

OPTIMIZING PATHFINDING: A NOVEL ALGORITHM FOR GRAPH THEORY

Aijaz Ahmad Magray¹, Ajjaz Maqbool Dar², Firdous Ahmad³, Aafaq A. Rather^{4,*}, R. Vijaykrishnaraj⁵, D. Vedavathi Saraja⁶, Syed M. Parveen⁷, Raeesa Bashir⁸

¹Department of Mathematics, Govt. Degree College Premnath Dogra, Samba 184121, India

²Jammu and Kashmir Institute of Mathematical Sciences Srinagar -190008, J&K, India.

³Department of Electronics & IT, Cluster University (S.P College), Srinagar 190006, J&K, India

^{4,*}Symbiosis Statistical Institute, Symbiosis International (Deemed University), Pune-411004, India

⁵Department of Mathematics, Bharath Institute of Higher Education and Research, Chennai, India

^{6,7}Department of Statistics, Maris Stella College, Vijayawada, India

⁸Department of Statistics, Cluster University, Jammu and Kashmir, India

¹magrayaijaz2@gmail.com, ²ajjazmaqbool013@gmail.com,

³firdousahmed15@gmail.com, ^{4,*}Corresponding author: aafaq7741@gmail.com, ⁵drvjmaths@gmail.com,

⁶sarajayoganand@gmail.com, ⁷parveen@marisstella.ac.in, ⁸raisabashir52@gmail.com

Abstract

In this paper, a new proposed algorithm called the Pentacode-algorithm is successfully designed and implemented to solve the space and time problems associated with previous algorithms. Several parameters, such as Shortest Distance (SD), Time to Search Shortest Distance (TSSD), and Loss of Information Packets (LIP) have been checked and verified using the proposed algorithm. The simulation results achieved from the MATLAB tool authenticate the correct functionality of the proposed Pentacode-algorithm. Advance improvements have been proved in the proposed algorithm in terms of the space search and time issues. In addition, First-Node-Die (FND), Half-Node-Die (HND) and Last-Node-Die (LND) are determined in our results. The comparative results have shown that the Pentacode-algorithm is more proficient than the previous algorithm. This Pentacode-algorithm can be applied to warless communication systems, network routing and protocol multidirectional transmission and Quantum computing research.

Keywords: Pentacode-algorithm, Active Nodes, Shortest Distance (SD), Time to Search Shortest Distance (TSSD), Loss of Information Packets (LIP)

1. Introduction

The well-studied research topic in graph theory is the shortest-path. The shortest path problem is a major issue for finding the minimal cost between the two nodes in a graph. No such general algorithm has proven so far that it can solve all variants of the shortest path in a graph. Even though several algorithms like the Greedy algorithm, Flood-fill algorithm, Floyd Warshall algorithm, Dijkstra's shortest path algorithm and Bellman Ford algorithm have been implemented for optimizing the route between the two nearest nodes in a graph. Despite these algorithms, the space and time complexities are associated with them. Therefore, shortest-path problems have moulded us to think of new alternatives. It has led gingus to lead a new challenge to rule out this issue in graph theory. The study of shortest path problems between the vertices (nodes) and edges (lines) in a graph is an

ambient research topic for research community. Therefore, a great deal of attention has been paid in graph theory for an optimizing shortest-path, with the minimum length criteria between two nodes. Therefore, numerous algorithms have been redesigned and implemented for finding a shortest path between two nodes [1]. Due to its enormous wide applications which include network routing protocols, route planning, traffic control, path finding in social networks, computer games, and transportation systems, forging us an exclusive state of shortest path issue in the graphs. The purpose to find the shortest path with all variants between two nodes in a graph is a complex act that needed to be resolved with the special techniques in mathematics. Several numbers of techniques suggested by the research scholars includes Greedy algorithm [1, 2], M. Fredman [3], Floyd Warshall algorithm [4, 5], Dijkstra's shortest path algorithm [6] Bellman Ford algorithm [7, 8] and Genetic algorithm [10] for optimizing the route between the two nodes in a graph. These algorithms' relies to reach shortest path in a space of graph. The Shortest path algorithms are the circle of relatives of algorithms used for solving the hassle of shortest route.

Several research solutions have been implemented to overcome the short path problems in a graph. However, no one address all the variant vulnerabilities of space and time that are associated with the shortest paths [11-12]. Our aim is to present a new model that handles all inherited limitations and vulnerabilities that occurred in previous algorithms [13-14]. The new model has been proposed to sort out complex variants of space and time issues for graph in this paper. It has an ability to find the shortest between two vertices or nodes in graph theory and we put the proposed model name as Pentacode-algorithm. The suitable functionality of the presented Pentacode-algorithm has been verified with suitable comprehensive comparisons with the previous algorithms [15].

2. Background of the shortest paths

Since several algorithms have been introduced which can be helpful in the weighted- graph to determine the shortest-path from the source node to every other destination node within the graph. These algorithms work to search the shortest-path to every single reachable node, provided the graph does not change. These algorithms can run until and unless they visit all reachable nodes, increasing space and time issue problems. However, the graph can be either directed or undirected when it needs to embrace a non-negative value on its every edge. In addition, these algorithms cannot be suggested for weighted- graphs. Since 1959 a Dutch computer scientist, Edsger-Dijkstra's, has suggested an algorithm that can be useful to the weighted- graph. The graph can be either directed or undirected in the situation that the graph needs to embrace a non-negative value on every edge. This algorithm stands out from the shortest path from the source node to every other destination node within the same graph data structure. Rather than searching the shortest path from the source node to the specific node, the algorithm works to search the shortest path to every single reachable node, provided the graph does not change. The algorithm runs until and unless all reachable nodes. The work plan of Dijkstra's algorithm can be understood as:

Let G be a graph in 2-D Plane. It may be directed or undirected.

Therefore, $G = f(V, E)$

"s" is source vertex in the graph plane

"w" is a weight function of graph target search space.

$w = E \rightarrow R$

Fig. 1 depicted below shows the shortest path direction of Dijkstra's algorithm. Here, dotted line shows the shortest path direction.

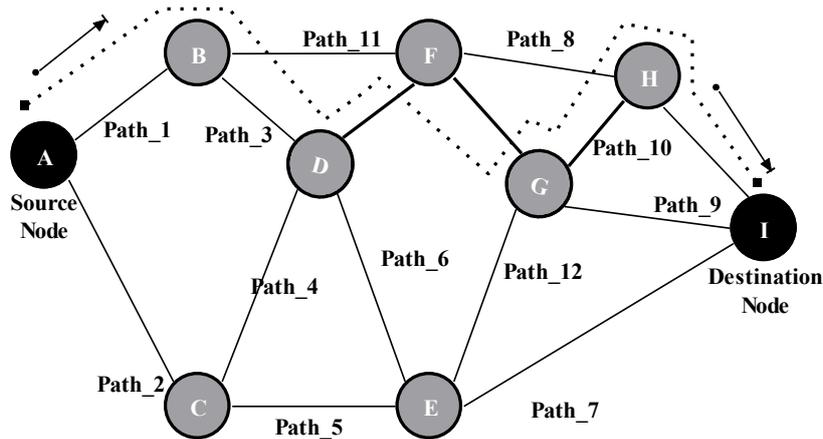


Fig. 1: Shortest Path Detection in Dijkstra's algorithm

Despite certain advantages, there are vulnerabilities of Dijkstra's algorithm for negative weights. These vulnerabilities can be illustrated in Lemma 1.

Lemma 1: Dijkstra's algorithm does not work with negative edges?

Proof: Negative weights can be well explained with these cycles, as shown in Fig. 2. Consider the following proof:

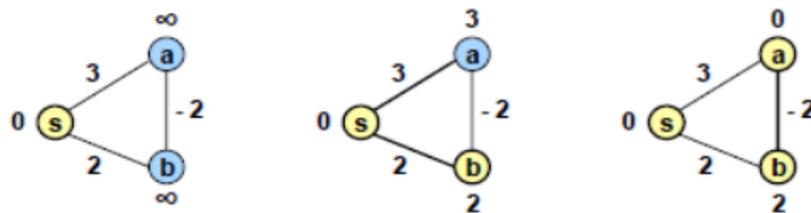


Fig. 2: Negative weight cycle

Let Dijkstra's algorithm visit node b and then at node a. Let it leave node b with a distance of 2 instead of the correct distance 1. The problem is that when we consider the closest vertex v not in the visited set, its shortest path may be through only the visited set and then extended by one edge out of the visited set, say to v . To find the shortest paths on a graph with negative weights, we should think of some inductive hypothesis. In Dijkstra's algorithm, the hypothesis was that if we have found the i nearest neighbours, we can add one more to find the $i+1$ nearest neighbours. As discussed earlier, Dijkstra's algorithm does not work on negative weights.

While in the contract, the Bellman–Ford algorithm [7, 8], sometimes called the Label Correcting Algorithm, computes single-source shortest paths in a weighted digraph (some edge weights may be negative). Its basic structure is very similar to Dijkstra's algorithm. However, instead of greedily selecting the minimum-weight node not yet processed to relax, it simply relaxes all the edges and does this $|V| - 1$ times, where $|V|$ is the number of vertices in the graph. These repetitions allow minimum distance to propagate accurately throughout the graph, since the shortest path can only visit each node at most once without negative cycles. Unlike the greedy approach, which depends on certain structural assumptions derived from positive weights, this straightforward approach extends to the general case. It has the advantage of resolving such a single source shortest-path commonly denoted as (SSSP) problem, and is better applied to be the parallelized for many core architectures [9]. A high degree of parallelism is guaranteed at the cost of the low work efficiency, which, compared to similar algorithms, involves much more redundant work and a consequent waste of power consumption. In short Bellman–Ford algorithm offers a number of advantages such as

negative weight handling, parallel operation support, fast and simple for small application. At the same time, this algorithm inherits number of limitations as the size of vertex and size of search space increases, which are discussed in Lemma 2.

Lemma 2: In Bellman–Ford algorithm, how does the vertex size and search space increase?

Proof: Bellman Ford algorithm can generate all pair's for shortest paths, if one runs the Bellman Ford algorithm and try to find the shortest path. This algorithm has to search all nodes, which increases complexity and requires more time to reach the final results. It consumes major resources of the system. Changes in the network topology are not reflected rapidly since the updates are spread node by node. Hence, the time required to reach the results is considerable when many parallel operations participate. There is no commitment for the optimal results. In some cases, the algorithm may fail due to complex operations.

3. Proposed Pentacode-Algorithm

The proposed Pentacode-algorithm is the Hybridization of integrating two or more meta-heuristic optimization algorithms for searching for the most suitable and optimal solution in a specific search space. It relies on combining the algorithms to resolve various. The proposed Pentacode-algorithm works on the parallel operations of both positive and negative weights. Therefore, parallel operations of the proposed algorithm hold good properties, making it capable of working on different methods and resolving problems efficiently. For example, a search is efficient when many solutions can solve problems.

Prime objectives of Hybridization are as:

- 1) To address the vulnerability of specific meta-heuristic optimization algorithm
- 2) To address limitations of the specific meta-heuristic optimization algorithm
- 3) To counter the disadvantages of the specific meta-heuristic optimization algorithm
- 4) To integrate the advantages of specific meta-heuristic optimization algorithm

Fig. 3 depicts the hybridization of two meta-heuristic optimization algorithms, which addresses the limitations and vulnerabilities by integrating the advantages of other meta-heuristic optimization algorithms and facilitates commitment for convergence with fewer time parameters.

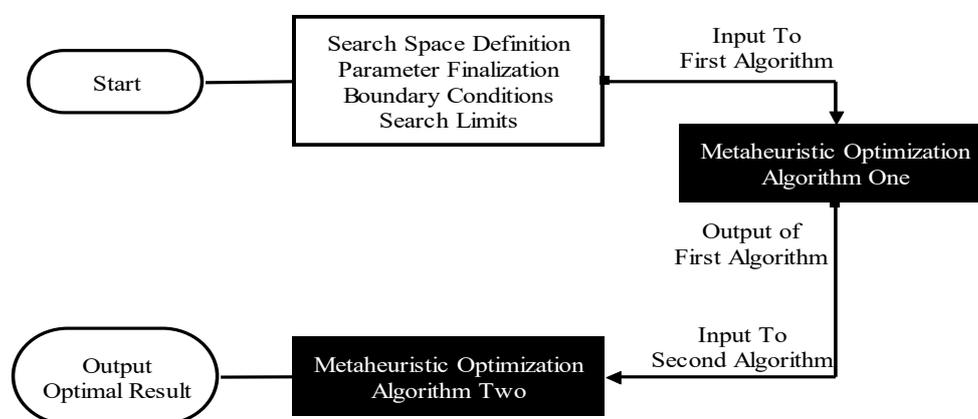


Fig. 3: Hybridization of Two Optimization Algorithm

Hence, the proposed Pentacode-algorithm is a possible method for solving the negative weights in the search space. It can overcome the limitations which are constituted by the previous designs [6, 10]. The proposed algorithm can perform parallel searches of various nodes in space, which can consume less time in space to achieve target. This ability will become a benchmark to be the fastest algorithm.

Dijkstra's shortest-path algorithm has a blind search. It calculates the entire path from source to destination, which makes the algorithm inefficient because it wastes more time to reach the necessary resources. In addition, if the space search of an algorithm increases, the time required for the optimal path also increases exponentially. In addition, it works with non-negative weights for the edges with slight modifications but fails for negative weight in the space graph, which swings the system and hangs the final results. However, nested loops taken by Dijkstra's algorithm lead to an increase the search time in space. Due to some limitations of Dijkstra's algorithm, like using nested loops, which increase the search time, it becomes necessary for the research community to implement a new algorithm that can span less time in space, using limited loops. In this regard a new Pentacode-Algorithm algorithm has been proposed.

The proposed Pentacode-algorithm is committed exact and more realistic in nature in the context of real world live wire problems. The Pentacode-algorithm works with waited/non-negative weights. It overcomes the limitations of Dijkstra's Algorithm. Initially, Bellman-Ford-algorithm is initiated with parameters subjected to target search space; the output of this search will be input for Dijkstra's Algorithm and is search space for the second algorithm. However, the proposed algorithm sown in Fig. 4, which is free of negative edges and is smaller in size than previous designs [6, 10].

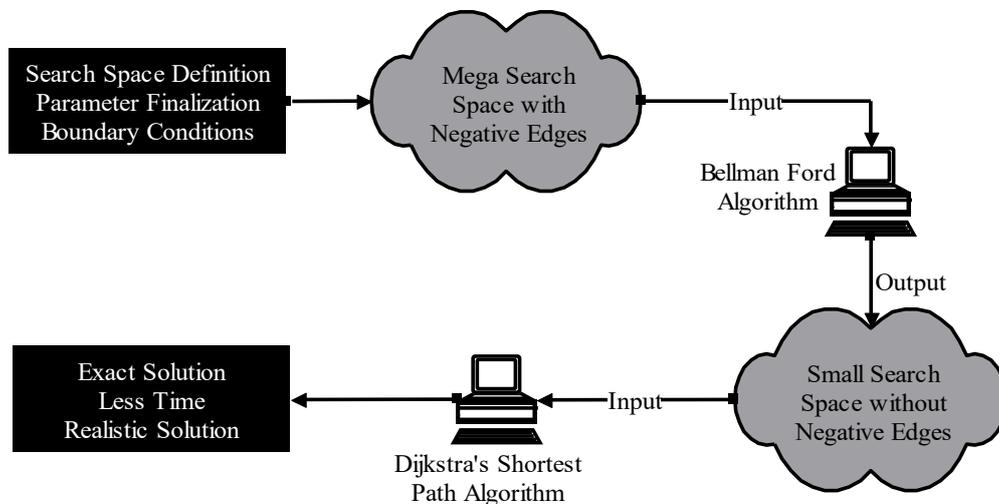


Fig. 4: The proposed Pentacode-Algorithm

The proposed Pentacode-Algorithm facilitates various advantages like 1) Negative weight or edge present in the search space is a major issue in Dijkstra's algorithms which has been countered in initial stage in PENTACODE-Algorithm such that new search space is suggested for optimal result. 2) Dijkstra's algorithm implements recursive search technique which calls another function for optimization, this recursive calling results increases searching time. However, in the proposed Pentacode-Algorithm already shortened the search space, now Dijkstra's algorithm required confining the search in smaller space, this technique saves time and resources. 3) There is fewer possibility of trapping the proposed algorithm for infinite loop which Confirms convergence within search space.

4. Comparison of the proposed Pentacode-Algorithm

- 1) Negative weight or edge present in the search space is a major issue in Dijkstra's algorithms, which has been countered in initial stage in Pentacode-Algorithm such that new search space is used for optimal results.

- 2) Dijkstra's algorithm implements recursive search technique which calls another function for optimization, this recursive calling results increases searching time. However, in the proposed Pentacode-Algorithm already shortened the search space. This advantages that Dijkstra's algorithm required confining the search in smaller space; this technique saves time and resources.
- 3) There is fewer possibility of trapping the proposed algorithm for infinite loop which Confirms convergence within search space. The detailed work plan of the proposed Pentacode-algorithm is mentioned in Fig.5.

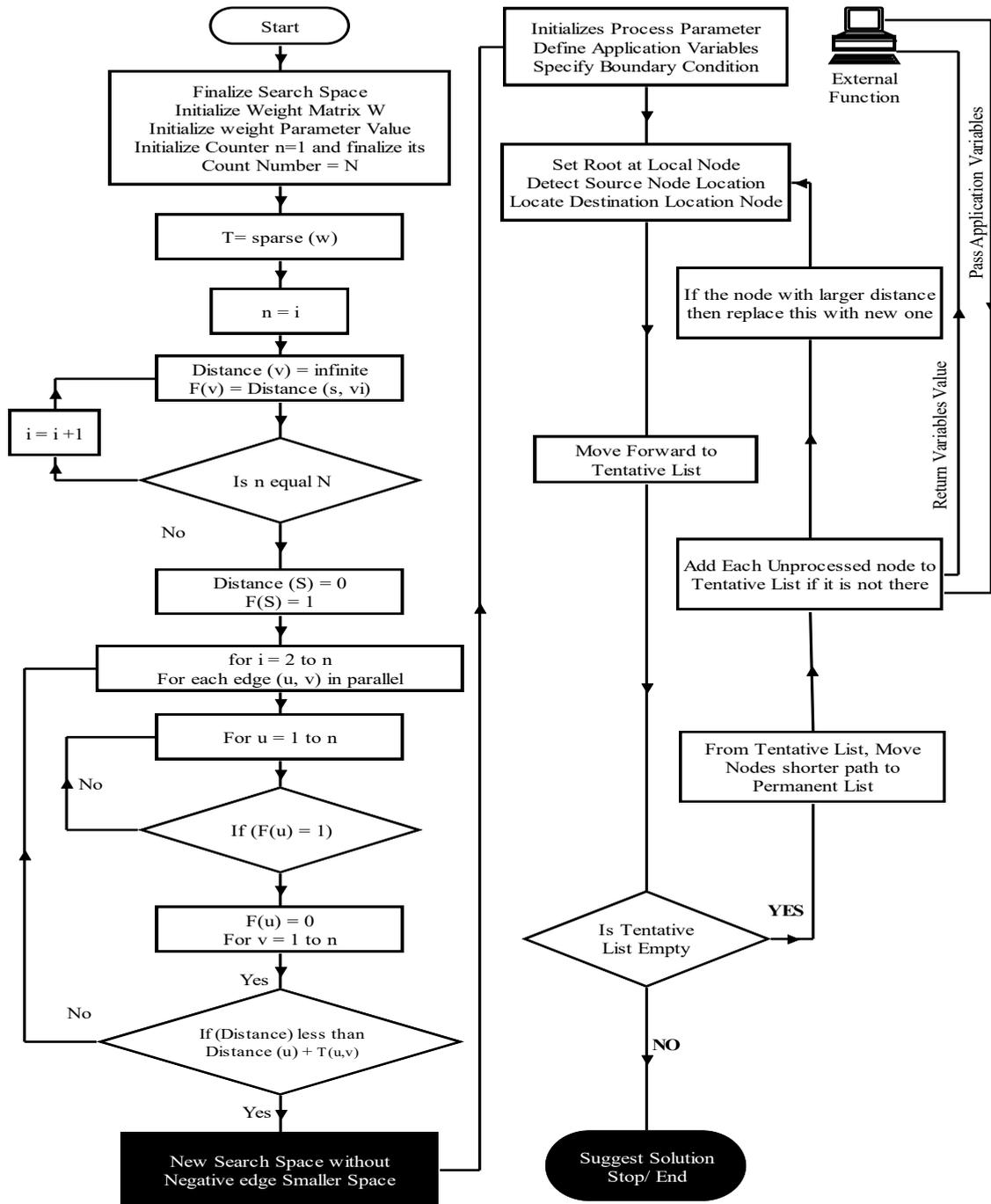


Fig. 5: The Proposed Pentacode-Algorithm Flow Chart

5. The Java Code Segment used for the proposed Pentacode-Algorithm

A systematic approach has been used to implement Algorithm to optimize information protocol. Java software coding technique has been used to test the proposed Pentacode-algorithm. To authenticity, the functionality and visualization of graphs have been determined using MATLAB-6.0. This is an independent platform which supports java code in which various parameters such as shortest distance (SD), time to search shortest distance (TSSD), Loss of information packets (LIP) have been verified by using the java code. In addition, first node die (FND), Half node die (HND) and last node die (LND) have been determined in our results. Optimal Shortest Path selection process for the proposed Pentacode-algorithm has been checked in the java cod as:

```

Declare Int N_D Number_of_Node, N_C Number_of_Cluster, D_N Distance Between Nodes
      Int NL1 Loop_1_Count, NL2 Loop_2_Count
      Start Clustering of Phase_One Level_1
      Start Clustering of Phase_One Level_2
      Start Clustering of Phase_One Level_3
Repeat-Step-3 for i =1, 2- ----- nStart Loop One
Repeat-loop for i =1, 2- ----- n.Start Loop Two
      Optimal Cluster Selection Process Start
      Check if a[Vs, i] =0
      Optimal Shortest Path Selection Process Start
      In case no direct-edge among Vs&Vi
      Then: leng[i]=Infinity [Infinity value is taken to be 999]
      path[i]=0 [Invalid path]
      Else: leng[i]=a[vs,i]and path[i]=vs
      End Clustering of Phase_One Level_3
      End Clustering of Phase_One Level_2
      End Clustering of Phase_One Level_1
Second Level Clustering Start with a new non-negative search space, shortened search space
      Start Clustering of Phase_Two Level_1
      Start Clustering of Phase_Two Level_2
      Start Clustering of Phase_Two Level_3
Repeat-Step3-for i =1, 2- ----- nStart Loop One
Repeat-loop for i =1, 2- ----- n.Start Loop Two
      Optimal Cluster Selection Process Start
      Check if a[Vs, i] = 0
      Optimal Shortest Path Selection Process Start
      In case no-direct edge amongVs&Vi
      Then:leng[i] = Infinity[Infinity-value is taken to-be 999]
      path-[i] = 0 [Invalid path]
      -Else:leng[i] = a[Vs, i]and path[i] = vs
      End Clustering of Phase_Two Level_3
      End Clustering of Phase_Two Level_2
      End Clustering of Phase_Two Level_1
Optimal Shortest Path Selection Process End: Print Result.
    
```

6. The Result and Discussion

The geographical map, considered an objected area for the testification of simulation results, has been shown in Fig. 6(a). The total test area is 10 x10 kilometres area, which h is 10 km long and 10 km wide, which means 100 square kilometres (100 Sq. Km). The test area has been taken from the Google Map web site. Various locations and places have been shown in this map, in which a few cluster and cluster member nodes have been chosen for testification. The proposed work plan has been verified using 10000 active nodes, which occupy an area of 10000 sq. meters.

Further, these active nodes have been grouped into 100 clusters. Fig. 6(b) depicts the graphical representation of the test model considered for simulation and verification. This is a state space presentation of Model in graph theory so that the shortest path can be searched from the possible combinations of the hundreds of options in the space.

The proposed Pentacode-algorithm has been compared with the various previous algorithms [6, 10] as shown in Table-1. It indicates the shortest distances between source and destination nodes. In our practical, we have considered various test distances which are-5.0-KM, -5.5-KM, -6.0-KM, -6.5-KM, -7.0-KM, -7.5-KM, -8.0-KM, -8.5-KM&-9.0-KM. In actual sense test has been done with 10Km x 10Km (Sq.100km). The result in the table 1 clearly reflects that the shortest path suggested by the Pentacode-algorithm is the minimum physical distance among path suggested by different algorithms. 9% to 11% better result with the A-algorithm. It is clear from the graphical comparison, including various parameters, that our proposed plan is superior to all other existing techniques, as shown in Fig. 6.

Table 2 shows the time taken of various shortest path algorithms. The graphical picture has proven that our proposed Pentacode-algorithm spans very less time and is more persistent than existing designs [6, 10]. The time taken by Pentacode-algorithm has shown minimum indication from every simulation results. Various distances have been considered between the source and destination node. The detailed graphical comparison of the shortest distance between the source and destination node by different algorithm is shown in Fig. 7, and the timer taken to search the shortest path by different algorithms is shown in Fig. 8:

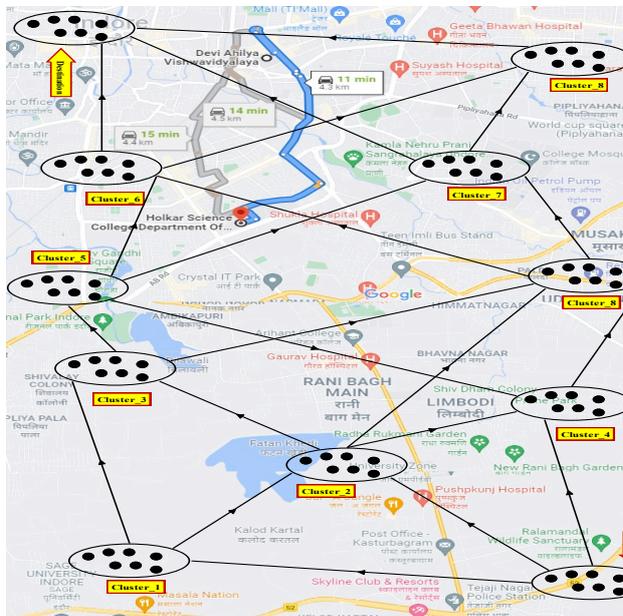


Fig. 6(a): Model Network Geographical Area

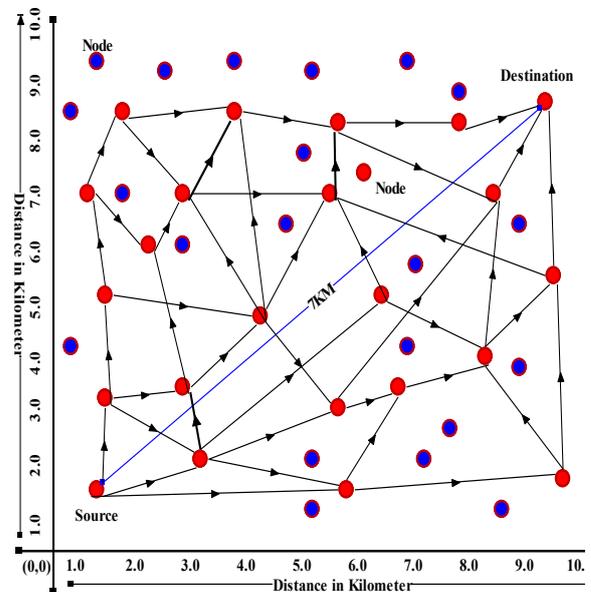


Fig. 6(b): The Test Model Area (100 Sq. Km)

Table-1: Shortest Distances Suggested by Different Algorithm

No.	Actual Distance Between Source Node and Destination Node	Physical Path by Dijkstra's Algorithm [6]	Shortest Suggested Path by Genetic Algorithm [10]	Shortest Suggested Path by Pentacode-algorithm
1	5.0 KM	6.1 KM	5.7 KM	5.4 KM
2	5.5 KM	6.9 KM	5.9 KM	5.7 KM
3	6.0 KM	7.1 KM	6.9 KM	6.5 KM
4	6.5 KM	7.6 KM	7.2 KM	6.9 KM
5	7.0 KM	8.2 KM	7.9 KM	7.4 KM
6	7.5 KM	8.7 KM	8.4 KM	7.9 KM
7	8.0 KM	9.4 KM	8.9 KM	8.6 KM
8	8.5 KM	10.2 KM	9.9 KM	8.9 KM
9	9.0 KM	11.4 KM	10.1 KM	9.8 KM

Figure 7, shows the shortest distance between source and destination node by different algorithm and

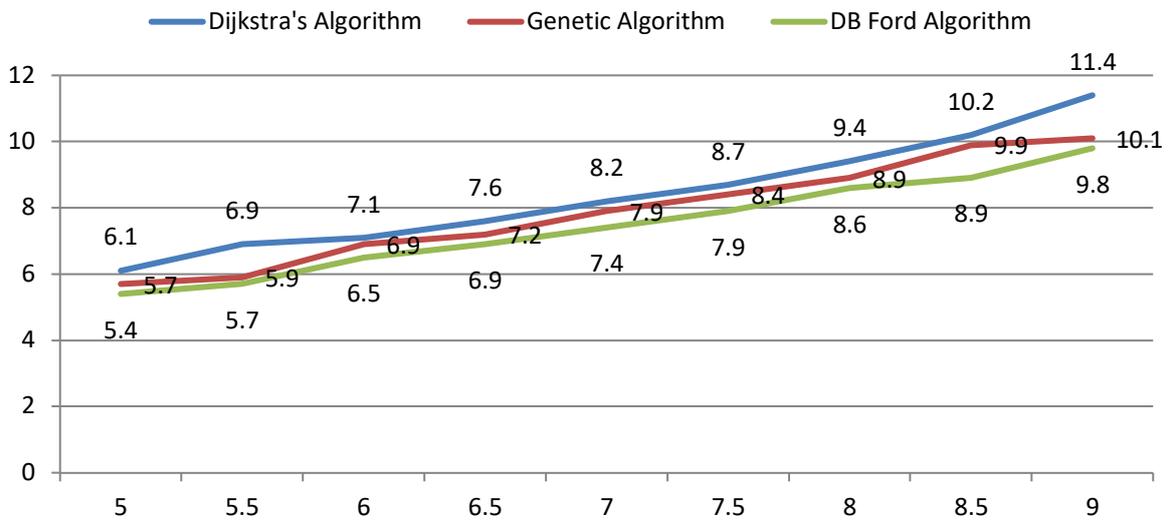


Fig. 7: Shortest Distance between Source and Destination Node by Different Algorithm

Table-2: Time taken to search Shortest Distance by Different Algorithm

No.	Actual Distance Between Source Node And Destination Node	Time Taken by Dijkstra's Algorithm To Reach Shortest Path Value (Nanoseconds) [6]	Time Taken by Genetic Algorithm To Reach Shortest Path Value (Nanoseconds) [10]	Time Taken by Pentacode-algorithm To Reach Shortest Path Value (Nanoseconds)
1	5.0 KM	110 Nanoseconds	90 Nanoseconds	85 Nanoseconds
2	5.5 KM	125 Nanoseconds	110 Nanoseconds	98 Nanoseconds
3	6.0 KM	136 Nanoseconds	119 Nanoseconds	104 Nanoseconds
4	6.5 KM	141 Nanoseconds	126 Nanoseconds	113 Nanoseconds
5	7.0 KM	151 Nanoseconds	132 Nanoseconds	121 Nanoseconds
6	7.5 KM	162 Nanoseconds	141 Nanoseconds	132 Nanoseconds
7	8.0 KM	171 Nanoseconds	154 Nanoseconds	139 Nanoseconds
8	8.5 KM	182 Nanoseconds	163 Nanoseconds	145 Nanoseconds
9	9.0 KM	198 Nanoseconds	174 Nanoseconds	152 Nanoseconds

Figure 8, depicts the timer taken to search shortest path by different algorithm.

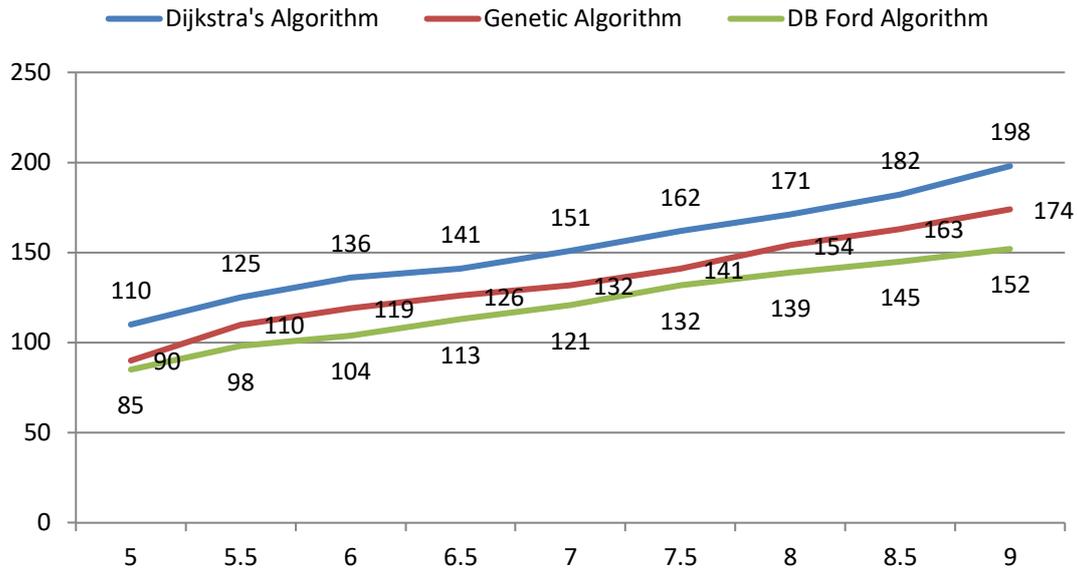


Fig. 8: Timer taken to search Shortest Path by Different Algorithm

7. Percentage Information Loss

The loss of Information inherits network protocols such as HTTP, FTP and SMTP I, because these protocols are associated with distance. If the distance between source and destination node increases, time will increase accordingly. The percentage loss has been tabulated in Table 3. Clearly, the Table-3 information loss of packets is minimal in the proposed Pentacode-algorithm compared with the previous algorithm [6, 10] Dijkstra's-Algorithm & Genetic-Algorithm. In actuality, the distance travelled and time taken to complete the distance in the proposed algorithm is minimal, as shown in Fig.9.

Table 3: Percentage Loss of Information

No	Test Distance Between Source & Destination Node	% in Information Protocol with Dijkstra's Algorithm [6]	% in Information Protocol with Genetic Algorithm [10]	% in Information Protocol with Pentacode-Algorithm
1	5.0 Km	16.0 %	11.5 %	10.2 %
2	5.5 Km	16.6 %	11.8 %	10.9 %
3	6.0 Km	17.3%	12.4 %	11.3 %
4	6.5 Km	17.5 %	12.9 %	11.7 %
5	7.0 Km	18.2 %	13.2 %	11.9 %
6	7.5 Km	18.8 %	13.8 %	12.5 %
7	8.0 Km	19.2 %	14.7 %	13.1 %
8	8.5 Km	19.6 %	15.4 %	13.9 %
9	9.0 Km	19.9 %	15.9 %	14.2 %
10	9.5 Km	20.2 %	16.2 %	14.6 %

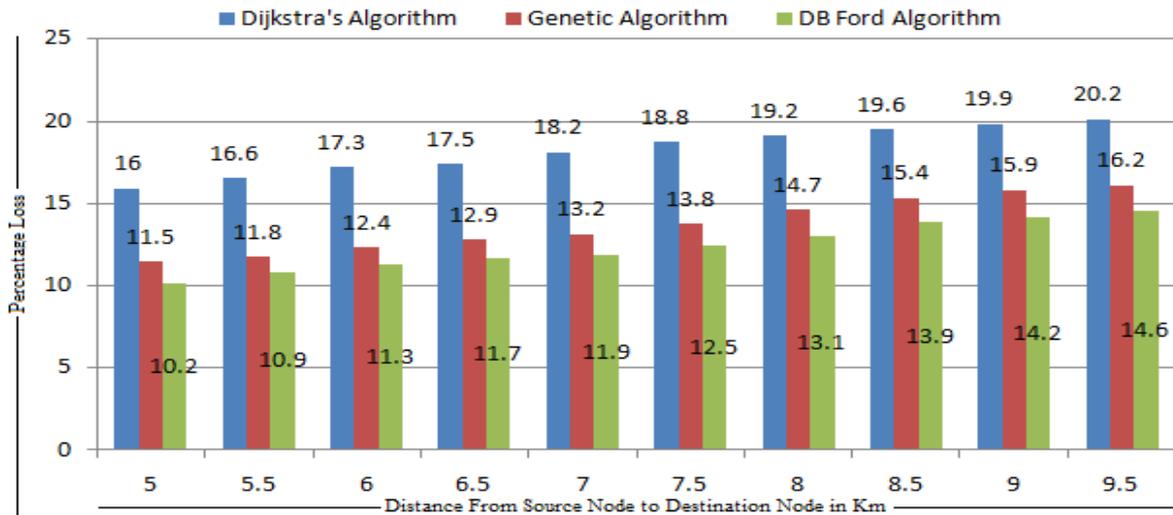


Fig. 9: Percentage loss in Information Packets with respect to distance in different Algorithm

8. First-Node-Die (FND), Half –Nodes-Die (HND) and Last-Node-Die (LND) Parameter

Evaluation of the Wireless-sensor-network can be-done by using the various parameters-for the system energy. Therefore, number of the nodes die in each round because of-FND, -HND &LND. Since when the-first node starts dissipating energy-its energy and the performance will starts-degrading. Therefore, it is required to-mark the death of the very first-node in-the-network. Hence the same cause of time for HND and LND is marked. Briefly, the lower the number of nodes, the better the performance of wireless communication system. The result clearly reflects that best performance is achieved by the proposed Pentacode-algorithm for shortest paths. The detailed comparisons of various parameters [6, 10] are listed in Table 4. The graphical picture depicted in Fig. 10, Fig. 11 and Fig. 12, shows the better performances achieved in the proposed design [6, 10].

Table 4: The Comparison of FND, HND & LND Parameter

No.	Protocol Count	Dijkstra's Algorithm [6]			Genetic Algorithm [10]			Pentacode-algorithm		
		FND	HND	LND	FND	HND	LND	FND	HND	LND
1	5000	900	880	820	870	840	810	840	820	800
2	6000	960	920	895	950	890	875	940	870	860
3	7000	990	935	910	965	910	895	950	900	875
4	8000	1000	980	950	990	950	900	960	930	895
5	9000	1050	998	978	1040	960	940	990	940	910
6	10000	1065	1020	995	1050	1030	955	1010	990	920
7	11000	1100	1050	1030	1060	1040	990	1020	910	940
8	12000	1150	1090	1050	1090	1060	1010	1030	940	930
9	13000	1190	1110	1070	1110	1090	1040	1040	960	940
10	14000	1200	1150	1090	1120	1110	1050	1055	970	960

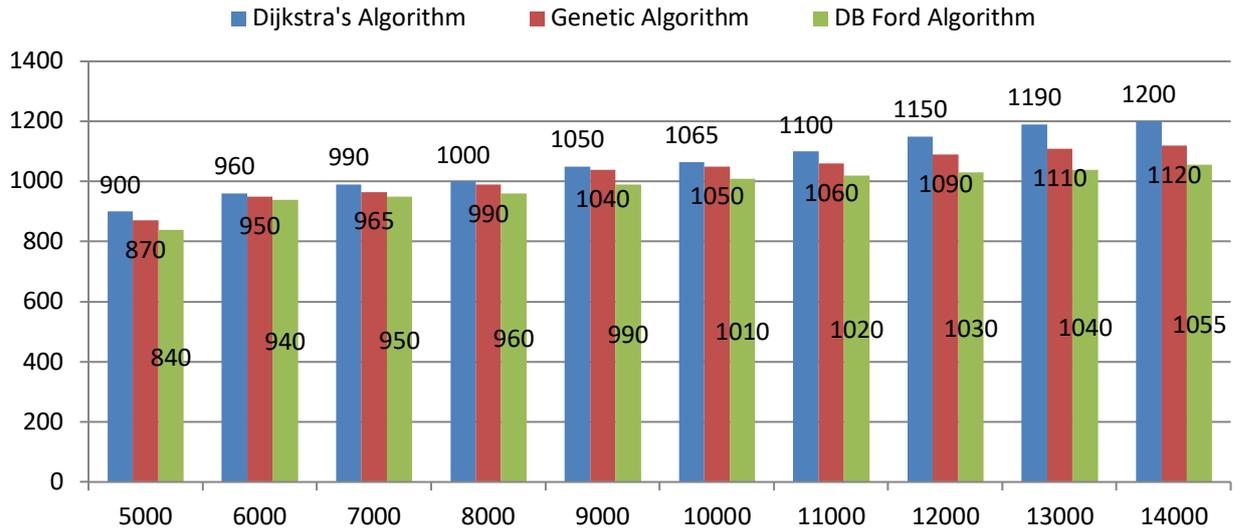


Fig. 10: First-Node-Die (FND) Pointer in Different Algorithm with respect to Different Protocol Number

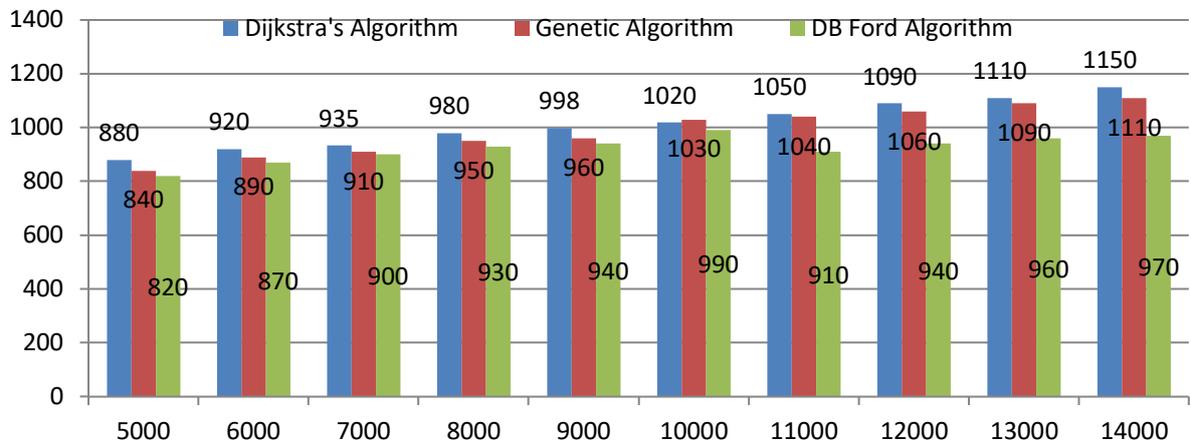


Fig. 11: Half-Node-Die (HND) Pointer in Different Algorithm with respect to Different Protocol Number

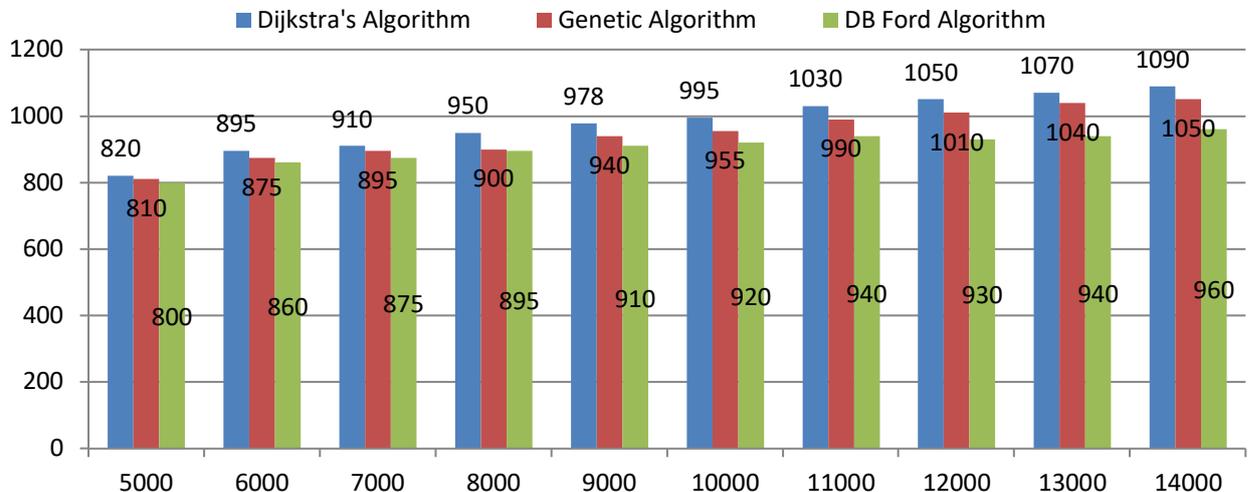


Fig. 12: Last-Node-Die (LND) Pointer in Different Algorithm with respect to Different Protocol Number

9. Future Scope

Shortest path algorithm is of great importance in modern industrial applications, corporate applications and even in the daily application of socio-economic activities of modern age at present time; when there are numbers of challenges in front of human beings.

Hence there are numbers of area in different sectors in which there will be great potential for further research in the respective field.

These core areas are documented as below in which the research can be extended:

- 1) Telecommunication Companies
- 2) Internet Service Provider Companies
- 3) Cloud Computing Infrastructure Companies
- 4) Live wire Channel Banking Applications
- 5) Missile Launching Application
- 6) Missile Defence Application
- 7) Rocket Trajectory Application
- 8) Agriculture and Environment Assisting Applications.

10. Conclusion

This paper presents a new Pentacode-algorithm for shortest path problems. The proposed Pentacode algorithm has been designed to enhance performance by effectively handling the challenges associated within Dijkstra's, Bellman-Ford, and Genetic algorithms. Shortest-Path-Algorithm. Various test simulation has been discussed and modeled in graphical mathematical formant. The simulation results clearly show that the proposed Pentacode-algorithm is superior to previous algorithms. Test simulation has been done with the input parameter value, including distances and time. Test parameters such as area which includes 10Km x 10Km (100 Sq. Km) in which 100 clusters have been grouped for calculations. Each cluster contains 100 nodes. The new Pentacode-algorithm has been designed to enhance the performance and optimize the distance for shortest path problems. Furthermore, the proposed algorithm has been implemented to improve upon the limitations of Dijkstra's and the Bellman-Ford shortest path algorithms. Moreover, the time taken by Pentacode-algorithm for searching path in space is minimal, indicates highly proficiency of shortest path optimization than previous algorithm. Verification of results has been determined using Java code and MATLAB-6.

References

- [1] R. Bellman (1957). *Dynamic Programming*. Princeton University Press.
- [2] R. Bellman (1958). On a routing problem. *Quarterly of Applied Mathematics*.
- [3] M. Fredman (1976). New bounds on the complexity of the shortest path problem. *SIAM*, pages 83–89.
- [4] R. Floyd. (1962). Algorithm 97: Shortest Path. *Communications of the ACM*, pages 344–348.
- [5] S. Warshall (1962). A theorem on boolean matrices. *Journal of the ACM (JACM)*, (1).
- [6] E. W. Dijkstra (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271.
- [7] R. Bellman (1957). *Dynamic Programming*. Princeton University Press.
- [8] R. Bellman (1958). On a routing problem. *Quarterly of Applied Mathematics*.
- [9] Tim LaRocca, Khanh Thai, Robert Snyder, Richard Jai, Daniel Kultran, Owen Fordham, (2019), Secure Satellite Communication Digital IF CMOS Q -Band Transmitter and K -Band Receiver, *IEEE Journal of Solid-State Circuits*, Volume: 54, Issue: 5, pp: 1329 – 1338.

- [10] S. Chiang, S. C. Chu, Y. C. Hsin and M. H. Wang (2006). Genetic Distance Measure for K-Modes Algorithm, *International Journal of Innovative Computing, Information and Control ICIC*, ISSN 1349- 4198, Volume 2, Number 1, pp.33-40.
- [11] Hongmei. Wang, Ming. Hu and Wei. Xiao, (2010). A new public transportation data model and shortest-path algorithms, *Informatics in Control Automation and Robotics (CAR) 2010 2nd International Asia Conference on*, pp. 456-459, 6–7.
- [12] Na Wang, Jian Li (2019). Shortest Path Routing With Risk Control for Compromised Wireless Sensor Networks, *IEEE Access*, Volume: 7, PP 19303 –19311.
- [13] Pradumn Kumar Pandey, Venkataramana Badarla (2020). Small-World Regular Networks for Communication, *IEEE Transactions on Circuits and Systems II: Express Briefs*, Volume: 67 (8), 1409 – 1413.
- [14] Hussein Al-Zubaidy, Viktori Fodor, György Dán, Markus Flierl (2017). Reliable Video Streaming With Strict Play out Deadline in Multi hop Wireless Networks, *IEEE Transactions on Multimedia*, Volume: 19 (10), pp 2238 – 2251.
- [15] Ahmad M. Khasawneh, Omprakash Kaiwartya, Ala' Khalifeh; Laith M. Abualigah, Jaime Lloret (2020). Green Computing in Underwater Wireless Sensor Networks Pressure Centric Energy Modeling "IEEE Systems Journal", Volume: 14, Issue: 4, pp 4735 – 4745.