Optimal Inspection Routes Problem

ASAE Use Case

Luís Henriques
Faculty of Engineering, University of Porto, DEI
Porto, Portugal
up202204386@fe.up.pt

Farzam Salimi
Faculty of Engineering, University of Porto, DEI
Porto, Portugal
up202204386@fe.up.pt

***Abstract*— In this paper we aim to apply genetic algorithm, simulated annealing, tabu search, and hill climbing to the vehicle routing problem. The routes were optimized by two different heuristics of greedy search and A-star method and these algorithms applied to the ASAE data and tested in four different sizes of establishments. Finally, the results of these algorithms compared to each other and conclusion has been made. (**_Abstract_**)**

Keywords—vehicle routing problem, genetic algorithm, simulated annealing, tabu search, hill climbing (**_key words_**)

I. INTRODUCTION

The operation research and optimization have been improved immensely during the past few years. One of the most challenging subjects and problems in this area of research is the Vehicle Routing Problem (VRP). In this problem we try to optimize the vehicle routes to support set of nodes or in ASAE problem sets of establishments. In these types of problem, the difficulty normally increases when we start to include more real-life conditions and situations such as specific working hours, or certain types of utility managements. The VRP has many different applications in the industry and it has many uses in the transportation and logistics.

In this article we present our work of solving a VRP problem using the metaheuristics on the set of data related to the ASAE. ASAE is a specialized authority in Portugal that focuses on ensuring the food safety and economic surveillance. We mainly focused on the 4 optimization algorithms of Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), and Hill Climbing (HC). These algorithms have been applied to the different scenarios and results of them compared to each other.

This article divided into different sections. In the first section we briefly go through the literature review and reviewing the most recent works in this area and the different techniques that have been used to solve this problem. Also, we will give the brief comparison between these methods and algorithms. In the methodology section we focus on the dataset and the scenarios. The ASAE dataset has specific features and the scenarios, as well, has its own challenges that we will discuss. In addition to that, in this section we will focus on how we applied different algorithms such as genetic algorithm, simulated annealing, tabu search and hill climbing to the data to solve the problem and heuristics that we used to optimize the routes internally and discuss the parameters we used in these methods.

In the result section, we present the result of applying different algorithms on the different data sizes (especially medium and large sizes of establishments) under different scenarios and compare them to each other. In this part, we will interpret the results and discuss the findings of applying different methods on different sets of data. In the conclusion section, we summarize our findings and the most important points of the article and talk about possible future improvements.

II. LITERATURE REVIEW

In the field of operational research and optimization vehicle routing problem has received a significant amount of attention among researchers in the previous years, and many different techniques has been proposed to solve this problem such as heuristic and meta heuristics. Heuristics are one of the most widely used algorithms that aims to find a good solution for the optimization problem in the acceptable amount of time [1]. One of the most popular heuristics for the VRP is the nearest neighbor algorithm which starts with an arbitrary node and select the nearest node that has not been visited yet. There are also other heuristics such as Clarke and Write algorithm, the sweep algorithm, and the saving algorithm. These types of methods cannot guarantee the global optimum and mostly gives the result close to global optimum [2].

On the other hand, metaheuristics are the types of algorithms that they aim to search the solution space and discover a good solution. As an example of metaheuristics, we can name genetic algorithms, simulated annealing, tabu search, and hill climbing, and these methods have been applied to solve the vehicle routing problem [3]. Genetic algorithm populates candidate solutions and apply the genetic operators such as mutation and crossovers to populated new solutions. These types of algorithms use a fitness function and do estimations based on that function. Simulated annealing is another creative way that uses the metallurgy process and cooling down the temperature concept to search the solution space. Also, tabu search is another metaheuristic that is based on the memory and we will use it in this study [4].

For the small scale problems and small sets of data, normally different exact methods can reach equally to the optimal solution quit fast, however, for the problems with the big sets of data or larger scale they most of the time become infeasible [5]. On the other hand, heuristics and meta heuristics can provide a good quality solution (most likely not global optimum) for the large-scale problems and in reasonable amount of time as well.

Overall, different meta heuristics such as genetic algorithms have been used for the large-scale problems because of their efficiency and time efficient characteristics and in most cases, they can find a good quality solution for these types of problems.

III. Methodologies

In this section we start with describing the data and the 2 scenarios/problems that we worked on, then we continue with the explanation of the 4 meta-heuristics selected for this project, those being Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) and Hill Climbing (HC). Finally, we discuss the parameters that we used for each meta-heuristic and how we chose them.

The dataset used in this project pertains to establishments in the Porto district of Portugal. The dataset comprises two CSV files, namely "establishment.csv" and "distances.csv". The first mentioned file contains comprehensive information on 1000 establishments, including their unique identifier (ID), name, geographical locations, latitude, longitude, estimated inspection duration, inspection utility value, and opening hours. In this file, the opening hours have a binary array representation with 24 indexes, where 0 means the establishment is closed, and 1 is the opposite. For example, if an establishment is open from 9 am to 12 am, then the indexes between the 9th and the 12th have a value of 1. In other words, the index value, starting from zero, is the actual hour, considering the 24-hour format.

The second file is the "distance.csv", which consists of a travel time matrix, in seconds, between each of these 1000 establishments and between the departure/arrival establishment (depot), shown by ID zero, and because of this, the size of the matrix is 1001x1001. For that same reason, the index shown as zero in the establishment's file was also considered as the data for the depot. The data structure enables the analysis of the establishment's inspection efficiency, considering the location, inspection duration, utility, and opening hours. During the evaluation process we divided this dataset into 4 sizes, that way the program could be tested for a simpler and easier-to-understand set of data, and after having the program correct, expand the analysis to more complex analysis with more and more data. The first set of data has 20 establishments, the second 100 establishments, the third 500 establishments, and the last had the full dataset. Notice that the depot is always included in each set of data. These datasets were called very small, small, medium, and large.

The presented problem includes finding the optimal inspection routes for ASAE, which is responsible for ensuring compliance with regulations among economic operators in Portugal. The problem is a simplified version of the actual ASAE inspection route problem and is classified as an instance of the Vehicle Routing Problem (VRP). ASAE uses inspection brigades, and each brigade is assigned a specific set of routes that they should go and inspect the operators. In this regard, we focused on two different scenarios and used metaheuristics to find solutions as close to the global optimum as possible.

From an artificial intelligence perspective, the first step in a COP is its formulation, composed of solution representation, neighborhood/mutation, crossover functions, rigid constraints, and evaluation functions. These will be defined for each problem solved

.

First scenario:

In the first scenario, we have a finite number of vehicles available, and our goal is to minimize the total travel time, which includes travel, waiting, and inspection time. The number of vehicles is assumed to be 10 percent of the data size.

Regarding the formulation, the solution representation is an array where the values are the vehicles (or route number) and the index number plus one the establishments to which each vehicle was assigned. In other words, for 20 establishments we have 2 vehicles available, and so a random solution for the problem could be [1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1]. This solution means that the two routes have the following set of establishments, 1: {1, 2, 3, 5, 8, 12, 13, 14, 16, 18, 19, 20} and 2: {4, 6, 7, 9, 10, 11, 15, 17}. Although the usual representation is with the id's of the economic operators, the solution representation can be changed to that exact representation by finding a way to order each set of establishments in each route. The solution representation chosen doesn't invoke any constraints considering unfeasible solutions after applying genetic or neighborhood operators; plus, this solution representation enables the possibility to compare the same distribution of establishments with the two objective functions created, as they are the ones that will set the route's order. These two objective functions will be explained below.

For SA, TA, and HC, we used a specific neighboring system. In this system, we define two neighboring methods in which one of which generates a new solution by randomly selecting an establishment and assigning it to different vehicles. The second system generates a new solution by exchanging the vehicles between two establishments.

In the genetic algorithm, two mutation functions were used. One takes a solution and swaps two random positions within it, and the other randomly changes one position in a solution to a value chosen uniformly at random between 1 and the number of vehicles available. For the crossover functions, we used three. One does crossover in the middle of the solution, the other does at a random point, and the other produces two new solutions by randomly swapping genes between two given solutions. This last operator is known as uniform crossover or UX crossover and is known for giving diversified populations.

In this first problem, we don't have any rigid constraints.

Looking at the objective functions, we created two that work in the exact same way considering the total time calculations, however they change in the way they define the route's order. So, the meta-heuristics focus in getting new ways of assigning establishments to routes, but then the objective functions orders the routes, extracted from the solution representation, and calculate their total time.

To set the order of each route two informed search methods were deployed, Greedy-Search and A*. The heuristic values were calculated considering the geocentric distance between two points on the planet. The geocentric distance is the shortest traveling distance between two points on the planet, and because we have the latitude and longitude of all the establishments, including the depot, we can get those values. To get seconds we just divided for an average velocity in km/h and multiplied by 3600 seconds; for the velocity we considered 70 km/h in order to get all the heuristic values lower than the correspondent real travel distances.

Notice that this is an optimistic way of assessing how close we are from a route to be ended because not only the best possible scenario is that after the following establishment, we end the route, but also the actual value used to assess the distance to the depot is a value better than the real one, and in fact the best possible time for the chosen velocity.

Regarding the inner-works of the objective functions that uses greedy-search starts by getting all the routes using that search method, and then calculates the travel time, the inspection time, and the waiting time for all the establishments, considering their order. For this to work it's necessary to keep track of the current hour at each step of the way so that we can check if when the brigades arrive the establishment is closed or open. The team should wait if the establishment is closed, however they can do inspections after closing time if they arrive while the establishment is open. During all of this all the total times for the routes are added together, resulting in the final score for the solution given by a meta-heuristic.

The objective function that uses the A* algorithm is more complex because instead of getting all the routes before calculating any times, like greedy-search, here the route is constructed while the objective function calculates the times. So, the same details on how to calculate times are the same as in the previous objective function, however here the A* algorithm chooses the next establishment than minimizes the total time to get there plus the heuristic value from that next establishment to the depot. In other words, for all the establishments that don't have been chosen to be part of the route, the algorithm uses the objective function to calculate the total travel time since the beginning of the route until that establishment, and then adds the proper heuristic value; then it chooses the establishment that has the value expressed by that sum.

Second scenario:

In the second scenario, everything was pretty much the same, regarding state representation, initial state, operators, inputs, and objective function, however, there were some changes. Here, the goal was to minimize the number of vehicles used, considering that a route stops in the last establishment inspected, as well as the fact that each team shouldn't work more than 8 working hours. Also, the inspection times for all establishments were considered 5 minutes. The problem formulation is the same, however now we have these rigid constraints, and the solution representation also includes the minimum number of vehicles necessary.

In the objective functions was added an 8-working hour limit (in seconds), as well as a change in the route adaptation, as well as in the inspection times.

Then for the vehicles, a loop was used where the number of vehicles increases and then the selected metaheuristic's algorithm runs and tries to find a solution. If not, it returns a very negative value, and the loop continues until an actual solution is found.

Although we solved this problem, the focus was on the first problem because it's the stepping stone for solving the second problem.

There are different parameters that can affect these three methods that need to be tested. For example, for the genetic algorithm population size, number of generations, crossover rate, mutation rate, selection method, crossover method and mutation methods are important parameters. For simulated annealing, initial temperature, cooling rate, number of iterations at each temperature, and neighbourhood size can be interesting parameters. In tabu search we can name tabu list size, number of iterations and neighbourhood size as an example.

IV. RESULTS

In this section we mostly focus on the results and findings of applying metaheuristics such as genetic algorithms, simulated annealing, tabu search, and hill climbing. In addition to that we will briefly present our user interface (UI) developed for this problem. In this section we will focus on comparing the score (total time) for each method to reach the optimum solution and the quality of the solution in different methods. In addition to that, we present the results of applying different methods in different size of establishments and show how the results of these metaheuristics and their internal informed search methods look like and how they improve the results.

For the small set of data we present the iteration results of 4 different metaheuristics under their own specific parameters in the annex. Figure 1 shows how the solutions for the tabu search (1000 iterations, tabu tenure of 10, greedy-search) evolve by each method. Different colours representing different runs and as we can see each run is different than the others, since it has the randomly generated initial solutions. The graphs for each metaheuristic for different sizes of establishments have been provided in the annex, and by comparing them we can conclude that these metaheuristics are more successful for the bigger sizes of establishments. In addition figure 2 shows the map related to one of the solutions of the same problem with same characteristics. All of these figures are provided by the specific user interface that we developed for this problem.

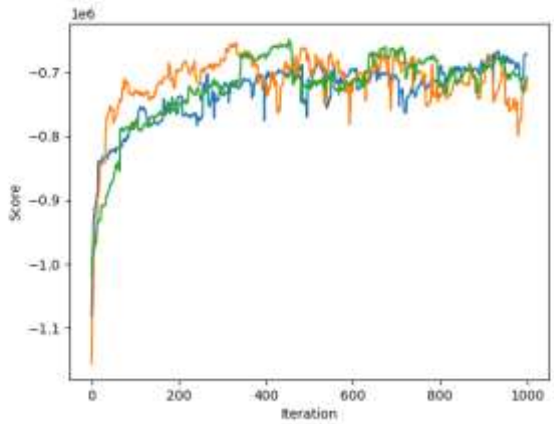

Figure 1. Evolution of tabu Search for the data with 100 establishment

Figure 2. Visualization of routes and establishments of the solution

In the annex we provided the results of the specific runs with the specific parameters and showed the evolution of each type of algorithm.

The next step is to compare these methods to each other. For a brief comparison of these methods we provided the figure 3 and 4. As we mentioned before these algorithms using different parameters to solve the problem and comparing to them while they can perform better by certain parameter changes may not be very accurate. However, in the following figure we provided an example comparison between these methods.

As we mentioned in the methodology, we used two different informed search methods, the greedy search and the A* algorithm, to optimize each route during its search by each metaheuristic. To have a good understanding of the behavior of each method under different conditions, instead on number of iterations, we set the time limit of 10 minutes for each run of each metaheuristic, and compare their best score to evaluate the performance of each metaheuristic.

Figure 3 and Figure 4 are presenting the comparison of the total travel time of our 4 metaheuristics for the medium (500 establishments) and large (1000 establishments) sizes under 2 different heuristics that used to optimize each route with the run time limit of 10 minutes.

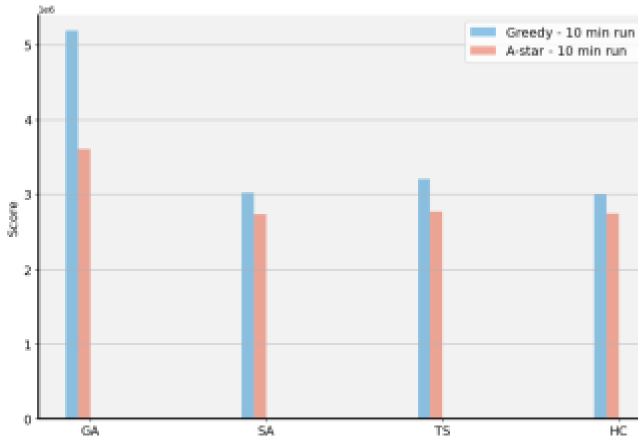

Figure 3. Comparing the 4 metaheuristics score and their heuristics in 10 minutes run for 500 establishments

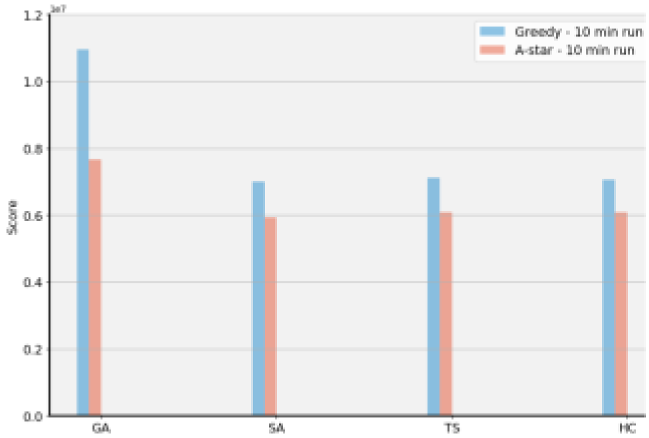

Figure 4. Comparing the 4 metaheuristics score and their heuristics in 10 minutes run for 1000 establishments

As we can see in the figure 3, genetic algorithm (with the score of 5192297 with greedy search and 3604196 with A-star) is the least effective among the 4 metaheuristics. In fact, the hill climbing (with the score of 2987812 for greedy search) is 42.62% more effective than the genetic algorithm in the that search, as well as 6.72% and 1% more effective than the tabu search (with the score of 3203251) and simulated annealing (with the score of 3018191), respectively, in the greedy search with 500 establishment. In addition, the A-star method has shown more promising result than the greedy search. For the medium size problem, this method has been 30.58% more effective than the greedy search for the genetic algorithm. These percentages for the simulated annealing, tabu search and hill climbing are 9.53%, 13.69%, and 8.36% respectively. Overall, for 500 establishments and for 10 minute runs, SA with the A* method has shown the best results among the other methods, while if we were only considering the greedy search hill climbing was the best method.

Now, although the other metaheuristics should give better results than HC, by changing the hyper-parameters of SA and TS we were able to find better results for the same time limit. In SA the cooling schedule is very important, but hard to find the sweet spot, the same with the tabu tenure for the TS. With changes, for example tabu tenure to higher values and cooling rate to values closer to one, we were able to improve the results, getting better ones than with HC.

These trends for the large size problem with the 1000 establishments is slightly different. While genetic algorithm is still showing the lowest performance among the 4 metaheuristics, simulated annealing has shown the most promising results among the other methods for both greedy search and A-star. This method has been 36.16%, 1.68%, and 1% more effective than genetic algorithm, tabu search, and hill climbing, respectively. These percentages for the A-star are 22.57%, 2.56%, and 2.67% respectively. Also, the same behavior exist between the A-star and greedy search and the latter one has been less effective. Table 1 shows the detailed scores related to the figure 3 and 4.

	Medium		Large	
	Greedy	A-star	Greedy	A-star
GA	5192297	3604196	10951731	7657468
SA	3018191	2730442	6990547	5928681
TS	3203251	2764716	7109883	6084350

HC	2987812	2738098	7058872	6091263

Table 1. Comparing the score of 4 metaheuristics and their internal heuristics in the 10 minutes run time for the medium and large sizes of establishments

To make all of these comparisons and see the effects of different changes we developed a user interface (UI) in python by the Tkinter library. It includes 4 metaheuristics of genetic algorithm, tabu search, simulated annealing, and hill climbing. It includes a dropdown for each of these algorithm that user can change the size of the data, number of iterations, population size, temperature, cooling rate, tabu tenure, as well as option to choose greedy search or A-star for each of these metaheuristics. Also, we provided the ability to show the map related to the optimal routes and location of each establishment for any size of the problem.

A very interesting result that we got, after checking all the calculations that the objective functions do, is that the A* algorithm minimizes waiting times by itself. This is a very fortunate result as this is also another sub-problem that was proposed. This makes sense because when choosing the best establishment to go next, the A* algorithm also checks if the teams need to wait for it to open, and a lot of times that waiting time can be more than 2 or 3 hours, and so the algorithm prioritizes establishments that are still open when the team arrives and leaves those establishments that open later for the end of the routes.

V. Conclusions

The goal of this project was to solve the vehicle routing problem for the data that is provided by ASAE. For this reason, we developed 4 different metaheuristics of genetic algorithm, simulated annealing, tabu search, and hill climbing. In addition, we used 2 heuristics of greedy search and A-star to find the optimum order of the routes when they are being randomly chosen by the metaheuristics. We developed and tested our models first on the very small and small size of the establishments and then we expanded it to the medium and large size.

For the comparison of these methods to each other several tests have been done and we presented the figure 3, figure 4, and table 1 related to the medium and large sizes of establishments, as they are more complicated and more meaningful in terms of results. As we mentioned before these algorithms using different parameters to solve the problem and comparing to them while they can perform better by certain parameter changes may not be very accurate. However, in the table 1 we provided the comparison between these methods for the 10 minuets run, as our goal was to evaluate the performance of each method in a fair condition. For instance as we can see in this table, the performance of metaheuristics with A-star is better than the greedy search. Also, for the medium and large size problem, genetic algorithm has been the least effective method. Simulated annealing shown the best performance among the other metaheuristics with the A-star integrated, and there was a slight difference in the trend of the effectiveness between the large and medium establishment sizes.

In conclusion, the goal of this study was to explore different metaheuristics, such as genetic algorithm, simulated annealing, tabu search, and hill climbing. For this reason we started to work on the ASAE data which in its core was a variation of the vehicle routing problem. We managed to solve 2 scenarios related to that dataset, and we presented the results of the first scenario in this article. In the 10 minuet runs of the medium and large size problem the simulated annealing, and hill climbing have shown more promising results in terms of scores.

In addition to that, we developed a user interface with different options of selecting the sizes of establishments, number of iterations, search type, temperature, cooling rate, and other parameters with the ability to show the graphs of the iterations and maps of the optimum routes.

References

[1] G. Clarke and J. W. Wright, “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points,” Operations Research, vol. 12, no. 3, pp. 568-581, 1964.

[2] M. Gendreau, G. Laporte, and F. Semet, “Vehicle Routing with Time windows; Part I – Route Construction and Location and Local Search Algorithms,” Transportation Science, vol. 31, no. 1, pp.49-64, 1997.

[3] C. Prins and J.-M. Proth, “A Polynomial Algorithm for the Vehicle Routing Problem with Time windows,” European Journal of Operational Research, vol. 88, no. 3, pp. 477-486, 1996.

[4] A. Corberan, G. Laporte, and M. A. Laguna, “Tabu Search for the Vehicle Routing Problem with Time Windows,” Annals of Operations Research, vol. 41, no. 4, pp. 469-488, 1993.

[5] A. Haghani and S. B. Malekly, “A Genetic Algorithm for the Vehicle Routing Problem,” European Journal of Operational Reseaarch, vol. 156, no. 2, pp. 443-455, 2004.

Annex

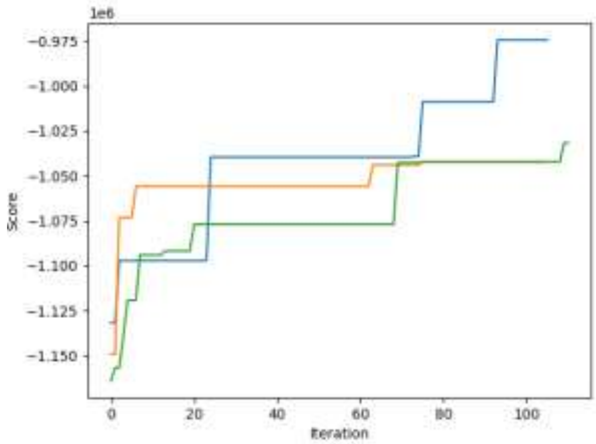

Figure 5. Evolution of genetic algorithm for the data with 100 establishments

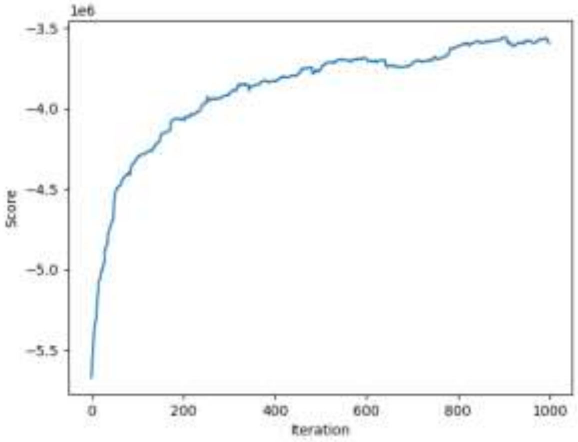

Figure 6. Evolution of hill climbing for the data with 100 establishment

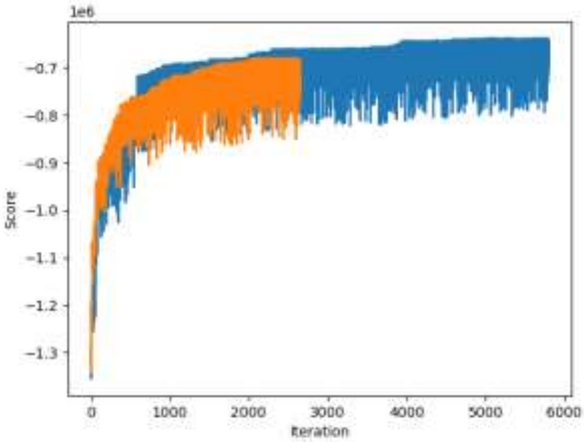

Figure 1. Evolution of simulated annealing for the data with 100 establishments

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.