

Solving the Vehicle Routing Problem with Genetic Tabu Search Algorithm

Iaia So

School of Information Technology and Engineering, Tianjin University of Technology and Education, Tianjin, China

Abstract: This paper aims to develop a genetic algorithm to solve a travel salesman problem (TSP). The algorithm is used to find the shortest path between the 25 cities of source and destination. In the literature the routing problem is solved by using search graph technique to find the shortest path. The main objective is to minimize the traveling cost and time. There are many researchers tried to solve this problem previously by using different methods and using different algorithms to get the better and effective solution. and in this ethic, I also work to get a better result than what they did before by using combination GA and TS algorithms. Our proposed GA has optimized implementation that makes our algorithm faster and efficient to solve the famous Traveling Salesman Problem (TSP). when compared with the state-of-the art Tabu Search Algorithm to solve the same TSP, our genetic algorithm achieved significant improvements in terms of accuracy. An efficient MATLAB implementation makes our code ready for the deployment in environment that has constraints on the system memory and speed. The results affirmed the potential of the proposed genetic algorithm. The obtained performance is better than genetic algorithm and Tabu search algorithm.

Keywords: Vehicle Routing Problem (VRP); Genetic Algorithm; Tabu search.

1. Introduction

The VRP is based on a very simple optimization problem which is called the traveling salesman problem. In this problem there are a given number of cities (n). A Traveling Salesman has to visit all of these (n) cities, each only once and he should return to the city he was starting his tour from. The goal is to optimize this problem, so to be able to ask the question: "which one is the shortest or cheapest, depending on the measure used path for the Salesman?" Solving this problem with a small number of cities seems to be very easy. Increasing the number of the cities, the problem is more complex and complex. This work will not focus on solving the TSP, just as a basis of the VRP, the work will boundary discuss it. TSP has applications in robotics, circuit board drilling, welding, manufacturing, transportation, and many other areas. An open n -city TSP has $(n-1)!$ Possible solutions. This number becomes impossibly large for even moderate values of n . For example, the number of possible solutions to a 50 – city TSP is $49! = 6.1 \times 10^{62}$. And 50 cities are not very many for a TSP. A circuit board could have tens of thousands of holes, and a drill need to be programmed to visit each of these holes while minimizing some cost function (time or energy, for example).

Zirour et al. [5] presented a genetic algorithm to solve the VRP with time windows and fuzzy Demand to minimize the aggregate distance for vehicles paths and the delay times at the customers due to any time violations. The GA is utilized to solve this problem which is formulated in two steps, in order to minimized the expenses in both steps due to initial solutions in the first step and the expenses of path failure in the second step. Their research showed that their solutions are close to best solutions.

Nazif et al. [6] present a genetic algorithm with an optimized crossover operator to solve the dynamic VRP with time window. Improved operator was applied by Aggarwal et al.

Bodin, Golden, Assad, and Ball (1983) propose the first solution approach for the OVRP. From the early 1980s to the late 1990s, the OVRP received very little attention in the operations research literature. However, recently OVRP has attracted more and more concern from practitioners and researchers. Since 2000, several researchers have used various heuristics and meta-heuristics to solve the OVRP with some success [21].

Brandão (2004) and Fu, Eglese, and O Li (2005) he carried out a Tabu Search Algorithm (TS) heuristic to solve the OVRP with constraints on vehicle capacity and maximum route length [10].

M.A. Mohamed, M.S. Ahmed, S.A. Mostafa, The VRP models are applied in a wide area of transportation and distribution such as transportation of individuals and items, conveyance service and garbage collection. The models have economic importance, particularly in developed countries. The economic factor in savings expenditures is a big motive for companies and researchers in an attempt to find the best way to resolve and improve transport efficiency [22].

S.A. Mostafa, M.S. Ahmed, The concept of VRP can be described as the issues of designing shortest paths routes from one location to a group of geographically distributed locations (customers, cities, universities, warehouses, schools, stores, etc.) [1-23].

G. Laporte, for example, a fleet of vehicles (for distribution of goods) starts from one location and visits a group of scattered cities or customers and return to the same location with less distance and costs on the conditions of [2]:

- Every city is visited by one vehicle only once within a single route.
- The capacity of each vehicle is enough for all cities included in the route.
- Routes begin and end at the same location.

The number of vehicles is supposed to be less than what can be proposed for routes as well as the number of routes is to be less than what can be provided to cover all cities.

2. Traveling Salesman Problem and Genetic Algorithm

Details of the solution by the problem to be solved in this TSP paper. where in the first place we give a brief general summary of TSP, and with the key words of our research. and with a general introduction of Solving Vehicle Routing Problem. And finally, we show the chosen method so that we can solve this TSP problem where we will use GA and is operators.

1) Traveling Salesman Problem (TSP) is an algorithm problem tasked with finding the shortest route between a set of points and locations that must be focused on optimization, TSP is often used in computer science to find the most efficient route for data to travel between various nodes. Applications include identifying network or hardware optimization methods. It was first described by Irish mathematician W.R Hamilton and British mathematician Thomas Kirkman in the 1800s through the creation of a game that was solvable by finding a Hamilton cycle, which is a non-overlapping path between all nodes. TSP has been studied for decades and several solutions have been theorized. The simplest solution is to try all possibilities, but this is also the most time consuming and expensive method. Many solutions use heuristics, which provides probability outcomes. However, the results are approximate and not always optimal. Rather than focus on finding the most effective route, TSP is often concerned with finding the cheapest solution. In TSPs, the large number of variables creates a challenge when finding the shortest route, which makes approximate, fast and cheap solutions all the more attractive.

In general, we will assume that an n -city TSP has cities denoted as city 1, city 2, ..., city n . we assume that there is a given distance $D(i, j)$ between cities i and j for all $i \in [1, n]$ and $j \in [1, n]$, and that $D(i, j) = D(j, i)$. This is called the symmetric TSP because the distance (or cost) from city i to city j is the same as the distance from city j to city i . We could imagine scenarios where $D(i, j) \neq D(j, i)$ (for example, it might cost more to go uphill than downhill) – such problems are called asymmetric TSP, but we do not discuss them further in this paper. In the TSP, we try to minimize total distance. Suppose that n cities in an open TSP are listed in the order $x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n$. Then the total distance is:

$$D_T = \sum_{i=1}^{n-1} D(x_i, x_{i+1})$$

Note that we use the term “distance” in a general sense. It might refer to physical distance, financial cost, or any other quantity that we want to minimize in a combinatorial problem.

2) Genetic algorithm (GA) is a method, based on the example of the natural selection in the “real natural world”. Ga uses several algorithms and methodologies to create better and better approximate solutions for a given problem,

using the older solutions (parents) to create new ones (children). This will be led to an approximate solution of the given solution space. It can be said, that – in the nature – over time it will lead to populations with individuals having better and better abilities of adaptation, and better chances of survivorship. The GA in nature: Evolutionary biology describes the methodology of adaptation of the different life structures and generations to their environment. To understand the method, some concepts have to be declared. At the beginning an initial generation has to be defined. This can be done using a random initialization or can use some kind of seeding which allows the algorithm to work in a search space where solutions are more likely. From now until a valid solution is found or the maximal level of allowed generations is reached, the following steps are performed.

a) Chromosome

In a TSP context, the chromosome encodes a solution to the problem (i.e., a tour). The fitness of the chromosome is related to the tour length, which in turn depends on the ordering of the cities. Since the Traveling Salesman Problem is a minimization problem, the tour lengths must be transformed so that high fitness values are associated with short tours, and conversely. A well-known approach is to subtract each tour length to the maximum tour length found in the current population. Other approach is based on the rank of the tours in the population.

b) Selection

In selection of individuals for reproduction to constitute a new a population (often called a new generation) is usually based upon fitness values. The higher the fitness, the more likely it is that the individual will be selected for the new generation. Some paradigms that are considered evolutionary, however, such as particle swarm optimization, can retain all population members from epoch to epoch. In fact, Bentley (1999) states that “the preferential selection of some parents instead of others is not essential to evolution”. (If there is no selection operator, then all population members produce offspring with equal probability). Termination of the algorithm is usually based either on achieving a population member with some specified fitness or on having run the algorithm for a given number of generations.

c) Reproduction

During the reproduction phase the next generation is created using the two basic methods, crossover and mutation. For every new child a pair of parents is selected from which the child inherits its properties. In the crossover process genotype is taken from both parents and combined to create a new child.

With a certain probability the child is further exposed to some mutation, which consists of modifying certain genes. This helps to further explore the solution space and ensure, or preserve, genetic diversity. The occurrence of mutation is generally associated with low probability. A proper balance between genetic quality and diversity is therefore required within the population in order to support efficient search.

d) Crossovers

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is an analogy to reproduction and biological crossover, upon which genetic algorithms are based. Both implemented crossovers do mutual exchange of genetic material between two parents. They take information from one individual and insert it in the other to create a new child. The probability which crossover method should be used can be configured.

e) Mutation

In genetic algorithms, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosome to the next. In the TSP mutation operator are aimed at randomly generating new permutations of the cities. As opposed to the classical mutation operator, which introduces small perturbations into the chromosome, the permutation operators for the TSP often greatly modifies the original tour.

Algorithm 1: Genetic Algorithm pseudo code

Step 1: Choose initial population
 Step 2: Evaluate the fitness of each individual in the population
 Step 3: Repeat until termination: (time limit or sufficient fitness achieved)
 a) Select best-ranking individuals to reproduce
 b) Breed new generation through crossover and/ or mutation (genetic operations) and give birth to offspring
 c) Evaluate the individual fitnesses of the offspring
 d) Replace worst ranked part of population with offspring
 e) Step 4: Get the best solution

3. Proposed Algorithm

In the third part of the work that is proposed algorithm to solve this problem of TSP to obtain a better result by our problem of SVRP. Where we created a new algorithm called GTS to obtain the best result. this new Genetic Tabu Search algorithm which is the result of the addition of Tabu Search Algorithm in Genetic Algorithm.

Tabu Search is an iterative procedure designed for the solution of optimization problems. Tabu Search has been used to solve a wide range of hard optimization problems. The Traveling Salesman Problem (TSP) and the capacitated are routing problem. Tabu Search is a meta-strategy for guiding known heuristics to overcome local optimality and has now become an established optimization approach that is rapidly spreading to many new fields. The method can be viewed as an iterative technique which explores a set of problem solutions, denoted by X , by repeatedly making moves from one solution S to another solution S' located in the neighborhood $N(s)$ of s . These moves are performed with the aim of efficiently reaching an optimal solution by minimizing some objective function $f(S)$.

Tabu search is implemented beginning with a random mapping as the initial solution, generated from a uniform distribution. To manipulate the current solution and move through the solution space, a short hop is performed. The intuitive purpose of a short hop is to find the nearest local minimum solution within the solution space. The basic

procedure to perform a short hop is to consider, for each possible pair of tasks, each possible pair of machine assignments, while the other assignments are unchanged. If the new makes pan is an improvement, the new solution is saved, replacing the current solution. The short hop procedure ends when (1) every part-wise remapping combination has been exhausted with no improvement, or (2) the limit on the total number of successful hops (limit hops) is reached. When the short hop procedure ends, the final mapping from the local solution space search is added to the to the tabu list.

Combined Heuristics

Genetic Algorithm can be combined with Tabu Search algorithm to create combinational heuristics. For example, The Genetic Tabu Search Algorithm (GTS) heuristic is a combination of the Genetic Algorithm and Tabu Search techniques. In general, GTA follows procedures similar to the GA outlined above. However, for the selection process, GTA uses the Tabu Search process.

These nature's heuristics were only relatively introduced into the scheduling area and more work needs to be done. There are a lot of interesting questions. First, the meaning of controlling measurements in such heuristics needs to be refined. For example, each possible pair of tasks, each possible pair of machine assignments, while the other assignments are unchanged.

If the new makes pan is an improvement, the new solution is saved, replacing the current solution. Second, there is a trade of between the search cost and the degree of optimality of solutions found. For example, in a genetic algorithm, historical knowledge can be used to guide the chromosome selection, crossover, or mutation process so that the search process can converge quickly. But this adjustment seems to contradict the philosophy of an evolutionary algorithm: randomization and diversity generate better results, and may not bring a better solution.

Experiments have shown that a good initial solution for Tabu Search improves both the quality of the solution as also execution time. We require m initial solutions for distributing among m nodes; we choose to combine Tabu Search with Genetic Algorithm.

In Genetic Algorithm, an initial population consisting of a set of solution is chosen and then the solution is evaluated. Relatively more effective solutions are selected to have more offspring's which are, in some way, related to the original solutions. If the genetic operator is chosen properly, the final population will have better solutions. GA improves the whole population. TS aims at producing one best solution. For the TS, we require several good initial solutions to ensure the required number of good initial solution.

And we use Tabu search operator to create new child by mutation operator, after we create a new child then we need to check if this child is in Tabu List just we drop it and go back to create a new one to check again with is not in Tabu List then we can add this new child.

4. Implementation

We have used MATLAB programming language to implement our system. The main advantages of MATLAB

include a clean object-oriented approach. The following figure describes the flowchart diagram of the system.

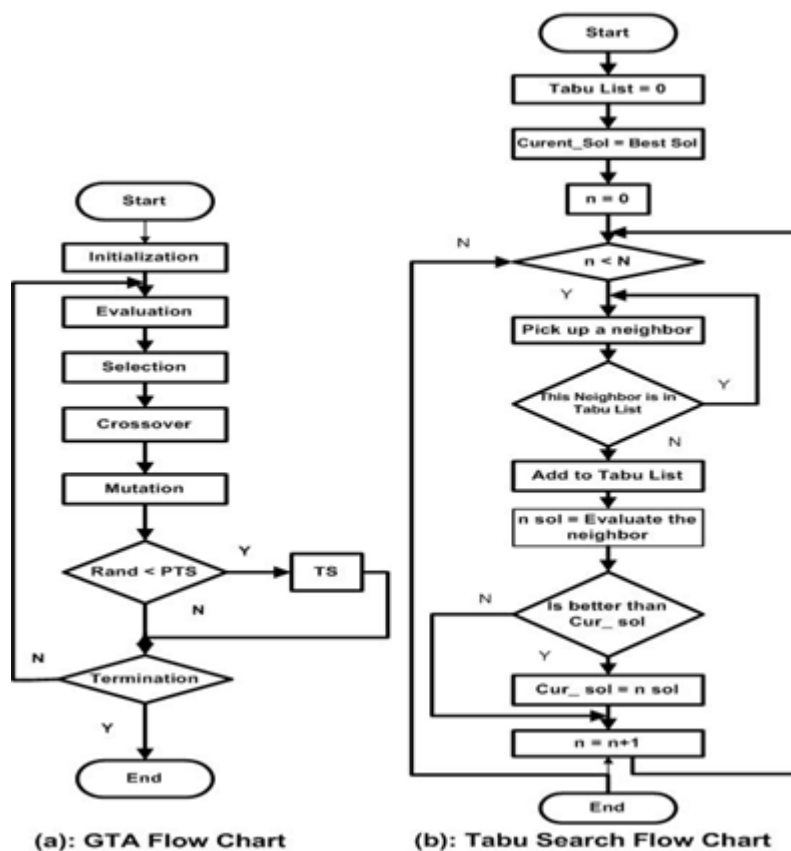


Figure 1: Flow Chart of solution of TSP combine Genetic Algorithm with Tabu Search

5. Experiments and Results

In this paper for experiment and result for our TSP we have been used the data set came from the paper result published on the internet with description “A hybrid GA-TS algorithm for Open Vehicle Routing Optimization of coal mines material”, with a data set the 25 Cities in China Zhengzhou with a population size 200, and 500 generation with 543 best path for the best final result.

In this part or section will contains the summary of all activities that haven been carried out during the experimental analysis process for the proposed algorithm. It will exclude the hardware requirement, software requirement to design the algorithm itself also, the section will contain the parameters, table of the data sets used by the proposed method and some output results including the graph that resulted by the new Genetic Tabu Search Algorithm.

The result we got with simple Genetic algorithm is shown below with same data file.

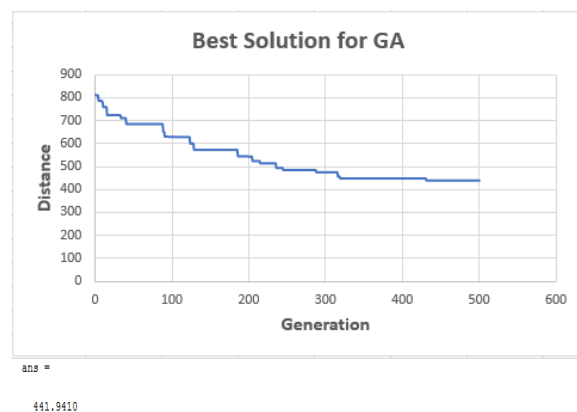


Figure 2: Best Solution for 25 cities with GA for TSP

In figure number two that is shown above the result obtained for the solution on this paper TSP problem. Which means finding the solution for the best solution with the least distance traveled. There in this figure we use genetic algorithm and its operators to obtain a better result of TSP in GA.

The result we got with simple Tabu Search algorithm is shown below with same data file.

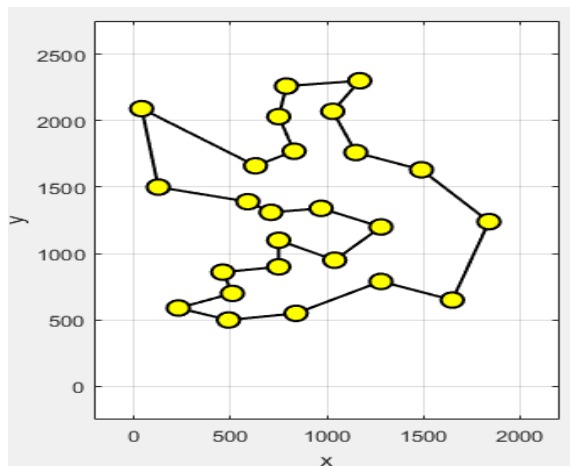
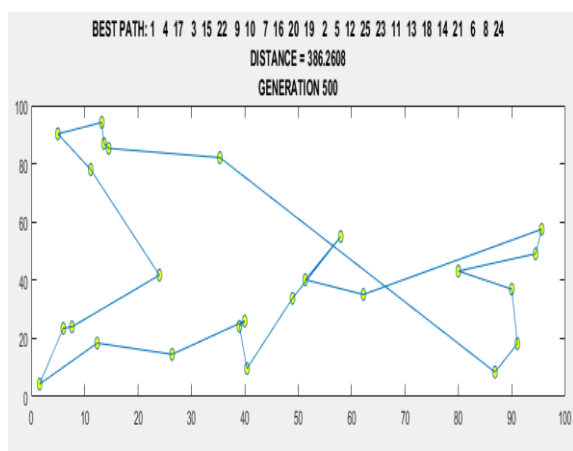


Figure 3: Best path with Tabu Search Algorithm for 25 cities for TSP

In figure number three that is shown above the result obtained for the solution on this paper TSP problem. Which means finding the solution for the best path with the least distance traveled. There in this figure we use Tabu Search Algorithm to obtain a better result of TSP.



ans =

415.5767

path6 10 18 5 12 15 9 17 19 25 22 21 3 14 11 7 24 13 23 16 2 20 4 8 1

Figure 4: Best Solution for 25 cities with GTS for (TSP)



In the figure number four that shown our final result of our Traveling Salesman Problem. In this figure that results we get from a combination of two algorithms GA and TS algorithm, where we combine the two algorithms and obtain our new algorithm that is GTS. Where our new algorithm allows us to obtain the best result for our TSP problem, and we can see in figure number four named Genetic Tabu Search Algorithm.



(a): Best Solution for GTS and GA



(b): Best Solution for Tabu Search

Figure 5: Representation of the three algorithms in twographs with different results and different colors

In the lastly two graphs, it is the combination of the three more algorithms, where I have two different figures the first figure is the illustration of GTS and GA with two different colors where all the two are using the same distance and the same iteration more with different results where we can see that GTS show the better result than GA as shown in the figure number one. In the second graph represent TS

algorithm result for TSP problem and we compare this result with GTS and GA according to our result GTS is better then

GA and TS Algorithm.

Table 1: Comparison of GA, GTS and TS performance for the TSP

Number of Cities	Algorithm	Parameters	Best Solution	Computing Time (s)
25	GA	Pop_size = 200; max gen = 500	441.9410	35.515 s
25	TS	ts_len = 4; ts_loop = 500	8934.3827	38.711 s
25	GTS	Pop_size = 200; max gen = 500	415.5767	34.214 s

In this general table shown above shows the comparison of the final results of our TSP obtained between the three algorithms represented in the graphs above. In figure number one of the first row of the table shows the results obtained by the genetic algorithm. In the second graph second line of the table shows us the results obtained by using TS algorithm and in the last figure and the last line of the table that shows us the best result of solving our problem that is our algorithm we created that is the addition of TS in GA algorithm where we called GTS algorithm.

6. Conclusion and Future Work

In the last part of my work that is the conclusion and future work in TSP. Then we did a general analysis and the comparison of the two algorithms that are GA and TS algorithm. Then our new algorithm we created that GA with addition of TS where we had the best final result that is better than GA and better than TS algorithm. And lastly where we tested all of them all in a single programming language called MATLAB and we can find the table above with all the information about our algorithm listed in table 1.

On this paper we have been presented a Genetic Tabu Search Algorithm for the Traveling Salesman Problem (TSP). The GTA algorithm is compared with the Tabu search algorithm (TS). The final result shows that the GTS give the better result than the existing problem GA, TS. The algorithm showed a very good result for the number of the nodes increased. The algorithm is tested in different work station. In future the GTS algorithm is experimented with the dynamic switching from Genetic Algorithm to Tabu Search and also the same algorithm will be implemented in network of heterogeneous work stations.

References

- [1] M. Fisher. Vehicle routing. Handbooks of Operations Research and Management Science, Chapter 1,8:1-31, 1995.
- [2] G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms, Eur. J. Oper. Res. 59 (3) (1992) 345–358.
- [3] O. Bräysy, M. Gendreau, Genetic algorithms for the vehicle routing problem with time windows, Arpakannus 1 (2001) 33–38.
- [4] M.A. Mohammed, M.K.A. Ghani, R.I. Hamed, M.K. Abdullah, D.A. Ibrahim, Automatic segmentation and automatic seed point selection of nasopharyngeal carcinoma from microscopy images using region growing based approach, J. Comput. Sci. 2017, <http://dx.doi.org/10.1016/j.jocs.2017.03.009>
- [5] M. Zirour, Vehicle routing problem: models and solutions, J. Qual. Meas. Anal. JQMA 4 (1) (2008)205–218.
- [6] H. Nazif, L.S. Lee, Optimized crossover genetic algorithm for vehicle routing problem with time windows, Am. J. Appl. Sci. 7 (1) (2010),p.95.
- [7] S. Zhong-yue, G. Zhong-liang, W. Qin, An improved adaptive genetic algorithm for vehicle routing problem, in: Logistics Systems and Intelligent Management, 2010 International Conference on 1, 116–120. IEEE, January,2010.
- [8] Y. Zhang, J. Liu, F. Duan, J. Ren, Genetic algorithm in vehicle routing problem, in: Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on 2, 578–581. IEEE., November,2007.
- [9] H.C. Lau, T.M. Chan, W.T. Tsui, W.K. Pang, Application of genetic algorithms to solve the multi depot vehicle routing problem, IEEE Trans. Autom. Sci. Eng. 7 (2) (2010) 383–392.
- [10] Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. Computation Operation Research, 10(2),3–211.
- [11] Brandão, J. (2004). A Tabu search algorithm for the open vehicle routing problem. European Journal of Operational Research, 157(3), 552–564. Cao, E., & Lai, M. (2010). The open vehicle routing problem with fuzzy demands. Expert Systems with Applications, 37(3),2405–2411.
- [12] Dantzing, G., & Ramser, J. (1959). The truck dispatching problem. Management Science, 10(6),8091.
- [13] Dautzère-Pérès, S., & Paulli, J. (1997). An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using Tabu search. Annals of Operations Research, 70(0), 281–3306.
- [14] Davis, L. (1985). Applying adaptive algorithms to epistatic domains. In Proceeding of 9th International Joint Conference on Artificial Intelligence (pp. 162–164). Citeseer, Greece.
- [15] Fleszar, K., Osman, I. H., & Hindi, K. S. (2009). A variable neighborhood search algorithm for the open vehicle routing problem. European Journal of Operational Research, 195(3), 803–809.
- [16] Fu, Z., Eglese, R. W., & Li, L. (2006). A new Tabu search heuristic for the open vehicle routing problem. Journal Operation Research Society, 56(3),263–273.
- [17] Fu, Z., Eglese, R., & O Li, L. Y. (2005). A new tabu search heuristic for the open vehicle routing problem. Journal of the Operational Research Society, 56(3), 267–274.

- [18] Glover, F. (1990). Artificial intelligence, heuristic frameworks and Tabu search. *Managerial and Decision Economics*, 11(5), 365–375.
- [19] Glover, F., Eric Taillard, E., & Eric Taillard, E. (1993). A user's guide to Tabu search. *Annals of Operations Research*, 41(1), 1–28.
- [20] Goldberg, D. E., Lingle, R. (1985). Alleles Loci and the traveling Salesman Problem. In *Proceedings of the 1st international conference on genetic algorithms* (pp. 154–159). <https://wwwproxy.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>
- [21] Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computation Operation Research*, 10(2), 3–211.
- [22] M.A. Mohammed, M.S. Ahmad, S.A. Mostafa, Using genetic algorithm in implementing capacitated vehicle routing problem, in: *Computer & Information Science (ICCIS), 2012 International Conference on* (Vol. 1, pp. 257–262). IEEE, June, 2012.
- [23] S.A. Mostafa, M.S. Ahmad, M.A. Mohammed, O.I. Obaid, Implementing an expert diagnostic assistance system for car failure and malfunction, *IJCSI Int. J. Computer Science. Issues* 9 (2) (2012), 1694–0814.

Author Profile



IAIA SO received the Bachelor Degree in Information System and Network Engineering from Dakar *IFAA* University in Senegal in 2011. Currently pursuing the M.E in Applied Computer Technology from Tianjin University of Technology and Education.

Mr. Iaia SO was working as Network assistant (IT) at Telecom Company MTN Guinea Bissau. from September 2013 – August 2016. He based on computer vision, Network as research areas.