A hybrid approach to enhanced genetic algorithm for route optimization problems

Mrinmoyee Chattoraj, Udaya Rani Vinayakamurthy
Department of Computing and Information Technology, REVA University, Bengaluru, India

Article Info

Article history:

Received Sep 28, 2022
Revised Jan 10, 2023
Accepted Jan 12, 2023

Keywords:

Crossover
Feasibility
Genetic algorithm
Graph reduction
Mutation
Optimized route

ABSTRACT

Shortest path problem has emerged to be one of the significant areas of research and there are various algorithms involved in it. One of the successful optimization techniques is genetic algorithm (GA). This paper proposes an efficient hybrid genetic algorithm where initially we use a map reduction technique to the graph and then find the shortest path using the conventional genetic algorithm with an improved crossover operator. On comparing this hybrid algorithm with other algorithms, it has been detected that the performance of the modified genetic algorithm is better as comparison to the other methods in terms of various metrics used for the evaluation.

Corresponding Author:

Mrinmoyee Chattoraj
Department of Computing and Information Technology, REVA University Kattigenahalli
Yelahanka, Bengaluru, India
Email: mrinmoyee2005@gmail.com

1. INTRODUCTION

In case of road network problems, one of the main task is to search the shortest path. We come across many problems such as total travel time, finding the minimum cost path and the path with lowest amount of traffic congestion. Dijkstra's algorithm finds the shortest path but it is not capable of finding the shortest path in case of difficult situation or crisis. Genetic algorithm (GA) is an evolutionary algorithm which give the solution to this problem using Darwin's evolution theory [1], [2]. Traditional genetic algorithm starts with a group of individuals which are known as chromosomes. Genetic algorithm is capable of solving any type of problem which can be represented by chromosome encoding [3].

This paper suggests an amalgamated algorithm which first reduces the size of the graph using a reduction algorithm and then uses the evolutionary algorithm to discover the optimized route from the starting to the ending vertex. In this case as the number of nodes are reduced hence the total time complexity is also less. Also since there are a number of feasible paths so in case of tough situation where a particular path cannot be selected due to some unforbidden circumstances like traffic jam, then in that case the user has the option of selecting the next most suitable feasible path. The arrangement of the remaining paper is as follows: Section 2 describes the associated works regarding the techniques used for finding the shortest distance. Section 3 describes the future framework or methodology used to find the best path. In section 4 we discuss and draw conclusions based on the final results. In section 5 we write the conclusions.

2. METHOD

In case of network analysis, numerous studies have been conducted which resulted in a number of approaches to find the shortest path under various conditions. Dijkstra's algorithm is a very common algorithms to which is used to search the shortest path [4], [5]. An improved version of Dijkstra's algorithm was proposed for sparse network [6]. An extended Dijkstra's algorithm was proposed which extends the conventional method in order to obtain the solution in a specified time [7]. In another technique a distributed streaming computing system and lightweight index is built to develop efficient shortest paths [8]. Genetic algorithm uses the vehicle travel distance to optimize the shortest path [9]-[11]. Another approach is to find an operative genetic algorithm which finds the optimal path using the method of fuzzy logic [12]. Generally genetic operation starts with a selection operator which picks the most suitable chromosomes to produce the ideal solution [13], [14]. Then the operators such as crossover and mutation are applied to prevent early convergence of solutions [15]-[18]. In order to develop a hybrid genetic algorithm to improve the effectiveness of the algorithm we consider Taxi dataset of New York City [19], which includes the pickup time, number of passengers, geo-coordinates, and several other variables. Millions of travel trajectories that occurred in New York City are included in this dataset.

3. PROPOSED FRAMEWORK/METHOD

The hybrid genetic algorithm consists of two strategies. In the first strategy we reduce the graph using a graph reduction technique in which we keep only the set of nodes which consists of the source, destination and the main intermediate nodes in order which calculates the shortest path. In the second strategy we propose a new mutation process which is considered as another local optimization technique. If the probability of mutation is set to a high value, then the genetic algorithm will hardly converge and on the other hand if the mutation probability is small then it will be trapped in local optima [20]-[23]. In this paper we have assigned the mutation probability to be 0.05. In this proposed mutation process we assign a random number to each node. If the number is less than the mutation probability, then corresponding node is changed with the second node whose value is lower than the mutation probability. If there are odd number of nodes with value less than the mutation probability, then the last node is discarded. In this process the mutated nodes are swapped in a sequential manner. In this method no duplicates are generated.

The computation results show that the method which is proposed gives us much improved paths as compared to the conventional genetic algorithm and it has an acceptable computation time. An algorithm is designed which is used to evaluate the set of shortest paths and select any two paths which are considered as chromosomes. These chromosomes are fused to find the next generation. The individual from the next generation will be the shortest path. Delay time is due to congestion in the path which reduces efficiency and increases the travel time. In case there is congestion in a path then we have multiple options to reach our destination which is not there in the case of Dijkstra's or Kruskal's algorithm. Our proposed work can handle delays as well as multiple options to reach our destination.

In our proposed algorithm we consider an arbitrary graph of 25 nodes where each node represents a point in the real-world coordinate specified by its latitude and longitude. The graph G (V, E) shows the collection of vertices and edges. The collection of vertices V= {A, B, C, D…., V, W, X, Y} and the collection of edges is considered as E. The arbitrary graph with the nodes is in Figures 1 and 2 whereas the tabular representation of the distance is shown in Table 1.

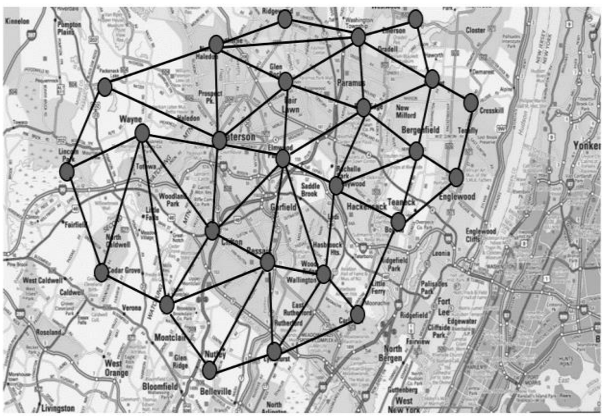

Figure 1. Nodes representing the various cities

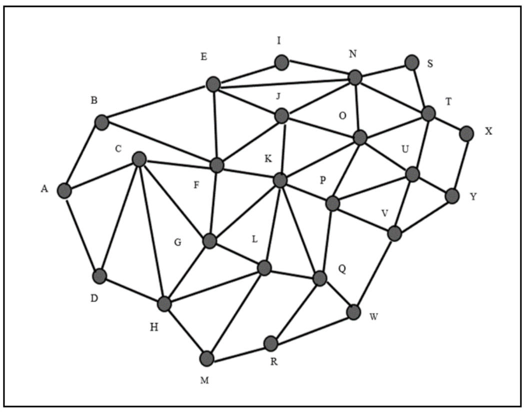

Figure 2. Arbitrary graph representing the various cities

The arbitrary graph given in Figure 2 which consists of 25 nodes is reduced into a subgraph by implementing the “graph reduction algorithm”. Since the number of nodes reduces from 25 to 10, the execution time to search the shortest path also decreases. The graph reduction algorithm consists of the first creating a Levelled graph, then deleting the dead nodes to remove the unwanted nodes and finally it uses the backtracking algorithm to keep only the required nodes. The algorithm 1 is as:

```
Proposed Algorithm 1 for Graph Reduction
Step 1: Start
Step 2: Select the temporary source vertex (SVertex) and the destination vertex (DVertex)
Step 3: The FIRST LEVEL is generated.
Step 4: Put the SVertex in this level.
Step 5: All the nodes connected to SVertex are added to the Queue.
Step 6: We find all the nodes connected to the SVertex in the same level. In case there are
still some untraversed nodes then SVertex will be the same level untraversed node and goto
Step 7 else goto Step4.
Step 7: Generate the next level.
Step 8: In this current level the queue is added.
Step 9: In case we find that the terminal node and all nodes are navigated in the same level
then goto Step 10 else SVertex is equal to the next node which is not traversed in the same
level and goto Step 5.
Step 10: Now we remove the dead nodes of the graph by deleting the nodes whose connecting
edges connects to nodes in the same level or the previous level.
Step 11: In the nth level we keep only the destination node and remove the remaining nodes.
Step 12: Delete the successor node from the previous level in case the surviving node does
not have a connecting edge otherwise goto Step 14.
Step 13: Repeat Step 12
Step 14: Repeat Step 11 so that in the nth level only the destination node is there, and all
other unwanted nodes are deleted.
Step 15: End
```

When we implement this algorithm on the arbitrary graph given in Figure 2 where ‘C’ is the source vertex and ‘Y’ is the destination, then we first get a levelled graph as shown in Figure 3, in which the dead nodes A, D, I, S and W will be deleted. By applying the backtracking technique, we remove the unwanted nodes from the levelled graph given in Figure 3, where we delete the dead nodes and the nodes which will not lead to the destination. Thus, we get a reduced graph consisting of 10 nodes as given in Figure 4.

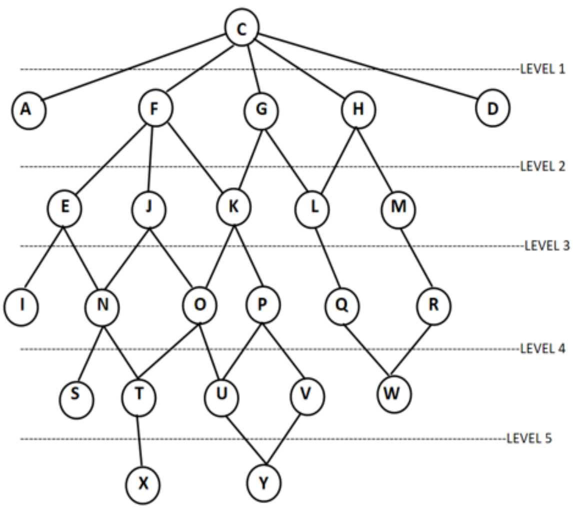

Figure 3. Levelled graph

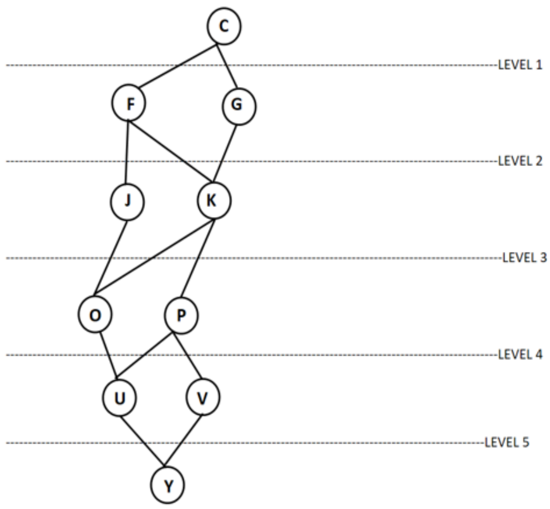

Figure 4. Reduced levelled graph

Since the subgraph contains a smaller number of nodes as compared to the main graph, the execution time as well as the search space is also reduced. In Figure 4 the possible paths from the source vertex ‘C’ to destination ‘Y’ will be represented by the chromosomes given in Figure 5. These are the chromosomes which are generated after implementing the graph reduction algorithm. All these chromosomes represent the possible feasible solution from the source vertex C to the destination Y. Since we have several feasible solutions, we have a variety of alternate choices in case of unpredicted conditions.

CHROMOSOME 1	C	F	J	O	U	Y
CHROMOSOME 2	C	F	K	O	U	Y
CHROMOSOME 3	C	F	K	P	U	Y
CHROMOSOME 4	C	F	K	P	V	Y
CHROMOSOME 5	C	G	K	O	U	Y
CHROMOSOME 6	C	G	K	P	U	Y
CHROMOSOME 7	C	G	K	P	V	Y

Figure 5. Representation of possible paths as chromosomes

From this group of shortest paths, we find the shortest path by implementing chromosomal fusion. The quality of the chromosome can be improved by the selection process which depends on the chromosome's fitness value. We apply a simple and effective selection operator which is the tournament selection method [24], [25]. To produce high-quality offspring, crossover, and mutation operators, as well as their probabilities, are very important. The frequency of crossover and mutation is expressed as a probability. In case the crossover probability is 0% then the offspring generated will be a duplicate copy of their parents and in case 100% is the probability then all offspring are made by crossover. In this case we will be using the China National Petroleum Corporation (CNPC) crossover operator [15]. The goal of the crossover of chromosomes is to pass on the good parts of the parent chromosomes to their offspring.

Mutation operator helps us to get stuck in local optima and improves the chances of achieving the global optima. We use a mutation operator with the intention to prevent it from getting stuck in the local optimum and it also increases the possibility to find the global optima. The frequency in which the parts of the chromosome will be mutated will be given by mutation probability. If the mutation probability is 0% then nothing is changed from the chromosome but if it is 100% then the entire chromosome is altered.

In our proposed method the mutation probability is 0.02 and the probability of crossover to be 0.8. In our mutation strategy we select a path from the chromosomes and assign two random integer numbers between 2 and (N-1) where the number of nodes is considered as N. All nodes between these numbers are reversed. If the new path is lower than the initial path, then the main selected route is reconstructed with the new arrangement otherwise this new path is discarded. The pseudo code of our proposed method is:

Step 1: Reduce the graph using a new graph reduction technique.
Step 2: Choose the initial paths.
Step 3: Evaluate each path length.
Step 4: Calculate the path's average length.
Step 5: Select the best ranking paths using tournament selection method.
Step 6: Apply CNPC crossover operator [15].
Step 7: Apply a mutation operator.
Step 8: Evaluate each path's length.
Step 9: Determine the average path length.
Step 10: Repeat Steps 5 to 9 until termination condition has reached.

4. RESULTS AND DISCUSSION

The outcome of the proposed algorithm is discussed in this unit. The planned system is exposed to various experiments for evaluation and comparative study is done for the proposed algorithm with other competitive algorithms such as conventional genetic algorithm, Dijkstra's-algorithm, particle swarm optimization (PSO) algorithm and artificial-bee-colony (ABC) algorithm. The comparison of these approaches is done with respect to processing time, travel time and average vehicle throughput. When compared to previous approaches, the proposed algorithm is observed to generate more encouraging results.

Table 1 depicts the performance of the modified genetic algorithms (GA) with the existing algorithms such as conventional genetic algorithm, Dijkstra's-algorithm, PSO algorithm and ABC algorithm. It has been observed that the modified genetic algorithm outperforms the other algorithms. In Table 2 we notice that the computing time of the modified GA is comparatively less than the other algorithms by doing a comparative study of optimal solutions. Table 3 shows the eight different values that we obtain for the modified genetic algorithm from which we select the optimal distance 118.21 (km). Figure 6 shows the processing time of the various algorithms, and it has been observed that the processing time of the modified GA is better.

Table 1. Comparison of the modified genetic algorithm's performance with those of the existing algorithms

System of neasurement	Number of nodes	Proposed modified GA	GA	Dijkstra's algorithm	PSO algorithm	ABC algorithm
Processing time (sec)	100	220	273	313	240	234
	200	371	411	482	403	398
	300	619	673	703	643	656
	400	785	871	920	882	892
	500	1002	1138	1270	1187	1183
Travel time (min)	100	233	284	341	276	260
	200	381	422	498	396	399
	300	671	704	771	723	735
	400	745	808	882	790	795
	500	1078	1140	1230	1108	1139
Average Vehicle throughput	100	190	219	275	284	270
	200	380	456	532	590	570
	300	568	678	720	696	690
	400	772	850	912	990	906
	500	962	1029	1076	998	1002

Table 2. Comparison of optimal solution

	Iterations	Computing time (S)	Optimal solution (Km)
Results of GA	1000	224.7	135.26
Results of Dijkstra's algorithm	1000	356.78	171.82
Results of PSO algorithm	1000	213.02	132.83
Results of ABC algorithm	1000	211.82	135.02
Results of Modified algorithm	1000	8.23	142.03
	6000	22.46	118.21

Table 3. Result obtained while performing 8 times

No. of Times	Result
1	120.29
2	123.42
3	122.34
4	119.43
5	118.21
6	123.24
7	119.82
8	118.21

The Figures 6-8 illustrates that the performance of the modified genetic algorithm with a two-step optimization technique of first using the graph reduction and then performing the optimization gave better results in contrast to other general conventional algorithms. In the network the number of nodes is denoted in the x-axis whereas y-Axis denotes the processing time, travel time and the average vehicle throughput.

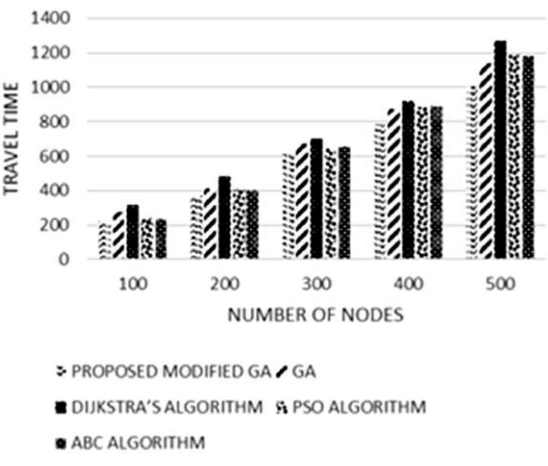

Figure 6. Analysis of the various algorithms' processing times

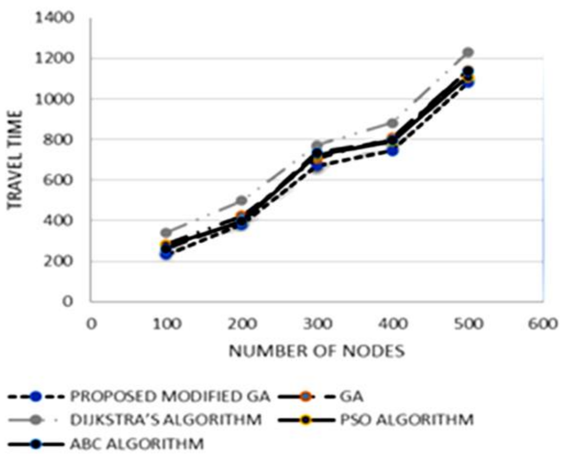

Figure 7. Comparison of the various algorithms' trip duration

In the case of the modified GA the initial graph was reduced using the graph reduction technique. The number of initial nodes was 25 which was reduced to 10 nodes. Around 156 samples were selected which were first reduced using the reduction technique and then the main parameters of GA were set as follows. The routes were selected using the tournament selection method. We used the crossover operator [15] and then we set the value of 0.05 to the mutation probability. The proposed method was tested on a New York City dataset and its efficiency was compared with the traditional GA methods, ABC algorithm and PSO algorithm as shown in Figure 7. Also, a comparative study of the average vehicle throughput for the different algorithms was done as shown in Figure 8 and the performance of the modified algorithm was comparatively better. The main parameter which was considered was its runtime and the accuracy of the answer and it was observed that we get an upgraded solution for this proposed genetic algorithm in contrast to the other algorithms. Figure 9 takes into consideration the tour length and the iteration number of the various methods, and it shows an improvement in the modified GA in comparison to other techniques.

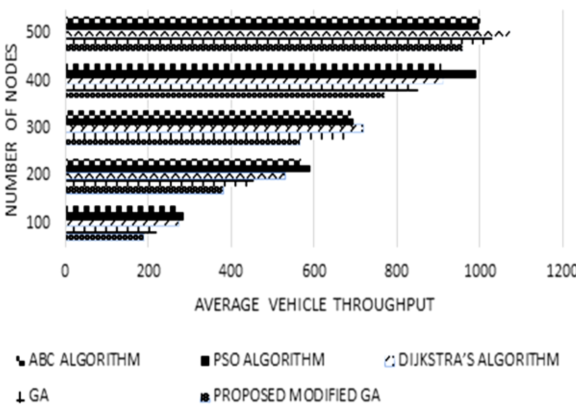

Figure 8. Comparison of the average vehicle throughput for the different algorithms

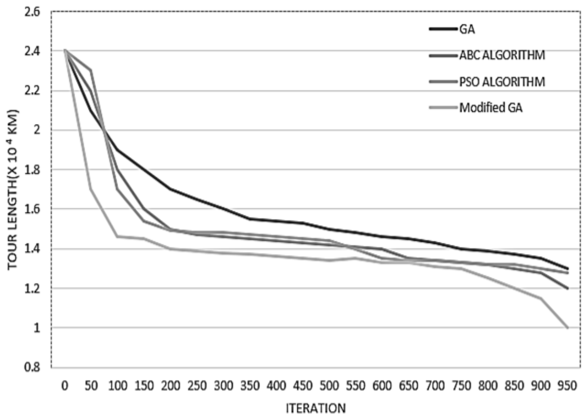

Figure 9. Comparison of the tour length Vs iteration number in the different methods

5. CONCLUSION

This paper consists of an improved hybrid genetic algorithm used to find the shortest paths. The proposed method consists of two local optimization strategies. The first strategy is a graph reduction technique which keeps only the important nodes used to find the shortest paths. The second optimization technique is a new mutation process whose intention is to avoid the path from getting stuck in local optima. The computation results show that the path which is proposed gives better results as compared to the other conventional algorithms.

REFERENCES

[1] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," *MIT press,* 1992, doi: 10.7551/mitpress/1090.001.0001.

[2] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *IEEE Trans. Industrial Electronics,* vol. 43, no. 5, pp. 519-534, 1996, doi: 10.1109/41.538609.

[3] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-A literature review," *In 2019 International Conference on Machine Learning, Big Data, Cloud And Parallel Computing (COMITCon)*, IEEE, 2019, pp. 380-384, doi: 10.1109/COMITCon.2019.8862255.

[4] A. A. Mohammed and G. Nagib, "Optimal routing in ad-hoc network using genetic algorithm," *International Journal of Advanced Networking and Applications*, vol. 3, no. 5, pp. 1323-1328, 2012.

[5] N. Makariye, "Towards shortest path computation using Dijkstra algorithm," *In 2017 International Conference on IoT and Application (ICIOT) IEEE*, 2017, pp. 1-3, doi: 10.1109/ICIOTA.2017.8073641.

[6] M. H. Xu, Y. Q. Liu, Q. L. Huang, Y. X. Zhang, and G. F. Luan, "An improved Dijkstra's shortest path algorithm for sparse network," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 247-254, 2007, doi: 10.1016/j.amc.2006.06.094.

[7] Y. Wang, J. Vrancken, and M. D. S. Soares, "Road network representation using Dijkstra's shortest path algorithm," *In 16th ITS World Congress and Exhibition on Intelligent Transport Systems and ServicesITS AmericaERTICOITS,* Japan, 2009.

[8] M. Noto and H. Sato, "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*', vol. 3, 2000, pp. 2316-2320.

[9] D. Zhang, D. Yang, Y. Wang, K. L. Tan, J. Cao, and H. T. Shen, "Distributed shortest path query processing on dynamic road networks," *The VLDB Journal*, vol. 26, no. 3, pp. 399-419, 2017, doi: 10.1007/s00778-017-0457-6.

[10] N. Shanmugasundaram, K. Sushita, S. P. Kumar, and E. N. Ganesh, "Genetic algorithm-based road network design for optimising the vehicle travel distance," *International Journal of Vehicle Information and Communication Systems,* vol. 4, no. 4, pp. 344-354, 2019, doi: 10.1504/IJVICS.2019.103931.

[11] M. Moza and S. Kumar, "Finding K shortest paths in a network using genetic algorithm," *International Journal of Computer Network and Information Security*, vol. 12, no. 5, 2020, doi: 10.5815/ijcnis.2020.05.05.

[12] S. Purusotham, T. Jayanth, T. Vimala, and K. Ghanshyam, "An efficient hybrid genetic algorithm for solving truncated travelling salesman problem," *Decision Science Letters*, vol. 11, no. 4, pp. 473-484, 2022, doi: 10.5267/j.dsl.2022.6.003.

[13] S. J. Parvez, S. Arun, R. Anandan, and H. M. Moyeenudin, "A study on geographical search and road network using progressive indicator for finding shortest path," *Journal of Computational Information Systems*, vol. 14, no. 6, 149, p. 152, 2018.

[14] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: reviews, implementations, and applications," *arXiv preprint arXiv:2007.12673*, 2020, doi: 10.36227/techrxiv.12657173.v1.

[15] E. G. Carrano, F. Campelo, and R. H. Takahashi, "Diversity-driven selection operator for combinatorial optimization," *In International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Cham, 2021, pp. 178-190, doi: 10.1007/978-3-030-72062-9_15.

[16] M. Chattoraj and U. R. Vinayakamurthy, "A self adaptive new crossover operator to improve the efficiency of the genetic algorithm to find the shortest path," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1011-1011, 2021, doi: 10.11591/ijeecs.v23.i2.pp1011-1017.

[17] R. Pinho and F. Saraiva, "A comparison of crossover operators in genetic algorithms for switch allocation problem in power distribution systems," *In 2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, IEEE, 2020, doi: 10.1109/CEC48606.2020.9185795.

[18] L. Beke, M. Weiszer, and J. Chen, "A comparison of genetic representations for multi-objective shortest path problems on multigraphs," *In European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*, Springer, Cham, 2020, pp. 35-50, doi: 10.1007/978-3-030-43680-3_3.

[19] A. P. Riascos and J. L. Mateos, "Networks and long-range mobility in cities: A study of more than one billion taxi trips in New York City," *Scientific Reports*, vol. 10, no. 1, pp. 1-14, 2020, doi: 10.1038/s41598-020-60875-w.

[20] V. Annepu and A. Rajesh, "Implementation of self adaptive mutation factor and cross-over probability based differential evolution algorithm for node localization in wireless sensor networks," *Evolutionary Intelligence*, vol. 12, no. 3, pp. 469-478, 2019, doi: 10.1007/s12065-019-00239-0.

[21] L. Manzoni, L. Mariot, and E. Tuba, "Balanced crossover operators in genetic algorithms," *Swarm and Evolutionary Computation*, vol. 54, p. 100646, 2020, doi: 10.1016/j.swevo.2020.100646.

[22] S. Mirjalili, "Genetic algorithm," *In Evolutionary algorithms and neural networks*, Springer, Cham, pp. 43-55, 2019, doi: 10.1007/978-3-319-93025-1_4.

[23] A. N. Nasir, A. A. R azak, R. M. Ismail, and M. A. Ahmad, "A hybrid spiral-genetic algorithm for global optimization," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-3, pp. 93-97, 2018.

[24] A. Bajaj and O. P. Sangwan, "Study the impact of parameter settings and operators role for genetic algorithm based test case prioritization," *In Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Amity University Rajasthan, Jaipur-India, 2019, doi: 10.2139/ssrn.3356318.

[25] F. A. Zainuddin, M. F. A. Samad, and D. Tunggal, "A review of crossover methods and problem representation of genetic algorithm in recent engineering applications," *International Journal of Advanced Science and Technology*, vol. 29, no. 6s, pp. 759-769, 2020.

BIOGRAPHIES OF AUTHORS

Mrinmoyee Chattoraj is a Research Scholar in "School of Computing and Information Technology", Reva University, Bangalore. She has 15 years of teaching experience. Currently she is working as an Assistant Professor in St. Joseph's University, Bangalore. Her research interests are computer network, machine learning and genetic programming. She can be contacted at email: mrinmoyee2005@gmail.com.

Dr. Udaya Rani Vinayakamurthy is Dr. Associate Professor in Reva University, Bangalore. She holds a Ph.D. degree in Computer Engineering with specialization in Data Mining and Warehousing. She has more than 22 years of experience and published more than 60 papers in National and International Journals. She has filed two patents and has been awarded one patent. Her research area includes data mining and warehousing, machine learning, big data analytics, big data and hadoop, genetic algorithms. She is currently guiding 8 students through their research under VTU and REVA University. Currently she is working as a Senior Associate professor in School of C&IT, Head IPR, and as a PG coordinator in REVA University. She can be contacted at email: udayarani.v@reva.edu.in.