Comparison Analysis Of Different Greedy Algorithms In Shortest-Path Optimization

Bhumika Sood[1], Aman Sharma[2]
[1]Dept of CSE
[2]Professor, Dept of CSE
[1, 2] HPU

Abstract- *When itcomes to optimization algorithm,the greedy algorithm is considered to be the best. Due to the tremendous development in the field of computer science, the theory of graphics has spread very quickly and widely. It is becoming one of the most important sciences that played a large role in solving many diverse applications and many problems. These applications include Google maps, Computer protocols, games, and many more. Many of the researchers have discussed the shortest path algorithm to solve the shortest path problem in these applications. In this study, a very popular algorithm i.e., a greedy algorithm is discussed with its various types and along with its types a comparison table is formulated in which the algorithm is compared based on performance, complexity, advantages, and drawbacks in terms of the shortest path optimization. The Analysis shows that Dijkstra's algorithm is better than the other four algorithms in terms of execution time and more efficient for solving the shortest path issue.*

Keywords- Greedy algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Prim's algorithm, Kruskal's algorithm, Selection sort algorithm, Comparative.

I. INTRODUCTION

Definitions of greedy algorithms vary in different surveys slightly, but most describe a greedy algorithm as one that makes a sequence of choices, the best choice available at that time. (Greedy term refers to choosing the best choice over the others). A greedy algorithm is efficient, and straight forward because of its simplicity. A greedy algorithm is adaptable and can be used for different problems in many areas of combinatorics (the branch of mathematics dealing with combinations of objects belonging to a finite set, by certain constraints, such as those of graph theory) and beyond. There are different applications of the greedy algorithm. These include compression of data, DNA sequencing, finding minimum cost spanning trees of a weighted graph, computational geometry, and routing through the network.[1] Some problems are difficult and impractical to solve exactly, but they may have a greedy algorithm that can be used to find solutions that are close to optimal for other problems. The greedy algorithm may produce an exact result for a particular problem, there might be no guarantee that a greedy algorithm exists to solve it exactly. This paper contains the following structure. Section 1 is the introduction. Section 2 covers the literature review. Section 3 presents the comparative analysis of the algorithms. Section 4 is the future scope and conclusion. This paper presents the comparative analysis of different greedy algorithms and these algorithms are selected based on their low complexity, their ability to find the shortest path, and their capability of handlin negative weights. [2]

The underlying algorithms are different types of Greedy algorithms used for the shortest path optimization.

1) **Selection Sort Algorithm**
2) **Bellman Ford Algorithm**
3) **Dijkstra's Algorithm**
4) **Prim's Minimal Spanning Tree Algorithm**
5) **Kruskal's Minimal Spanning Tree Algorithm**

1) **Selection Sort Algorithm: -** Selection sort algorithm is one of the easiest and simple algorithms used for sorting an array from the unsorted part. It is the arrangement of numerical values in statistical order. There are two sub-arrays in a given list of arrays.[3] The sorted array will be placed on the left, and the unsorted list will be placed on the right of the array. It is one of the simplest and most useful sorting algorithms for dealing with small amounts of data. If you have a large data area, then selection sort may be the quickest method.[4]

2) **Bellman Ford Algorithm: -** Bellman ford algorithm is also called a single shortest path algorithm.[5] This algorithm is used to find the shortest distance from one vertex to all other vertices in a weighted graph. Bellman ford's algorithm finds the distance in a bottom-up manner.[6] There are two path problems in the Bellman-Ford Algorithm:

The two types of path problems are [7]:

a) **Single Shortest Path Problem**

b) All Pair Shortest Path Problem

- **a) Single shortest path problem**: Single shortest path problem is the problem of finding the shortest path between a pair of vertices.[8] It is a weighted graph. This problem is solved by Dijkstra. In this case, a single result will be kept and other shortest paths are discarded.
- **b) All pair shortest path problem:**All pair shortest path algorithm is also known as Floyd-Warshall Algorithm.[9] This algorithm is used to find all pairs of shortest-path problems from a given weighted graph. It will find the distance from any node to all other nodes in a graph.

3) **Dijkstra's Algorithm: -** Dijkstra's algorithm is one of the best greedy algorithms when compared with other algorithms. It is the single shortest path algorithm.[10]. It is the best algorithm to find the shortest distance and is used in many areas like google maps, telephone networks, and so on. Dijkstra algorithm works correctly for both directed and undirected graphs and handles non-negative weights for directed graphs.[11] It will always find the best solution and shortest path. Dijkstra's algorithm is best because of its low complexity.[12]

Two types of graphs in Dijkstra's are: - [13]

- **a) Directed graph**
- **b) Undirected graph**

- **a) Directed graph:** In a directed graph user can go from one node to another node in a specific direction for every pair of connected graphs.[14]
- **b) Undirected graph:** In every pair of connected graph user can go from one node to any other node in both directions.[15]

4) **Prim's Minimal Spanning Tree Algorithm: -** Prim's algorithm is also known as Jarnik's Algorithm. This algorithm is used to find the minimum spanning tree from a weighted undirected graph.[16] A minimum spanning tree is a subgraph such that all the vertices in the original graph exist in the subgraph and the subgraph is connected and there will be no cycles in the graph. So essentially, a tree is constructed out of the vertices and edges in the graph such that all the vertices should be covered. Used in network designing, road networks, etc. [17] Prim's algorithm is helpful when dealing with dense graphs that have lots of edges.

Kruskal's Minimal Spanning Tree Algorithm: - Minimum spanning tree is a subgraph such that all the vertices in the original graph exist in the subgraph and the subgraph is connected and there will be no cycles in the graph. It is called a minimum spanning tree because of the edges that the user includes the sum of the weight of those edges should be minimum. Kruskal's algorithm is the algorithm that finds the minimum cost-spanning tree using the greedy approach. In Kruskal's algorithm, all the edges should be sorted in non-decreasing order of their weight. It will pick the smallest weight edge that does not create a cycle in the minimum spanning tree constructed. Various applications are telephone networks, LAN connections, etc. [18] Kruskal's algorithm is faster for the sparse graph.

II. LITERATURE REVIEW

The literature review has been done by studying various research papers. It helps in providing necessary information on the theoretical framework for conducting research work. Also, it will help in providing a summary, comparison, classification, and evaluation of a topic of interest. Many researchers are using the greedy algorithm in different techniques. A detailed literature review of greedy algorithms is given below.

Aditya Dev Mishra & Deepak Garg et al. analyzed the given problem i.e., type small numbers, and large numbers. After that, he applied the sorting algorithm by keeping in mind the minimum complexity, minimum comparison, and maximum speed. Also, discussed the advantages of the different sorting algorithms to choose the best algorithm. Various types of Sorting algorithms are developed in the past to make and improve sorting fast. No single sorting method is best for different situations. Various factors need to be considered while choosing a sorting algorithm. This includes the size of the list which needs to be sorted, the programming efforts and the number of words in main memory available, the size of disk or tape units, the extent to which the list is already ordered, and the distribution of values are some important factors.[19]

Ramcis N. Vilchez et al. presented a remedy for the noted deficiencies of the O(n2) sort algorithm for large data. Although selection sort is regarded as the most straightforward algorithm, it is also considered the second worst algorithm in terms of time complexity for large data. Several enhancements were conducted to address the inefficiencies of the selection sort. O (n2) time complexity is very efficient for a small list of elements. However, for enormous data, these algorithms are not efficient. A sorting algorithm refers to the arranging of numerical or alphabetical or character data in ascending or descending. However, for large data, these algorithms are very inefficient. This study presented a remedy for the noted deficiencies of the O(n2) sort algorithm for large data. Among the O(n2) algorithms, selection sort was the subject of the study considering its simplicity.[20]

Dinitz and Itzhak presented a new hybrid algorithm called Bellman-Ford–Dijkstra by combining Bellman-Ford and Dijkstra algorithms. This algorithm is for locating the shortest paths from a source node s in a graph G with general edge costs. To improve the runtime of the Bellman-ford algorithm and a sparse sharing of the destructive cost edges. They purposedamixed hybrid of Bellman-Ford and Dijkstra algorithms. The principle of the proposed algorithm is to repeat the Dijkstra algorithm multiple times without resetting the temporary value of d (v) to the vertices. They also suggested a new and simple proof that the Bellman-Ford algorithm can generate the shortest paths tree.[21]

Singh and Tripathi made comparative between two algorithms: Bellman-Ford and Dijkstra and discussed the results based on the number of nodes. The research enables us to define and suggest which algorithm is used in the shortest path problems for a specific variant. Their results show that with a very minimum quantity of nods, the Bellman-Ford algorithm is superior to Dijkstra's algorithm but Dijkstra is most effective for a massive variety of nodes.[22]

Rash and Tamimi described and compared the algorithms: Dijkstra, and Bellman-Ford for solving the shortest path problem. Their search shows that the Dijkstra algorithm is more effective in memory for sparse graphs because it does not involve the representation of the distance matrix as a dense matrix. The complexity of the Bellman-Ford algorithm concerning time is slower than the algorithm of Dijkstra. Dijkstra's algorithm can be used only with non-negative edge weights graphs. [23]

Wang et al. compared three algorithms that are Dijkstra, Bellman-Ford, and Floyd–Warshall algorithms based on time and space complexity. Their analysis showed that the Dijkstra algorithm is only useful in the shortest route issue of a single source and it is always optimized in a real implementation, such as heap optimization. The implementation of the Bellman-Ford algorithm is simple but the algorithm itself is inefficient for finding the shortest path. The Floyd-Warshall algorithm is the slowest and most inefficient space for redundancy while working with oversized different kinds of points and edges.[24]

Wang, H., Yu, Y et al.presented the shortest path planning methods of maze robot path planning, and algorithms Simulation results indicate that the application of the proposed path planning algorithm, the Dijkstra algorithm, can be correct and effective in mobile robot path planning.Dijkstra algorithm is used in robot path planning. The shortest path is selected in the process of the barrier. Simulation results prove the model valid; it can effectively solve the maze robot path planning.[25]

SamahW.Get al. compared Dijkstra's algorithm with the Bellman-Ford algorithm. Very popular algorithms called the Dijkstra algorithm and the Bellman-Ford algorithm are used to make a comparison between them based on complexity and performance in terms of shortest-path optimization. Our results show that Dijkstra is better than the Bellman-Ford in terms of execution time and more efficient for solving the shortest path issue, but the algorithm of Dijkstra works with non-negative edge weights. [26]

Artur Mariano et al.carried out a study on the performance-optimized realization of Prim's algorithm on reconfigurable hardware, which is increasingly present in such platforms.investigated two algorithmic variants and compare their performance against implementations on desktop-class and embedded CPUs. Results show that the raw execution time of an optimized implementation of Prim's algorithm is faster than the embedded processor.[27]

Feixue Huang et al. compared the two algorithms i.e., Prim's and Kruskal's algorithms. Minimum Spanning Tree is based on the basic principles of the Greedy act, namely, in a system, the tree structure is formed by n nodes and has n-1 edges, while the edges set is the combination of edges that has the minimum weight and guarantees no formation of loops. In the prim algorithm, the MST sets "grows" from a single root node and is none of the business of the network edges, so the prim algorithm is more suitable for the network MST with dense edges. The result shows that Prim's algorithm is faster than Kruskal's algorithm.[28]

Yu Wang et al. compared with the Prim algorithm, the Kruskal algorithm is more suitable for sparse-edged network MST. Kruskal algorithm considers every edge by its weight, and MST "grows" in a cluster, while the priority queues are used for storage and tables are used to store the clusters in sets, which has little to do with the nodes, when the map is nearly complete, its efficiency is not high compared to Prim algorithm.[29]

Sai Munzir et al. Kruskal's and Prim's algorithm will be used to determine the minimum weight of a complete broadcasting graph regardless of the minimum time unit $\lceil {}^{2}\log n \rceil$ and modified Prim's algorithm for the problems of the minimum time unit ⌈^{2}log n⌉ is done. As an example, in the case here, the training of trainer problem is solved using these algorithms.[30]

Krit Salahddine et al. compared the Kruskal algorithm and the Dijkstra algorithm, the following conclusions can be drawn: a strength that is the existence of weight sorting will facilitate the search for the shortest path, study also considered the characteristics of Kruskal's independent algorithm, as it facilitates and improves the formation of the path. The shortcoming of the Kruskal algorithm is that if the number of nodes is very high, it will be slower than Dijkstra's algorithm because it has to sort thousands of vertices first, then form a path. So, If the number of vertices is very large, it will be slower than Dijkstra's algorithm because it must sort thousands of vertices first, then form the path.[31]

III. COMPARATIVE ANALYSIS

This comparative study presents a complete analysis of greedy algorithm with its different types and these algorithms are compared to find the best greedy algorithm which is Dijkstra's algorithm based on different parameters like time complexity, and space complexity with their best, average, and worst case, and a comparison table is made to differentiate these algorithms with the help of different parameters. These parameters are merits, demerits, time complexity, and methods used in the algorithms.

Table 3.1 presents the comparative analysis of the different greedy algorithms for shortest path optimation. Table 3.0 contains five different shortest path algorithms, the parameters of analysis is Time complexity, the method used for the distance travel alongside Merits and Demerits. From Table 3.1 we can draw that however Dijkstra's, The algorithm has low complexity, it cannot hold negative Weights. The Bellman ford Algorithm can hold the negative weights but because of the changing network topologies, it does not scale well. Prim's algorithm is better than Krushkal's as it can deal with dense graphs, but the shortcoming it contains is that it searches the whole list of edges from the beginning if any new edge gets added. Kruskal's Algorithm runs faster in sparse graphs but it fails to detect the cycles in the directed graph. The advantage of using the selection sort algorithm is that it can work very well with small lists and there is no requirement for extra space for its work, however, when it deals with a large amount of data, it fails to perform up to the mark. Table 3.1 presents the comparative analysis of the different shortest path algorithms that the Greedy algorithm contains. Time complexity, the Method of distance travel, Merits, and demerits are some parameters that are chosen for the purpose of Comparision of these different algorithms. Observations reflect that the Dijkstra algorithm is much and commonly usedinreal-time applications.

Table 3.1: Comparison of various Greedy Algorithms

Algorithm	Method	Time complexity	Merits	Demerits
Dijkstra-based algorithm	The suggested strategy makes the algorithm appropriate for use in dynamic settings where distinct types of sensors can change the accessibility status of inner areas. [32]	O(|E|+|V|log|V|) using a binary heap. [38]	The main advantage is its low complexity.[43]	It is unable to handle negative weights. [48]
Bellman-Ford	The Bellman-Ford algorithm deals more effectively with a small number of nodes, and the Dijkstra algorithm is greater efficiency for a large wide of nodes.[33]	O (V * E) [39]	The capability of handling negative weights. [44]	It does not scale well because of changes in network topologies.[49]
Prim's algorithm	In the prim algorithm, the minimum spanning tree set "grows" from a single root node and is none of the business of the network edges, so prim's algorithm is more suitable for the network MST with dense edges.[34]	O (E log V) using binary Heap / O (E + V log V) using Fibonacci Heap [40]	The complexity of Prim's algorithm is better than Kruskal's algorithm. So, it is helpful when dealing with dense graphs. [45]	A list of edges has to be searched from the beginning if any new edge gets added.[50]
Kruskal's algorithm	Kruskal algorithm considers every edge by its weight, and MST "grows" in a cluster, while the priority queues are used for storage and tables are used to store the clusters in sets, which has little to do with the nodes, when the map is nearly complete, its efficiency is not high compared to Prim algorithm. [35]Thus, to compare with the Prim algorithm, the Kruskal algorithm is more suitable for sparse-edged network MST. [36]	O (E Log E or E Log V) [41]	Kruskal's algorithm runs faster in sparse graphs. It generates the spanning tree from the root vertex. [46]	It fails to detect the cycles in a directed graph.[51]

Selection Sort Algorithm	It is one of the easiest and most useful sorting algorithms for dealing with a small amount of data set. Even though it performs a lot of comparisons, it performs the least amount of data moving. Thus, if your data has small keys but a large data area, then selection sorting may be the quickest.[37]	O (N 2) [42]	It works very well on the small list and no additional storage is required. [47]	It has poor efficiency when dealing with a large amount of data.[52]

IV. CONCLUSION

This paper presented the comparative analysis in terms of shortest path optimization between different greedy algorithms. In this paper, we presented a detailed review and comparative study of greedy algorithms. On comparing all the papers, the one algorithm which has outperformed all others in most of the studies taken into consideration and that algorithm is Dijkstra's algorithm. Different algorithms are compared which are Dijkstra, Bellman-Ford, Selection Sort, Prim's, and Kruskal's algorithms to conclude which of them is more efficient for finding the shortest path between two vertices. Our results show that the Dijkstra algorithm is much and commonly used in real-time applications.

REFERENCES

[1] Liu, J., Zhao, C.C. and Ren, Z.G., 2016, August. The application of the greedy algorithm in real life. In Proc. Int. Conf. Manuf. Construct. Energy Eng. (MCEE) (pp. 1-7).

[2] Programiz (2022) Greedy Algorithm, Available at: https://www.programiz.com/dsa/greedy-algorithm (Accessed: 12/10/22).

[3] Jadoon, S., Solehria, S.F., Rehman, S. and Jan, H., 2011. Design and analysis of optimized selection sort algorithm. International Journal of Electric & Computer Sciences (IJECS-IJENS), 11(01), pp.16-22.

[4] Solehria, S.F., Rehman, S. and Jan, H., 2011. Design and analysis of optimized selection sort algorithm. International Journal of Electric & Computer Sciences (IJECS-IJENS), 11(01), pp.16-22.

[5] Bannister, M.J. and Eppstein, D., 2012, January. Randomized speedup of the Bellman-Ford algorithm. In 2012 Proceedings of the Ninth Workshop on Analytic Algorithmics and Combinatorics (ANALCO) (pp. 41-47). Society for Industrial and Applied Mathematics.

[6] Magzhan, K. and Jani, H.M., 2013. A review and evaluation of shortest path algorithms. International journal of scientific & technology research, 2(6), pp.99-104.

[7] Busato, F. and Bombieri, N., 2015. An efficient implementation of the Bellman-Ford algorithm for Kepler GPU architectures. IEEE Transactions on Parallel and Distributed Systems, 27(8), pp.2222-2233.

[8] Ramalingam, G. and Reps, T., 1996. An incremental algorithm for a generalization of the shortest-path problem. Journal of Algorithms, 21(2), pp.267-305.

[9] 1Seidel, R., 1995. On the all-pairs-shortest-path problem in unweighted undirected graphs. Journal of computer and system sciences, 51(3), pp.400-403.

[10] Noto, M. and Sato, H., 2000, October. A method for the shortest path search by extended Dijkstra algorithm. In Smc 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics.' cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0 (Vol. 3, pp. 2316-2320). IEEE.

[11] Makariye, N., 2017, May. Towards shortest path computation using the Dijkstra algorithm. In 2017 International Conference on IoT and Application (ICIOT) (pp. 1-3). IEEE.

[12] Barbehenn, M., 1998. A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. IEEE transactions on computers, 47(2), p.263.

[13] 1Ojekudo, N.A. and Akpan, N.P., 2017. An application of Dijkstra's Algorithm to the shortest route problem. IOSR Journal of Mathematics, 13(3), pp.20-32.

[14] Levi, G., 1973. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. Calcolo, 9(4), pp.341-352.

[15] Pettie, S. and Ramachandran, V., 2005. The shortest path algorithm for real-weighted undirected graphs. SIAM Journal on Computing, 34(6), pp.1398-1431.

[16] Kponyo, J.J., Kuang, Y., Zhang, E. and Domenic, K., 2013, October. VANET cluster-on-demand minimum spanning tree (MST) prim clustering algorithm. In 2013 International Conference on Computational Problem-Solving (ICCP) (pp. 101-104). IEEE.

[17] 1Subramanian, P., Ramkumar, N., Narendran, T.T. and Ganesh, K., 2013. PRISM: PRIority-based SiMulated annealing for a closed loop supply chain network design problem. Applied soft computing, 13(2), pp.1121-1135.

[18] Nirmala, G. and Uma, K., 2012. Fuzzy Shortest Route Algorithm for Telephone Line Connection. International Journal of Scientific and Research Publications, 2(8).

[19] Mishra, A.D., and Garg, D., 2008. Selection of best sorting algorithm. International Journal of intelligent information Processing, 2(2), pp.363-368.

[20] Vilchez, R.N., 2019. Bidirectional Enhanced Selection Sort Algorithm Technique. International Journal of Applied and Physical Sciences, 5(1), pp.28-35.

[21] Dinitz, Y., & Itzhak, R. (2017). Hybrid Bellman–Ford–Dijkstra algorithm. Journal of Discrete Algorithms, 42, 35–44. doi:10.1016/j.jda.2017.01.001

[22] Singh, J.B., Tripathi, R.C. (2018). Investigation of Bellman–Ford Algorithm, Dijkstra's Algorithm for suitability of SPP. IJEDR | Volume 6, Issue 1 | ISSN: 2321-9939

[23] AbuRyash, H.& Tamimi, A. (2015). Comparison Studies for Different Shortest path Algorithms. International Journal of Computers and Applications, Vol.14. 10.24297/ijct.v14i8.1857.

[24] WANG, X.Z. (2018). The Comparison of Three Algorithms in Shortest Path Issue. IOP Conf. Series: Journal of Physics: Conf. Series 1087 (2018) 022011 doi :10.1088/1742- 6596/1087/2/022011

[25] Wang, H., Yu, Y. and Yuan, Q., 2011, July. Application of Dijkstra algorithm in robot path-planning. In 2011 second international conference on mechanic automation and control engineering (pp. 1067-1069). IEEE.

[26] Samah, K.A.F.A., Sharip, A.A., Musirin, I., Sabri, N. and Salleh, M.H., 2020. Reliability study on the adaptation of Dijkstra's algorithm for gateway KLIA2 indoor navigation. Bulletin of Electrical Engineering and Informatics, 9(2), pp.594-601.

[27] Mariano, A., Lee, D., Gerstlauer, A. and Chiou, D., 2013, June. Hardware and software implementations of Prim's algorithm for efficient minimum spanning tree computation. In International Embedded Systems Symposium (pp. 151-158). Springer, Berlin, Heidelberg.

[28] Huang, F., Gao, P. and Wang, Y., 2009, November. Comparison of Prim and Kruskal on Shanghai and Shenzhen 300 Index hierarchical structure tree. In 2009 International Conference on Web Information Systems and Mining (pp. 237-241). IEEE.

[29] Wang, Y., 2009, November. Comparison of Prim and Kruskal on Shanghai and Shenzhen 300 Index hierarchical structure tree. In 2009 International Conference on Web Information Systems and Mining (pp. 237-241). IEEE.

[30] Garg, V.K., 2019. Comparison of Different Prim's Algorithm Techniques. Think India Journal, 22(12), pp.80-88.

[31] Chang, W.C., Salahddine, Chiu, Y.D. and Li, M.F., 2008, August. Learning Kruskal's Algorithm, Prim's Algorithm, and Dijkstra's Algorithm by a board game. In International Conference on Web-based Learning (pp. 275-284). Springer, Berlin, Heidelberg.

[32] Crauser, A., Mehlhorn, K., Meyer, U. and Sanders, P., 1998, August. A parallelization of Dijkstra's shortest path algorithm. In International Symposium on Mathematical Foundations of Computer Science (pp. 722-731). Springer, Berlin, Heidelberg.

[33] Dinitz, Y., & Itzhak, R. (2017). Hybrid Bellman–Ford–Dijkstra algorithm. Journal of Discrete Algorithms, 42, 35–44. doi:10.1016/j.jda.2017.01.001.

[34] Noto, M. and Sato, H., 2000, October. A method for the shortest path search by extended Dijkstra algorithm. In Smc 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics.' cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0 (Vol. 3, pp. 2316-2320). IEEE.

[35] Huang, F., Gao, P. and Wang, Y., 2009, November. Comparison of Prim and Kruskal on Shanghai and Shenzhen 300 Index hierarchical structure tree. In 2009 International Conference on Web Information Systems and Mining (pp. 237-241). IEEE.

[36] Moret, B.M. and Shapiro, H.D., 1991, August. An empirical analysis of algorithms for constructing a minimum spanning tree. In Workshop on Algorithms and Data Structures (pp. 400-411). Springer, Berlin, Heidelberg.

[37] Jadoon, S., Solehria, S.F. and Qayum, M., 2011. Optimized selection sort algorithm is faster than insertion sort algorithm: a comparative study. International Journal of Electrical & Computer Sciences IJECS-IJENS, 11(02), pp.19-24.

[38] Barbehenn, M., 1998. A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. IEEE transactions on computers, 47(2), p.263.

[39] Busato, F. and Bombieri, N., 2015. An efficient implementation of the Bellman-Ford algorithm for Kepler GPU architectures. IEEE Transactions on Parallel and Distributed Systems, 27(8), pp.2222-2233.

[40] Martel, C., 2002. The expected complexity of Prim's minimum spanning tree algorithm. Information processing letters, 81(4), pp.197-201.

[41] Li, H., Xia, Q. and Wang, Y., 2017. Research and improvement of the Kruskal algorithm. Journal of Computer and Communications, 5(12), p.63.

[42] Hayfron-Acquah, J.B., Appiah, O. and Riverson, K., 2015. Improved selection sort algorithm. International Journal of Computer Applications, 110(5).

[43] Samir Surya (2018) advantages and disadvantages of Dijkstra's algorithm in data structures, Available at: https://www.quora.com/What-are-advantages-and-disadvantages-of-Dijkstras-algorithm-in-data-structures (Accessed: 12/10/22).

[44] Study experts (2022) Dijkstra's algorithm, Available at: https://studyexperts.in/blog/tag/advantages-of-bellman-ford-algorithm-over-dijkstras/ (Accessed: 12/10/22).

[45] SNEHASISH KONGER (2022) Which One Is A Better Approach, Available at: https://scientyficworld.org/prims-algorithm-and-kruskals-algorithm/ (Accessed: 12/10/22).
[46] JanuszSzpilewski (2022) Advantages of Krushkal algorithm, Available at: https://www.quora.com/What-are-the-advantages-of-the-Kruskal-algorithm (Accessed: 12/10/22).
[47] Joe Wandy (2018) The Advantages & Disadvantages of Sorting Algorithms, Available at: https://sciencing.com/the-advantages-disadvantages-of-sorting-algorithms-12749529.html (Accessed: 12/10/22).
[48] Appelman-Dijkstra, N.M. and Papapoulos, S.E., 2018. Clinical advantages and disadvantages of anabolic bone therapies targeting the WNT pathway. Nature Reviews Endocrinology, 14(10), pp.605-623.
[49] AbuSalim, S.W., Ibrahim, R., Saringat, M.Z., Jamel, S. and Wahab, J.A., 2020, September. Comparative analysis between Dijkstra and bellman-ford algorithms in shortest path optimization. In IOP Conference Series: Materials Science and Engineering (Vol. 917, No. 1, p. 012077). IOP Publishing.
[50] Geeks for Geeks (2022) Minimum spanning tree, Available at: 50. https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/ (Accessed: 12/10/22).
[51] Dr. Sagunthala (2022) Disadvantages of kruskalsalg, Availableat:https://www.coursehero.com/file/p51ivf7u/Disadvantages-of-Kruskals-Alg-It-always-produces-a-MS-have-the-minimum-cost-T/ (Accessed: 12/10/22).
[52] Konologist (2022) Advantages And Disadvantages Of Selection Sorting Techniques, Available at: https://knologist.com/what-advantages-and-disadvantages-of-selection-sorting-techniques/ (Accessed: 12/10/22).