considerations and adjustments which allow A* to be efficient such as changing game representation. For example, navigation meshes are used widespread in 3D games, but it can radically change the game world in unpredictable ways during gameplay. Rabin [17] stated that solutions for the navigation mesh are existed but are not well identified or revealed by the game industry or the solutions are challenging to execute.

In terms of path quality, Rabin [17] added that Artificial Intelligence (AI) characters tend to favour the shortest paths rather than realistic or genuinely optimal paths. Hence, this situation leads to plenty of opportunities for improvement in the algorithm itself.

In conclusion, common challenges for pathfinding in games can be concluded as:

i. Time-consuming in execution
ii. Massive memory consumption
iii. Unrealistic path quality

3. RESEARCH PERSPECTIVES ON THE INFORMED SEARCH ALGORITHMS FOR PATHFINDING IN GAMES

Although many researchers discovered that A* algorithm is ideal, Cui & Shi [19] disagreed and stated that A* could speed up by enhancing it from various perspectives. There are four suggested by Cui & Shi [19] to boost the performance of A* which are search space optimization, memory usage reduction, heuristic functions improvement, and new data structure implementation.

Several researchers aimed to optimize informed search algorithms for pathfinding in games and this paper classified the optimization approaches by four identified aspects as shown in Figure 1.

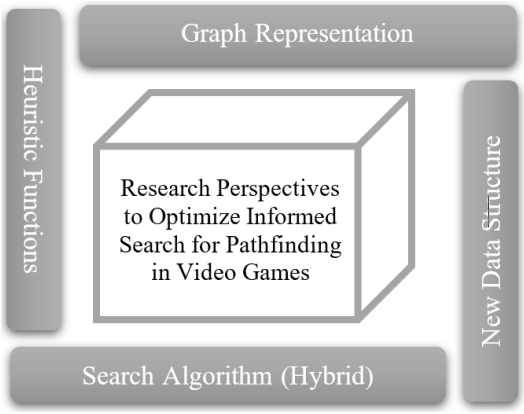

Figure 1: The Research Perspectives for Informed Search Pathfinding Optimization in Video Games

Four main aspects of optimization for informed search pathfinding will be discussed in the following section, and the phrase 'research perspectives' will be used in this study to describe the aspects or techniques for optimization. The following section is devoted to reviewing research perspectives for the optimization of informed search pathfinding. It is divided into four different subsections concentrated on: (i) graph representation, (ii) heuristic

functions, (iii) search algorithms, and (iv) data structure used to enhance pathfinding algorithm.

A preliminary search from the https://scholar.google.com suggested a massive quantity of studies correlated to "pathfinding" in games. After an initial screening procedure, some keywords from the abstract were referred to the delineation of the chosen papers or publications to obtain the highly significant and notable articles. The focused publications ought to principally represent pathfinding techniques; including single or multi-agent navigation; static or dynamic; pathfinding optimization. The search terms "pathfinding AND games AND (modification OR optimization OR improvement OR enhancement)" were also stemmed from the review articles of procedure search in pathfinding.

Based on Table 1, a typical graph representation for the most prominent pathfinding algorithm, which is A* is by using grid maps [22]. However, since grid maps faced poor performances in term of memory in a larger environment, Subrando et al. [22] implemented A* in navigation mesh. They have reported that the resulting solution able to solve poor performance issues in grids and found no further issues.

Cui, Harabor, & Grastien [23] introduced any-angle pathfinding in NavMesh, namely Polyanya, which expands their previous work in grids, Anya [24]. The expansion is tested on three different types of NavMesh which are constrained Delaunay triangulation (CDT), a merge neighboring polygon in CDT (M-CDT), and a rectangle mesh [23]. As a result, Polyanya+M-CDT performed best and surpassed Anya in terms of search time and node expansions.

In tactical pathfinding which intends to locate a secure path, Brewer [25] mentioned that the usual way to represent the graph is by the use of waypoint or grid. He introduced a cover representation for tactical pathfinding in NavMesh and indicated that fast pathfinding is achieved without large memory consumption.

One study by Cui & Shi [19] examined n-sided-poly-based NavMesh that is much easier to automatically produce compared to triangle-based NavMesh. They claimed that their new implementation in n-sided-poly-based NavMesh has an advantage in memory usage.

From Table 1, it can be concluded that NavMesh is highly selected to be used in optimizing the pathfinding algorithm to reduce memory consumption.

Table 1: Summary of the reviewed literature on pathfinding optimization by changing graph representation

Reference	Graph representation	Pathfinding algorithm	Advantages
Subrando et al. (2018) [22]	Grids → NavMesh	A*	Less memory overhead in NavMesh for a larger environment
Cui et al. (2017) [23]	Grids → NavMesh	Polyanya	Speed up over one magnitude of search time and node expansions on average
Brewer (2013) [25]	Grids/Waypoint→ NavMesh	A*	Combine cover representation with NavMesh for tactical pathfinding without huge memory usage
Cui & Shi (2011) [19]	(NavMesh) n-sided-poly →triangle	A*	Reduce memory consumption and no post-processing is needed

3.1 Optimization by Changing Graph Representation

In pathfinding, environments are often represented as a graph or map. Although the term differs in the logical state, the term 'graph representation' is often used interchangeably with search space [15], map representation [11] [12], or game world [20]. Throughout this study, the term 'graph representation' will be used to refer to the pathfinding environment. Graph representation is one of the essential attributes in determining navigation performance in most pathfinding algorithms [11].

For the optimization purpose, the primary step is to consider changing the game's graph representation [15]. Selecting the best graph representation is often the first solution to optimize the pathfinding algorithm because it brings a considerable impact on the algorithm's implementation.

In games, three popular graph representations are grid maps, waypoint graphs, and navigation meshes. A debate has long prevailed as to whether which graph representation is the most excellent, however, Rabin & Sturtevant (2019) [20] made it clear that each graph representation has its advantages and disadvantages based on the type of game it represents. They have summarized a few benefits and limitations for common graph representation used in games in their paper.

Graph representation in games can be modified to indirectly optimize pathfinding performance. Table 1 outlines several optimizations of informed search pathfinding by changing the graph representation.

3.2 Optimization by Improving Heuristic Functions

The heuristic search function is a key field of AI, and it has been extensively utilized in game-play, path-planning and agent control [26]. The definition of the heuristic search function can be simplified as a set of guidelines for selecting pathways in a state-space that resulted in an agreeable possible resolution by decreasing the number of options movements [12]. The heuristic itself employs a practical pathfinding algorithm that is not guaranteed to be finest or rational, but which is nevertheless adequate for the pathfinding solution. The selection of heuristic function acts as a substantial part of the overall pathfinding process to help in reducing computation. There is two general heuristic function used in previous literature, which is Manhattan

distance and Euclidean distance. Table 2 summarizes related research for pathfinding by optimizing the heuristic function.

From Table 2, modification of heuristic function in pathfinding is proposed by Smołka et al. [27] who introduced two modifications in their technique. To optimize A*, they apply a Chebyshev distance as its heuristic and add vertex penalties to shorten the time in the search technique. Besides, they also remove extra vertices in a straight line to increase execution time during the search. The optimization of A* is named A*MOD by them and is aimed to produce a natural and realistic path and remove excessive corners in the path.

mentioned that although time reduction has been achieved compared to the existing search technique, further reduction is needed to enhance search technique.

Additionally, Mathew [31] considered direction-based heuristic by combining eight possible directions and its heuristic key. For the evaluation purpose, Mathew [31] used a 150x150 grids and 2D array and employ direction-based heuristic to the A* algorithm. As a result, he demonstrated that A* with the use of direction-based heuristic generates a better path in terms of quality and utilize a smaller amount of time and memory compared to the traditional A*.

Table 2: Summary of the reviewed literature on informed search pathfinding by improving heuristic functions

Reference	Objective Function(s)	Optimization Approach(es)	Characteristics
Smołka et al. (2019) [27]	Time; Realistic path	Chebyshev distance	Alter metric and apply vertex penalties for diagonal move
Yiu et al. (2018) [28]	Time; Complexity	Multi-Weighted-Heuristic function	Optimize heuristic by utilizing Genetic Algorithm (GA)
Cao (2018) [29]	Time	Check From Closed List	Check closed list instead of an open list in A* to reduce time and Djikstra's algorithm [16]
Abd et al. (2017) [30]	Time	Weight-based recurrence	Dependent on the quantity of repetitions for search algorithm and w-value
Abd et al. (2017) [30]	Time	Weight-based-coordinate distance	Rely on length of start-end nodes related to w-value
Abd et al. (2017) [30]	Time	Weight-based-travel cost	Refer to travel cost which refers to a weight set, α
Mathew (2015) [31]	Time; Memory; Path	Direction based heuristic	Introduce heuristic key to represent left right up down
Kim et al. (2011) [32]	Time	Visibility-test heuristic	Modify Euclidean heuristic by searching visible vertices for obstacles

In 2018, Yiu, Du, & Mahapatra [28] presented a Multi Weighted-Heuristic (MWH) functions that make use of genetic algorithm (GA) to automatically optimize by selecting four heuristic functions from a gene pool of 10 different heuristics. They embed the MWH function in A* search and name it as Evolutionary Heuristic A* search (EHA*). It then tested on two well-known problems in Artificial Intelligence which are Blocks World Problem and Sliding Tile Puzzle. The memory usage is proved to be significantly improved, and they reported that the time complexity has a dependency on the chromosome's size. As the length of the chromosome increases, the time is increasing exponentially.

Cao [29] proposed a consistent and admissible heuristic function, "Check From Closed List" which targets to reduce time complexity in the specific operation for A* open list.

In addition, Abd et al. [30] introduced a new weighted heuristic search pathfinding algorithm. The proposed weighted heuristic search is then tested by implementing it in popular pathfinding algorithms: A*, Bidirectional A*(Bi-A*), and Jump Point Search (JPS) algorithms. In conclusion, the weighted heuristic search technique portrayed some flaws that need to be enhanced. Flaws in their proposed weighted heuristic can be summarized as follows: i) optimality of solutions is compromised ii) performance's stability is proportional to the weight value, and they also

According to Kim et al. [32], they introduced a visibility test heuristic and applied A* in a NavMesh. They aim to reduce the search space on a vast number of polygons in the NavMesh with an extensive terrain. Their experiment has shown that the visibility test heuristic outmatches the plain A* by reducing execution time during the search with a smaller number of visited nodes.

From Table 2, it can be summarized that most of the heuristic functions aim to minimize the time to execute the algorithm and it is the only research work [27] aimed for producing a realistic path.

3.3 Optimization by Combining Search Algorithms

Besides graph representation and heuristic functions, the search algorithm is the next significant process in optimizing the pathfinding algorithm. In general, search algorithms can be separated into informed and uninformed search [7]. An uninformed search contradicts an informed search where the former does not require the use of a heuristic function. When the target location is unknown, all directions were searched in the pathfinding process, and it is also called as blind searches [33]. Examples of standardly used uninformed pathfinding algorithms are the iterative-deepening depth-first search (IDD FS) [34].

Informed pathfinding utilizes a heuristic function to trace the target location through pathfinding. Since the target location is known, an informed search usually is quicker than an uninformed search [33]. Various informed pathfinding algorithms were created in the past literature which includes the A*search [35] and the Iterative-Deepening-A* (IDA*) search [34]. To date, the A* algorithm has dominated the field for many decades. Recently, a considerable literature has grown up around the theme of hybrid pathfinding algorithms. Table 3 summarizes the hybrid search pathfinding algorithm in games.

Cui & Shi [19] stated that potential research in optimizing the A* algorithm is to combine various optimization methods into a single solution.

Table 3: Summary of hybrid search pathfinding algorithms for optimization in games

Reference	Environment "system"	Agent's type	Highlight(s)	Pathfinding technique	Memory Complexity	Time Complexity
Sazaki et al. (2018) [36]	Dynamic	-	Compare pathfinding using a combined algorithm for NPC in avoiding dynamic obstacles	Combine Dynamic Pathfinding (DPA) and A* algorithm	-	Shorter time than single DPA
Harsani et al. (2018) [38]	Dynamic	Multi-agent	Add Fuzzy logic in game rule's implementation to increase difficulty and behaviour to look natural	Fuzzy logic and Algorithm A*	-	-
Sabri et al. (2018) [41]	Static	Single-agent	Assess the conventional algorithm, A* and Bee algorithm in pathfinding	Bee Algorithm	Inconsistent	In free obstacle play, a * algorithm scans ten times faster; Bee outperformed A * in the obstacles and complex game world.
Díaz & Iglesias (2017) [42]	Dynamic	Multi-agent	Present a swarm-based intelligence to reflect and animate those behavioral routines for the NPC's AI	Swarm intelligence (SI) – Particle Swarm Optimization	-	CPU time improved
Johan Hagelbäck (2016) [43]	Dynamic	Multi-agent	A hybrid method where A* is used without the presence of an enemy unit or building. When the units	A* combine Flocking with boid algorithms	-	A*/boid outperform A*/potential

In Table 3, considering of hybrid algorithm, Sazaki, Primanita, and Syahroyni [36] combined A* algorithm and Dynamic Pathfinding Algorithm (DPA) in their study and Sazaki, Satria, and Syahroyni, [37] evaluated the hybrid algorithm and provided a comparison with only DPA in car racing game. The hybrid algorithm reduces the time needed to achieve the target from the beginning node. Besides, it improves the chance of successfully reaching the goal node with or without obstacles. DPA is used to prevent static or dynamic obstacles for the NPC's navigation.

Harsani, Mulyana, & Zakaria [38] developed a platform game with three levels of difficulties named Goat Foraging Game. Two algorithms including Fuzzy logic and A* were used in the game rules and the conclusion is made for both algorithms. It is concluded that NPC's character behavior obtained by fuzzy logic can enrich difficulties and natural NPC's behavior, and A* successfully found the shortest path between the player and NPCs.

Ant Colony Algorithm is first introduced in 1991 by Colorni, Dorigo, & Maniezzo [39]. It is then applied to vehicle navigation for travelers in Iran which involves a multivariate such as medical care, entertainment, traffic light numbers, accidents, and traffic flow [40]. The algorithm combined two algorithms which are Ant Colony and A* algorithm. A* acts as a first task in the navigation by enlivens paths pheromones in ant's algorithm. The results from both combinations yield a very promising route in terms of cost average.

Similarly, a study on the Bee algorithm has been conducted by Sabri et al. [41] and has been compared with A* on two different maps and two distinct measurements, performance, and memory consumption. For performance, on average, the A* algorithm surpasses the Bee algorithm on the free obstacle's map (200x200 and 500x500). However, for a simple environment with obstacles, Bee algorithm performs

three times better, while in a complex environment, A* slightly outmatch Bee algorithm. Memory consumption for the Bee algorithm shows promising results since it performs three times better on average for free obstacle map (200x200 and 500x500). Memory for the Bee algorithm also outperformed A* in both map size with the obstacle.

Another bio-inspired algorithm is swarm intelligence (SI). Díaz & Iglesias [42] illustrated one of SI techniques which is Particle Swarm Optimization (PSO) in the FPS game name Isolated. Two experiments were conducted to determine PSO efficiency in pathfinding and to assess the behavioral patterns for pathfinding and action planning of NPC. A promising result in terms of CPU time was yield, however, no benchmark is included in this experiment.

Additionally, Johan Hagelbäck [43] presented a hybrid algorithm in a Real-Time Strategy (RTS) game named StarCraft. The hybrid algorithm combined A* and boid flocking algorithm to tackle the chore of the unit's placement efficiently in battle events. Previously, he has coupled A* with a potential field for the same purpose [20]. As a result, hybrid pathfinding A*/boid able to target the enemy in RTS game with better performance than the hybrid solution of A*/potential field.

3.4 Optimization by Changing the Data Structure

One of the approaches to optimize search algorithms in pathfinding is to change data structure in the implementation process. Basically, any data structure has two sides: "abstraction" and "implementation" [44]. For the abstraction side, it indicated its concept or action performed while implementation explains how the concept is being implemented. For example, in maintaining an open list in the A* algorithm, a priority queue is the best abstraction and can be implemented using a binary heap [18].

By using the A* algorithm, Cao [29] has presented a comparison of several data structures which are array, hash table, min-heap, and hot queue. All four data structures were discussed in his study and time complexity is measured by comparing it using four primary procedures in A* open list, namely "insertNew", "deleteMin", "whetherContains" and, "decreaseKey". However, every data structure present benefits and limitations, depending on what type of optimization it tends to focus on. There is various data structure such as an array, hash table, binary min-heap, and Fibonacci heap multilevel buckets. Each data structure has its own advantages and disadvantages. The common data structure used in the implementation of the pathfinding algorithm is an array and heap. Surprisingly, only a few studies have been found for optimization through data structure modification. Table 4 provides a review of pathfinding optimization by changing the data structure.

In another significant study, Tavakoli [45] proposed two new data structures called partial min-heap and cache min-heap in the implementation of the A* algorithm and assess it to compare with other common data structures. By limiting the size of the heap, partial min-heap able to accomplish the number of operations performed and the time needed. Since the size of the data structure is reduced, the trade-off was the success rate in the pathfinding itself. Hence, the author suggests a few conditions that are suitable with the partial min-heap for the best-case scenario. Basically, the cache min-heap use a splitting concept for both insert operations in a single array, while combining both concepts of a min-heap and an unsorted array. Although the cache min-heap is unstable, Tavakoli [45] mentioned that it ensures to discover the optimal path.

Besides the proposal of a new implementation of its heuristic function, Cao [29] also suggested a new data structure named multi-stack heap to be implemented with a traditional A* algorithm in the open list. The results indicated that the multi-stack heap only needs $O(1)$ for insertion and deletion, while typical data structure used in A*, min-heap previously takes about $O(n)$ in the most terrible case scenario.

Table 4: Summary of pathfinding optimization by data structure modification

Reference	New Data Structure	Benchmark	Highlight(s)	Pathfinding technique	Performance	Time Complexity
Tavakoli (2019) [45]	Partial min-heap	Unsorted array and min-heap	Limit the size of the data structure to reduce time in executing an operation. Suitable for small map size.	A* algorithm	Compromise success rate	Better in several cases
Tavakoli (2019) [45]	Cache min-heap	Unsorted array	Split the data insertion to two concepts of the unsorted array and min-heap	A* algorithm	Unstable, but guarantee an optimal path	Shorten time but varies in several cases
Cao (2018) [29]	Multi-stack heap	Min heap	Tackle two main operations in an open list for A* to reduce time complexity from $O(n)$ to $O(1)$	A* algorithm	-	$O(1)$

Even though there is only a minimal study in this optimization perspective, the evidence reviewed here seems to suggest a pertinent role of data structure in the optimization of the pathfinding algorithm.

4. DISCUSSIONS

Overview of pathfinding algorithms is presented and briefly described in each section. One of the well-known algorithms which are A* algorithm also has been explained in detail. Besides, numerous variants involved in the pathfinding algorithm are discussed in detail which presents potential in further optimization.

4.1 Additional Research Directions for Pathfinding Optimization in Video Games

Results from earlier studies demonstrate a consistent and robust association between pathfinding and games. It has been estimated that the billion-dollar games industry will continue to grow every year towards the year 2022 [5]. With the advent of technology, demands were increased in developing immerse game experiences. Hence, more effort should be spent on the design of a realistic path in the navigation techniques, while maintaining the performance and reasonable memory usage. For example, many hybrid algorithms discussed in Section 3.3 such as [28] [29] [31] [32] were focusing on time and memory, instead of path optimality. Besides the considerable demands in the game industry, it is also essential to acknowledge the need for AI programmers in pathfinding optimization [46].

4.2 Summary and Future Trends

Since the year 2010, there has been a growing and diminishing trend towards pathfinding optimization in games over the past decades. Even though the only focus of this paper is video games, however, it is observable during the selection of publications that pathfinding applications in other fields outside the game's navigation such as robotics is also dominant research.

There are several studies using swarm intelligence in the pathfinding technique for the game's navigation in comparison with the others [41] [42] [43]. The potential of the hybrid algorithm shall not be undervalued since the algorithmic structure of swarm intelligence achieves well in increasing CPU performance.

This review considers publications to evaluate and assess the trend for optimization of informed search pathfinding in video games over the last decade. In the observation, most of these pathfinding optimization studies have been limited to navigation for games in a personal computer (PC) and mobile, however, it was not possible for it to be extended in virtual reality. During the paper selection phase, by far, the application of pathfinding in virtual reality exists [47], but lack of the optimization part and thus excluded in this review. As an alternative to gamification [48] for the immersive game experience, future trends in pathfinding optimization are expected to explore further on practical applications and optimization in virtual reality.

5. CONCLUSION

In this review, pathfinding in games engaged significant experts and optimizations with tightening resources. Given the inadequate resources for navigating the characters in the game with future game's demand, optimization of informed search pathfinding remains a huge potential study. This review synthesized the recent research progress in the game's navigation and emphasizes the optimization using four different perspectives. Hence, it can be concluded that research on the pathfinding technique for navigation in games is a promising area. For example, although A* algorithm is a classic pathfinding solution since 1968, A* is still being implemented and benchmarked, and it is further optimized in most of the current researches. This study also found out that limited optimization is performed by changing the data structure while most optimization is carried out by combining search algorithms and exploiting heuristic functions. Swarm intelligence algorithms such as bee colony and ant colony are somewhat new research areas in games that can be tackled in the future.

ACKNOWLEDGMENT

Appreciation is expressed to the Ministry of Education Malaysia and Universiti Teknologi MARA for generous funding and support. This research was carried out during the study leave of the main author under the 2019 Academic Training Scheme for Bumiputera (SLAB) scholarship scheme. Postgraduate members of Media and Game Innovation Centre of Excellence (MaGICX), Universiti Teknologi Malaysia (UTM) which assisted this work are hereby acknowledged by the authors.

REFERENCES

1. Z. Abd Algfoor, M. S. Sunar, A. Abdullah, and H. Kolivand. **Identification of metabolic pathways using pathfinding approaches: a systematic review**, *Brief. Funct. Genomics*, Vol. 16, March 2016. https://doi.org/10.1093/bfgp/elw002
2. P. Diao and N. J. Shih. **MARINS : A Mobile Smartphone AR System for Pathfinding in a Dark Environment**, Sensors, Vol. 18, No. 10, 2018. https://doi.org/10.3390/s18103442
3. L. Cai, Y. Zhou, Y. Liang, and J. He. **Research and Application of GPS Trajectory Data Visualization**, *Ann. Data Sci.*,Vol. 5, No. 1, pp. 43–57, 2018. https://doi.org/10.1007/s40745-017-0132-1
4. A. Botea, B. Bouzy, M. Buro, C. Bauckhage, and D. Nau. **Pathfinding in Games**, *Artif. Comput. Intell. Games*, Vol. 6, pp. 21–32, 2013.

5. Newzoo. **2019 Global Games Market Report**, 2019. [Online]. Available: https://newzoo.com/inhttps://newzoo.com/insights/trend-reports/newzoo-global-games-market-report-2019-light-version/ng-37/.
6. D. Aversa. **Smart Pathfinding: Extending Pathfinding with Agent Capabilities**, Ph.D. dissertation, Department of Computer, Control and Management Engineering, Sapienza – University of Rome, 2017.
7. A. Bleiweiss. **GPU Accelerated Pathfinding**, in *Proc. 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, 2008, pp. 65-74.
8. S. N. Z. Ahmmad and F. Muchtar. **A review on applications of optimization using bat algorithm**, *Int. J. Adv. Trends Comput. Sci. Eng.*,Vol. 9, No. 1.1, pp. 212–219, 2020.
9. T. Cazenave. **Optimizations of data structures, heuristics and algorithms for path-finding on maps**, in *Proc. 2006 IEEE Symp. Comput. Intell. Games, CIG'06*, Vol. 06, 2006, pp. 27–33. https://doi.org/10.1109/CIG.2006.311677
10. S. Rabin and N. R. Sturtevant, **Pathfinding Architecture Optimizations**, in *Game AI Pro: Collected Wisdom of Game AI Professionals*, CRC Press, 2013, ch. 17, pp. 241–252. https://doi.org/10.1201/b16725
11. Z. Abd Algfoor, M. S. Sunar, and H. Kolivand. **A comprehensive study on pathfinding techniques for robotics and video games**, *International Journal of Computer Games Technology*, April 2015.
12. A. N. Sabri, N. Haizan, M. Radzi, and H. Hassan. **The State of Art Heuristic Pathfinding in Games**, *Advanced Science Letters*, Vol. 24, No. 2, pp. 1273–1278, 2018.
13. D. Sigurdson, V. Bulitko, W. Yeoh, C. Hernandez, and S. Koenig. **Multi-Agent Pathfinding with Real-Time Heuristic Search**, in *IEEE Conference on Computatonal Intelligence and Games (CIG*), pp. 1–8, August 2018. https://doi.org/10.1109/CIG.2018.8490436
14. N. H. Barnouti, S. S. M. Al-Dabbagh, and M. A. Sahib Naser. **Pathfinding in Strategy Games and Maze Solving Using A* Search Algorithm**, *J. Comput. Commun.*, Vol. 04, No. 11, pp. 15–25, 2016.
15. A. Noori and F. Moradi. **Simulation and Comparison of Efficency in Pathfinding algorithms in Games**, *Ciência e Nat.*, Vol. 37, pp. 230, 2015. https://doi.org/10.5902/2179460X20778
16. E. W. Dijkstra, **A Note on Two Problems in Connexion with Graphs**, *Numerische mathematik*, Vol. 1, No. 1, pp. 269–271,1959.
17. S. Rabin, *Game Ai Pro2: Collected wisdom of game AI professionals*, 2015. https://doi.org/10.1201/b18373
18. X. Cui and H. Shi. **A * -based Pathfinding in Modern Computer Games**, *Int. J. Comput. Sci. Netw. Secur.*, Vol. 11, No. 1, pp. 125–130, 2011.
19. X. Cui and H. Shi. **Direction Oriented Pathfinding In Video Games**, *Int. J. Artif. Intell. Appl.*, Vol. 2, No. 4, pp. 1–11, 2011.
20. J. Hagelbäck. **Potential-field based navigation in StarCraft**, in *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 388–393, 2012. https://doi.org/10.1109/CIG.2012.6374181
21. S. Rabin and N. R. Sturtevant. **Choosing a Search Space Representation**, *Game AI Pro 360*, Vol. 1, pp. 13–18, 2019.
22. T. Subrando, F. A. Prasetyatama, and D. Fitrianah. **Implementation of a* algorithm within navigation mesh in an artificial intelligence based video games**, *Int. J. Eng. Technol.*, Vol. 7, No. 4, pp. 3249–3254, 2018.
23. M. L. Cui, D. D. Harabor, and A. Grastien, **Compromise-free pathfinding on a navigation mesh**,*IJCAI Int. Jt. Conf. Artif. Intell.*,pp. 496–502,2017,doi: 10.24963/ijcai.2017/70.
24. D. Harabor and A. Grastien. **An optimal any-angle pathfinding algorithm**, *ICAPS 2013 - Proc. 23rd Int. Conf. Autom. Plan. Sched.*, pp. 308–311, 2013.
25. D. Brewer. **Tactical Pathfinding on a NavMesh**, *Game AI Pro*, pp. 361–368, 2013.
26. V. Bulitko, Y. Björnsson, and R. Lawrence. **Case-based subgoaling in real-time heuristic search for video game pathfinding**, *J. Artif. Intell. Res.*, Vol. 39, pp. 269–300, 2010. https://doi.org/10.1613/jair.3076
27. J. Smołka, K. Miszta, M. Skublewska-Paszkowska, and E. Łukasik. **A* pathfinding algorithm modification for a 3D engine**, *MATEC Web Conf.*, Vol. 252, pp. 03007, 2019.
28. Y. F. Yiu, J. Du, and R. Mahapatra. **Evolutionary Heuristic A∗ Search: Heuristic Function Optimization via Genetic Algorithm**, in *Proceedings - 2018 1st IEEE International Conference on Artificial Intelligence and Knowledge Engineering, AIKE 2018*, pp. 25–32, Sep. 2018.
29. Q. Cao. **Exploiting Problem Structure in Pathfinding**, *Electron. Theses Diss.*, 2018.
30. Z. Abd, M. Shahrizal, and A. Abdullah, **A new weighted pathfinding algorithms to reduce the search time on grid maps**, *Expert Syst. Appl.*, Vol. 71, pp. 319–331, 2017. https://doi.org/10.1016/j.eswa.2016.12.003
31. G. E. Mathew. **Direction based heuristic for pathfinding in video games**, *Procedia Comput. Sci.*, Vol. 47, pp. 262–271, 2015. https://doi.org/10.1016/j.procs.2015.03.206
32. H. Kim, K. Yu, and J. Kim. **Reducing the search space for pathfinding in navigation meshes by using visibility tests**, *J. Electr. Eng. Technol.*, Vol. 6, No. 6, pp. 867–873, 2011.
33. K. Li, K. Phooi, L. Seng, L. Ang, and S. Inn. **Uninformed pathfinding : A new approach**, *Expert*

Syst. Appl., Vol. 42, No. 5, pp. 2722–2730, 2015.

34. R. E. Korf. **Depth-First Iterative-Deepening : An Optimal Admissible Tree Search***, *Artificial intelligence*, Vol. 27, pp. 97–109, 1985.
35. P. E. Hart and J. Nils. **A Formal Basis for the Heuristic Determination of Minimum Cost Paths**, *IEEE transactions on Systems Science and Cybernetics*, Vol 4, No. 2, pp. 100–107, 1968. https://doi.org/10.1109/TSSC.1968.300136
36. Y. Sazaki, A. Primanita, and M. Syahroyni. **Pathfinding car racing game using dynamic pathfinding algorithm and algorithm A∗**,in *Proc. - ICWT 2017 3rd Int. Conf. Wirel. Telemat. 2017*, pp. 164–169, July 2018.
37. Y. Sazaki, H. Satria, and M. Syahroyni, **Comparison of A∗ and dynamic pathfinding algorithm with dynamic pathfinding algorithm for NPC on car racing game**, in *Proceeding 2017 11th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2017*, pp. 1–6, January 2018. https://doi.org/10.1109/TSSA.2017.8272918
38. P. Harsani, I. Mulyana, and D. Zakaria. **Fuzzy logic and A∗ algorithm implementation on goat foraging games**, *IOP Conf. Ser. Mater. Sci. Eng.*, Vol. 332, No. 1, 2018.
39. A. Colorni, M. Dorigo, and V. Maniezzo. **Distributed Optimization by ant colonies**, in *Proc. First Eur. Conf. Artif. Life*, pp. 134–142, 1991.
40. H. Salehinejad and H. Nezamabadi-pour. **Combined A*-ants algorithm: a new multi-parameter vehicle navigation scheme**, *arXiv preprint arXiv:1504.07329*, pp. 154–159, 2015.
41. A. N. Sabri, N. H. M. Radzi, and A. A. Samah. **A study on Bee algorithm and A∗ algorithm for pathfinding in games**, *ISCAIE 2018 IEEE Symp. Comput. Appl. Ind. Electron.*, pp. 224–229, 2018. https://doi.org/10.1109/ISCAIE.2018.8405474
42. G. Díaz and A. Iglesias. **Swarm intelligence scheme for pathfinding and action planning of non-player characters on a last-generation video game**, in *Advances in Intelligent Systems and Computing*, Vol. 514, pp. 343–353, 2017.
43. J. Hagelbäck. **Hybrid Pathfinding in StarCraft**, *IEEE Trans. Comput. Intell. AI Games*, Vol. 8, No. 4, pp. 319–324, 2016. https://doi.org/10.1109/TCIAIG.2015.2414447
44. J. Bentley. **Programming pearls: Thanks, heaps**, *Commun. ACM*, Vol. 28, No. 3, pp. 245–250, 1985.
45. M. Tavakoli. **Performance Evaluation of Competing Data Structures in Pathfinding**, *ProQuest Diss. Theses*, 2019.
46. S. Rabin and N. R. Sturtevant. **Pathfinding Architecture Optimizations**,*Game AI Pro Collect. Wisdom Game AI Prof.*, pp. 241, 2013.
47. M. Faisal, H. Nurhayati, Y. M. Arif, F. Kurniawan, and F. Nugroho. **Immersive bicycle game for health virtual tour of uin maulana malik ibrahim Malang**, *J. Teknol.*, Vol. 78, No. 5, pp. 325–328, 2016. https://doi.org/10.11113/jt.v78.8330
48. R. M. Andrias and M. S. Sunar, **User/player type in gamification**,*Int. J. Adv. Trends Comput. Sci. Eng.*, Vol. 8, No. 1.6 Special Issue, pp. 89–94, 2019. https://doi.org/10.30534/ijatcse/2019/1481.62019