

Solving a Logistics System for Vehicle Routing Problem Using an Open-Source Tool

Filipe Alves[1,2(✉)], Filipe Pacheco[2], Ana Maria A. C. Rocha[1], Ana I. Pereira[2], and Paulo Leitão[2]

[1] ALGORITMI Center, University of Minho, 4710-057 Braga, Portugal
arocha@dps.uminho.pt,a32657@alunos.ipb.pt
[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
{filipealves,apereira,pleitao}@ipb.pt

Abstract. The growing demand for logistics services for deliveries, collections, or home health services, have significantly increased. However, there is a need to have a technologically innovative information system for digitizing data in the operational logistics of these services, required for an increasingly better vehicle route planning. Unsurprisingly, for many years, there has been an increasing and steady growth in the interest and development of optimization tools to solve real-world problems, namely in the logistic domain. The evolution and support of computational power and the fact that advances in optimization solvers have allowed many of them to be developed as free or open-source software, to the detriment of some classic numerical calculation software. The main issue arises in the dynamic search for solutions obtained by open-source solvers and how they can be useful in solving complex combinatorial problems in real life, such as the optimal allocation of routes in logistics planning services. This work proposes an application that integrates the Google OR-Tools software and the Google Maps and Distance Matrix API. The approach developed in this work uses a VRP mathematical model to minimize the maximum route (considering as objective function the time or the distance) and provide a workload balancing, with the use of a cloud application to reduce costs and an online map service. Experimental results were obtained on simulated VRP instances in the district of Porto, where the quality of the computational solution is analyzed for training and easy usability in logistics problems.

Keywords: Logistics system · VRP · Open-source solver

This work has been supported by FCT – Fundação para a Ciência e a Tecnologia within the R&D Units Projects Scope: UIDB/00319/2020. Filipe Alves is supported by FCT Doctorate Grant Reference SFRH/BD/143745/2019.

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12953, pp. 397–412, 2021.
https://doi.org/10.1007/978-3-030-86976-2_27

1 Introduction

Nowadays, sharing and "on demand" economy, the number of available services and the demand for them have increased. New services, such as Uber have emerged. One of the benefits of such a system is the ability to develop optimization logic that increases profits for partners, and decreases the costs for customers [14]. It is no different in logistics, where solving routing problems makes delivery of goods more efficient by optimizing fleets of vehicles - allowing them to deliver more orders and helping each driver to complete more deliveries. Problems such as the Travelling Salesman Problem (TSP) or Vehicle Routing Problem (VRP) have always received a lot of attention in the literature. TSP was first explored by Flood [6] and VRP was introduced using the example of optimum routing of a fleet of gasoline delivery truck by Dantzig et al. [5]. Ever since, these two problems have been important in the field of operations research.

Any process that requires transportation and visit planning is among the most important tasks of operational and strategic management, whether in a product company or in home visits by informal caregivers. The complex problem of route planning consists of the proper and optimal rationalization of the process of allocation and distribution of resources offered by the network in question.

In operational research, the route planning is usually included in the class of VRP. The VRP is a problem of combinatorial optimization and integer programming, which normally tries to organize a set of routes to satisfy constraints and requirements, minimizing the cost of global transportation (monetary, distance or other context). VRP, nowadays, is already studied taking into account several factors that can affect it, such as vehicle capacity, number of depots, traffic conditions or time restrictions (time windows). These factors make route optimization and scheduling increasingly difficult in reasonable time. In fact, VRP is an NP-hard problem, which means that, in practice, it is not possible to build an expert algorithm that will always provide optimal solutions in an acceptable amount of time. However, recent advances on Information and Communications Technologies (ICT) - increasing use of GPS, traffic map, distributed systems - open up new possibilities to optimize the process of planning and scheduling routes in several domains. New technologies combined with optimization techniques allow the practical development and implementation of new solutions to support decision-making in the operational logistic. The increase in computing power and the development of new solvers using classical algorithms, have enabled the development of efficient tools capable of solving many routing issues, especially those that can be represented in the form of VRP.

This research work focuses on the development of a cloud application based on an open-source tool for solving VRP. The application integrates the Google OR-Tools for VRP optimization, the Google Maps and the Google Distance Matrix API as external services. The Google OR-Tools software is detailed, and has gained great popularity as an efficient optimization package for several combinatorial problems. Another contribution of this work is the experimental results of the application on simulated instances of VRP in the district of Porto.

The paper is divided as follows: Sect. 2 briefly introduces the related literature, while Sect. 3 shows the architecture of the system to be developed. Section 4

presents the mathematical formulation of the VRP. The open-source solver tool used to solve the VRP is described in Sect. 5. The computational experiments, data collection and strategies of VRP are presented and discussed in Sect. 6. Finally, Sect. 7 rounds up the main conclusions and future work.

2 Related Literature

The vehicle routing problem [3], a well-known combinatorial optimization [17] problem, is defined as a fleet of vehicles in a network seeking to serve a set of customers, starting from a single depot. The typical VRP objectives are to minimize the total distances travelled by all vehicles or to minimize the travel costs while serving every customer. The VRP has been studied in applied mathematics and computer science for decades. Dantzig et al. [5], have the first record in the VRP literature, where they studied a relatively large scale travelling salesman problem and proposed the mathematical programming formulation and algorithmic approach to solve the problem of delivering gasoline to service stations.

VRP is known to be a NP-hard problem for which many exact and heuristic algorithms have been proposed. However providing fast and reliable solutions is still a challenging task. Basically, a single or a fleet of vehicles are responsible for delivering items or perform something to multiple customer nodes, starting and returning at a given node, called the depot. The main objective is to optimize the set of routes in order to attain the maximum possible reward, optimizing the use of resources and try to obtain the best workload balance. The VRP problem is computationally difficult to solve to optimality, but throughout the time, there have been many proposed solutions, such as, exact methods. For example, one of the first exact method proposed was the branch-and-bound algorithm [2]. However, many of the literature in exact methods relate the work only for up to 360 crossing points, which may be insufficient in some cases [15]. For an overview of the VRP, see, for example, [8,11,12,19].

The evolution in the operational research was considering metaheuristics, in order to improve the sub-optimal solutions. Examples of efficient metaheuristics, include genetic algorithms [10], local search [13] or ant colony optimization [7].

In terms of software, there are some classic and commercial solutions, such as Gurobi [9], LocalSolver [20] and IBM CPLEX [4], which are used to solve VRP problems. However, they require high computational effort, are commercial licenses and generally used in an academic context. Currently, there are cloud optimization applications that solve VRP problems, such as OptimoRoute,[1] Badger Maps,[2] Route4Me,[3] Routific tools,[4] and many others. These software provide systems for optimizing routes, supported by API, but they are in a commercial scope, with closed source and subscription plan fees. In this sense, Google OR-Tools [16], an open source Google optimization tool for VRP issues with an active community and several libraries, has recently emerged. The tool is completely free, customizable and capable of commercial and academic applications.

[1] https://optimoroute.com/.
[2] https://www.badgermapping.com/.
[3] https://www.route4me.com/.
[4] https://routific.com/.

3 System Architecture

This project aims to develop a dynamic and innovative application based on a robust system for digitized data manipulation and integrated use of open-source tools as external services to use optimization algorithms and real maps. The architecture leverages back-end data models, with integration into a database, and integrates a well-structured tool, namely Google OR-Tools as external service, to contribute to the optimized route planning and simulation of logistical cases. The structure of the system has a cloud architecture to facilitate flexibility between the different modules and components. On the other hand, the creation of the application programming interface (API) allows routines and patterns of access to the system by the developed dashboard (front-end). In this way, the architecture provides a service capable of being used by different users, adapting the data according to the problems and monitoring/accessing the routing and scheduling solutions through an interface. The communication of the system to its different parts can be seen in the Fig. 1.

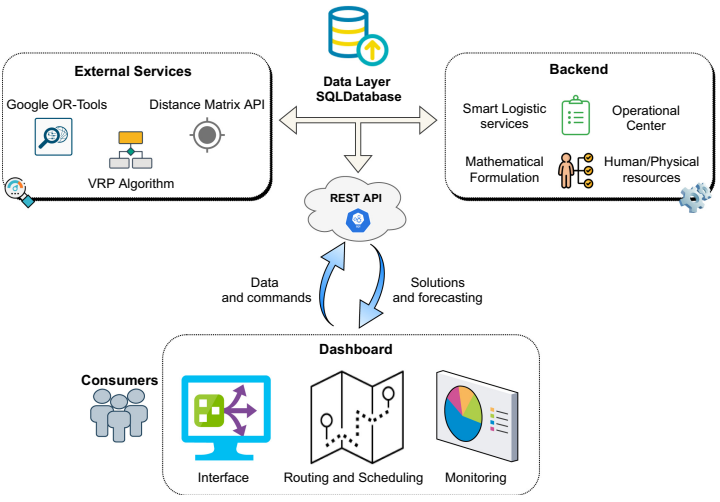

Fig. 1. System architecture approach.

The module integrated with external services, will have optimization skills to provide mechanisms for optimized solutions, using VRP algorithms implemented on the Google OR-Tools platform, with easily edited cognitive models and with different solution search strategies. On the other hand, the use of Google Distance Matrix service allows to compute the distance and duration of the trip between multiple origins and destinations for a given problem. In this sequence, the Google Maps API will offer customization and use of static and dynamic maps, where users will be able to obtain real-time data and advanced information from specific locations for a greater reality of the project. The mathematical

formulation on the back-end will assist the VRP engine to search for the optimal solution taking into account the needs, distance or duration of the routes under study. The dashboard module provides a connection to the platforms using a user-friendly interface, which allows real-time monitoring of the solution state.

4 VRP Formulation

The main idea is to solve a logistics system for VRP using a well-known algorithm embedded in an open-source solver. The approach is to consider a scenario where a number of points (locations) are distributed over a certain area (Portugal), each of which demands some kind of services. The service provider has a fleet of vehicles, typically homogeneous and initially located at a single location, called the depot. Each of these vehicles can provide service trying to guarantee equity of services, that all vehicles have an assigned route and that it has a good workload-balancing. Keeping in mind the recent concern towards the environment and the fact that the service provider wishes to distribute work equally among all the vehicles, a mathematical model has been applied to minimize the total time and distance for the routes. The problem is to find a set of routes, one route for each of the vehicles such that the route distance or the route time are minimized.

The routes consist of a set of nodes which are arranged sequentially in that particular order in which they are visited. Each route starts and ends at the same depot, and each node is serviced by one vehicle. The classic VRP formulation was taken into account and adapted to guarantee the proper flow of vehicles [18]. The mathematical model to describe this approach is presented below:

- n: number of nodes to be serviced (locations),
- v: total number of vehicles used,
- d_{ij}: cost (distance) to go from i to j,
- t_{ij}: cost (time) to go from i to j.
- u_i^k: place in the sequence of visits of vehicle k for the customer i.

The binary variables take the form:

$$x_{ijk} = \begin{cases} 1, \text{ if the vehicle } k \text{ visits } j \text{ immediately after } i; \\ 0, \text{ otherwise.} \end{cases}$$

The objective function that could be minimized is

$$\min \quad D = \sum_{k=1}^{v} \sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk} d_{ij} \tag{1}$$

or based on the minimization of the maximum routing time

$$\min \quad T = \sum_{k=1}^{v} \sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk} t_{ij} \tag{2}$$

Some constraints are assumed as listed below:

$$\sum_{k=1}^{v}\sum_{j=0}^{n} x_{ijk} = 1, \quad \forall i \in N; \tag{3}$$

$$\sum_{j=0}^{n} x_{0jk} = 1, \quad \forall k \in K; \tag{4}$$

$$\sum_{i=0}^{n} x_{jik} - \sum_{j=0}^{n} x_{ijk} = 0, \forall i \in N; \forall k \in K; i \neq j \tag{5}$$

$$u_j^k \geq u_i^k - C(1 - x_{ij}^k) + 1, \quad \forall i, j \in N; \forall k \in K; i \neq j \tag{6}$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in N; \forall k \in K; \tag{7}$$

where N represents the index set of all locations $N = \{1, ..., n\}$ and K represents the index set associated to the vehicles $K = \{1, ..., v\}$.

The constraints in (3) state that each customer is visited exactly once, while constraints (4) and (5) ensure that each vehicle is used exactly once and that flow conservation is satisfied at each customer vertex. Finally, the sub-circuit elimination are expressed by constraints (6). Alternative formulations of VRP are already explored by Bektas [1]. Thus, taking into account this mathematical model, it will be able to optimize the use of vehicles and minimize the assigned routes, leveraging a better balancing workload and empowering the user to select the route based on their preferred objective function (time or distance).

5 Open Source Solver

There are many barriers and difficulties related to the development of algorithms to solve classic VRP problems, either due to the different characteristics, specific requirements, or even the closed source code of commercial software. These barriers also hamper the development of distributed and decentralized systems with little cooperation in supplier software. Nowadays, more open source tools and libraries are available to solve different types of VRP. One of the great examples of this, with recent activity and high trust from users and researchers, is Google OR-Tools. This suitable library to solve a particular and practical VRP, presents statistical data of reasonable speed, programming environment, flexibility, functionality and easy to use in different programming languages. Many of the criteria chosen for the use of this platform, are related to the access and applicability in many real-world scenarios, such as:

- Adjusted to handle the most difficult problems in vehicle routing, flows, integer and linear programming, and constraint programming.
- Flexibility in choosing and adding the modified algorithm itself (generating an initial solution) and repairing or evolving an algorithm (heuristic and/or metaheuristic strategies).

- Choice and definition the objective function by the user.
- Flexibility in defining soft or specific constraints, and execution speed (especially for large problem instances).
- Quality of produced solutions.
- Integration skills with external systems or online services (Google Maps API, and others) as well as visualization systems.
- Testing tools with abilities of use benchmark sets and different datasets in several formats (csv, xlsx, json, sql, and others).

By having a programming environment and integration of different external services, the tool provides customization and mechanisms for different test scenarios in dynamic environments. Thus, it is also interesting to understand the steps of the procedure for the proper functioning of the model and its algorithmic execution in the open-source solver (see the flowchart presented in Fig. 2).

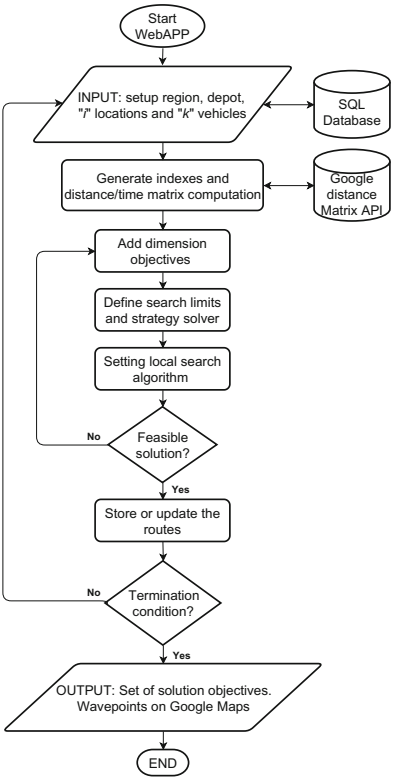

Fig. 2. Sequential procedure for executing the VRP adapted on the OR-Tools platform.

In general, OR-Tools is an open-source package from Google, fast and portable software suite for solving complex and simple combinatorial optimization

problems, which allows the use for specific real-world problems [16]. It supports various several programming languages, including C++, C#, Java and Python. OR-Tools can solve many types of VRP, including problems with pickups and deliveries, heterogeneous fleets, multiple depots, time windows, different start and end locations, multiple capacity dimensions, initial loads, skills, scheduling problems, and so on. Obviously, there are some limitations on solving VRP as they are inherently intractable for larger instances. Therefore, OR-Tools sometimes returns solutions that are good, but are not optimal. Sometimes better solutions can be found by changing the search options for the solver. The first solution strategy is used to find an initial solution (one can also specify a set of already found initial routes).

The platform presents several first solution strategies. Some strategies let the solver detect the strategy to be used according to the model being solved, where other strategies iteratively build a solution by inserting the cheapest node at its cheapest position; the cost of insertion is based on the arc cost function. In the topic of decision strategies, the user is also given the possibility to incorporate local search options. This strategy considers six examples of search options, such as greedy descent, simulated annealing, and tabu search.

The documentation of OR-Tools is available with plenty of examples and API documentation. Google OR-Tools is very promising, recent, with an extensive and intensive network of development in operational research by one of the largest companies in the world, which makes available the software for solving various VRP problems.

6 Computational Experiments

This section describes the Google OR-Tools optimization tool applied to different designed and simulated cases of logistics VRP.

6.1 Data Collection

The dataset was prepared to allocate different locations in the district of Porto, as an example, where operational and logistics services in the delivery or collection of products and/or provision or follow-up of health care are increasingly urgent and necessary. However, the experiments can be easily replicated in other regions or countries. Even more with the pandemic situation of COVID 19, more and more people are requesting these logistics services, either from local units or from social solidarity institutions. The data allow replicating daily cases of operational visit planning and are inserted and saved according to the development of a database on the platform (SQLserver). The datasets can be chosen at random during the analysis.

Each dataset generated randomly comprises the locations and the depot and represents cases of a logistics system, being able to provide simulations of routing and task planning in operational services of the kind. The dataset was generated

using the database and it was possible to generate a set of tests with different instances, which can be assigned to a single day or divided over several days.

To evaluate the robustness of the open-source tool (OR-Tools), multiple problem sizes with different number of vehicles were provided, making a total of 3 instances, where the dimension of each instance can be seen in Table 1.

Table 1. List of test instances.

Parameters		
Dataset	Locations (n)	Vehicles (v)
Case 1	10	2
Case 2	20	4
Case 3	30	6

It should be noted that the different locations are dynamically instantiated in a time/distance matrix between all start and end points.

In this sense, a sensitivity analysis will be conducted considering two optimization strategies. The first one involves heuristics and the other uses local search strategies based on metaheuristcs.

The tests are conducted on a Lenovo Yoga, with an Intel(R) Core(TM) M-5Y71 CPU 1.4GHz with 8.0 GB of RAM. The development of a user-friendly web application makes it possible to digitize the entire process of managing information and routes, enabling the system to dynamically insert data, vehicles, locations and other information. It should be noted that the calculations of distance and time, use the Google Maps cloud, which provides the facility to keep accurate locations and routes through comprehensive data and real-time traffic. In turn, the distance matrix provides travel times and distances to one or more locations, useful for the combinatorial calculation of the OR-Tools.

6.2 Strategies to Solve VRP

The method involves going through all the crossing points and health or technical professionals represented by each vehicle, along with the constraints to the Google Optimization Tools solver, choosing the main strategy and treating its output as the final solution. In the development platform, unit tests were structured for the entire application, both in data entry and in the output of solutions. The use of this approach allows, in general, different strategies to be tested using the OR-tools library. Thus, when invoking the Google Optimization Tools Routing library, different parameters were used, namely, the description of some options for the Routing Solver, in order to enhance the diversification of the solutions used.

In this sense, the first aspect to be taken into account is the search limits, which end when the solver reaches a specific limit, such as the maximum length

Table 2. Summary of search limits.

Parameter	Value	Description
Solution limit	∞	Limit to the number of solutions generated during the search
Time limit	5000	Limit in seconds of the time spent in the search
Lns time limit	100	Limit in seconds of the time spent on the completion search for each local search neighbor

of time, or number of solutions found. It is possible to set a search limit using the search parameters of the solver, as shown in Table 2.

In this context, after the search limits definition, it is important to address the first solutions strategies, that is, the methods that the solver uses to find an initial solution. Table 3, lists the six different types of options (out of a total of thirteen, some similar to each other), chosen for the problem under study.

Table 3. Summary of first solution strategies.

Strategy option	Description
Automatic	Lets the solver detect which strategy to use according to the model being solved
Path_Cheapest_Arc	Starting from a route "start" node, connect it to the node which produces the cheapest route segment, then extend the route by iterating on the last node added to the route
Savings	Savings algorithm (Clarke & Wright)
Christofides	Christofides algorithm (Nicos Christofides)
Best insertion	Iteratively build a solution by inserting the cheapest node at its cheapest position
Global cheapest arc	Iteratively connect two nodes which produce the cheapest route segment

The user can check the status returned by the method while the search unfolds, such as, "routing_not_solved", "routing_success - problem solved successfully", "routing_fail - no solution found for the problem", or for example, "routing_invalid".

In turn, the user also has local search options at his disposal. These options can change the search approach and provide better results, especially for more complex problems, than the first search options listed above. Table 4 presents the three local search strategies chosen as the object of study (out of a total of six), which can be used to set and cover a larger space for searching for solutions.

As a final parameter to end the call to Google Optimization Tool library, set the parameter that controls the propagation "Use Light Propagation" as true. This parameter describes the use of the constraints with light propagation in routing model. The "Extra propagation" control is only necessary when using depth-first search or for models which require strong propagation to finalize the

Table 4. Summary of local search strategies.

Local search	Description
Guided Local Search	Uses guided local search to escape local minima
Simulated annealing	Simulated annealing algorithm
Tabu search	Tabu search algorithm

value of secondary variables. Changing this setting to true will slow down the search in most cases and increase memory consumption in all cases.

The setting of the different parameters and options will generate different outputs and routes. In the same sequence, the performance will also be distinct, which will contribute to a sensitivity and metric analysis: runtime, maximum time/distance of the routes and the number of vehicles that are used. The main goal is to optimize the maximum time/distance, at a reduced runtime, without much impact on the workload balance between the vehicles.

6.3 Results

In this section, the solutions found using the open-source Google optimization tool (OR-Tools) are presented.

Thus, different first solution strategies, from Table 3, were applied to VRP instances. After the first layer of tests, the optimal routes were generated to minimize the length of the longest route among all vehicles. In the three instances of the tests, the results corresponding to the routing of operational logistics using the first solutions strategies based on heuristics are presented in Table 5.

Table 5. List of results for the first solution strategies based on Heuristics.

Heuristics	Case 1		Case 2		Case 3	
	Route time (min)	Runtime (ms)	Route time (min)	Runtime (ms)	Route time (min)	Runtime (ms)
Automatic	72	75	77	133	100	238
Best insertion	–	–	–	–	–	–
Christofides	82	13	77	100	78	375
Global cheapest arc	72	8	76	87	86	313
Path cheapest arc	72	12	77	67	100	186
Savings	72	16	100	54	86	325

Considering the objective function selected by the user ($\min T$), a set of routes for the vehicles are generated with the total distances of the operational planning. It is possible to verify that the runtime increase as long as the complexity of the instances increase. In addition, there was a slight increase in terms of the time of the route and scheduling solution as long as the complexity and instances increase, mostly in Case 3.

When considering the local search strategies based on metaheuristics, the operational solutions obtained are shown in Table 6.

Table 6. List of results of the local search strategies based on metaheuristics.

Metaheuristics	Case 1		Case 2		Case 3	
	Route time (min)	Runtime (ms)	Route Time (min)	Runtime (ms)	Route Time (min)	Runtime (ms)
Guided Local Search	72	1001	74	5001	75	5001
Simulated annealing	72	1001	77	5001	75	5000
Tabu search	72	1002	77	5001	77	5005

The results are consistent when compared to the ones from Table 5, although were obtained better solutions for Case 2 and Case 3 in terms of route time. Note that, the OR-Tools allows to specify a runtime limit when using metaheuristcs, allowing to increase the chances of finding better solutions. Thus, different runtime values were selected. In general, the "Guided Local Search" metaheuristic strategy presents the best results in terms of routing time.

Figure 3 presents graphically the obtained results in Table 5. The graph shows on the x-axis the different strategies for first solutions (heuristics) with their respective color and grouped by the 3 cases under study. In turn, the y-axis presents the results of each of the strategies in the order of the route time (in minutes).

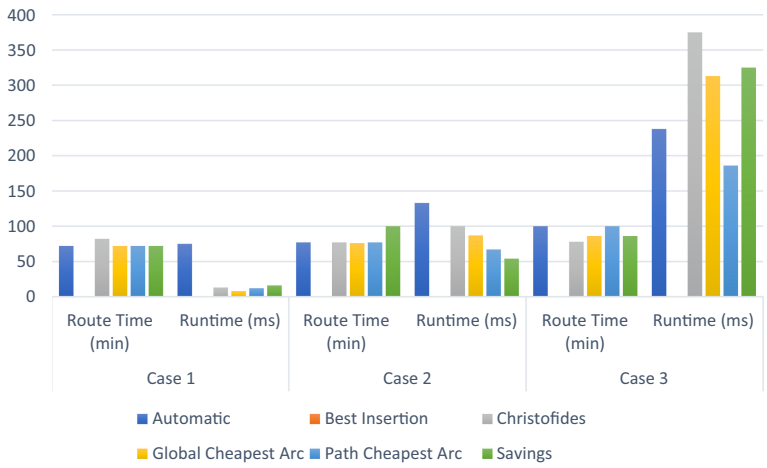

Fig. 3. Heuristics results for each case study.

The "Best Insertion" heuristic was not able to solve any case problem proposed. In a simplified way, it is possible to verify that the maximum route

solutions obtained by the different strategies do not differ much from case to case. However, it is possible to verify that the runtime of Case 3, when compared to Cases 1 and 2, has undergone a noticeable increase, possibly due to the increased in dimension and complexity.

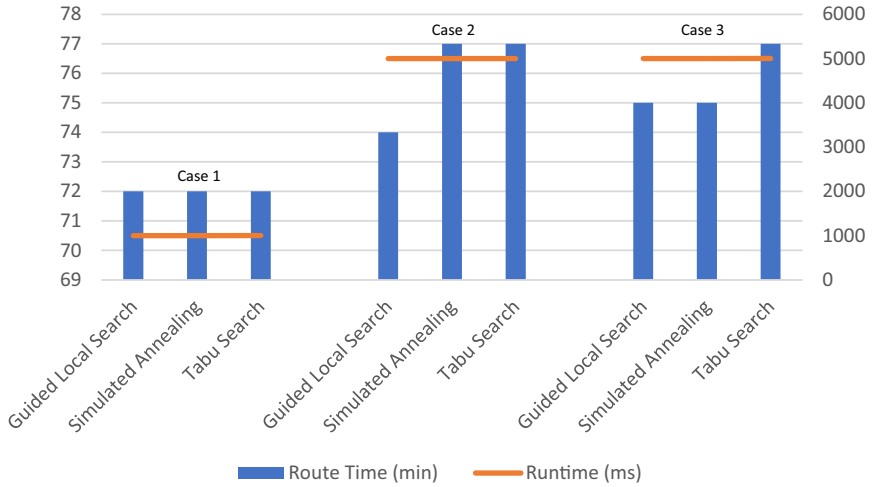

Fig. 4. Metaheuristics results for each case study.

The solutions obtained by the local search strategies (metaheuristics), in Table 6, will be represented for a better visualization of the optimization process, in Fig. 4. The values on the left vertical axis are related to the routing time, while the values on the right vertical axis refer to the runtime. Therefore, there is a slight increase in the maximum routes from Case 1 to the remaining cases.

Regarding the metaheuristic, it is possible to suggest the "Guided Local Search" as the best choice, due to its better performance (essentially in Case 2) relatively to the others. According to this suggestion, the scheduling and routing for Case 2 is presented in Table 7.

Table 7. Scheduling and routing using Guided Local Search for Case 2.

Guided Local Search		
Vehicles	Routes	Route time
Vehicle 0	0 ->17 ->16 ->11 ->2 ->3 ->0	72 min
Vehicle 1	0 ->15 ->12->13 ->4 ->7 ->10 ->0	72 min
Vehicle 2	0 ->8 ->1 ->18 ->19 ->0	74 min
Vehicle 3	0 ->5 ->9 ->6 ->14 ->0	73 min
Maximum route time: 74 min		
Runtime: 5001		

Location 0 represents the depot, where all vehicles start and end routes. As soon as the results are obtained, it is available for the user to view any of the VRP solutions and also has the availability to access any of the gantt graphics for recall via WiFi and inform each of the vehicles, of their respective route. It should be noted that all requests seek to establish an improved workload-balance between all the vehicles, to facilitate distribution and acceptance by logistics managers.

Since access via the app is facilitated with quick access to solutions, the user will be able to, after viewing the schedule, interact dynamically by sending the specific coordinates of the route of one of the vehicles to its user and customize the real-time information according to google maps and places to visit. Figure 5 illustrates the route and tracking of the locations to be visited by one of the vehicles for Case 2. The experience of using this app becomes real, with the possibility in obtaining updated reports, optimized directions and geolocation (using google maps API service included in the system).

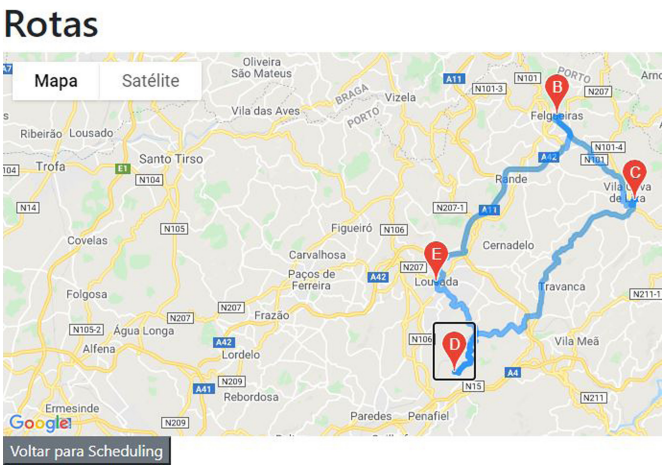

Fig. 5. Route map wave-points.

The user will always be able to monitor any of the routes and interact with it, being able to return to the main interface, viewing the remaining solutions of the different strategies.

In turn, it is important to highlight the fact that the results presented above are according to the search and optimization of problems according to the objective of minimizing the time in routes. The user when defining the dimensions of the objectives, defining the solver strategies and respective algorithms will have the opportunity to decide whether to compute the matrix dimension between the localities, according to distance or time. The Google matrix service automatically generates the matrix according to the user's decision, being able to change the objectives of the problem whenever he/she wishes. In this case, minimizing

the distance, and with the possibility of data input in real-time, the results would be different, but in any case, the whole process of calculations and combinations in obtaining solutions is the same. The idea is to enhance a system that provides the best solutions and in turn enables the user to support decision making in logistical VRP problems.

The proposed system architecture has significant potential to be used in real-world problems with further improvements, even in the application of the other VRP variants.

7 Conclusion

This research shows the extreme usefulness of a recent and popular open-source optimization tool for solving a logistics system for VRP in a specific region in Portugal, namely in the district of Porto. The developed system revealed some interesting properties of the considered optimization toolkit, as well as in the digitization of information in a user-friendly way.

The different case studies and instances revealed a very balanced solution behavior, with very good quality in the results and in the runtime, the latter often considered as the essential performance parameter for the choice and adoption of any optimization solver for real application. The system also presented different and valid configurations according to the optimization strategies of the initial solution, heuristics and metaheuristics, providing a weighted choice to the user.

The ease of integration in programming environments (in this case C#), plus the use of the Google Maps and Distance Matrix API service, integrated with the development of the database for the engine, retrieve and manage information, created a potential future prototype for more application cases. The application can be used to provide guidance on real VRP in the Porto district or replicated in other places. Embedding this system and optimization resources in the cloud, combined with artificial intelligence, namely multi-agent systems, may in the future, enable the distribution of dynamic and reactive VRP, thus managing emergencies and unexpected events in real-time.

References

1. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega **34**(3), 209–219 (2006)
2. Christofides, N., Eilon, S.: An algorithm for the vehicle-dispatching problem. J. Ope. Res. Soc. **20**(3), 309–318 (1969)
3. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
4. Cplex, I.I.: V12.1: user's manual for CPLEX. Int. Bus. Mach. Corp. **46**(53), 157 (2009)
5. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manage. Sci. **6**(1), 80–91 (1959)
6. Flood, M.M.: The traveling-salesman problem. Oper. Res. **4**(1), 61–75 (1956)

7. Gambardella, L.M., Taillard, É., Agazzi, G.: MACS-VRPTW: a multiple colony system for vehicle routing problems with time windows. In: New Ideas in Optimization. Citeseer (1999)
8. Golden, B.L., Raghavan, S., Wasil, E.A.: The Vehicle Routing Problem: Latest Advances and New Challenges, vol. 43. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-77778-8
9. Gurobi Optimization, L.: Gurobi optimizer reference manual (2021). http://www.gurobi.com
10. Karakatič, S., Podgorelec, V.: A survey of genetic algorithms for solving multi depot vehicle routing problem. Appl. Soft Comput. **27**, 519–532 (2015)
11. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. Eur. J. Oper. Res. **59**(3), 345–358 (1992)
12. Laporte, G., Gendreau, M., Potvin, J.Y., Semet, F.: Classical and modern heuristics for the vehicle routing problem. Int. Trans. Oper. Res. **7**(4–5), 285–300 (2000)
13. Nagata, Y., Bräysy, O.: Efficient local search limitation strategies for vehicle routing problems. In: van Hemert, J., Cotta, C. (eds.) EvoCOP 2008. LNCS, vol. 4972, pp. 48–60. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78604-7_5
14. Nguyen, H.N., Rintamäki, T., Saarijärvi, H.: Customer value in the sharing economy platform: the Airbnb case. In: Smedlund, A., Lindblom, A., Mitronen, L. (eds.) Collaborative Value Co-creation in the Platform Economy. TSS, vol. 11, pp. 225–246. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8956-5_12
15. Pecin, D., Pessoa, A., Poggi, M., Uchoa, E.: Improved branch-cut-and-price for capacitated vehicle routing. Math. Programm. Comput. **9**(1), 61–100 (2016). https://doi.org/10.1007/s12532-016-0108-8
16. Perron, L., Furnon, V.: OR-Tools (2019). https://developers.google.com/optimization/
17. Sbihi, A., Eglese, R.W.: Combinatorial optimization and green logistics. 4OR **5**(2), 99–116 (2007). https://doi.org/10.1007/s10479-009-0651-z
18. Toth, P., Vigo, D.: An overview of vehicle routing problems. In: The Vehicle Routing Problem, pp. 1–26 (2002)
19. Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM (2002)
20. Zhang, Y., Qi, M., Miao, L., Wu, G.: A generalized multi-depot vehicle routing problem with replenishment based on LocalSolver. Int. J. Ind. Eng. Comput. **6**(1), 81–98 (2015)