# 

# The Critique on THE efficiency of new path-finding algorithms

Ege Bulut; Master’s Computer Science Student; [D22124401@mytudublin.ie](mailto:D22124401@mytudublin.ie)

Table of Contents

[The Critique on THE efficiency of new path-finding algorithms 1](#_Toc122149944)

[Introduction 2](#_Toc122149945)

[evaluation 3](#_Toc122149946)

[References 4](#_Toc122149947)

ABSTRACT

New algorithms are being developed by improving the old ones or merging multiple ones to achieve efficient, faster, and better-optimized algorithms. In most cases, the choice on which algorithm to further develop is either made by choosing a specific algorithm and testing its performance against the set of problems it is known to solve and make a conclusion through benchmarking, or by choosing a specific type of problem and developing a better algorithm by either improving an existing one or merging multiple ones, then to be compared by benchmarking with the original ones. In this article, five example journals that study new algorithms’ performances by following the described procedures will be reviewed. The papers will be discussed by the choice of their research method, experimentation, and finally, the gaps in the research.

Keywords: Shortest Path Algorithm, Ant Colony Algorithm, Floyd Algorithm, Dijkstra’s Algorithm, Efficiency, Optimization

## Introduction

Pathfinding problems are very widely known and the solve these problems many algorithms can be chosen. Some of these algorithms are chosen based on the network, but the most important aspect of these algorithms is their performance. Pathfinding problems can be found in the real world very easily, whether it is a physical path such as train networks or a virtual network between the computers on a system. The efficiency of these algorithms that solve these problems is very important for the companies who will be implementing these solutions as it means more profit. An efficient train network means better transportation for people, and a better virtual network means faster data transfer for every request. There already are some known algorithms to solve these problems such as Dijkstra’s algorithm or Ford-Fulkerson Algorithm. However, even though these algorithms provide exemplary performance results, there are versions of these algorithms that provide slightly better results. These new algorithms most of the time encapsulate the original ones or merge the original algorithm with a different path-finding algorithm to decide the efficient one to use based on the real-life case rather than being strict with a single way of solution. These solutions are accepted as better when they achieve equivalent or slightly better results than the originals. The reason for this is that there is a mathematical constraint to the complexity of these algorithms. Running through a network to understand the path has a certain weight on the runtime complexity, and it is difficult to increase the performance beyond that. Additionally, the data types remain the same. Networks are either cyclic or non-cyclic graphs and they either are weighted or not. All of this means that there is a mathematical limit to how efficiently these problems can be solved. And the well-known algorithms are very close to this limit. Therefore, the studies of the kind that will be discussed in this paper must be reviewed thoroughly to make sure that the newly declared algorithm is strictly equivalent or more efficient universally. Because there could be many factors why the algorithm may seem better than its predecessors. One of these factors is that the system used to test the new algorithm might be a better-equipped system than the one that is used to test the older algorithm. Another reason might be the specific problem used to run the benchmark. The newly developed algorithm might be better at solving the problem used in the experiment to do the benchmarking. This would be a misleading conclusion pointing out that the new algorithm is better than the original one, but the hypothesis might be wrong. Thus, the research about creating more efficient and optimized algorithms needs to be evaluated very carefully to understand whether the results are precisely proving the hypothesis and whether the results are significant universally.

## Evaluation

There are five journals to compare the questions that research a more efficient algorithm for path-finding algorithms. Some of these papers research an algorithm based on a previously developed algorithm, whereas some research an algorithm based on a problem. One of these is about finding an adaptive reliable shortest-path algorithm for gaussian process-regulated environments. This research compares the newly found algorithms against the ones existing. The paper runs a simulation based on real-life networks and recognized examples. The real-world test was done by using the Singapore Arterial Road Network. If the same experiment was done in different networks, it would highly strengthen the findings of the experiment. Moreover, another gap is the system information on which the experiment was conducted. To replicate the test data, it would be very useful to have information on what sort of setting was used to run the experiment. Moreover, the paper goes into detail about the mathematical foundations of the algorithms. There are some terminologies and background information that is required to understand the contents of the paper, some of them might have been explained in the paper. The experiment is extensive, and it proves that the aim of the research was met.

Another paper is about researching an algorithm that can be more efficient than Dijkstra's Algorithm while having a polynomial time complexity. The algorithms were tested only against one problem with one other algorithm which is well-known to solve single-constrained shortest-path problems. Other tests must be done with different solutions and compared with other algorithms such as Dijkstra. Additionally, the paper seems to give an extensive amount of information about the coding of the algorithms and explain them in detail. The mathematical depth might be argued as too much. The content was supported by graphs and these graphs also visualize the results that were gathered from the simulation. A missing part is the specifications of the system used for the simulation so that the data can be tested again.

This paper which is trying to achieve an optimal solution for mobile robots’ pathfinding by improving the convergence of the ant colony algorithm that is already being used has achieved its aim by finding an improved version of the ant colony algorithm. However, this has been proven for only one problem, therefore, it should be tested for a variety set of problems. Also, to get a more significant understanding of the improvement made in the performance, the algorithm should be compared with the others such as Dijkstra or A-star. Furthermore, the research seems to study the chosen algorithm extensively and studies the possible improvements. The benchmark test seems to be the best option to get results and the evaluation shows a clear conclusion.

The research that was testing the Floyd algorithm’s efficiency in solving an example problem called the desert crossing game, questions the effectiveness of the algorithm against the experiment’s problem, but it only proves that the algorithm satisfies a solution for the given problem. To be able to determine the effectiveness of the algorithm, other algorithms that can solve the problem should be tested as well. Finally, benchmarking could be done to get information on the efficiency of the algorithm. Moreover, the research seems to be more focused on the experiment. The problem used to test out the algorithm seems to be very complex, and the success of the algorithm was determined by its ability to solve this problem. This approach seems to be enough to prove that this algorithm can solve weighted shortest-path problems. However, it does not give information about its efficiency as there is no comparison between other algorithms’ results that can solve this problem.

The article that is aiming to study the improved optimization genetic algorithm and test its efficiency against Dijkstra and genetic algorithm by running it against k shortest paths problem on a graph in large scale routing problems, seems to be inconclusive about the conclusion because it seems to be when the algorithm works it either works equivalent or better than Dijkstra, but it also does not work for all the tested cases. So, the most important further study to carry out would be to compare these two algorithms for different problems. Additionally, the paper seems to be written poorly. There seem to be inverted sentences and misspelled words. There is also missing contextual information about the purpose of the research, the study of the algorithm, and the experiment conducted. On the contrary, the experiment set was described such as the language the code is written in and the processor that was used to do the test, which is crucial to replicate the test.

## References

HOU, X., GUO, H., & ZHANG, Y. (2020). Adaptive reliable shortest path in gaussian process regulated environments. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. doi:10.1109/iros45743.2020.9341736

Kou, C., Hu, D., Yuan, J., & Ai, W. (2020). Bisection and exact algorithms based on the lagrangian dual for a single-constrained shortest path problem. *IEEE/ACM Transactions on Networking,* *28*(1), 224-233. doi:10.1109/tnet.2019.2955451

Su, Q., Yu, W., & Liu, J. (2021). Mobile robot path planning based on improved ant colony algorithm. *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*. doi:10.1109/acctcs52002.2021.00050

Wang, S., Liu, B., Liu, W., Hu, C., Tang, Y., & Yang, J. (2021). Research on the shortest path for crossing desert based on Floyd algorithm. *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. doi:10.1109/icftic54370.2021.9647205

Yassine, H. M., & Zahira, C. (2020). An improved optimization algorithm to find multiple shortest paths over large graph. *2020 Second International Conference on Embedded & Distributed Systems (EDiS)*. doi:10.1109/edis49545.2020.9296433

Word Count: 1388