HBSP Method to Enhance Optimum Shortest Path of the Graphs

Sadhana H. Barkund[1] Ashutosh Sharma[2], H. R Bhapkar[3]

[1]sadhana.barkund@gmail.com, [2]drashutosh8200@gmail.com, [3]hrbhapkar@gmail.com

[1,2]Department of Mathematics, Jagdishprasad Jhabarma Tibrewala University, Rajasthan, India

[3]Department of Mathematics, MIT ADT University's, MIT School of Engineering, Pune, India

1. Abstract

A network in mathematics is merely a group of connected objects that is formalized through the use of graphs. The shortest path or route problems can be applied to search for a classic sequence of options with a fixed target state. In this paper, a new philosophy on the shortest path is defined by a pragmatic approach. It deals with the possible variable scaling of the merits without changing the ranks. The ranks are constant, but the scales are varied. A great number of shortest-path algorithms are studied in the literature. This work defines a new HB Optimum Method, an innovative and more suitable method. There are four types of HB Optimum methods- One-sided Increment, One-sided Decrement, Two-sided Change in two elements, and Two-sided change to n elements. The HB shortest path method is defined by taking the existing shortest path algorithms as a base or input, and weights of distinct edges of the first and the second path are revised without changing the ranks of the paths. The generic mathematical model and an algorithm of the same are presented in this work. This paper explains two new concepts, the Maxmin path and the Minmax path of any weighted connected graph. Moreover, HBSPM is more advanced than Dijkstra's and Prim's Algorithms. The HBSPM works well for Dense and Sparse graphs also. This work has numerous applications in science, technology, and real-life circumstances. The HBSPM helps to decide the increments/decrements in the price of an article, profit margins, wireless networks, possible variation in grades, and many more.

Keywords: Shortest path algorithm, HB Optimum Method, Dijkstra's Algorithm, Prim's Algorithms

DOI Number: 10.14704/nq.2022.20.13.NQ88093 **Neuro Quantology 2022; 20(13):713-721**

2. Introduction

Mathematics has been crucial in every aspect of science and technology. Almost all branches of mathematics have been used for the development of mankind and to protect the life of an individual from abrupt circumstances. In the present COVID-19 pandemic era, mathematics helps to shorten the pandemic and guides to enrich harmony in the world. [1, 2, 3]

Any type of virus is defined using the partite graphs, and growth rates are decided by the philosophy of the corona product of graphs [4]. But the corona product of graphs was defined in 1970 by great mathematicians Frucht and Harary [5].

An approach for finding the shortest path or route between two graph vertices is known as the shortest-path or route algorithm. In shortest path algorithms, a graph may be directed, undirected, or mixed [6, 7]. In a network the problem of searching the shortest path or route where the edges are road segments and the vertices are intersections with weights based on segment length can be used.

When looking for routes between real locations, the shortest path or route algorithms are utilized. This shortest path or route problem always said the min-delay path problem in networking or telecommunications circles is sometimes paired with the widest path problem. Plant and facility planning, robotics, transportation, and VLSI design are further applications that are frequently investigated in operations research. Using the shortest path approach, word ladder puzzles can be resolved. Find the shortest route between machines 1 and

2 in a network. To discover the shortest route between two places on a map, video games use shortest path algorithms. The shortest path or route problem in a social network is locating the route between two people [7].

3. Literature Review

Author Madhumita Panda presented a survey of the various shortest paths calculated using different algorithms & how it has been used for different applications [8]. This paper's major goal was to research and test several shortest path algorithms that are applied to navigation systems. Prior to introducing the algorithm and emphasizing the efficiency of its searching activity, Abdelfattah IDRI et al. offered an overview of the current solutions [10].The principles of the shortest path problem were developed and recognized by the author Kairanbay Magzan and Hajar Mat J ani. They described the fundamental ideas and practical applications of the Dijkstra's, Floyd-Warshall, Bellman-Ford, and Genetic algorithms. They assessed each algorithm and then provided the findings [11, 12, and 13]. A new shortest path or route algorithm for undirected graphs was assessed for practical effectiveness by Seth Pettie et al. Since robust shortest path algorithms were the main topic of this research, it is important to define the word clearly. The Pettie-Ramachandran shortest path or route algorithm for undirected graphs has been simplified, and it has been put to the test against Dijkstra's algorithm [11].

In addition to earlier presented algorithms for distance evaluation, Haysam Selim and Justin Zhan have proposed a quick method and algorithm for the approximation and reduction of a large network as well as a quick estimate computation for the shortest path or route. This article aims to reduce the network that will have an effect on finding the shortest path or route in order to complete the study as quickly as possible [10, 14]. In a network with few distinct edge lengths and edges of positive lengths, James B. Orlin and M. Williamson proposed an effective way to execute Dijkstra's algorithm. This work [11] investigated the SSSPP for tiny K of unique edge lengths.

The National City traffic advisory methods were established and implemented by Arun Kumar et. al. In order to establish the storage structure of transportation networks and determine the shortest path or route among two cities, they attempted to examine the Dijkstra algorithm and Floyd algorithm executed in Microsoft Visual Studio in this research [11, 12].

In this research, A.H. U.C. and Eneh Due to its quick & reliable performance traits with linear run time complexity, Arinze tested a number of shortest path algorithms, including Dijkstra's SPA executed with double bucket data structure, to construct their route guiding application for the best results.Other shortest path algorithms evaluated included Bellman-Ford, Floyd-Warshall, A* search, restricted search, and Gallo Pallotino graph growth algorithm. In this work, a web-based vehicle route planning tool is suggested. The findings of a thorough computational investigation on dynamic methods for all pairs of shortest path or route problems were provided in this paper by Camil Demestrescu and Giuseppe F. Italiano. They have tested, developed, and assessed three different dynamic shortest path algorithms in this paper: On machines with low memory bandwidth, D-RRL appeared to be more advantageous.In this paper, Thorat Surekha and Rahane Santosh explained & implemented Dijkstra’s algorithm & Bellman-Ford algorithm. Regarding their overall performance in resolving the shortest path problem, algorithms are satisfactory. There is only one result produced by any of these algorithms [11, 12]. To address both issues, Lin Qi and Markus Schneider in this study proposed two novel algorithms, RSP (Realtime Shortest Path) and RSP+ (Realtime Shortest Path Plus). To guarantee a constant response time while searching the shortest route from any given node to a necessary set of destinations, they pre-processed the network [14].

This article, according to authors Ivan Beker, Vesna Jevtic, and Datibor Dobrilovic, focuses on improving efficiency and business processes to cut expenses of all kinds, particularly by removing waste. This paper's major objective was to study the application of one of the various models of rules selection now in use, the shortest path algorithms for forklift routing optimization. The issue of choosing a few web services from the candidate list was investigated by authors Jie Yang and Xingdong Yin. They created a single source of web services for "bright things" that focus on the shortest path problem. They presented the SSMTkSP problem & its corresponding algorithms. They analyzed

the performance of their algorithms. They showed experimental results & reviewed the related work.

In this paper, author Rhyd Lewis reviewed many well-known algorithms for resolving the shortest route problems in graphs with edge weights. He subsequently concentrated on a different version of this issue where extra fines are levied at the Vertices; these penalties can be applied to control things like traffic jam delays and delays caused by transfers on public transportation. By comparing three strategies—shortest Dijkstra's path algorithm, Restricted search algorithm, and A* algorithm—the author Liang Dai researched the single source shortest path or route problem and hopes to draw some general conclusions. The three algorithms were implemented & visually demonstrated [10, 14]. For a complete understanding of the Graphs, the coloring of graphs, and shortest path algorithms, the readers can refer [5-7, 21-23]

4. Basic Terms

- **Connected Graph:** A graph H having a path from a point (vertex or node) to all remaining vertices is known as a connected graph. Else, it is known as a disconnected graph or not connected graph.
- **Weighted Graph:** A graph H in which each edge has assigned a number (weight) is said to be a weighted graph.
- **First Shortest Path:** A path or route of a connected graph H is said to be the first shortest path of H if it has the minimum weight among all existing paths or routes of H. The path weight is the addition of the weight of total edges of the path.
- **Second Shortest Path:** A path of a connected graph H is said to be the second shortest path of H if its weight is more than the weight of the first shortest path or route and less than the weight of all remaining paths, excluding the first shortest path of H. Similarly, the nth shortest path can be defined.

5. Main Work

In this work, a new philosophy of the ranking of merits is defined by a pragmatic approach. It deals with the possible variable scaling of the merits without changing the ranks. The ranks are constant, but the scales are varied.

5.1 HB Optimum Method

In real life, we come across situations in which there is a notable difference between the first smallest and the second smallest numbers or the first greatest and the second greatest numbers of the data. For example, in a class of 100 students, suppose the percentages obtained by the first topper T_1 and the second topper T_2 are 92.5% and 86%, respectively. How many alternate possibilities are there so that the ranks of the topper students remain the same? This philosophy is known as the HB Optimum method. Depending upon the nature of percentages of rankers, there are four types of HB Optimum methods.

i. **One-sided Increment:** If T_2 scores a percentage more than 86 % but less than 92.5 %, then also the first and the second ranks of students remain the same. The percentage of T_1 is constant, and T_2 is flexible. It is known as the left increment or one-sided increment.

ii. **One-sided Decrement:** In this case, even if T_1 scores a percentage less than 92.5 % but more than the percentage of T_2. So, the first and the second ranks of students remain the same. The percentage of T_2 is constant, and T_1 is variable (not constant). It is called the right decrement or one-sided decrement.

iii. **Two-sided change in two elements:** In this type, if there is a decrement in the percentage of T_1 and an increment in the percentage of T_2, their ranks are the same. The percentages of both rankers are flexible, but their ranks are the same. It is called two-sided change.

iv. **Two-sided change to n elements:** This type deals with change in n elements by two sides as per requirements, without changing the sequence of ranks. This type is applicable where two or more ranks are essential.

This philosophy applies to numerous real-life problems or circumstances. We come across the same philosophy in the graph theory, particularly in the shortest path algorithms. Several algorithms find the shortest paths among all vertices of the weighted graph. All such algorithms deal with the first shortest path or route among each pair of vertices. The second shortest path is not considered in all existing literature on the algorithms. The weight of each edge is fixed or constant.

5.2 HB Shortest Path Method (HBSPM)

In the HB Shortest Path Method, the weights of the edges in the first and second shortest paths of the weighted connected graphs are flexible. The weights can be changed as per need. Let us consider the weighted connected graphs having the first and second shortest paths, P_1 and P_2, respectively. Any existing algorithm of literature can be applied to obtain the first and second shortest paths. Therefore, $w(P_1) < w(P_2)$.

These two paths may or may not have some common edges. Now, as per the above increments, reduce and/or increase the weights of those edges in the first or second paths but not in both. So, it is of the types of one-sided increment, one-sided decrement, or two-sided change depending upon the philosophy used. This method is known as the **HB shortest path method.**

5.3 Mathematical Model of HBSPM for One-sided Increment

Let H be any weighted connected graph. Find the first and second shortest paths among the desired vertices by using any existing algorithms, i.e., Dijkstra's algorithm, Prim's algorithm, Kruskal's algorithm, etc.
Let P_F be the set of edges in the first shortest path with length n.

$P_F = \{ E_{xi} ; 1 \le i \le n\}$

Where E_{xi} denotes edges of the first shortest path. Let $W(P_F)$ be the weight of the path P_F.
Then, to find the second shortest path's edges. Let P_S be the set of the second shortest path's edges. Therefore, $P_S = \{ E_{yi} ; 1 \le i \le m\}$.
Where E_{yi} denotes the second shortest path's edges. Let $W(P_s)$ be the weight of the path P_s.
Let P be the set of edges in P_F but not in both, i.e. $P_F \cap P_S$.

$P = P_F - \{ P_F \cap P_S \}$
$= \{E_{zt} / 1 \le t \le n\}$

Let, $D = W(P_S) - W(P_F) - 1$

Case 1: If $E_{zt} \varepsilon P$, $W(E_{zt}) = W_t$
Then, the revised weight of the edge is
$W^*(E_{zt}) = W_t + D$

Case 2: Make at most t parts of D say D_1, D_2 …Dt such that $D_1 + D_2 + D_3 + \ldots . D_t = D$.
And
$W^*(E_{zt}) = W_i + D_j$; i=1,2,…..n, j=1,2,3……t

In this way, revise the weights of the first path's edges without changing the rank.

5.4 Mathematical Model of HBSPM for One-sided Decrement

Let H be any weighted connected graph. Find the first and second shortest paths among the desired vertices using the existing algorithms. Let P_F and P_S be the first and the second shortest paths among vertices of the weighted graph.
In this method, perform the similar process like Mathematical Model of HBSPM for one-sided increment. Let P be the set of edges which are in P_S but not in P_F. After finding the difference D, there are two cases.

Case 1: If $E_{zt} \varepsilon P$, $W(E_{zt}) = W_t > D$
Then, the revised weight of the edge is
$W^*(E_{zt}) = W_t - D$

Case 2: Make at most t parts of D say D_1, D_2 …Dt such that $D_1 + D_2 + D_3 + \ldots . D_t = D$.
And
$W^*(E_{zt}) = W_i - D_j$; j=1,2,3……t

In this way, revise the weights of the second shortest path's edges without changing the rank.

Maxmin: - Let $W(P_F)$ be the weight of the path P_F and $W(P_s)$ be the weight of the path P_s.

A path is said to be a local minimum path if its revised weight is less than the weight of the second shortest path. There are several local minimum paths according to the distributions of the weights to edges in the first shortest path. The weight of the greatest path among all possible local minimum paths is called the Maxmin of the first shortest path of the graph.

Therefore, Maxmin= Max $\{ W(P_{Fi}) / W(P_{Fi}) < W(P_s) \}$.

In general, the maxmin is associated with the one-sided increments of the HB optimal method.

Minmax: - Minmax is associated with the one-sided decrement of the HB optimal method. Let G_F and G_s be the first and the second greatest paths, respectively. By the analogy of the maxmin, the Minmax is defined as follows.

Minmax = Min $\{ W(G_{Fi}) / W(G_{Fi}) > W(G_s) \}$.

5.5 Algorithm of HBSPM

Considering all algorithms' properties, the steps of the proposed algorithm of HBSPM are given below.

1. Start
2. Select the first and second shortest paths using any existing algorithm.
3. Declare Array EF and ES to store first and second shortest path edges, respectively.
4. Declare Array PEF and PES to store a set of edges from EF and ES Array.
5. Declare Array WEF and WES to store weight in the First and Second shortest path edges, respectively.
6. Calculate the weight for edges in the PEF and PES Array to store it in the variable WPEF and WPES.
7. Declare Array PEZ to find the set of edges which are in the PEF Array, but not in both the PEF and PES Array.
8. Store the result of the declared Array PEZ in step 7 and the weight of edges in WPEZ.
9. Calculate the difference between WPES and WPEF. Also, Declare variable D and assign the calculated difference -1. (i.e. WPES-WPEF-1).
10. If PEZ contains only one edge, then calculate the new weight. Calculate the weight of PEZ and store it in variable WPEZ. Calculate the total revised weight by adding D and WPEZ. Store it in W (i.e. W=WPEZ +D).
11. If PEZ contains more than one edge, then distribute the weight difference D. Distribute weight D on each edge as a variable in PEZ like D_1+ D_2+ D_3+.....+ D_t= D. Calculate revised weight for each edge in PEZ using the WPEZ array. Revised weight of the edge in PEZ is calculated like below, and stored in a new array WPEZR. For i=0; i< count of edges in PEZ. WPEZR[i]=WPEZ[i] + DD[i]. Calculate the sum of weights in WPEZR and store it in variable W as the new revised weight.
12. End

6. Illustration for one-sided increment

Consider a weighted undirected graph. In this graph, first, take a path or route from Node A to Node M. While going from path A-M, obtain the shortest distance path or route and second shortest path or route. After finding the shortest and second shortest paths, increase the weight to less than one of the second shortest paths. But, while increasing the weight of the edges, remember that the path does not change.

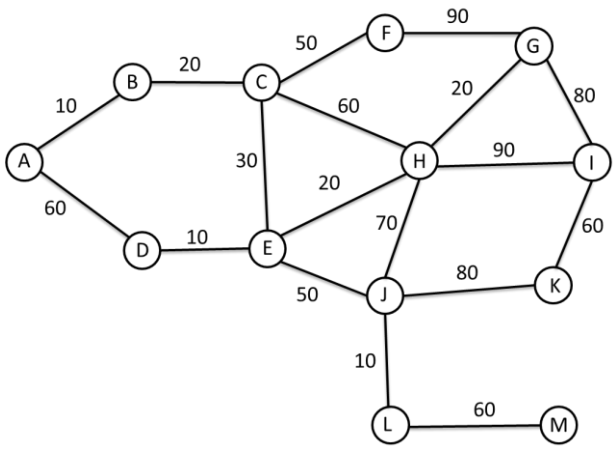

Figure 1.1 Undirected weighted graph

To find the weight between Nodes A to M through various paths.

Path (A-M)	Weight of path
A – D – E – J – L - M	190
A – B – C – E – J – L - M	180

By observing the above weights, A – B – C – E – J – L - M is the shortest path whose weight is 180, and A – D – E – J – L - M is the second shortest path whose weight is 190. So, increase the weight less than the second shortest path. It means we can increase the weight of A – B – C – E – J – L - M up to 189. But, at the same time, remember that the path should not be changed. First, check the edges of the shortest and second shortest paths to check whether they are same or different. If any edge is the same, skip that particular edge and take the remaining edges.

There is only one second shortest path. So first, find the shortest path's edges.

$E_{xi}= E_1$=A - B, $E_{xi}=E_2$=B - C, $E_{xi}=E_3$=C - E, $E_{xi}=E_4$=E - J, $E_{xi}=E_5$=J - L, $E_{xi}=E_6$=L - M

P_F = { E_{xi} } (Here, P_F be the set of edges in the first shortest path.)

= {A - B, B - C, C - E, E - J, J - L, L - M}

Then, to find the second shortest path's edges.

$E_{yi}=E_1$=A-D, $E_{yi}= E_2$= D-E, $E_{yi}=E_3$= E-J, $E_{yi}=E_4$=J-L, $E_{yi}=E_5$=L-M

$P_S = \{ E_{yi} \}$ (Here, let P_S be the set of edges in the second shortest path.)

= {A - D, D - E, E - J, J -L, L - M}

Here,

$P = P_F - \{ P_F \cap P_S \}$ (Here, P is the set of edges which are in P_F but not in both i.e. $P_F \cap P_S$.)

= {A - B, B - C, C - E, E - J, J - L, L - M}-{E - J, J - L, L -M}

= {A - B, B - C, C - E}

So, take edges A - B, B - C, and C - E to increase the weight. Here, we will use HBSPM for One-sided Increment.

$D = W(P_S) - W(P_F)-1$

Here, $W(P_s)$ be the weight of the path P_s.

$W(P_F)$ be the weight of the path P_F.

D= 190-180-1= 9

Now, increase the weight of the shortest path's edges so that the first shortest path remains the first. Increase the weight 9 on the A-B edge. After increasing the weight, the first shortest path remains the same. This way, divide this weight (D) on the edges of set P.

These ways increase the weight. But, the path should not be changed. Next, take another path and apply the same rule to that path also. In the same way, you can show the HBSPM for one-sided decrement by illustration.

7. Dijkstra's Algorithm

Negative weight cycle graphs are not compatible with this technique. Directed and undirected graphs exist in Dijkstra's algorithm. Dijkstra's algorithm always shows the shortest route for positive edge weights.

Algorithm

i. Create a set known as "shortest path tree set" (sptSet) that records the vertices that are part of the tree with the shortest distance to the source.
ii. Finalized. This set is initially empty.
iii. Give each vertex in the input graph a weight. Set all weight to INFINITE at the beginning. Set the start vertex's weight to 0.
iv. While not all vertices are present in sptSet

A) Choose a vertex u with a minimum distance value that is absent from the sptSet.
b) Append u to the sptSet.
c) Upgrade the weight among each of u's neighbouring vertices. Iterate through all nearby vertices to upgrade the weight. If the weight of the edge u-v plus the weight of u (from source) for each neighbouring vertex v is less than the weight of v, then the weight of v should be updated.

Dijkstra's Algorithm working steps-

- Initialize distances according to the algorithm.
- Determine the first node and the distances between it and nearby nodes.
- Choose the subsequent node with the smallest distance, and then repeat the calculations for nearby nodes.
- The outcome of the shortest-path tree.

8. Prims Algorithm

Prim's algorithm is a greedy method for searching the least spanning tree for a weighted undirected graph. In other words, it selects a subset of the edges that compose a tree, containing each vertex, and ensuring that the weight of every edge taken together in the tree is kept to a minimum.

Algorithm

i. Use a single vertex drawn at random from the graph to start a new tree.
ii. Choose the edge that connects the tree to vertices that are not presently in the tree that has the lowest weight, and then add it to the tree to grow it by one edge.
iii. Redo step 2 (till all vertices are in the tree).

Beginning with a single vertex, Prim's algorithm creates edges by adding the ones with the fewest weights until the objective is accomplished. The steps to put the prim algorithm into practice are as follows:

- Pick a random vertex as the first node of the minimal spanning tree.
- Find the smallest edge among all those joining the tree to fresh vertices, and then include it in the tree.
- Redo step 2 as essential to find a minimum spanning tree..

9. Comparison with Dijkstra's and Prim's algorithms

The one-to-one comparison of the Algorithm of HBSPM with Dijkstra's and Prim's Algorithm is as follows.

9.1 With Dijkstra's Algorithm

- This algorithm and Dijkstra's algorithm don't support a negative weight cycle.
- This algorithm and Dijkstra's algorithm don't support negative weight edges.
- In some cases, the Maxmin path and Minmax path are the same in this algorithm, but the Minmax path and Maxmin path are not present in Dijkstra's algorithm.
- This algorithm and Dijkstra's algorithm work out the single-source shortest path problem and all pair shortest path problems.
- Dijkstra's algorithm can run on undirected and directed graphs, but this algorithm runs on undirected graphs only.
- In this algorithm, if the difference between the second and first shortest paths is not more than 1, then we can't increase the weight. The second shortest path is not present in Dijkstra's algorithm.

9.2 With Prim's Algorithm

- This algorithm works well for Dense and Sparse graphs, whereas Prim's algorithm works well for dense graphs.
- This algorithm doesn't support negative weight edges, but Prim's algorithm supports a negative weight edge.
- In some cases, the Maxmin path and Minmax path are the same in this algorithm but Minmax path and Maxmin path are not present in Prim's algorithm.
- This algorithm and Prim's algorithm run on undirected graphs only.
- Prim's algorithm makes the minimum spanning tree, whereas this algorithm doesn't make the minimum spanning tree.
- In this algorithm, if the difference between the second and first shortest paths is not more than 1, then we can't increase the weight. The second shortest path is not present in Prim's algorithms.

10. Applications of HBSPM

There are more applications in the HBSPM. Some of them are as follows.

- This algorithm is used by the dealers to increase the cost of the product.
- It is used to find the maximum profit (by using first max and second max).
- This algorithm is used in the quotation of any product (Car, Fridge, House, Paint etc.).
- It is used to find student's class toppers.
- It is helpful to search the locations on the map.
- It is used in the telephone network.
- This algorithm is essential for computer networks like the Internet.

11. Conclusion

This paper has presented a new method to enhance the Optimum Shortest Path of the graph called as the HB Optimum method. The proposed method has explained four types- One-sided increment, One-sided decrement, two-sided change in two elements, and Two-sided change to n elements. The generalized formula and an algorithm of the method are given with illustrations. The comparison of the new method with Dijkstra's algorithm and Prim's Algorithm is explained meticulously. The new method would be helpful in many applications such as deciding the price of an article, internet connectivity, telephonic networks, etc.

Conflict of Interest: Authors have no conflict of interest.

Funding: The authors did not receive support from any organization for the submitted work.

Ethical Conduct: The work presented in the paper is the author's own original work and has not been communicated or published elsewhere.

Data Availability Statements: Not applicable.

Informed Consent Statement: Not available.

12. References

1. Bhapkar, H.R., Mahalle, P.N., Shinde, G.R. and Mahmud, M. , 2021. Rough sets in COVID-19 to predict symptomatic cases. In: K.C. SANTOSH and A. JOSHI, eds., COVID-19: prediction, decision-making, and its impacts. Lecture notes on data engineering and communications technologies (60). Singapore: Springer, Singapore, pp. 57-68. ISBN 9789811596810.

2. Bhapkar, H. R., Mahalle, P. N., Dey, N., & Santosh, K. C. (2020). Revisited COVID-19 Mortality and Recovery Rates: Are we Missing Recovery Time Period?. Journal of medical systems, 44(12), 202. https://doi.org/10.1007/s10916-020-01668-6
3. Shinde G.R., Majumder S., Bhapkar H.R., Mahalle P.N. (2022) Impact of COVID-19 Pandemic on Education Sector. In: Quality of Work-Life During Pandemic. Studies in Big Data, vol 100. Springer, Singapore. https://doi.org/10.1007/978-981-16-7523-2_4
4. Bhapkar H.R., Mahalle P.N., Dhotre P.S. (2020) Virus Graph and COVID-19 Pandemic: A Graph Theory Approach. In: Hassanien AE., Dey N., Elghamrawy S. (eds) Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach. Studies in Big Data, vol 78. Springer, Cham. https://doi.org/10.1007/978-3-030-55258-9_2.
5. Harary, Frank (1969). Graph Theory. Reading, Massachusetts: Addison-Wesley.
6. Bondy, J. A.; Murty, U. S. R. (2008). Graph Theory. Springer. ISBN 978-1-84628-969-9.
7. Deo, Narsingh (1974). Graph Theory with Applications to Engineering and Computer Science (PDF). Englewood, New Jersey: Prentice-Hall. ISBN 0-13-363473-6.
8. Abraham, Ittai; Delling, Daniel; Goldberg, Andrew V.; Werneck, Renato F. research.microsoft.com/pubs/142356/HL-TR.pdf "A Hub-Based Labeling Algorithm for Shortest Paths on Road Networks." Symposium on Experimental Algorithms, pages 230–241, 2011.
9. Derniame, Jean Claude; Pair, Claude (1971), Problèmes de cheminement dans les graphes (Path Problems in Graphs), Dunod (Paris)
10. Olya, Mohammad Hessam (2014). "Applying Dijkstra's algorithm for general shortest path problem with normal probability distribution arc length." International Journal of Operational Research. **21** (2): *143–154. doi:*10.1504/IJOR.2014.064541.
11. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 24.3: Dijkstra's algorithm". Introduction to Algorithms (Second ed.). MIT Press and McGraw–Hill. pp. 595–601. ISBN 0-262-03293-7.
12. *Floyd, Robert W.* (June 1962). "Algorithm 97: Shortest Path". *Communications of the ACM*. **5** (6): 345. *doi*:*10.1145/367766.368168*.
13. Bellman, Richard (1958). "On a routing problem." Quarterly of Applied Mathematics. **16**: 87–90. doi:10.1090/qam/102435. MR 0102435 .
14. *Zeng, W.; Church, R. L. (2009).* "Finding shortest paths on real road networks: the case for A*." *International Journal of Geographical Information Science.* ***23****(4): 531–543.* doi*:*10.1080/13658810801949850*.* S2CID 14833639*.*
15. Edelkamp, Stefan; Jabbar, Shahid; Lluch-Lafuente, Alberto (2005). *"Cost-Algebraic Heuristic Search" (PDF)*. The Twentieth National Conference on Artificial Intelligence (AAAI) Proceedings: 1362–1367.
16. Cook, Stephen A. (1985), "A Taxonomy of Problems with Fast Parallel Algorithms," Information and Control, **64** (1–3): 2–22, doi:10.1016/S0019-9958(85)80041-3.
17. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 24.3: Dijkstra's algorithm". Introduction to Algorithms (Second ed.). MIT Press and McGraw–Hill. pp. 595–601. ISBN 0-262-03293-7.
18. Dijkstra, E. W. *(December 1959),* "A note on two problems in connexion with graphs" (PDF)*, Numerische Mathematik, **1** (1): 269–271,* CiteSeerX 10.1.1.165.7577, doi*:*10.1007/BF01386390, S2CID 123284777.
19. Prim, R. C. *(November 1957),* "Shortest connection networks And some generalizations", Bell System Technical Journal*, **36** (6): 1389–1401,* Bibcode*:*1957BSTJ...36.1389P, doi*:*10.1002/j.1538-7305.1957.tb01515.x.

20. Gass, Saul; Fu, Michael (2013). Gass, Saul I; Fu, Michael C (eds.). "Dijkstra's Algorithm". Encyclopedia of Operations Research and Management Science. Springer. **1**. *doi:10.1007/978-1-4419-1153-7*. *ISBN 978-1-4419-1137-7* – via Springer Link.

21. Bhapkar, H. R., and J. N. Salunke. "Proof of four colour map theorem by using PRN of graph." J Bullet Soc Math Serv Stand 3.2 (2014): 35-42.

22. A A Bhange and H R Bhapkar *"Perfect Coloring of Graph with its kinds" J. Phys.: Conf. Ser. 1663 012024, 2020.*

23. Bhange, A. A., and H. R. Bhapkar. *"α, β Colouring of Graphs and Related Aspects." J Emerg Technol Innovative Res (2019): 6-4.*

Biography of Authors:

Mrs. Sadhana Hambirrao Barkund is pursuing her Ph. D. (Graph Theory) in Mathematics at Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan.She has registered copyright of her research method. She has published two research papers. She has 8 years of teaching experience. She has attended National and International Conferences.

Email: sadhana.barkund@gmail.com

Dr. Ashutosh Sharma has completed his Ph.D. from MDU Rohtak(Haryana) in 1999 . He has worked with the Haryana Government for more than 31 years. During that time, he taught mathematics in UG and PG colleges of different districts of Haryana. In last 5 years in government PG College for women sec 14 Panchkula, where he got the opportunity to work as officiating Principal and Vice Principal. He is retired as Principal from GCG Kurukshetra. He has guided to 12 M.Phil. And 6 Ph.D. students. He has good experience of educational administration. He has published 12 research papers

Email: drashutosh8200@gmail.com

Dr. H. R. Bhapkar has completed his Ph. D. (Graph Theory) in 2016. He is working as Professor in Mathematics at MIT Art, Design, and Technology University's MIT School of Engineering, Pune. He has published 38 research papers in national and international journals and conferences. He has developed his own academic credit system and defined a universal conversion formula to convert CGPA to an equivalent percentage. He has authored 141 books on Engineering Mathematics for different universities in India. He is the Research Guide in Mathematics at MIT Art, Design and Technology University, Pune. Additionally, he is a Board of Studies Member at MIT ADT University and Modern Arts, Commerce and Science College, Pune. He has 20 years of teaching experience.

Email: haribhau.bhapkar@mituniversity.edu.in