1.

Reference:

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

Aim:

The aim of the research is to test and compare the newly introduced two algorithms against the state-of-the-art algorithm to solve adaptive reliable shortest path problems in a Gaussian process regulated environment.

Methods:

A simulation is run with the two newly proposed algorithms on Sioux Falls Network. Afterwards, the two algorithms were tested on Singapore Arterial Road Network. After obtaining data from a canonical testbed for transportation studies and a real-world setting, the results were compared with state-of-the-art algorithms’ results.

Conclusion:

Given that the environment is suitable for Gaussian process, the two algorithms prove to be more reliable and effective than other state-of-the-art solutions.

Gaps:

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 which the experiment was conducted on. In order to replicate the test data, it would be very useful to have information on what sort of setting was used to run the experiment.

Critique:

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, which 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.

2.

Reference:

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

Aim:

The aim of the paper is to prove that the newly developed two algorithms from Dijkstra are more efficient than the original Dijkstra’s Algorithm’s values and that can be proven to have a polynomial time complexity.

Methods:

Benchmarking is used in this research, which is the appropriate methodology to prove the efficiency of the algorithms when compared with the other known solutions such as Dijkstra.

Conclusion:

The result of the simulation and benchmarking proves that the newly developed algorithm satisfy the polynomial time complexity while having better performance than their competitors to solve single constrained shortest path problems.

Gaps:

The algorithms were tested only against one problem with one other algorithm which is well known to solve singe constrained shortest path problems. Other tests must be done with different solutions and compared with other algorithms such as Dijkstra.

Critique:

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.

3.

Reference:

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

Aim:

This paper is trying to achieve an optimal solution for mobile robots’ path finding by improving the convergence of the ant colony algorithm that is already being used.

Methods:

A simulation was run on improved ant algorithm and traditional ant algorithm to compare the performance and the number of nodes chosen for the path found.

Conclusion:

The conclusion of the experiment is that the improved ant algorithm reduces the redundant paths and enhances the convergence process while reducing its time. Moreover, node optimization method that is introduced proves to be reducing the number of nodes in the chosen path.

Gaps:

The research has achieved its aim by finding an improved version of the and 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.

Critique:

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.

4.

Reference:

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

Aim:

The aim of the research is to test the Floyd algorithm’s efficiency on solving an example problem called desert crossing game.

Methods:

An experiment is used for this method, using a desert crossing game. This method is the most suitable method as the research is trying to prove that Floyd algorithm is suitable to solve problems such as desert crossing game.

Conclusion:

The results from the experiment on the desert crossing game, using the various modifiers such as weather conditions or resource replenishment, shows that the Floyd algorithm satisfies the requirements by being able to find the shortest path strategy to get across the desert.

Gaps:

The research 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, a benchmarking could be done to get information on the efficiency of the algorithm.

Critique:

The research seems to be more focused on the experiment. The problem used to test out the algorithm seems to be very complex, and success of the algorithm was determined in 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.

5.

Reference:

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

Aim:

This article’s aim is 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 graph in large scale routing problems.

Methods:

As the method of this research, an experiment with random generated values were conducted. Afterwards, the results were benchmarked with other algorithms to derive the conclusion of the research.

Conclusion:

The newly found algorithm seems to find solutions that are equivalent or better than Dijkstra’s answer. However, it is also noted that there are several cases that the algorithm could not find any solutions, whereas Dijkstra was able to find.

Gaps:

The research 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.

Critique:

The paper seems to be written poorly. There seems to be inverted sentences and misspelled words. There are also missing contextual information about the purpose of the research, study of the algorithm and the experiment conducted. On the contrary, the experiment setting was described such as the language the code is written in and the processor that was used to do the test, which are crucial to replicate the test.