electronics

MDPI

Article

Applying a Genetic Algorithm to a m-TSP: Case Study of a Decision Support System for Optimizing a Beverage Logistics Vehicles Routing Problem

David E. Gomes [1], Maria Inês D. Iglésias [1], Ana P. Proença [1], Tânia M. Lima [1,2] and Pedro D. Gaspar [1,2,*]

[1] Department of Electromechanical Engineering, University of Beira Interior, Rua Marquês de D'Ávila e Bolama, 6201-001 Covilhã, Portugal; david.emanuel.gomes@ubi.pt (D.E.G.); maria.ines.iglesias@ubi.pt (M.I.D.I.); ana.proenca@ubi.pt (A.P.P.); tmlima@ubi.pt (T.M.L.)

[2] C-MAST-Center for Mechanical and Aerospace Science and Technologies, Rua Marquês de D'Ávila e Bolama, 6201-001 Covilhã, Portugal

* Correspondence: dinis@ubi.pt

Abstract: Route optimization has become an increasing problem in the transportation and logistics sector within the development of smart cities. This article aims to demonstrate the implementation of a genetic algorithm adapted to a Vehicle Route Problem (VRP) in a company based in the city of Covilhã (Portugal). Basing the entire approach to this problem on the characteristic assumptions of the Multiple Traveling Salesman Problem (m-TSP) approach, an optimization of the daily routes for the workers assigned to distribution, divided into three zones: North, South and Central, was performed. A critical approach to the returned routes based on the adaptation to the geography of the Zones was performed. From a comparison with the data provided by the company, it is predicted by the application of a genetic algorithm to the m-TSP, that there will be a reduction of 618 km per week of the total distance traveled. This result has a huge impact in several forms: clients are visited in time, promoting provider-client relations; reduction of the fixed costs with fuel; promotion of environmental sustainability by the reduction of logistic routes. All these improvements and optimizations can be thought of as contributions to foster smart cities.

Keywords: genetic algorithms; m-TSP; VRP; decision support system; case study

check for updates

Citation: Gomes, D.E.; Iglésias, M.I.D.; Proença, A.P.; Lima, T.M.; Gaspar, P.D. Applying a Genetic Algorithm to a m-TSP: Case Study of a Decision Support System for Optimizing a Beverage Logistics Vehicles Routing Problem. *Electronics* **2021**, *10*, 2298. https://doi.org/10.3390/electronics10182298

Academic Editor: Chilukuri K. Mohan

Received: 2 August 2021
Accepted: 16 September 2021
Published: 18 September 2021

1. Introduction

The concern for the development of smart cities has been growing exponentially in recent years. The smart city concept may vary, but it is mostly focused on the environment, and its preservation, and not only on the technology, as described by Aletà et al. [1]. It can also be based on three main concepts: economic efficiency, social equity, and, the most relevant, environmental quality [2]. The increasing need to make cities more sustainable and designed for people has led to the search for alternatives that allow for a better rationalization and optimization of resources and means. These developments in cities have not only had an effect on their design and organization but also play a very important role in promoting development, for example, in industry and in the way communities perceive their activities. Smart Mobility, as some authors refer to it, may be perceived through the optics of the transportation of people or goods. The growth of the cities and the importance of mobility within the urban area has led to research on ways to minimize traffic, its impact on the environment, and increase the efficiency of travel. The industry's evolution has also increased the concern to improve the profitability of resources, in order to reduce costs and increase efficiency, keeping up with the main concerns of the evolution of smart cities. Despite being a constantly changing problem, due to instabilities of a different kind, business success can often be dictated by the applied methodology. With the growing need for the enhancement of the efficiency associated with the transportation of goods, logistics has become a very important economic activity.

Logistics can be defined as the art and science of acquiring, producing, and distributing all products considering the usefulness of time and place, that is, the timing of delivery at the right place and not only in the shortest time possible but also at the right time [3]. Companies specialized in logistical processes can focus on different areas, such as transporting goods and some additional services, such as warehouse management or stock control. Although transportation is the main focus of this sector, it is also very important to control the stock and manage the storage of products. Thus, decision support systems play a very important role in this management.

It is possible to observe an increasing interconnection between linear programming and problems in goods transportation, most of the time related to the question: What would be the best shipping method that allows a minimized cost of shipping of n units to m destinations and a maximization of the profit? Several methods are available. Vehicle Route Problem (VRP) is based on the assignment of distribution routes for a set of clients scattered on the map at the lowest possible cost [4]. The Traveling Salesman Problem (TSP) seeks to define routes that cover all points of sale, passing through each one only once, returning to the point of origin. The Multiple Traveling Salesman Problem (m-TSP) is a variant of the TSP. The difference is the existence of m sellers for n destinations. Since the beginning of the TSP study, many approaches and methods have been developed [5]: classic methods, mainly based on linear programming [6] and branch-and-bound [7], and artificial intelligence methods, such as Tabu Search, Genetic Algorithms or Artificial Neural Networks. Artificial intelligence methods based on neural networks represent a good approach to the problem in this study, as they are not less effective than others. Their characteristics allow the needed calculations to occur, even when conditions change over time [8]. If a complex problem of the shortest path is not approached using a neural network or genetic algorithms, the complexity of the problem grows exponentially.

Sakharov et al. [9] consider that a large number of empirical problems can be formulated as problems of finding the shortest path and can be solved using network models, once again, recurring to artificial intelligence methods. A universal computer model of the Floyd–Warshall algorithm was used to approach a shortest path problem. All steps that the algorithm takes to return a solution were described, providing a tree of points. However, the Floyd–Warshall algorithm is not the best approach to this problem, specifically because of the characteristics of the user data. This type of algorithm works better in cases of weighted, oriented or not, graphs, opposed to this problem in which the simplification of the roads and establishments to nodes and edges could return a non-applicable solution.

Azis et al. [10] conducted a study whose main objective was to compare the conventional methods, the Floyd–Warshall Algorithm and the Heuristic Method, the so-called Greedy Algorithm. Through the data analysis and interpretation, it was possible to conclude that the conventional method, despite its "temporal" limitations in defining the shortest route, compared to the greedy algorithm, is more accurate. Although taking longer, it considers all stopping points and possible routes, while the greedy algorithm exclusively recognizes the one with the lowest weight in each interaction, so that the time frame is smaller and consequently always leading to the optimal final result. Hybrid methods can also be applied. Jiang et al. [11] proposed a hybrid method, merging the Ant Colony Algorithms (ACO) and the Partheno Genetic Algorithms (PGA), similar to the Genetic Algorithms, but lacking the crossover operators. The ACO are known for being suitable for NP-hard problems, such as m-TSP, and the PGA are a great approach to this kind of problem [12]. The complexity level of the current case study does not require such a method.

Fujdiak et al. [13] applied genetic algorithms in a real case study of waste collecting in smart cities context as an optimization method. A crossover and mutation of the Floyd–Warshall algorithm was performed. This algorithm was chosen since a metric system was used and the negative values of edges are not used. The crossover allowed to develop changes in the new population and consequently find a better solution. The mutations allowed to exchange nodes. As in our case study, asymmetric matrices considering the existence of one-way streets were resorted to. Despite being similar case studies, this algorithm does not apply to the current problem since despite generating the solution for the shortest distance between all pairs of vertices, it was applied to a problem with a smaller sample size.

The m-TSP problem is an approach that softens the VRP problem when it comes to capacity but is nevertheless able to address other needs, such as single or multiple warehouse constraints, numbers of vendors, work schedule constraints or limits on locations per vendor, and even time window constraints. These types of problems are not only useful for managing vendor routes, but also for other purposes where more informed decision-making is useful. Solving this problem can be used, for example, to distribute work among several workers with restricted functions, by managing their schedules, by planning missions or production, or by distributing tasks efficiently [14]. We seem to be moving towards times when, for example, unmanned aerial vehicles, known as drones, will play a fundamental role in short distribution routes for light load or for transportation within industrial complexes. The m-TSP problem using aerial vehicles is addressed in [15]. New approaches will be required to deal effectively with these kind of problems.

It is also usual to observe that some of these routing problems are dynamic, in which the cities or places to visit are known over time and not in advance as in conventional vehicle routing problems. Gam [16] addressed two different heuristics, proposing in one the assignment of the vehicle with the shortest distance traveled to the client, and in the other, the vehicle that is closest to the new location that arose in the problem. These heuristics also pay attention to the number of locations to visit so that all salesmen visit roughly the same number.

There are numerous applications for dynamic m-TSP problems, such as pickup and delivery services where several time windows and vehicle waiting times are present [17]. This is already an evolution from one of the pioneering studies [18] in which it was not taken into account that deliveries can all influence each other directly by changing all delivery times.

Based on methods already developed, such as VRP, TSP, m-TSP, combined with Genetic Algorithms (GA), it was possible to obtain the best route to follow in order to minimize costs [19]. A GA is a meta-heuristic method that does not work with single optimal solutions, but with populations of solutions [20].

2. Methodology

The m-TSP is a well-known extension of the traveling salesman problem, in which n cities are assigned to m sellers, where each knot can only be visited once by only one seller, thus preventing route overlapping. It is necessary to pay attention to the limitations to which the problem is subject, such as, for example, damaged roads or tolls, timetables and temporal windows, precedence in deliveries, etc. In this case study, the biggest constraint is the minimum number of places to be visited [15].

The mathematical model of m-TSP can be represented through the following formulation:

$$Minimization : F = (f_1, f_2) \tag{1}$$

$$f_1 = \sum_{k=1}^{m} \sum_{i=0}^{n} \sum_{j=1}^{n} c_{ij} x_{ijk} \tag{2}$$

$$f_2 = \underset{0 \le x \le 1}{max} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk} - \min_{1 \le k \le m} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk} \tag{3}$$

where,

$$x_{ij} = \begin{cases} 1, \; salesmene\ K\ goes\ from\ city\ i\ to\ city\ j \\ 0, \; otherwise \end{cases} \tag{4}$$

$$s.t.\begin{cases} \sum_{k=1}^{m}\sum_{i=0}^{n} x_{ijk} = 1;\ \forall j = 1\ldots\ldots,n \\ \sum_{k=1}^{m}\sum_{j=0}^{n} x_{ijk} = 1;\ \forall j = 1\ldots\ldots,n \\ \qquad\quad \sum_{k=1}^{m}\sum_{i=1}^{n}\ x_{i0k} = m \\ \qquad\quad \sum_{k=1}^{m}\sum_{j=1}^{n}\ x_{i0k} = m \\ \sum_{i\in S}\sum_{j\notin S} x_{ijk} \geq 1;\ \forall k \in V, \forall S \subseteq C \end{cases} \tag{5}$$

The information above is represented by the set of n cities and m sellers and is expressed by

$$C = (i),\ i = 0,\ 1,\ 2,\ \ldots,\ n;\ V = (k),\ k = 1,\ 2,\ \ldots,\ m. \tag{6}$$

where c_{ij} defines the distance between cities i and j, and x_{ijk} for the salesman k from cities i to j, considering that each city is visited only once [21].

Equations (2) and (3) represent, respectively, the two objective functions that are intended to be minimized: the total distance from the seller and the difference between the longest and shortest route. Equation (5) concerns the restriction that all salesmen depart from the same city. So, except at the point of origin, there is only one salesman at each point.

Through a previous analysis of the various types of algorithms and methods available, it is concluded that, for problems of this type, the best approach is to use Genetic Algorithms. Genetic Algorithms are meta-heuristic methods that are based on scientific phenomena, such as the Theory of the Evolution of Species, in which each one has a variety of individuals, with some characteristics being maintained in generations and others recombined. This type of algorithm does not work with single optimal solutions, but with a population of solutions, which are improved over time, as only individuals with the best characteristics survive and predominate [20]. The Genetic Algorithm developed by Kirk [22] finds a viable solution close to the optimal solution for m-TSP, one of the TSP variations, which looks for the shortest route passing through all points of a network. This algorithm is based on two important premises: (a) it is a closed m-TSP, that is, the starting point is also the return point, in this case, the company's headquarters; (b) Except for the starting point, each location is only visited once. Before the iterative process, it is necessary to define the characteristic inputs of this algorithm:

- Generate XY, an $N \times 2$ matrix, where N is the number of places, in which it was chosen to generate a "random" matrix of $N \times N$. Thus, the results of the solution will be later portrayed on a map, instead of having to insert a matrix with the coordinates of each place.
- Insert DMAT, which is the $N \times N$ matrix of distances or costs. Distances between locations were looked up on Google Maps and imported in Microsoft Excel matrices.
- NSALESMEN represents the number of vendors visiting the locations, which differ in this problem between one salesman on the large experimental routes, two salesmen on the routes in the North and South zones and four in the Central area.
- MINTOUR is the minimum number of places to be visited by each salesman, in this case 10 locations on all routes except Central Route which was defined as 15 as there are more places. It was through this input that the number of places to visit per day, for each salesman, was kept balanced.
- POPSIZE was set to 80, which represents the number of solutions with which the algorithm starts iterations and mutations in search of a better solution. This parameter must be, by obligation of the algorithm, divisible by 8.
- NUMITER is the maximum number of iterations that the algorithm will do, in search of the best solution, in which case it was defined as 5000.
- The remaining inputs, were kept unchanged, remaining by default.

It was also necessary to make an adjustment to the code of the algorithm, in which a "disp(rte)" was inserted, a function that allows the viewing of the order of the points of the returned routes. Contrary to the original code, where only the visual format of the connections and the total sum of kilometers of the solution's routes existed, this not being viable because a random matrix was used instead of a matrix of geographic coordinates.

The algorithm flowchart shown in Figure 1 helps to understand how the algorithm works, in which the necessary inputs are initially inserted. After defining the inputs, the algorithm will choose the best solution from the initial population of eight. With this more restricted group of solutions, it will start mutating some parts of the solution through Flip, Swap, Slide and Modify Breaks. With the best solution found and with the adaptation included in the algorithm it is possible to visualize the direction in which to go.

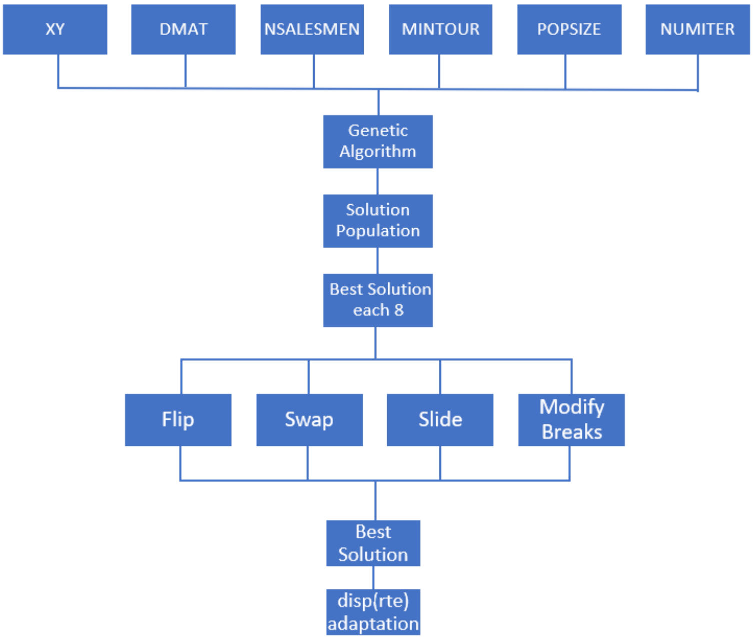


Figure 1. Algorithm flowchart.

The algorithm was reproduced on an Asus VivoBook computer with Intel(R) Core (TM) i7-8565U CPU @ 1.80 GHz 1.99 GHz, 16.0 GB RAM, Windows 10 Home 64-bit OS. The response time of the various routes was always between 2 s and 10 s. Thus, the computation time is acceptable for real case applications.

3. Case Study

The problem under analysis is the creation of a decision support system (DSS) that allows for the optimization of distribution routes of a small food and beverage distribution company, headquartered in Covilhã, Portugal. The main goal of the DSS is to optimize routes, for the five days of the week, which minimizes travel costs and the distances traveled by workers, considering the approximately 270 establishments provided by the company. These locations can be seen in Figure 2.

Currently, the routes are covered by four workers: one that only carries out sales, another which carries out sales and only distributes the respective orders, one that carries out sales only one day of the week and distributes the rest, and the last that only distributes. However, these task assignments are not reflected on an optimization of the time and distance covered by workers and, therefore, the applied methodology undergoes a reorganization of the distribution of the workers in question. Thus, two workers are allocated to exclusively carry out the sales and the remaining two workers to exclusively carry out the delivery of orders. Following this approach, it was intended to optimize eight routes, one for each seller/distributor, for four days a week, with Friday being destined for locations that, for some reason external to the company, could not be visited.

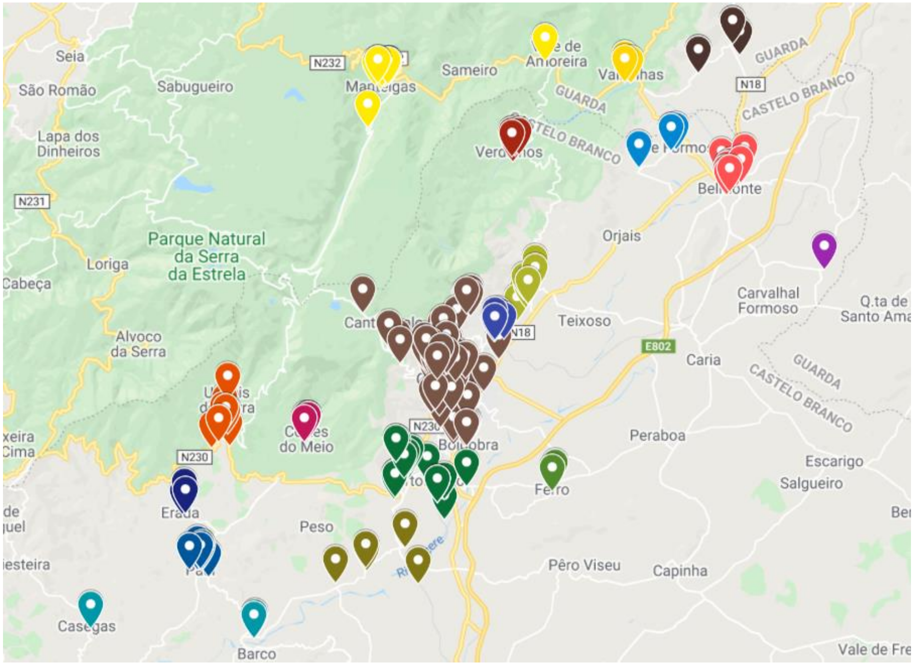


Figure 2. Locations in study.

Currently, there are many factors external to the company that may lead to the need to revisit some of the locations at the end of the week. This way, two routes to the North and South Zones and four routes to the Central Zone will be considered. It is assumed that the distributor can travel the same route that the seller traveled and that, therefore, there will not be a need to create personalized routes. These routes must allow workers to cover the minimum number of total kilometers, but also consider the characteristics of the roads, particularly the orography since the city is located in a mountain region with crooked roads with high slopes. This condition may lead to high fuel consumption and consequently to higher fuel costs besides the significant emission of combustion gases that must be reduced to promote environmental sustainability.

4. Model Formulation

Initially, in order to reduce the complexity of data processing provided by the company, the approximately 270 establishments were divided into three main zones: North Zone, South Zone and Central Zone. A simplification was sought, similarly to the division into three main zones, within the North and South zones because, although there is a wide dispersion of establishments in the zones, it is possible to define some clusters within the locations, since the distances between establishments, compared to the distances between locations, are negligible. Using the Google Maps distance measurement tool, the exact distances between each point were obtained, thus allowing the construction of distance matrices for the respective zones. These matrices of dimension $N \times N$, with N being the number of key points associated with each of the three zones in question, were adapted, resorting, when necessary, to the maximization of distances that would represent conditioned sections, either by the distance they represent or by the conditions of the road necessary to go through. These maximizations were the result of a study of the pattern through the exhaustive iterative running of the algorithm, which was observed in the various routes. Mostly, the algorithm returned a solution that presented a smaller route and a larger one, in the case of the North and South zones. It is considered that these matrices are symmetric, to ease problem-solving. For example, the distance from point A to B is always equal to the distance from point B to A, therefore not considering any types of traffic constraints that may cause significant differences between A to B and vice-versa.

4.1. North Zone

The zone defined as the North Zone has 26 key points. It is possible to observe in Figure 3 and Table 1 the many key points associated with the Clusters in question.

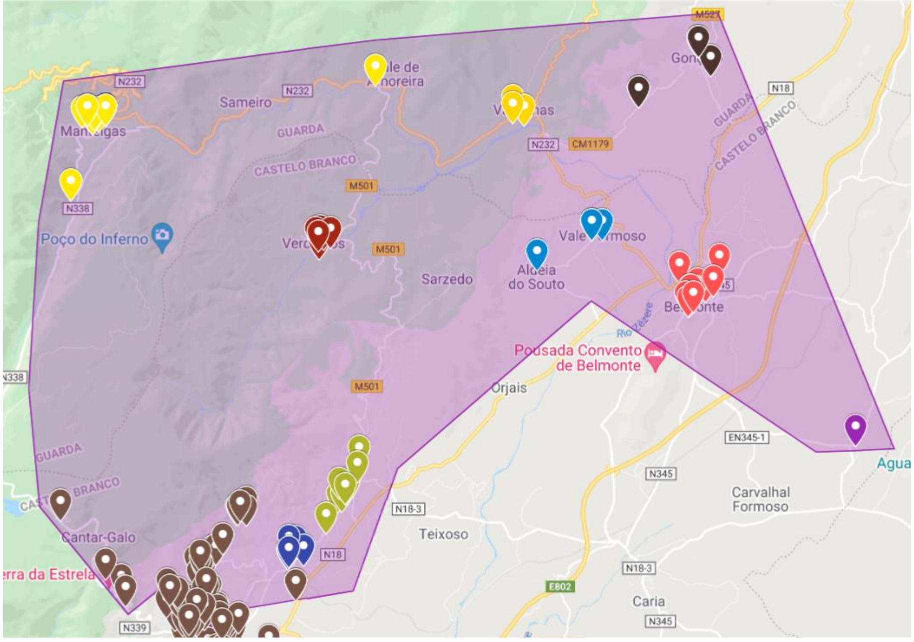


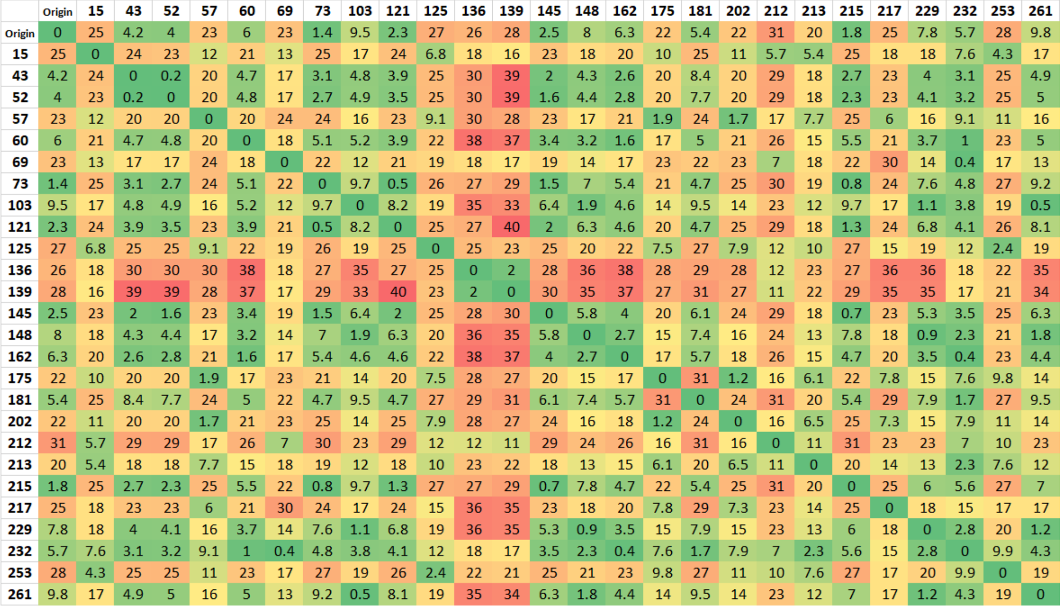
Figure 3. North zone.

Table 1. Clusters of North zone.

Clusters [(1)]	Key Points
Inguias (1)	217
Belmonte (12)	202, 57, 175
Gonçalo (4)	125
Cruz da Pedra (1)	253
Vale Formoso (3)	213
Valhelhas (3)	15
Verdelhos (6)	69
Vale da Amoreira (1)	212
Manteigas (11)	136, 139
Penhas da Saúde (4)	230, 169, 180
Penedos Altos (16)	121, 145, 215, 73
Parque Industrial Canhoso (1)	60
Canhoso (7)	162
Teixoso (10)	148, 229, 261
Gibraltar (1)	103
Vila do Carvalho (5)	52

[(1)] Number of establishments associated with the cluster.

Initially, using the 26 × 26 matrix shown in Figure 4, and considering a single route, with only one salesman, it was evaluated how the algorithm would proceed in the connection between the points. Due to the orography of the area, the first matrix, when used for two salesmen, did not return a viable solution. The solution considered splitting the 26 × 26 distance matrix in two: an outer Route, including the points far away with crooked roads with high slopes, and a smaller Route, encompassing the points closer to the city.



	Origin	15	43	52	57	60	69	73	103	121	125	136	139	145	148	162	175	181	202	212	213	215	217	229	232	253	261
Origin	0	25	4.2	4	23	6	23	1.4	9.5	2.3	27	26	28	2.5	8	6.3	22	5.4	22	31	20	1.8	25	7.8	5.7	28	9.8
15	25	0	24	23	12	21	13	25	17	24	6.8	18	16	23	18	20	10	25	11	5.7	5.4	25	18	18	7.6	4.3	17
43	4.2	24	0	0.2	20	4.7	17	3.1	4.8	3.9	25	30	39	2	4.3	2.6	20	8.4	20	29	18	2.7	23	4	3.1	25	4.9
52	4	23	0.2	0	20	4.8	17	2.7	4.9	3.5	25	30	39	1.6	4.4	2.8	20	7.7	20	29	18	2.3	23	4.1	3.2	25	5
57	23	12	20	20	0	20	24	24	16	23	9.1	30	28	23	17	21	1.9	24	1.7	17	7.7	25	6	16	9.1	11	16
60	6	21	4.7	4.8	20	0	18	5.1	5.2	3.9	22	38	37	3.4	3.2	1.6	17	5	21	26	15	5.5	21	3.7	1	23	5
69	23	13	17	17	24	18	0	22	12	21	19	18	17	19	14	17	23	22	23	7	18	22	30	14	0.4	17	13
73	1.4	25	3.1	2.7	24	5.1	22	0	9.7	0.5	26	27	29	1.5	7	5.4	21	4.7	25	30	19	0.8	24	7.6	4.8	27	9.2
103	9.5	17	4.8	4.9	16	5.2	12	9.7	0	8.2	19	35	33	6.4	1.9	4.6	14	9.5	14	23	12	9.7	17	1.1	3.8	19	0.5
121	2.3	24	3.9	3.5	23	3.9	21	0.5	8.2	0	25	27	40	2	6.3	4.6	20	4.7	25	29	18	1.3	24	6.8	4.1	26	8.1
125	27	6.8	25	25	9.1	22	19	26	19	25	0	25	23	25	20	22	7.5	27	7.9	12	10	27	15	19	12	2.4	19
136	26	18	30	30	30	38	18	27	35	27	25	0	2	28	36	38	28	29	28	12	23	27	36	36	18	22	35
139	28	16	39	39	28	37	17	29	33	40	23	2	0	30	35	37	27	31	27	11	22	29	35	35	17	21	34
145	2.5	23	2	1.6	23	3.4	19	1.5	6.4	2	25	28	30	0	5.8	4	20	6.1	24	29	18	0.7	23	5.3	3.5	25	6.3
148	8	18	4.3	4.4	17	3.2	14	7	1.9	6.3	20	36	35	5.8	0	2.7	15	7.4	16	24	13	7.8	18	0.9	2.3	21	1.8
162	6.3	20	2.6	2.8	21	1.6	17	5.4	4.6	4.6	22	38	37	4	2.7	0	17	5.7	18	26	15	4.7	20	3.5	0.4	23	4.4
175	22	10	20	20	1.9	17	23	21	14	20	7.5	28	27	20	15	17	0	31	1.2	16	6.1	22	7.8	15	7.6	9.8	14
181	5.4	25	8.4	7.7	24	5	22	4.7	9.5	4.7	27	29	31	6.1	7.4	5.7	31	0	24	31	20	5.4	29	7.9	1.7	27	9.5
202	22	11	20	20	1.7	21	23	25	14	25	7.9	28	27	24	16	18	1.2	24	0	16	6.5	25	7.3	15	7.9	11	14
212	31	5.7	29	29	17	26	7	30	23	29	12	12	11	29	24	26	16	31	16	0	11	31	23	23	7	10	23
213	20	5.4	18	18	7.7	15	18	19	12	18	10	23	22	18	13	15	6.1	20	6.5	11	0	20	14	13	2.3	7.6	12
215	1.8	25	2.7	2.3	25	5.5	22	0.8	9.7	1.3	27	27	29	0.7	7.8	4.7	22	5.4	25	31	20	0	25	6	5.6	27	7
217	25	18	23	23	6	21	30	24	17	24	15	36	35	23	18	20	7.8	29	7.3	23	14	25	0	18	15	17	17
229	7.8	18	4	4.1	16	3.7	14	7.6	1.1	6.8	19	36	35	5.3	0.9	3.5	15	7.9	15	23	13	6	18	0	2.8	20	1.2
232	5.7	7.6	3.1	3.2	9.1	1	0.4	4.8	3.8	4.1	12	18	17	3.5	2.3	0.4	7.6	1.7	7.9	7	2.3	5.6	15	2.8	0	9.9	4.3
253	28	4.3	25	25	11	23	17	27	19	26	2.4	22	21	25	21	23	9.8	27	11	10	7.6	27	17	20	9.9	0	19
261	9.8	17	4.9	5	16	5	13	9.2	0.5	8.1	19	35	34	6.3	1.8	4.4	14	9.5	14	23	12	7	17	1.2	4.3	19	0

Figure 4. Distance matrix of North zone.

Also, for this main zone, due to the different responses returned by the algorithm, it was necessary to study the possibility of conditioning, through the maximization of values, a specific location of key points and decide to which route it belongs. Thus, through a preliminary analysis, the matrix of this zone was divided into two, with only the points of the routes previously returned, and divided into two approaches: Verdelhos (red dots in Figure 3, corresponding to key point 69) in the inside and the outside route. By applying the algorithm to only one salesman for both approaches, adding the distances of both routes, the solutions were 152.95 km and 139.10 km, for key point 69, inside or outside, respectively. With these data, it was possible to conclude that there was a need to maximize the distances from the cluster associated to point 69 to points further south, thus enforcing it to be inserted in the route of the localities on the outside.

4.2. South Zone

The South Zone also has 26 key points. These can be seen in Figure 5 and Table 2.

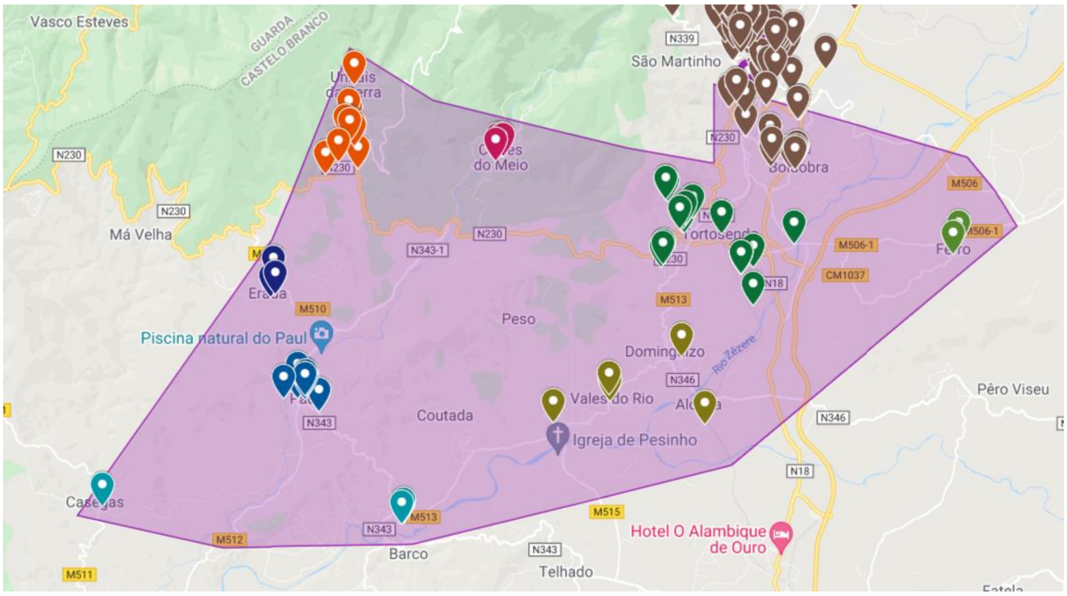


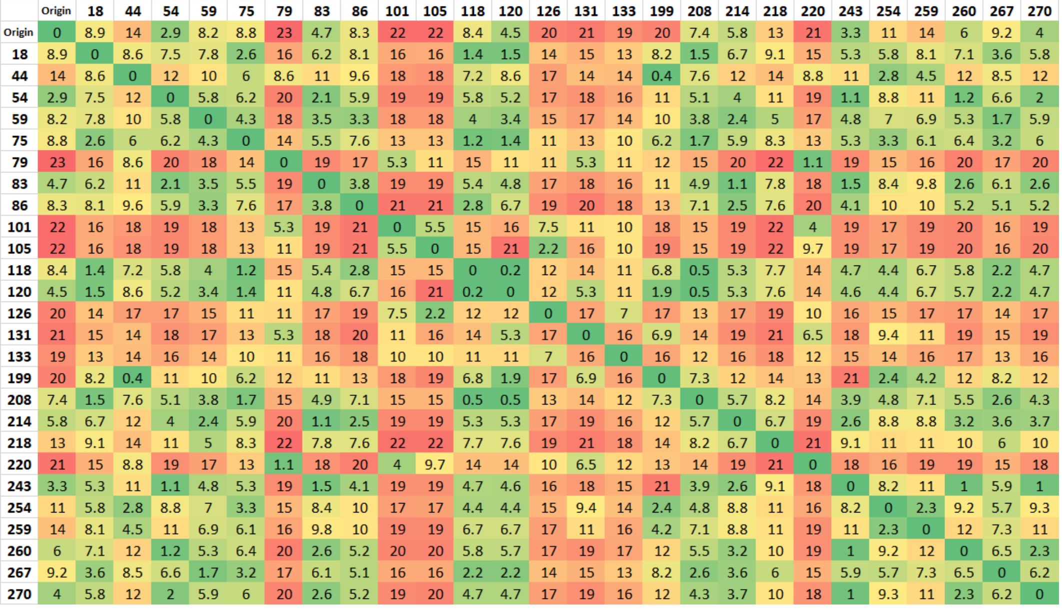
Figure 5. South zone.

Table 2. Clusters South zone.

Clusters [(1)]	Key Points
Covilhã Sul (4)	260, 54
Ferro (2)	218
Alcaria (1)	259
Dominguizo (1)	254
Vales do Rio (2)	199
Peso (1)	120
Barco (3)	131
Paul (12)	79, 220
Casegas (1)	44
Erada (3)	101
Unhais da Serra (15)	105, 126
Cortes do Meio (5)	133
Bairro do Cabeço (4)	75
Tortosendo (9)	86, 118, 208
Casal da Serra (3)	18
Parque Industrial do Tortosendo (3)	267
Ponte Pedrinha (1)	59
Boidobra (11)	214, 83
Refúgio (4)	243, 270

[(1)] Number of establishments associated with the cluster.

Similarly, to the North Zone, the behavior of the algorithm was first studied through a single route, including all points, using the 26 × 26 matrix (see Figure 6). Then, after maximizing some distances, and conditioning some paths, a solution was obtained that met the first one attained. A particular case in this zone is Casegas, key point 44. Due to the restriction of a single passage through each point associated with the m-TSP, it was necessary to condition the Paul-Casegas zone (79-220-44), for the algorithm to triangulate the surrounding locations. Like the previous case, it was necessary to carry out a previous analysis of the situation at point 44 alone, concluding that it was necessary to return to points 79 and 220, and then follow the rest of the route.



	Origin	18	44	54	59	75	79	83	86	101	105	118	120	126	131	133	199	208	214	218	220	243	254	259	260	267	270
Origin	0	8.9	14	2.9	8.2	8.8	23	4.7	8.3	22	22	8.4	4.5	20	21	19	20	7.4	5.8	13	21	3.3	11	14	6	9.2	4
18	8.9	0	8.6	7.5	7.8	2.6	16	6.2	8.1	16	16	1.4	1.5	14	15	13	8.2	1.5	6.7	9.1	15	5.3	5.8	8.1	7.1	3.6	5.8
44	14	8.6	0	12	10	6	8.6	11	9.6	18	18	7.2	8.6	17	14	14	0.4	7.6	12	14	8.8	11	2.8	4.5	12	8.5	12
54	2.9	7.5	12	0	5.8	6.2	20	2.1	5.9	19	19	5.8	5.2	17	18	16	11	5.1	4	11	19	1.1	8.8	11	1.2	6.6	2
59	8.2	7.8	10	5.8	0	4.3	18	3.5	3.3	18	18	4	3.4	15	17	14	10	3.8	2.4	5	17	4.8	7	6.9	5.3	1.7	5.9
75	8.8	2.6	6	6.2	4.3	0	14	5.5	7.6	13	13	1.2	1.4	11	13	10	6.2	1.7	5.9	8.3	13	5.3	3.3	6.1	6.4	3.2	6
79	23	16	8.6	20	18	14	0	19	17	5.3	11	15	11	11	5.3	11	12	15	20	22	1.1	19	15	16	20	17	20
83	4.7	6.2	11	2.1	3.5	5.5	19	0	3.8	19	19	5.4	4.8	17	18	16	11	4.9	1.1	7.8	18	1.5	8.4	9.8	2.6	6.1	2.6
86	8.3	8.1	9.6	5.9	3.3	7.6	17	3.8	0	21	21	2.8	6.7	19	20	18	13	7.1	2.5	7.6	20	4.1	10	10	5.2	5.1	5.2
101	22	16	18	19	18	13	5.3	19	21	0	5.5	15	16	7.5	11	10	18	15	19	22	4	19	17	19	20	16	19
105	22	16	18	19	18	13	11	19	21	5.5	0	15	21	2.2	16	10	19	15	19	22	9.7	19	17	19	20	16	20
118	8.4	1.4	7.2	5.8	4	1.2	15	5.4	2.8	15	15	0	0.2	12	14	11	6.8	0.5	5.3	7.7	14	4.7	4.4	6.7	5.8	2.2	4.7
120	4.5	1.5	8.6	5.2	3.4	1.4	11	4.8	6.7	16	21	0.2	0	12	5.3	11	1.9	0.5	5.3	7.6	14	4.6	4.4	6.7	5.7	2.2	4.7
126	20	14	17	17	15	11	11	17	19	7.5	2.2	12	12	0	17	7	17	13	17	19	10	16	15	17	17	14	17
131	21	15	14	18	17	13	5.3	18	20	11	16	14	5.3	17	0	16	6.9	14	19	21	6.5	18	9.4	11	19	15	19
133	19	13	14	16	14	10	11	16	18	10	10	11	11	7	16	0	16	12	16	18	12	15	14	16	17	13	16
199	20	8.2	0.4	11	10	6.2	12	11	13	18	19	6.8	1.9	17	6.9	16	0	7.3	12	14	13	21	2.4	4.2	12	8.2	12
208	7.4	1.5	7.6	5.1	3.8	1.7	15	4.9	7.1	15	15	0.5	0.5	13	14	12	7.3	0	5.7	8.2	14	3.9	4.8	7.1	5.5	2.6	4.3
214	5.8	6.7	12	4	2.4	5.9	20	1.1	2.5	19	19	5.3	5.3	17	19	16	12	5.7	0	6.7	19	2.6	8.8	8.8	3.2	3.6	3.7
218	13	9.1	14	11	5	8.3	22	7.8	7.6	22	22	7.7	7.6	19	21	18	14	8.2	6.7	0	21	9.1	11	11	10	6	10
220	21	15	8.8	19	17	13	1.1	18	20	4	9.7	14	14	10	6.5	12	13	14	19	21	0	18	16	19	19	15	18
243	3.3	5.3	11	1.1	4.8	5.3	19	1.5	4.1	19	19	4.7	4.6	16	18	15	21	3.9	2.6	9.1	18	0	8.2	11	1	5.9	1
254	11	5.8	2.8	8.8	7	3.3	15	8.4	10	17	17	4.4	4.4	15	9.4	14	2.4	4.8	8.8	11	16	8.2	0	2.3	9.2	5.7	9.3
259	14	8.1	4.5	11	6.9	6.1	16	9.8	10	19	19	6.7	6.7	17	11	16	4.2	7.1	8.8	11	19	11	2.3	0	12	7.3	11
260	6	7.1	12	1.2	5.3	6.4	20	2.6	5.2	20	20	5.8	5.7	17	19	17	12	5.5	3.2	10	19	1	9.2	12	0	6.5	2.3
267	9.2	3.6	8.5	6.6	1.7	3.2	17	6.1	5.1	16	16	2.2	2.2	14	15	13	8.2	2.6	3.6	6	15	5.9	5.7	7.3	6.5	0	6.2
270	4	5.8	12	2	5.9	6	20	2.6	5.2	19	20	4.7	4.7	17	19	16	12	4.3	3.7	10	18	1	9.3	11	2.3	6.2	0

Figure 6. Distance matrix of South zone.

4.3. Central Zone

The zone defined as the Central Zone represents around 100 establishments. However, as some are in the same buildings or at a distance of less than 10 m (distance defined as the limit for considering clusters in this zone), some points were grouped together, thus reducing to 72 key points. These points can be seen in Figure 7.

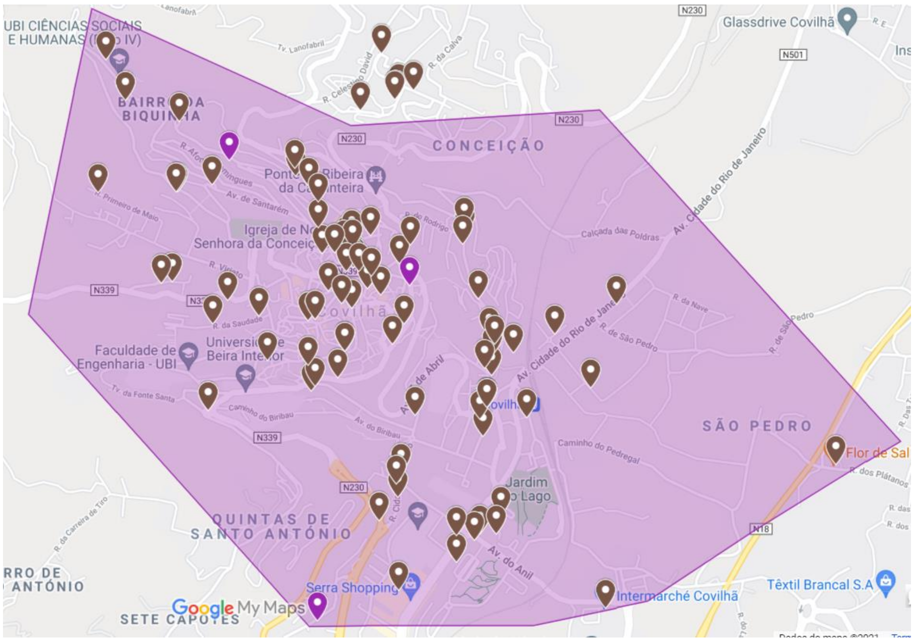


Figure 7. Central zone.

As the distances between points are very small, there is no need to obtain an optimal solution as all points are close and relatively close to the Headquarters, within a radius of about 5 km. Thus, it was not necessary to carry out an equally detailed analysis of the 72 × 72 matrix, as it happened in the other zones. It was only necessary to run the algorithm for four salesmen, representing the two days of the week associated with the two workers. The 72 × 72 matrix can be seen in Figure 8.

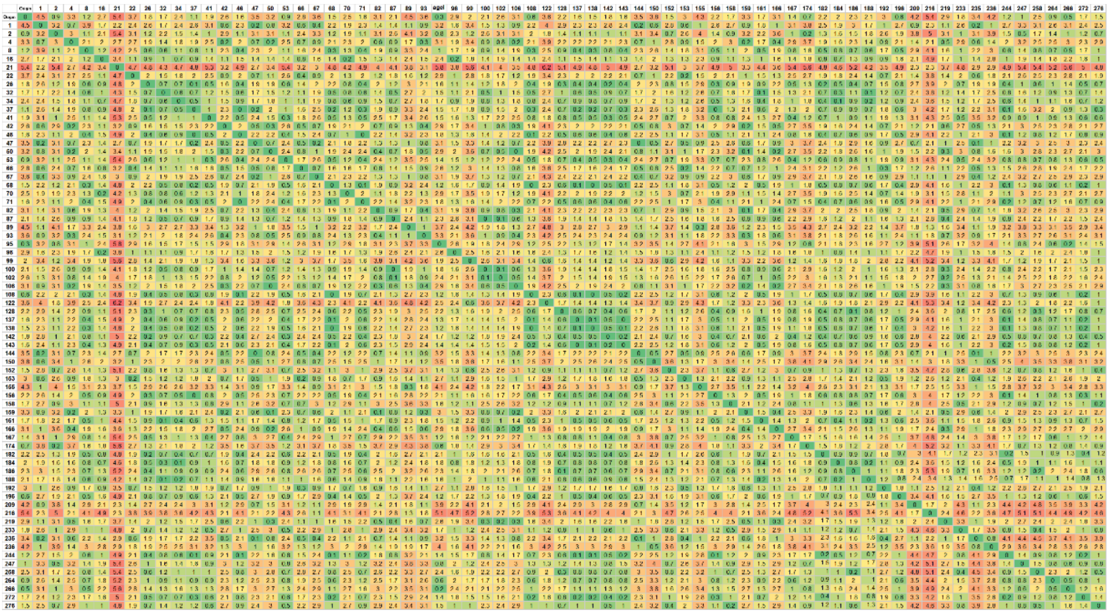


Figure 8. Distance matrix of Central zone.

5. Analysis of Results

5.1. North Zone

Figure 9 shows the route that covers all points in the North Zone, which served as the base for the analysis that led to the maximization of some distances. After applying the algorithm for two salesmen, the routes observed in Figures 10 and 11 were returned.

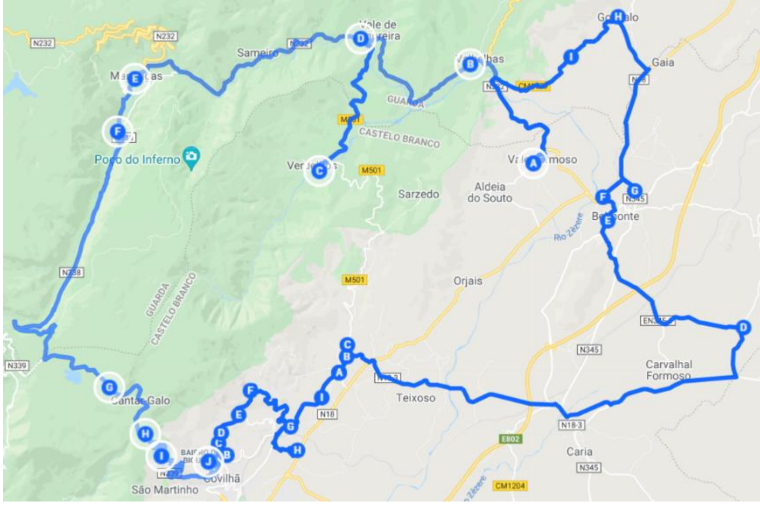


Figure 9. Long route of North zone.

Figure 10 represents the route that was referred to as the inside route, which runs through the points in the North Zone closest to the city.

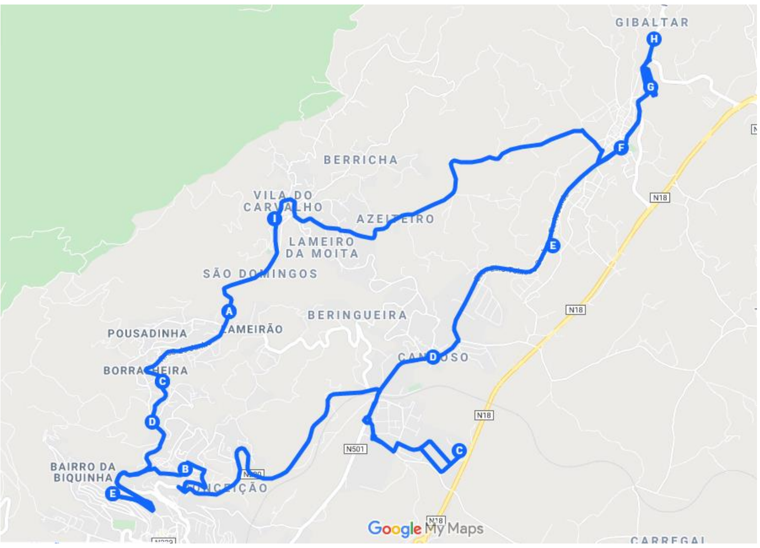


Figure 10. Route 2 of North zone.

In Figure 11, the route identified as the outside route is represented, which includes the remaining locations. As defined, this route already includes the key point 69 cluster.

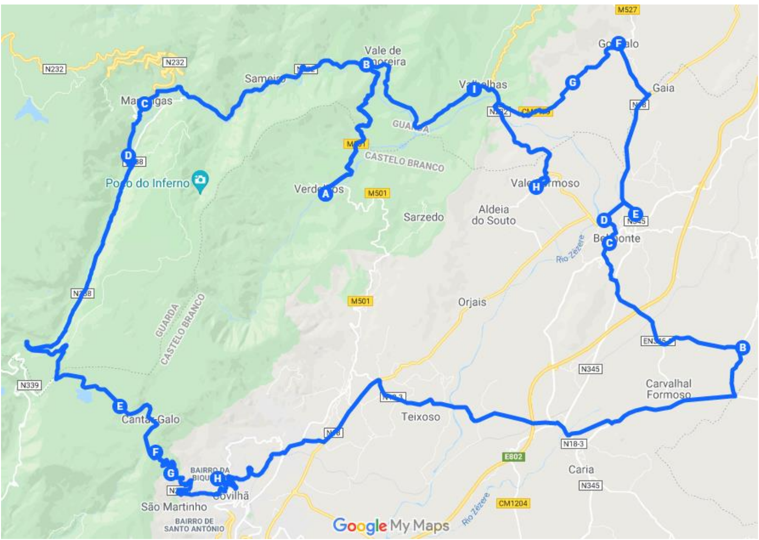


Figure 11. Route 1 North zone.

Table 3 shows the distances of the various routes mentioned above.

Table 3. Routes optimization for the North zone.

Route	Key Points	Distance (km)
Long route	0→121→73→215→145→52→162→60→148→229→ 261→103→217→202→57→175→125→253→213→15 →69→212→139→136→230→169→180→0	126.30
1st route	0→217→202→57→175→125→253→213→15→69 →212→139→136→230→169→180→0	118.60
2nd route	0→121→60→162→148→229→261→103→52→145 →215→73→0	21.10

5.2. South Zone

Figure 12 shows the route that covers all points in the South zone, which, through its analysis, led to the maximization of some distances. After applying the algorithm for two salesmen, the routes observed in Figures 13 and 14 were returned.

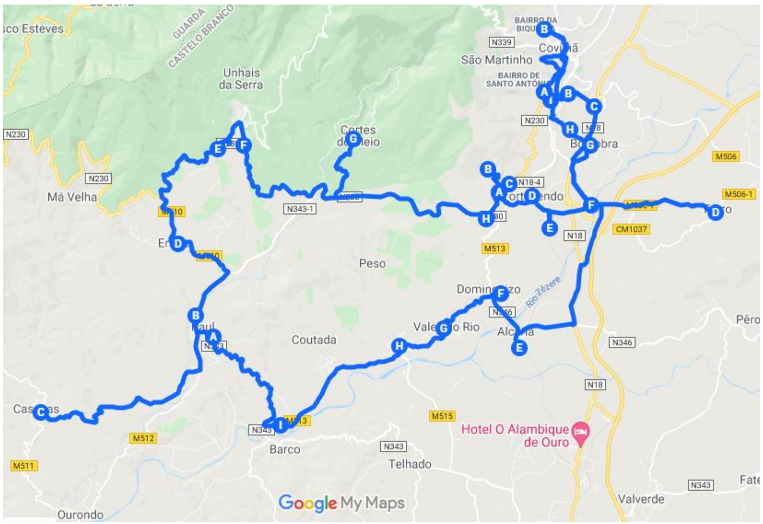


Figure 12. Long route of South zone.

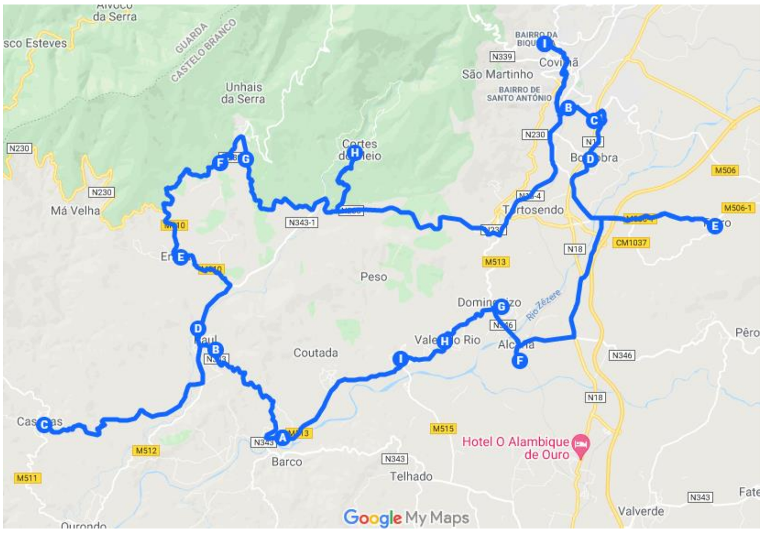


Figure 13. Route 1 South zone.

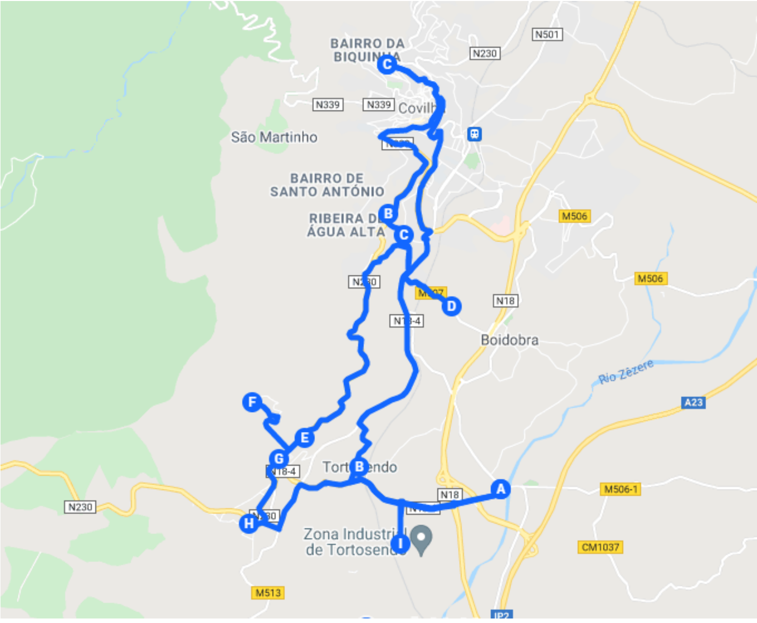


Figure 14. Route 2 South zone.

Once again, it is possible to conclude that the algorithm itself always opts for a larger route shown in Figure 13, and a smaller one shown in Figure 14, to minimize the distances from the first and last point of the routes to the origin.

In Figure 13 it is possible to observe the triangulation that occurs in the Paul-Casegas section. The distances between these points had to be maximized so that the algorithm would return the route that the worker will take.

In this particular case, it is beneficial that the road that connects Paul to Casegas (44) is traveled in both directions, although there is an alternative connection between 44 and 3 (Erada, the location that would be visited next, if the matrix was not adapted). The use of this route would result in a greater distance covered.

Figure 14 shows the route that runs through the establishments in the South zone of the city.

Comparing Figures 13 and 14, it is possible to see that, when returning to the headquarters, the sellers will go through very close roads (top of the outside route and bottom of the inside route) that could even be visited in an alternative route that encompassed all these points. However, in order for daily routes to be balanced and sellers to cover roughly the same number of establishments per day, the minimum number of points to visit on each route has a significant impact on the distribution of points.

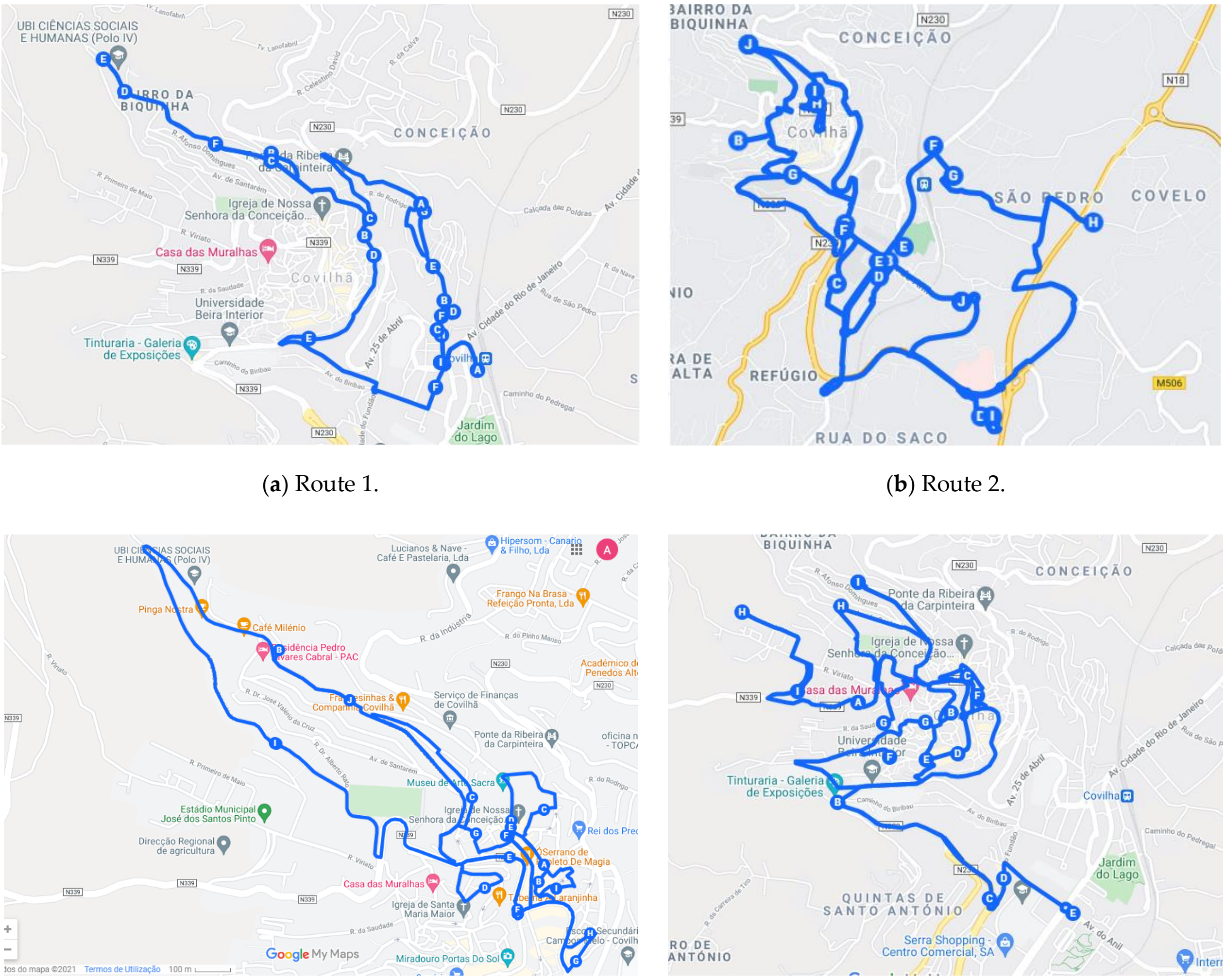
Table 4 includes the distances of the various routes mentioned above.

Table 4. Routes optimization for the South zone.

Route	Key Points	Distance (km)
Long route	0→260→54→218→259→254→199→120→131→79→44 →220→101→105→126→133→75→118→18→208→86→ 267→59→214→83→243→270→0	119.65
1st route	0→260→54→214→218→259→254→199→120→131 →79→44→220→101→105→126→133→0	100.50
2nd route	0→270→243→83→208→18→118→75→267→59→86→0	31.95

5.3. Central Zone

For the Central zone of the city, four routes were generated: two for each worker, representing the two days of the week that will be assigned in this zone. These routes are shown in Figure 15a–d.


UBI CIÊNCIAS SOCIAIS E HUMANAS (Polo IV)
BAIRRO DA BIQUINHA
CONCEIÇÃO
Igreja de Nossa Senhora da Conceição
Casa das Muralhas
Covilhã
Universidade Beira Interior
Tinturaria - Galeria de Exposições
Av. 25 de Abril
Av. Cidade do Rio de Janeiro
Jardim do Lago
N230
N339
N18
M506
SÃO PEDRO
COVELO
REFÚGIO
RUA DO SACO
Hipersom - Canaria & Filho, Lda
Lucianos & Nave - Café E Pastelaria, Lda
Pinga Nostra
Café Milénio
Frango Na Brasa - Refeição Pronta, Lda
Serviço de Finanças de Covilhã
Ponte da Ribeira da Carpinteira
Museu de Arte Sacra
Estádio Municipal José dos Santos Pinto
Direcção Regional de agricultura
Igreja de Santa Maria Maior
Miradouro Portas Do Sol
Google My Maps
QUINTAS DE SANTO ANTÓNIO
Serra Shopping - Centro Comercial, SA
A
B
C
D
E
F
G
H
I
J

(**a**) Route 1.

(**b**) Route 2.

(**c**) Route 3.

(**d**) Route 4.

Figure 15. Routes in central Zone.

Thus, taking into account the proximity of all points, there was no need to optimize the routes, so the 72 × 72 matrix was considered (Figure 8). As can be seen in Table 5, the distances of the routes within this zone have much smaller values than the other zones.

Table 5. Routes optimization for the Central zone.

Route	Key Points	Distance (km)
1st route	0→196→161→188→16→42→4→50→159→93→82 →106→166→219→102→87→100→32→8→266→95→0	8.24
2nd route	0→158→66→1→67→155→89→216→21→209→144→150 →236→235→192→70→46→108→0	15.40
3rd route	0→264→167→41→142→26→184→34→182→244→156 →272→68→138→137→143→2→174→0	5.99
4th route	0→53→71→128→186→258→233→122→99→152→96 →22→153→47→37→276→247→0	9.03

6. Discussion

The current algorithm allows for the best optimization to this kind of problem. The singularity of the orography in this case study requires special attention and the adaptability of the algorithm. The current adaptation to the algorithm proposed by Kirk [22] represents a versatile answer to the need for routing. Because of the simplicity of the inputs, some can easily be changed to accommodate the current needs of its user.

Given the inability to predict when changes will occur, punctual or definitive, in the number of workers, the flexibility of the developed tool represents a very important feature, as it allows for high adaptability to the day-to-day of the company in which it is applied.

The fact that the routes were previously divided in a way so that Fridays are reserved for unforeseen events and trips to establishments that, for some reason, were not visited, gives the company a larger margin, as the routes associated with the worker or workers missing can be carried out on that day. The solutions presented for the division of zones and places by days have a large scope for responding to unforeseen events, however, the possibility of changing routes was never excluded. This way, given the need for this change, it is only necessary to change the parameter referring to the number of available workers, thus allowing a new route solution to be returned, adapted to the new problem, for the whole week or just for a specific day. Other parameters, like the minimum number of visited establishments and the matrix size, can be changed in order to obtain different solutions, only needing to adapt the distances' matrix accordingly to the weekly needs.

Considering the optimizations proposed in this article, the total sum of distances traveled would decrease to 309 km per week for sellers, and these routes would be replicated by distributors, for a total of 618 km per week. Although the shorter intra-cluster distances were not considered in this study, as they are relatively short distances and often end up not being actually covered, a decrease of more than half of the covered kilometers by the workers is expected. This significant reduction was possible due to adjustments made in the functions of workers in the company and through the application of the m-TSP resolution algorithm [22] with some adaptations in the distance matrices. Despite the good results obtained, it is important to highlight the limitations characteristic of this type of problem. This type of algorithm does not consider constraints, such as traffic, orography of the area, road directions and, perhaps most significantly, does not take into account the possibility of visiting a path and returning by the same route if distances so justify because it is based on the m-TSP approach. This limitation leads to the need of adapting the distance's matrix for each zone, preventing the choice of certain roads, which the users may know are not reliable.

7. Conclusions

Given the lack of algorithm and methods applicable to this case study in particular, due to the orography, an adapted Genetic Algorithm was applied to the Multiple Traveling Salesman Problem (m-TSP) as a decision support system for optimizing a beverage logistics vehicle routing problem. A case study was developed in a mountain region city, Covilhã (Portugal) to optimize routes, for the five days of the week, which minimize travel costs and the distances traveled by workers, considering the approximately 270 establishments. The optimization of the daily routes for the workers assigned to distribution was divided into three zones: North, South and Central. A reduction of 618 km per week of the total distance traveled was determined by the algorithm. This result promotes provider-client relations by ensuring that clients are visited in time; reduces the fixed costs with fuel by the lower distance traveled and simultaneously promotes environmental sustainability by the shortening of logistical routes. All these improvements and optimizations can be thought of as contributions to foster smart cities.

Author Contributions: Conceptualization, T.M.L.; methodology, D.E.G., M.I.D.I., A.P.P., T.M.L., P.D.G.; validation, T.M.L., P.D.G.; formal analysis, D.E.G., M.I.D.I., A.P.P.; investigation, D.E.G., M.I.D.I., A.P.P.; resources, D.E.G., M.I.D.I., A.P.P., T.M.L.; data curation, D.E.G., M.I.D.I., A.P.P.; writing—original draft preparation, D.E.G., M.I.D.I., A.P.P.; writing—review and editing, T.M.L., P.D.G. and T.M.L.; supervision, P.D.G. and T.M.L.; project administration, P.D.G. and T.M.L.; funding acquisition, P.D.G. and T.M.L. All authors have read and agreed to the published version of the manuscript.

Funding: P.D.G. and T.M.L. acknowledge Fundação para a Ciência e a Tecnologia (FCT—MCTES) for its financial support via the project UIDB/00151/2020 (C-MAST).

Acknowledgments: Authors acknowledge the support of the company that provided the routes.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aletà, N.B.; Alonso, C.M.; Ruiz, R.M.A. Smart mobility and smart environment in the spanish cities. *Transp. Res. Procedia* **2017**, *24*, 163–170. [CrossRef]
2. Melo, S.; Macedo, J.; Baptista, P. Guiding cities to pursue a smart mobility paradigm: An example from vehicle routing guidance and its traffic and operational effects. *Res. Transp. Econ.* **2017**, *65*, 24–33. [CrossRef]
3. Jacobs, F.R.; Chase, R.B. *Operations and Supply Chain Management*, 15th ed.; McGraw Hill: New York, NY, USA, 2018.
4. Mandal, S.K.; Pacciarelli, D.; Løkketangen, A.; Hasle, G. A memetic NSGA-II for the bi-objective mixed capacitated general routing problem. *J. Heuristics* **2015**, *21*, 359–390. [CrossRef]
5. Öncan, T.; Cordeau, J.-F.; Laporte, G. A tabu search heuristic for the generalized minimum spanning tree problem. *Eur. J. Oper. Res.* **2008**, *191*, 306–319. [CrossRef]
6. Wong, R. Integer programming formulations of the traveling salesman problem. In Proceedings of the IEEE International Conference of Circuits and Computers, New York, NY, USA, 1–3 October 1980; pp. 149–152.
7. Theurich, F.; Fischer, A.; Scheithauer, G. A branch-and-bound approach for a Vehicle Routing Problem with customer costs. *EURO J. Comput. Optim.* **2021**, *9*, 100003. [CrossRef]
8. Sazonov, A.; Derjabin, V. Forecasting to paths of the motion ship with the help of neyronnoy network. *Vestn. Gos. Univ. Morskogo Rechn. Flot. Im. Admirala S. O. Makarova* **2013**, *3*, 6–13.
9. Sakharov, V.; Chernyi, S.; Saburov, S.; Chertkov, A. Automatization Search for the Shortest Routes in the Transport Network Using the Floyd-warshell Algorithm. *Transp. Res. Procedia* **2021**, *54*, 1–11. [CrossRef]
10. Azis, H.; Mallongi, R.D.; Lantara, D.; Salim, Y. Comparison of Floyd-Warshall algorithm and Greedy Algorithm in determining the shortest route. In Proceedings of the 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT), Makassar, Indonesia, 6–7 November 2018; pp. 294–298. [CrossRef]
11. Jiang, C.; Wan, Z.; Peng, Z. A new efficient hybrid algorithm for large scale multiple traveling salesman problems. *Expert Syst. Appl.* **2020**, *139*, 112867. [CrossRef]
12. Wang, Z.; Fang, X.; Li, H.; Jin, H. An improved Partheno-Genetic Algorithm with reproduction mechanism for the multiple traveling salesperson problem. *IEEE Access* **2020**, *8*, 102607–102615. [CrossRef]
13. Fujdiak, R.; Masek, P.; Mlynek, P.; Misurec, J.; Olshannikova, E. Using genetic algorithm for advanced municipal waste collection in Smart City. In Proceedings of the 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Prague, Czech Republic, 20–22 July 2016; pp. 1–6.

14. Bektas, T. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* **2006**, *34*, 209–219. [CrossRef]
15. Cheikhrouhou, O.; Khoufi, I. A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy. *Comput. Sci. Rev.* **2021**, *40*, 100369. [CrossRef]
16. Garn, W. Balanced dynamic multiple travelling salesmen: Algorithms and continuous approximations. *Comput. Oper. Res.* **2021**, *136*, 105509. [CrossRef]
17. Fabri, A.; Recht, P. On dynamic pickup and delivery vehicle routing with several time windows and waiting times. *Transp. Res. Part B Methodol.* **2006**, *40*, 335–350. [CrossRef]
18. Caramia, M.; Italiano, G.F.; Oriolo, G.; Pacifici, A.; Perugia, A. routing a fleet of vehicles for dynamic combined pick-up and deliveries services. *Oper. Res. Proc.* **2002**, 3–8. [CrossRef]
19. Mosayebi, M.; Sodhi, M.; Wettergren, T.A. The Traveling Salesman Problem with job-times (TSPJ). *Comput. Oper. Res.* **2021**, *129*, 105226. [CrossRef]
20. Hillier, F.S.; Lieberman, G.J. *Introduction to Operations Research*, 9th ed.; McGraw Hill: New York, NY, USA, 2010.
21. Yang, S.; Shao, Y.; Zhang, K. An effective method for solving multiple travelling salesman problem based on NSGA-II. *Syst. Sci. Control Eng.* **2019**, *7*, 108–116. [CrossRef]
22. Kirk, J. Fixed Start/End Point Multiple Traveling Salesmen Problem-Genetic Algorithm. 2021. Available online: https://www.mathworks.com/matlabcentral/fileexchange/21299-fixed-start-end-point-multiple-traveling-salesmen-problem-geneticalgorithm (accessed on 2 August 2021).