

A Balanced Traffic Routing Using the Bio-inspired Traversing and Marking Metaheuristics

Cheikh Mouilah^{1*}, Abdellatif Rahmoun²

¹ Biomathematics Laboratory, Univ. Sidi Bel-Abbes, Sidi Bel Abbes P.B. 89, 22000, Algeria

² LabRI-SBA, ESI SBA, Higher School of Computer Science, Sidi Bel Abbes P.B. 73, 22016, Algeria

Corresponding Author Email: cheikh.mouilah@univ-sba.dz

<https://doi.org/10.18280/ria.340105>

Received: 5 October 2019

Accepted: 13 December 2019

Keywords:

nature-inspired metaheuristics, balanced traffic routing, marking algorithm, geographic information system

ABSTRACT

Recent research focuses on crossing and marking the nodes of a directed graph when it comes to shortest path algorithms. Such behavior is common to animals. Each animal marks its territory using different types of techniques such as smell and sound. In this work, the main idea is to create four populations of marking individuals, two of them looking for the “source—destination” route, and the other two looking for the opposite route. The simultaneous search of these two itineraries (alternating) or one after the other (sequential) are the two variants of this algorithm. To find a route in the urban road network, two populations cooperate, one is installed in the source node and the other is placed in the destination node. As long as there are no meetings between individuals from these two populations, they progress alternately. The first meeting points between individuals form feasible and optimal paths between the starting point and the ending point. Experimental results show that work-linked research is faster, but its risks being blocked, due to competition between individuals on the same path. However, the sequential search is more efficient and it has managed to have all the optimal solutions.

1. INTRODUCTION

The spatial data of a road network can be represented using a huge graph where its vertices are the crossroads and its edges are the segments of the street. The problem of the shortest path in a network is a big concern for several economic, military and even social areas.

Methods dealing with the shortest path go back to the 1950's, and precisely in 1957. Prim [1] published the first algorithm that performs a tree of minimal size. In 1959, Moore [2] proposes an algorithm of the shortest path, often known as “Bellman-Ford-Moore”. This algorithm is based on Ford's centralized solution and Bellman's equation [3].

In the same year, the Dijkstra algorithm is proposed to solve the problem of the shortest path in the case of positive weighting [4].

Several works in the current literature pointed that the “hub-labeling” is considered as being the fastest algorithm in this field [5].

Metaheuristics have emerged to solve all kinds of discrete problems and may also be adapted to continuous problems [6]. Hence, several metaheuristics are proposed to solve the problem of the shortest path like Tabu Search (TS) [7, 8]. Other methods based on evolutionary algorithms are also proposed such as genetic algorithms (GA) [9-12], as well as swarm intelligence techniques such as colony optimization (ACO) [13-18] and particle swarms optimization (PSO) [19, 20].

In order to improve the performance in terms of runtime; many researchers propose a hybridization scheme of some metaheuristics [21-24].

In this article, we propose a novel algorithm inspired by the behavior of striking animals. Territory marking is a natural behavior of most animals so to conquer territories for hunting, living, attracting females and forbidding to other males to trespassing such area.

2. ALGORITHM

2.1 Presentation

The main idea of this algorithm is to launch two populations in two waves; the first one is set at the starting point and the other at the end point. As long as there is no crossroad, the two waves will progress alternately. The points of crossroad are set between them (two waves) form optimal paths between the starting and ending point (Figure 1).

Two colored populations (black, red) represent the two waves in this situation.

The population is composed of a group of individuals which reproduce and kill. Each population is divided into two subpopulations (Figure 2):

- Individuals that are advancing to the destination must use the same direction of the edges.
- Other Individuals are moving in the opposite direction of the edges.

Finally, we get four (4) classes of subpopulations:

- BPA: the black population advancing (forward);
- BPB: the black population in a backward direction;
- RPA: the red population advancing (forward);
- RPB: the red population in a backward direction;





Figure 1. The evolution of both populations

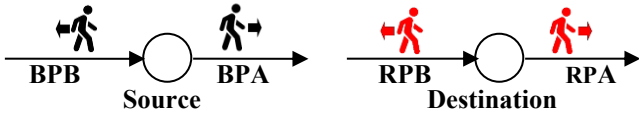


Figure 2. The four subpopulations classes

To find a path between the source and the destination, it is required to have a meeting between an individual from the BPA subpopulation and an individual from the RPB subpopulation, as well as a meeting point of one RPA individual with another from BPB, that is a path between the destination and the source. (Table 1) summarizes all of the possible situations.

Table 1. Possible meetings between individuals from the four subpopulations

	BPA	BPB	RPA	RPB
BPA	✗	✗	✗	✓
BPB	✗	✗	✓	✗
RPA	✗	✓	✗	✗
RPB	✓	✗	✗	✗

The green sign represents "a solution is found" and the red sign represents "a meeting situation to be ignored".

Any individual from the BPA or RPB subpopulation may meet other individual from the RPB or BPA respectively, thus a source-destination route is found. To have a destination-source route, a crossroad must be found between a BPB or RPA individual and another RPA or BPB individual respectively.

The advantage of this approach is to find paths between source and destination and the opposite direction.

2.2 The algorithm

The principles that manage the evolution and control of the two populations are as follows:

2.2.1 Reproduction of the individuals

An individual must reproduce if it meets more than one unmarked adjacent street; it creates new individuals in these adjacent streets. As of the moment, when these individuals (offspring) are born, the genealogical table is updated, and the "offspring" individual inherits the population characteristics from its parent (color, destination ... etc.).

However, if the individual encounters only one adjacent street, it continues its way without any reproduction.

2.2.2 The process of killing individuals

To limit massive population reproduction, any individual considered as unnecessary must be removed. This may be summarized as follows:

- After any reproduction, the "parents" must disappear;
- Another individual from the same population has already marked the adjacent street.
- Any individual in a dead end is blocked.

2.2.3 The marking

Each street in the network has two marking information, one regarding individuals that move from source to destination and the others that move in the opposite direction (Figure 3). These two marking methods act as a communication media between individuals, thus as a means of developing new solutions.



Figure 3. Marking streets scheme

The first individual marks the street, its fingerprint (individual ID) is saved in an appropriate table.

2.2.4 Competition of individuals

During exploration, individuals may encounter footprints of other individuals in the adjacent streets, this individual is considered blocked if and only if all the adjacent streets are marked; hence we come up with three possible situations:

- If the marking belongs to this individual, then it will be automatically removed;
- If its marking belongs to an individual of the same subpopulation, then this individual must be killed;
- Finally, if the marking belongs to the adjacent street of an individual from another competing population, then this consists of a new solution that should be saved.

2.2.5 Composing a path

An individual's path is a group of crossed from the starting point to the meeting point of another individual. The fusion of the two segments is the route between the source (starting point of the first individual) and the destination (starting point of the second individual).

To compose an itinerary of an individual, the algorithm uses a genealogical table (Figure 4) that groups all identifiers of the individuals created with their parents.

The search for the roots of a given individual using a recursive function from the smallest child to the largest father.

Child	Parent
①	①
2	1
③	①
4	2
⑤	③
6	3
⑦	⑤
8	5
9	⑦

Figure 4. The search for the root in the genealogical table

The root of individual 9 is: 9 - 7 - 5 - 3 - 1

If the identifier of the child individual equals the parent's identifier, then the search is completed and the root of that individual is then found.

An individual's route vertices check the adjacent streets of its location, if there is a street that is already marked by that individual or its parent; that street is retained and becomes the new position; and the process continues up to the starting point.

The following organizational chart (Figure 5) explains the principle of composition of a given individual's path based on the genealogical table and adjacent streets:

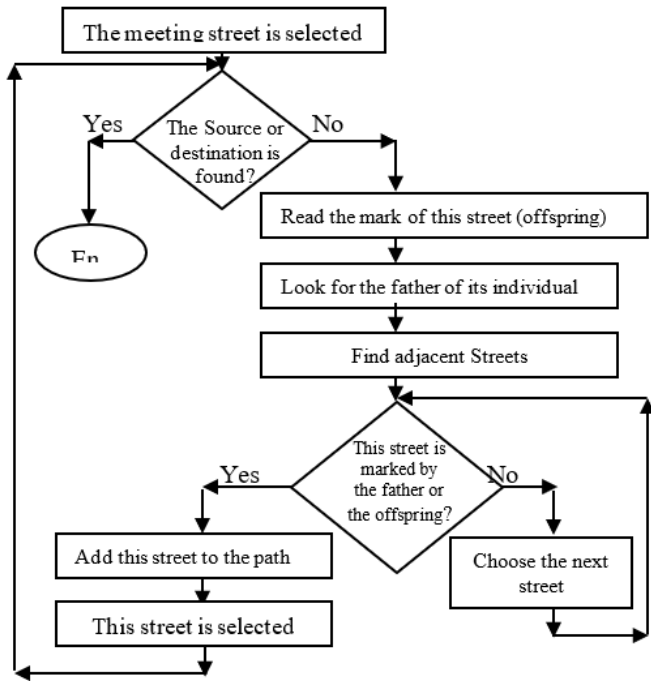


Figure 5. Organization chart of an individual's path search

This Organization chart represents a recursive function, which makes it possible to create a path for any individual, and the fusion between the two paths of two individuals makes up the itinerary between the starting point and the ending point.

2.2.6 Enhancement of the solution

The first meetings between two concurrent subpopulations (BPA—RPB or RPB—BPA) give optimal paths in relation to the number of edges between the starting and ending point. Each route that is found is seen as the shortest path in the non-weighted graph (the point of all edges equals 1).

The exploitation of the routes of the individuals encountered (that provided the solution) reveals that there are individuals (from the same subpopulation) that are blocked and then eliminated. The paths of these individuals constitute other potential solutions (as shown in Figure 6).

The paths between the starting vertex and the meeting point from the previous figure are listed in Table 2.

The exploitation of the paths of the two individuals gave us several possible paths (some of them are rejected because they are circuits); the fusion between two paths of the two individuals we met gives us a route between the starting and the ending vertex (as shown in Table 3).

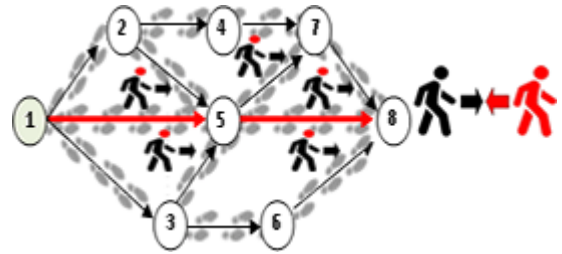
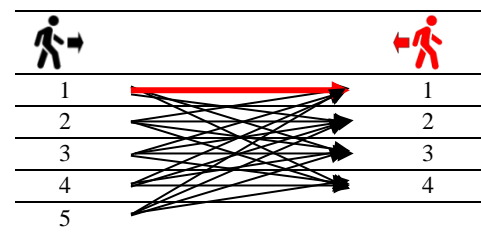


Figure 6. Itinerary “1–5–8” is the shortest path compared to the number of edges, and the paths of blocked individuals (with red heads in the figure) may lead to other feasible solutions

Table 2. List of roads found

N°	Paths	observation
1	1-5-8	Minimal path
2	1-2-5-8	
3	1-2-5-7-8	
4	1-2-4-7-8	
5	1-5-7-8	
6	1-3-5-8	
7	1-3-6-8	

Table 3. The fusion between the paths found constitutes several realizable routes



Let:

- N: is the number of paths found by the 1st individual.
- M: is the number of paths found by 2nd individual.

The total number of routes is equal to N x M.

2.2.7 Sequential or alternating research

The method of finding solutions is very important, either in the quality of the solution or in the speed of the responses. Therefore, the algorithm proposes two techniques of search and which are:

- Sequential search: the algorithm searches first for routes between the source and destination, as it ends and the results are obtained. It searches for routes between the destination and the source. To determine the direction of the search (source—destination or destination—source), it is sufficient to create appropriate subpopulations for each direction during initialization.
- Alternate research: all subpopulations participate in this research. Individuals in two directions move alternately (one iteration for source—destination and the other for destination—source). The competition between individuals on the streets may cause saturation or blockages. This critical situation is affected by time and quality of the solutions and also the progress of the algorithm. To unblock this situation; if the algorithm which determines a subpopulation is stable (there are no new births or individuals eliminated) for a specified

number of iterations, it must eliminate this subpopulation to give other individuals a chance in the opposite direction.

2.3 Pseudo code

```

Initialisation();
While (!all_population_is_empty
OR !params.SolutionFound)
{
for(inti = 0 ; i< 5 ; i++) {
params.Type = i;
params = Discover(&params);
}
}
Function Discover(void *p)
{
thisGeneration = GetGeneration(params->Type);
for (inti = 0 ; i< (int) thisGeneration.size() ; i++) {
thisPeroffspring = thisGeneration[i];
Neighbors = GetNeighbors(thisPeroffspring->Road);
For (int j = 0 ; j < (int) Neighbors.size() ; j++)
{
thisRoad = Neighbors[j];
if ((thisPeroffspring->MyDirection&&
graph[thisRoad].MyMarked == -1) ||
(!thisPeroffspring->MyDirection&&
graph[thisRoad].ItsMarked == -1))
{ AddNewPeroffspring(); }
else {
if (graph[thisRoad].MyMarked != -1 &&
graph[thisRoad].ItsMarked != -1)
{
if (Same_Population(graph[thisRoad].MyMarked,
graph[thisRoad].ItsMarked)
{thisPeroffspring->Kill = true;}
}
}
else {
GenereSolution();
} } } } }
}

```

3. RESULTS

The experiments were applied to real data from the city of Sidi Bel Abbes (a medium scale city in the west of Algeria); issued from a GIS repository(www.openstreetmap.com)

This city is represented by an oriented graph of 7581 edges[segments] and 5113 vertices[nodes]. Each edge is weighted by its distance.

This work is Inspired from grouting's work, the algorithm is programed in C++ of Microsoft Visual Studio 2010, then is converted to DLL and added to the Postgresql Kernel.

To evaluate this algorithm, we created four sets of 100 randomized routes.

The number of routes in a graph of 7581 segments is a huge number. To evaluate performance of this algorithm, we select randomly only 400 routes put in 04 sets.

The evaluation depends on two important factors that are:

- Optimality: is the shortest path between source and destination and in the opposite way.
- The time required to find an optimal solution.

An algorithm is considered optimal if it is capable of finding the shortest path between its end vertices in both directions.

The application of this algorithm with its two techniques on the 04 sets of routes (800 experiments) gave the following

results (Figure 7):

- The two techniques have found the same optimal solutions (routes) (set_1= 98.5%, set_2= 99%, set_3= 98.5% and set_4= 96%).
- The solutions of the sequential technique are much better compared to the solutions of the alternative technique in all four sets (set_1= 1.5 %, set_2= 1%, set_3= 1.5% and set_4= 4%).

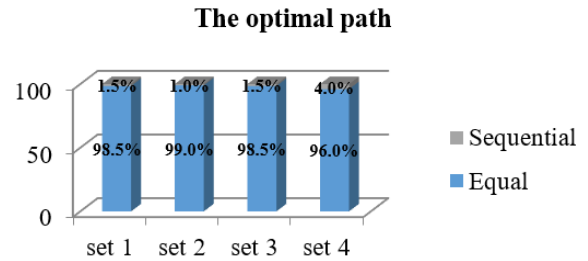


Figure 7. Optimal paths using the two techniques (sequential and alternately)

On the contrary, the alternately technique is faster than the sequential technique. In all four sets, the alternately method performs better (set_1=85%, set_2=84.5%, set_3=88.5% and set_4=80.5%), but there exist as well routes where the sequential technique is faster (set_1=14%, set_2=14.5%, set_3=10.5% and set_4=16%) (Figure 8).

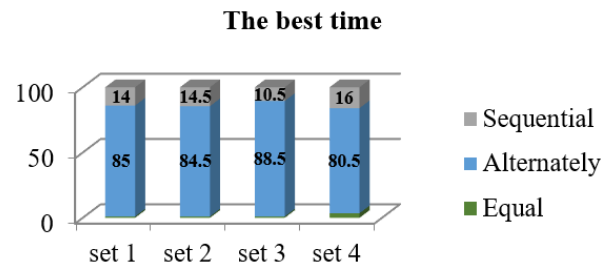


Figure 8. The speed between the two techniques

Although the alternately technique is faster than the sequential technique, it causes network saturation. This situation is due to the competition of individuals in the streets. Experiments have shown that the network is saturated once in the 3rd set, and 2 times in the 4th set (Figure 9). This saturation makes the algorithm stuck.

The saturation rate in the alternately method

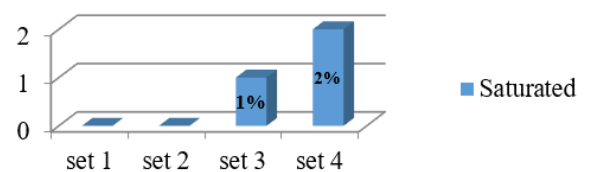


Figure 9. The saturation rate of the network in the alternately technique

In the 800 experiments, the sequential method succeeded in finding all the optimal solutions, but the alternative method was blocked in 0.75% of the experiments.

The alternative method was faster in 84% of the experiments and slower in 14%.

The two techniques found the same optimal solutions (routes) in 98% of the experiments and the solutions of the

sequential technique are much better in the remaining 02%.

In (Table 4), we resume the results obtained from the 03 experiments of the comparison.

Table 4. The summary table of the comparison between the two techniques

Methods	Optimality	The best time	Saturation
Alternately	00 %	84,625%	0,75 %
Sequential	02 %	13,75%	0%
Equal	98%	1,625	---

To properly evaluate the solutions found by this algorithm, we compared results to other well-known algorithm in route optimization (Dijkstra).

Dijkstra is an optimization algorithm to find the optimal path (the shortest path) between two vertices in an oriented and weighted graph with a single positive cost.

The proposed algorithm allows to find several feasible solutions between the two vertices of the extremity in an oriented graph, and then it chooses the optimal route between them instead; the disjkstra algorithm gives a single path, that is the shortest between two vertices.

We compare the solutions obtained by the two algorithms on the same routes.

All solutions of the sequential technique are as good as Dijkstra’s algorithm. On the contrary, the alternately technique failed to find the optimal routes in several cases (set_1 =1.5%, set_2 =1%, set_3 =1.5% and set_4 = 4%) (Figure 10).

Error rate in optimality compared to the dijkstra algorithm

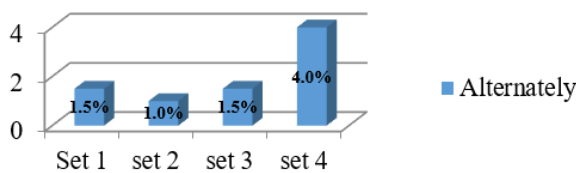


Figure 10. The failure rates of the alternately method

We then use the sequential method to solve the problem of the shortest path in an oriented and weighted graph despite the fact that is slower than the alternately method.

This metaheuristic has found all the optimal paths required in 400 different (randomly selected) routes, depending on the results obtained from the experiments. Additionally, it gave us other realizable solutions near the optimal, which is an advantage that can be exploited to solve multi-objective problems.

Another advantage of this approach is that the complete path of the graph is not necessary, to find a shorter path. Just launch two waves in the two vertices (source and destination) of a route and the intersection points of these waves can give optimal paths.

4. CONCLUSIONS

The goal of this article is to optimize routes for vehicles in the urban road networks of an Algerian city. Shortest path algorithms in the literature rely on mathematical modeling such road networks which come to be a difficult task. We therefore attempted in this work to use a metaheuristic that is inspired by the behaviors of landmark animals where each animal (an individual) marks the street it traverses with its

footprint, so that this individual reproduces, die and inherit from their parent in a given population from source to destination.

The meeting of two individuals from two different populations; constitutes an optimal route between the source and the destination and the exploitation of these optimal routes may lead to other possible paths.

The algorithm uses two vertices techniques:

- The sequential method: it has succeeded in finding all the solutions without any failure, but it is slow compared to the alternately method.
- The alternately method is faster, sensitive to saturation and less optimal compared to the sequential method.

This algorithm comes up with an optimal (“shortest”) route as well as some possible paths between source and destination in an oriented graph. Further work may emphasize on solving problems of “multi-objective” shortest path.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments and suggestions that greatly improved the presentation of this work.

This work was partially supported by the MESRS (Ministry of Higher Education and Scientific Research, Algeria) (DGRSDT, PRFU: C00L03UN220120200004 and PRFU: C00L07ES220120180003).

REFERENCES

- [1] Sullivan, H., Bashkow, T.R., Klappholz, D. (1977). A large scale, homogenous, fully distributed parallel machine, II. ACM SIGARCH Computer Architecture News, pp. 118-124. <https://doi.org/10.1145/633615.810660>
- [2] Hain, T. (2002). An IPv6 provider-independent global unicast address format. Internet Draft.
- [3] Zhang, L., Berson, S., Herzog, S., Jamin, S. (1997). Resource reservation protocol (RSVP)--Version 1 functional specification. Resource. <https://doi.org/10.17487/rfc2205>
- [4] Floyd, S., Kohler, E. (2006). RFC 4341-profile for datagram congestion control protocol (DCCP) congestion control ID 2: TCP-like congestion control. Internet Engineering Task Force, Request for Comments. <https://doi.org/10.17487/rfc4341>
- [5] Abraham, I., Delling, D., Goldberg, A.V., Werneck, R.F. (2011). A hub-based labeling algorithm for shortest paths in road networks. International Symposium on Experimental Algorithms, pp. 230-241. https://doi.org/10.1007/978-3-642-20662-7_20
- [6] Ayari, N. (2010). Métaheuristiques parallèles hybrides pour l’optimisation combinatoire: Problème de règles de Golomb. Université de Jendouba TUNISIE: 15.
- [7] Hertz, A., Laporte, G., Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. Operations Research, 48(1): 129-135. <https://doi.org/10.1287/opre.48.1.129.12455>
- [8] Brandão, J., Eglese, R. (2008). A deterministic tabu search algorithm for the capacitated arc routing problem. Computers & Operations Research, 35(4): 1112-1126. <https://doi.org/10.1016/j.cor.2006.07.007>
- [9] Ahn, C.W., Ramakrishna, R.S. (2002). A genetic

- algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6): 566-579. <https://doi.org/10.1109/TEVC.2002.804323>
- [10] Dey, A., Pradhan, R., Pal, A., Pal, T. (2018). A genetic algorithm for solving fuzzy shortest path problems with interval type-2 fuzzy arc lengths. *Malaysian Journal of Computer Science*, 31(4): 255-270. <https://doi.org/10.22452/mjcs.vol31no4.2>
- [11] Tsujimura, Y., Gen, M. (1998). Entropy-based genetic algorithm for solving TSP. 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat..No.98EX111), pp. 285-290. <https://doi.org/10.1109/KES.1998.725924>
- [12] Nagata, Y., Soler, D. (2012). A new genetic algorithm for the asymmetric traveling salesman problem. *Expert Systems with Applications*, 39(10): 8947-8953. <https://doi.org/10.1016/j.eswa.2012.02.029>
- [13] Głałowski, M., Musznicki, B., Nowak, P., Zwierzykowski, P. (2012). Shortest path problem solving based on ant colony optimization metaheuristic. *Image Processing & Communications*, 17(1-2): 7-17. <https://doi.org/10.2478/v10248-012-0011-5>
- [14] Ok, S.H., Seo, W.J., Ahn, J.H., Kang, S., Moon, B. (2011). An ant colony optimization approach for the preference-based shortest path search. *Journal of the Chinese Institute of Engineers*, 34(2): 181-196. <https://doi.org/10.1080/02533839.2011.565574>
- [15] Bezerra, L.C., Goldbarg, E.F., Buriol, L.S., Goldbarg, M.C. (2011). Grace: A generational randomized ACO for the multi-objective shortest path problem. *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 535-549. https://doi.org/10.1007/978-3-642-19893-9_37
- [16] Wang, C.X., Cui, D.W., Wang, Z.R., Chen, D. (2005). A novel ant colony system based on minimum 1-tree and hybrid mutation for TSP. *International Conference on Natural Computation*, pp. 1269-1278. https://doi.org/10.1007/11539117_168
- [17] Skinderowicz, R. (2013). Ant colony system with selective pheromone memory for SOP. *International Conference on Computational Collective Intelligence*, pp. 711-720. https://doi.org/10.1007/978-3-642-40495-5_71
- [18] Mouilah, C., Belkadi, K. (2012). Application of "AntNet" in an urban road network: Case of Algerian cities. *International Conference on Education and e-Learning Innovations*, pp. 1-7. <https://doi.org/10.1109/ICEELI.2012.6360648>
- [19] Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, Q. (2007). Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, 103(5): 169-176. <https://doi.org/10.1016/j.ipl.2007.03.010>
- [20] Liao, Y.F., Yau, D.H., Chen, C.L. (2012). Evolutionary algorithm to traveling salesman problems. *Computers & Mathematics with Applications*, 64(5): 788-797. <https://doi.org/10.1016/j.camwa.2011.12.018>
- [21] Mohemmed, A.W., Sahoo, N.C. (2007). Efficient computation of shortest paths in networks using particle swarm optimization and noising metaheuristics. *Discrete Dynamics in Nature and Society*, 2007(4): 1-25. <https://doi.org/10.1155/2007/27383>
- [22] Davoodi, M., Golsefidi, M.M., Mesgari, M. (2019). A hybrid optimization method for vehicle routing problem using artificial bee colony and genetic algorithm. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42: 293-297. <https://doi.org/10.5194/isprs-archives-XLII-4-W18-293-2019>
- [23] Dudeja, C. (2019). Fuzzy-based modified particle swarm optimization algorithm for shortest path problems. *Soft Computing*, 23: 8321-8331. <https://doi.org/10.1007/s00500-019-04112-1>
- [24] Maleki, F., Yousefikhoshbakht, M. (2019). A hybrid algorithm for the open vehicle routing problem. *Iran University of Science & Technology*, 9(2): 355-371.