Execution Time Experiments to Solve Capacitated Vehicle Routing Problem

Adriano S. Silva[1,2,3,4,5(✉)], José Lima[1,3], Ana I. Pereira[1,3], Adrián M. T. Silva[3,4], and Helder T. Gomes[2,3]

[1] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
{adriano.santossilva,jllima,apereira}@ipb.pt
[2] Centro de Investigação de Montanha (CIMO), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
htgomes@ipb.pt
[3] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
adrian@fe.up.pt
[4] Laboratory of Separation and Reaction Engineering - Laboratory of Catalysis and Materials (LSRE-LCM), Faculty of Engineering, University of Porto, Porto, Portugal
[5] ALiCE - Associate Laboratory in Chemical Engineering, Faculty of Engineering, University of Porto, Porto, Portugal

Abstract. Studies dealing with route optimization have received considerable attention in recent years due to the increased demand for transportation services. For decades, scholars have developed robust algorithms designed to solve various Vehicle Routing Problems (VRP). In most cases, the focus is to present an algorithm that can overcome the shortest distances reported in other studies. On the other hand, execution time is also an important parameter that may limit the feasibility of the utilization in real scenarios for some applications. For this reason, in this work, a Guided Local Search (GLS) metaheuristic available in open-source OR-Tools will be tested to solve the Augerat instances of Capacitated Vehicle Routing Problems (CVRP). The stop criterion used here is the execution time, going from 1 s (standard) to 10 s, with a last run of 360 s. The numerical results demonstrate that increasing the execution time returns significant improvement in distance optimization. However, the optimization found considering high execution times can be expensive in terms of time, and not feasible for situations demanding faster algorithms, such as in Dynamic Vehicle Routing Problems (DVRP). Nonetheless, the GLS has proven to be a versatile algorithm for use where distance optimization is the main priority (high execution times) and in cases where faster algorithms are required (low execution times).

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/05757/2020, UIDP/05757/2020, UIDB/00690/2020, UIDB/50 020/2020, and UIDB/00319/2020. Adriano Silva was supported by Doctoral Grant SFRH/BD/151346/2021 financed by the Portuguese Foundation for Science and Technology (FCT), and with funds from NORTE 2020, under MIT Portugal Program.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14111, pp. 273–289, 2023.
https://doi.org/10.1007/978-3-031-37126-4_19

Keywords: Route planning · Open-source · Versatile algorithm

1 Introduction

In the current scenario, companies and manufacturing services are constantly facing logistics challenges, one of them associated with delivery time [14]. In this regard, research on Vehicle Routing Problems (VRP) is gaining attention [16,17]. VRP is a fundamental combinatorial problem in distribution, logistics, and transportation. The first problem in this strand reported in the literature dates from 1959 when Dantzig and Ramser solved a truck disposing of problem [5]. Since then, several formulations and solutions have been reported in the literature, considering various scenarios [3]. Since the first formulation, the objective of these problems was always focused on cost reduction via finding the shortest path to execute a determined service [13]. In brief, the problem is used in cases where an agent needs to find the best way to serve different customers with more than one vehicle available to perform the task [26]. To increase the compatibility of the problem with a determined real scenario, constraints are considered.

In this regard, modifications in the original formulation of VRP were made along time, with new formulations arising. The most known formulation considers the capacity of the trucks and customer's demands as constraints to find the shortest path, the Capacitated Vehicle Routing Problem (CVRP) [19]. Another well-studied formulation is the Vehicle Routing Problem with Time Windows (VRPTW), which considers time windows the main constraint for route planning [6]. The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) is another formulation example, considering capacity and time constraints. Other formulations besides the above-mentioned can be found in this work [18]. All the formulations shown so far have one thing in common apart from the objective of reducing traveled distances: they are static. In other words, the data required to execute the algorithm is known before the decision-making procedure, and the route will not change once planned [10]. In recent years, the Dynamic Vehicle Routing Problem (DVRP) has received attention due to the more realistic nature of algorithms operation [17]. In this approach, the customers' set is not completely known once the truck leaves the central depot to execute the task, and the next stop is dynamically planned based on demands. For this application, the execution time of algorithms is precious since the stop times of trucks can range from seconds to minutes. Despite the difference regarding decision-making for route planning, algorithms developed for static approach can easily be adapted to execute the task in a dynamic environment.

Therefore, this work will assess the Guided Local Search (GLS) metaheuristic's performance in OR-Tools, considering different execution times as stop criteria. To perform this evaluation, Augerat dataset [2], available in the literature and used by other authors, was considered. Most authors dealing with algorithm development are only focused on optimization performance without considering the feasibility of their algorithm in terms of execution time. In light of the necessity to develop faster solvers, the performance evaluation will consider not

only the algorithm's ability to find the optimal and shortest distance. Still, it will consider the execution time as well. The rest of the paper will bring relevant literature on algorithms development to solve CVRP considering execution times in Sect. 2, along with the applications of this problem for real life. Section 3 shows the method adopted to perform the study, Sect. 4 presents the numerical results and discussions, and Sect. 5 brings the conclusions and future work.

2 Related Literature and Applications

The relevant literature was gathered upon searching in Web of Science and Scopus for works dealing with "Capacitated Vehicle Routing Problem" and "Execution Time". The word cloud with the most relevant author's keywords is shown in Fig. 1 for both databases.



(A) (B)

Fig. 1. 50 most relevant keywords in (A) Web of Science and (B) Scopus database.

The word clouds obtained with literature analysis using data gathered from the databases shows similar tendencies for studies in this field. In this case, metaheuristic algorithms have a significant presence in both clouds, which is comprehensible due to the fast response achieved using this class of algorithms to solve CVRP. In the next sections, the strategies used by works in this strand will be reviewed, along with real-case applications of CVRP.

2.1 Algorithm's Execution Time in Capacitated Vehicle Routing Problems

The VRP is the most important problem encountered at the operational level for transportation systems. The family of problems in this strand is classified as NP-hard problems, meaning the solution by exact methods can take a long time depending on the size of the problem considered. Since the first formulation in

1959, the family of VRP problems has increased through time, with the emergence of more formulations approaching different aspects of real transportation systems [15]. Improving the algorithm's performance through execution time has been recognized as an important task in recent years. Authors exploring this particular path often explore the hybridization of a particular algorithm as a strategy, considering the assembly of a new algorithm changing the time-consuming step for another mechanism available in other algorithms. This strategy is based on the identification of time-consuming steps during the execution of the algorithm and modification of this stage with the aid of other mechanisms used by other algorithms. It is important to highlight that this is the only possible way to solve the problem. There are other solutions as well, such as using parallel computation technologies.

For instance, Ant Colony Optimization (ACO) has stagnation property, which reduces the exploration of new routes and increases the processing time to return the solution. To tackle this issue, a study has considered developing a hybrid algorithm based on ACO with Crowd Search Algorithm (CSA) to achieve better solutions with reasonable execution time [7]. The authors have considered the Augerat database to run experimental tests to validate their approach. Despite the fast execution time shown by the results (828 to 2374 ms), the authors did not report the execution time obtained with their ACO, which makes it difficult to observe the improvement in execution time.

Some works are devoted to exploring the currently available heuristics to find the one with the best performance for their case of study. Dubois *et al.* [9], for example, have assembled a new model of VRP named Vehicle Routing Problem with Deadlines (VRPD), which includes several constraints from typical formulations. Among the studied heuristics, the authors found that the Best Flowtime Insertion with Order Questioning (BFIOQ) routine can improve by more than 55% with affordable computational time (around 600 ms). More sophisticated heuristics were also used to evaluate the resource dispatch criteria to solve the VRP. The comparison between Greedy Resource Dispatch 1 (GRD1) and Constant Objective Detection Resources Dispatch (CODRD) revealed a better performance of GRD1 compared to CODRD. However, despite the improved performance, GRD1 can result in infeasible situations for real scenarios, raising questions regarding the usage of this algorithm in a final tool.

Most solutions for time improvement in execution times are related to algorithm modification to increase the feasibility. However, other strategies must be explored to utilize the current hardware to run algorithms. Abdelatti *et al.* [1] employed parallel computation to solve a large instance CVRP taking advantage of the special CUDA GPU architecture on an NVIDIA GPU grid. In brief, the authors distributed the array elements over the GPU grid and used the GPU kernel functions to find high-quality solutions with acceptable execution times. To validate their strategy to speed up the execution of the algorithm, the authors have compared the solutions obtained in GPU with the ones achieved via CPU. The numerical results obtained by the authors show that the execution time of the GPU is 52.7, 162.6, 368.6, 321.5, 339, 454.4 times faster than the CPU

for problems comprised of 21, 31, 40, 50, 60, and 76 nodes. The gap increase observed for the comparison between the two processing agents is related to the capacity of the GPU to handle problems with a higher number of nodes, which is not observed for the CPU. In other words, execution times are not affected similarly according to the number of nodes in the CPU and GPU.

2.2 Applications of CVRP

Over the last decades, the utilization of optimization packages based on mathematical programming for the logistics sector has increased significantly. The increased number of publications has proven the ability of computerized processes to save resources up to 20% in the transportation sector. For instance, most works dealing with optimization in this context often refer to a real-case situation to explain the motivations behind the study. The applications of CVRP extend to any situation in which it is possible to identify a set of customers with associated demands requiring service and a set of vehicles that needs to serve the customers.

Silva *et al.* [22] have studied route optimization in municipal solid waste collection. In their work, three metaheuristics available in open-source OR-Tools were studied to optimize waste collection tasks, considering execution time as an important parameter. For instance, several works are devoted to this variant of CVRP, which has enabled the emergence of a new class of VRPs named Capacitated Waste Collection Problems (CWCP). The formulations, in this case, are similar to CVRP ones, with the addition of new constraints related to the service addressed. The obnoxious effect of organic waste, heterogeneous fleets collecting different types of waste, and multi-compartment trucks collecting more than one type of waste are examples of considerations taken into account. Additionally, works focus on collecting and transporting other residues, mostly from industrial sites.

Another class of real problems emerging in recent years is the Capacitated Electric Vehicle Routing Problem. The increase in electric vehicles fleet and the viability of using those vehicles in cargo transportation systems, especially deliveries, has created this new variant of CVRP. The problem is based on the premise that electric vehicle consumption is directly connected with the weight they are carrying. Several works have been reported on route planning, considering the best delivery path using electric vehicles. The main considerations in this formulation are related to the necessity of those vehicles to recharge during the path and the consideration of a dynamic consumption based on the cargo. For instance, Jia *et al.* [12] proposed a confidence-based bilevel ACO algorithm to solve the problem of delivery service and vehicle charging. The ACO was adapted to solve the delivery problem at the upper level, and a new metaheuristic was used to determine the need to recharge the vehicle (simple enumeration).

Other real-world applications for CVRP are less frequently found in the literature, such as relief systems. Despite fewer studies in this field, the relevance is not diminished. These works are fundamental to studying new strategies that could save lives. The principal difficulty in studying the systems is associated

with the lack of information regarding disasters to model new formulations and explore solutions in simulated environments. Data gathering is not the priority in these situations. Regardless, the study by Dubois *et al.* [9] evaluates the execution time and efficiency of different algorithms in the context of flooding crises.

3 Methodology

This work aims to analyze the influence of execution time in the performance of GLS, available in OR-Tools, to solve the Augerat instances of CVRP. Previous works demonstrated that Guided Local Search (GLS) is the best heuristic for solving CVRP [20,22]. For this reason, we used GLS to solve the Augerat instances of CVRP, defining the execution time as stop criteria. In these instances, $(n-1)$ represents the number of nodes, and k is the number of trucks considered to find the optimized path. In most real-life cases where VRP emerges as problem seeking solution, execution time is a key tool to present feasible solutions. In this context, the execution times considered in this work were chosen in the time interval $t = [1, 10]$; with $\Delta t = 1$. By solving the instances, a comparison of the gap between execution times and solution improvement was made to demonstrate the applicability of this algorithm in cases that require faster solutions and also for cases demanding high performance. The solutions for each instance were further gathered in a single file with the aid of the pandas' package, and the complete algorithm was written in python. Figure 2 brings an overview of the complete work.

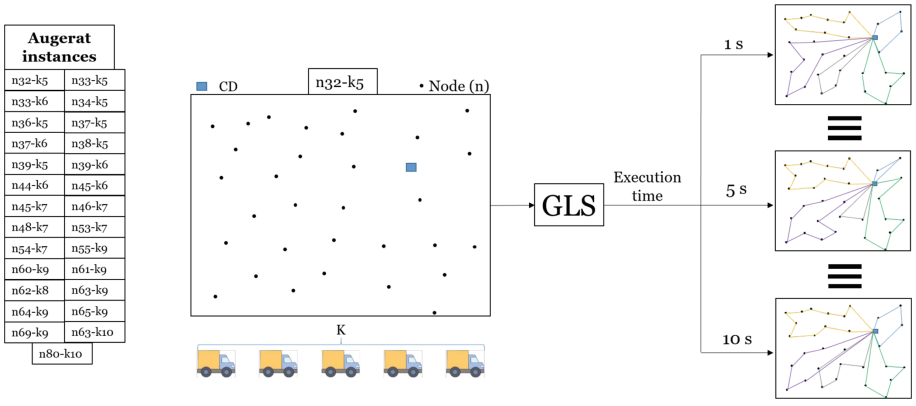

Fig. 2. Representation of the study performed.

3.1 Capacitated Vehicle Routing Problem

The formulation presented here is based on the literature [21]. In this problem, the customers have requested services of pick-up or deliveries, and the demands are deterministic and known in advance. The fleet of vehicles ascribed for the task can be homogenous or heterogeneous, directly impacting vehicles' capacity and route planning. The objective is to find the shortest routes and distances to serve all the customers. The problem can be described formally as a graph theoretic problem. Let $G = (V, A)$ be a complete graph, for which $V = \{0, 1, 2, \ldots, n\}$ is the vertice set, and $A = \{(i, j) : i \neq j \text{ for } i, j = 0, \ldots, n\}$ represents the set of arcs. Each arc $(i, j) \in A$ is associated with a non-negative cost $c_{i,j}$ that expresses the travel cost spent from vertex i to j. In this representation, node 0 represents where vehicles depart and return at the end of the routes (central depot). Correlated to a location l, there is a demand d_l, defined as null to the central depot. The problem will be solved assuming that the set of trucks k is homogeneous and trucks have the same capacity Q. Therefore, the assumptions summarizing the problem are:

- The number of trucks is supposed to be limited (K) and homogeneous.
- Every location is visited once by exactly one truck.
- Trucks have limited capacity Q.
- Trucks must empty their load in the central depot once full capacity is reached and remain there.
- Full demand cannot exceed the total capacity of the fleet.
- Each route begins and ends at the central depot.
- The routes are calculated with all information available in a static approach.
- Demands from customers cannot be split.

The mathematical formulation that was considered to solve the CVRP described above was written based on the general formulation in the literature [25]. The visit sequence for the routes for each truck was found based on the definition of decision variables as shown next:

$$\text{Minimize} \sum_{l \in N} \sum_{j \in N} c_{lj} \sum_{k \in K} x_{ljk} \tag{1}$$

subject to:

$$\sum_{l \in N} d_l y_{lk} \leq Q, \quad \forall k \in K \tag{2}$$

$$\sum_{k \in K} y_{0k} \leq K \tag{3}$$

$$\sum_{j \in N} x_{ljk} = \sum_{j \in N} x_{jlk} = y_{lk}, \quad \forall l \in N, \forall k \in K \tag{4}$$

$$\sum_{k \in K} y_{lk} = 1, \quad \forall l \in N \setminus \{0\} \tag{5}$$

$$u_{jk} \geq u_{lk} + Q(x_{ljk} - 1) + d_j, \quad \forall l, j \in N \setminus \{0\}, l \neq j, \text{ such that } d_l + d_j \leq Q, \forall k \in K \tag{6}$$

$$d_l \leq u_{lk} \leq Q, \quad \forall l \in N \setminus \{0\}, \forall k \in K \tag{7}$$

$$x_{ljk} \in \{0, 1\}, \quad \forall l, j \in N, \forall k \in K \tag{8}$$

$$y_{lk} \in \{0, 1\}, \quad \forall l \in N, \forall k \in K \tag{9}$$

The decision variables are defined as follows:

$$x_{ljk} = \begin{cases} 1, \text{ if truck } k \text{ visits location } j \text{ after location } l \\ 0, \text{ otherwise.} \end{cases}$$

$$y_{lk} = \begin{cases} 1, \text{ if truck } k \text{ visits location } l \\ 0, \text{ otherwise.} \end{cases}$$

The objective function represented in (1) enables minimizing the total cost to serve all the customers, respecting the stated constraints. Constraint (2) is the main constraint in this problem, stating that the sum of demands of vertices visited in a route is equal or smaller to the truck's capacity. Constraints (3) and (4) are related to vehicle circulation, ensuring that every truck leaves the depot only once and keeps performing the service as long as the capacity constraint is not violated (continuity of the route). Constraint (5) guarantees that every location is visited exactly once by only one truck, illustrating that demands are not split in this formulation. The constraint in (6) ensures all routes are connected to the central depot. The next constraint, (7), is connected to the previous and represents the truckload k after visiting the location l (u_{lk}). At last, constraints (8) and (9) are responsible to specify the domain of decision variables x_{ljk} and y_{lk}.

3.2 Guided Local Search

Guided Local search is a successful metaheuristic algorithm for solving many problems since the latest '90s. The algorithm is a penalty-based implementation whose efficiency is penalizing some solution features along the iterations.

The mathematical formulation presented here was based on the literature [27]. Denoting by I_i the following indicator function and assuming minimization:

$$I_i(x) = \begin{cases} 1, \text{ if solution } x \text{ has feature } i \\ 0, \text{ otherwise.} \end{cases}$$

Let p_i be a penalty function associated with a feature i of the solution, denoting as f the original objective function. The GLS will use the following augmented objective function g:

$$g(x) = f(x) + \lambda \sum_i I_i(x) \times p_i \tag{10}$$

The GLS will use the function shown in (10) to find new solutions. The penalty factor (λ) can be used to tune the search to find completely different solutions (high λ value, diversification) or similar solutions (low λ value, intensification). Penalties start with a null value and are incremented by 1 with each local optimum found. The mechanism behind penalties in GLS is defined upon the assessment of the utility of a feature, being the penalization attributed in the case of large utilities. The utility function is calculated as follows for a feature i in a solution x:

$$u_i(x) = I_i(x) \times \frac{c_i(x)}{1 + p_i} \tag{11}$$

In Eq. 11, the value $c_i(x)$ is the cost associated with a feature i in a solution x. If the feature i does not apply to a solution x, its utility will be defined as 0 ($I_i(x) = 0$). On the other hand, the utility $u_i(x)$ is proportional to the cost $c_i(x)$ of the feature i in solution x. However, the utility tends to disappear if a determined feature is often penalized because a feature that appears with frequency in a local optimum might be part of a good solution. Based on the general formulation of GLS introduced here, the algorithm used to solve the CVRP in this work is based on the following description. Denoting by $d_{i,j}$ the cost for traveling the arc (i, j), this will be considered as the cost of the objective function for that arc, and then $c_{ij} = d_{i,j}$. With this in mind and considering the general definition presented above, the augmented objective function of the GLS here used will be:

$$g(x) = \sum_{(i,j)} d_{i,j}(x) + \lambda \sum_{(i,j)} c_{ij}(x) \times p_{i,j} \times I_{ij}(x) \tag{12}$$

Penalty factor λ starts with value 0.1 by default.

4 Numerical Results

This section will discuss the results obtained for route optimization using GLS available in OR-Tools to solve the Augerat CVRP dataset. The main goal is to

determine the best-case execution time feasible in the context of dynamic scenarios of CVRP and also evaluate the algorithm performance for cases demanding for higher performance regarding distance shortening. In this regard, the execution time should be as low as possible. For this reason, here we tested the performance of GLS to solve the Augerat instances considering as criteria stop execution times between 1 and 10 s.

4.1 Optimal Distances

The Augerat instance considered for this work comprises 27 datasets, each exploring a different scenario. The most important difference between each dataset is the number of customers and trucks since the truck's capacity is the same for all datasets ($Q = 100$). For all sets, increasing the execution time from 1 to 360 s returned the best improvement in distance optimization. Comparing the results obtained using GLS with the literature revealed that GLS can overcome the literature in 62.96% of the datasets. Figure 3 brings the comparison between optimal values and the results obtained by GLS in the maximum execution time considered here.

The sets for which the distance obtained was worse than the optimal had a ratio of nodes/truck above 7, which reveals that the algorithm would need more execution time to achieve better results. Although a good result was observed in increasing the execution time from 1 to 360 s, it is important to consider the need to study the execution times as well. For instance, the maximum execution time considered in this study would not be feasible for applications in real and dynamic scenarios.

4.2 Execution Time and Optimal Distances

The best result observed for all tests was obtained with the maximum execution time. This can be explained due to the higher time available for the algorithm to process the information and search for the optimal solution, escaping local minima. On the other hand, it is important to evaluate the improvement in the solution according to the execution time. Figure 4 brings the sum of optimal distances found for each set considering execution times from 1 to 10 s and the execution time of 360 s.

The summary of the optimal distances found according to the execution time shows how powerful this parameter can be in tuning the algorithm's performance. However, the distance shortening by increasing the execution time is clearly not linear, as reported in other works that utilize metaheuristic algorithms for route optimization. For this reason, it is important to consider the algorithm's future application to choose the proper execution time better. For instance, if one considers it more important to achieve better solutions no matter the execution time, a higher execution time would return better solutions. On the other hand, considering dynamic scenarios (i.e. dynamic route planning in waste collection) would demand faster algorithms since time is also precious in these activities. For instance, a deeper evaluation of how execution time impacts the algorithm's

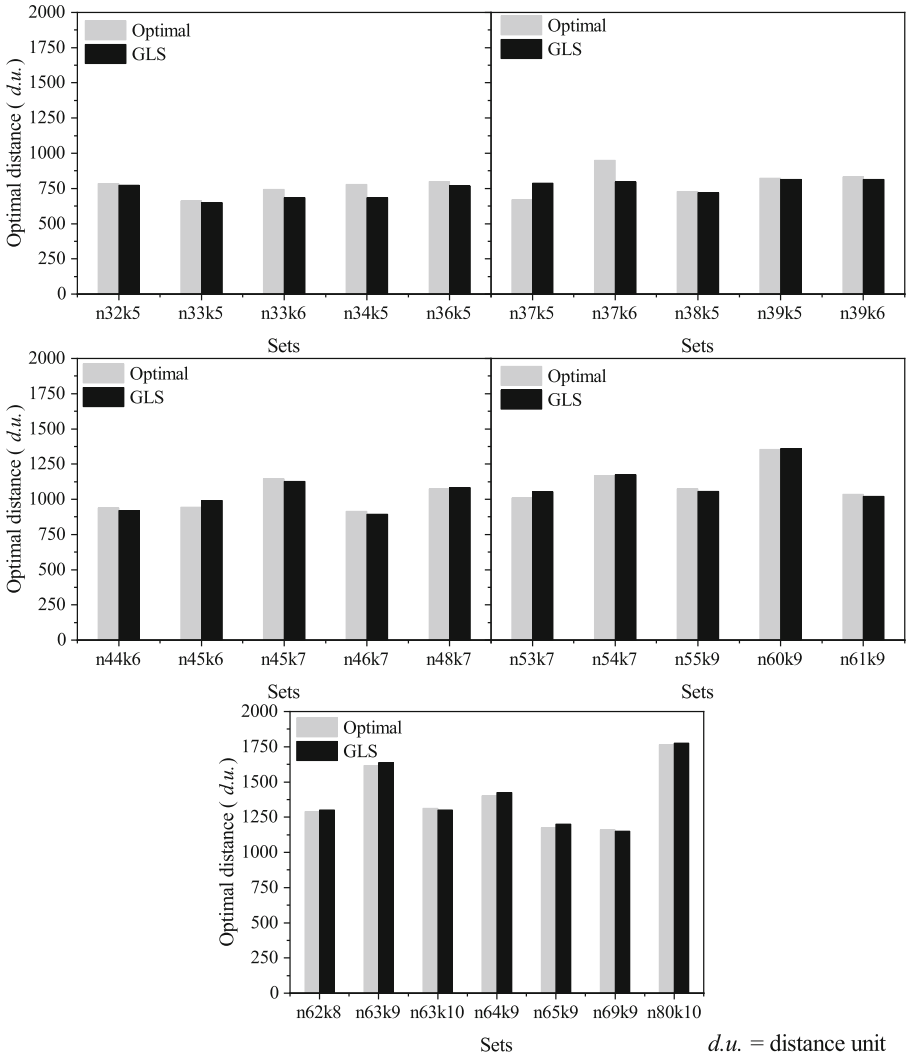

Fig. 3. Best results obtained with GLS compared to optimal reported on the literature

.

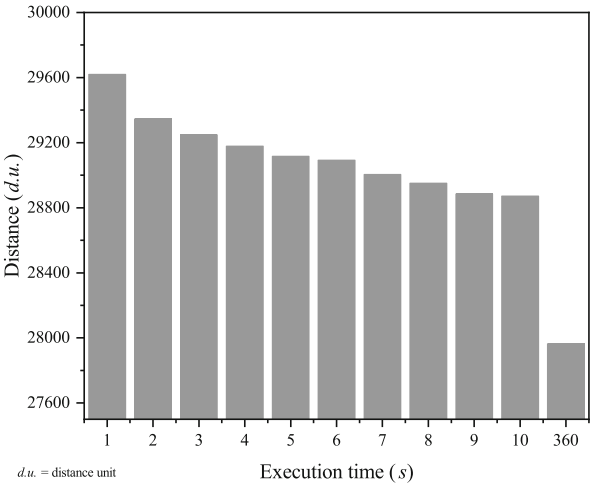

Fig. 4. Sum of optimal distance found for Augerat instances according to the execution time used (time interval in the X-axis is not proportional to the actual time interval).

performance is expressed in Fig. 5a showing the percentage of optimal distances found according to the execution time. Additionally, an evaluation of distance improvement in consonance with execution time increase, based on the standard execution time (1s), is expressed in Fig. 5b.

The evolution of better results compared to the literature according to the execution time considered to run the algorithm was already expected due to the observed differences reported in previous sections. Both summarized results in Figs. 4 and 5a have demonstrated that increasing the execution time from 1 to 360 s returns more significant optimization. However, Fig. 5b demonstrates that this improvement is expensive in execution time. For instance, the high execution time would not be feasible nor efficient in situations where time is more valuable than distance shortenings. For instance, increasing the execution time from 1 to 360 s returned an average improvement of less than a unit (0.11 d.u./s).

Some studies reported a correlation between the performance of their algorithm and the set's tightness [8,11]. The parameter is a correlation between the total demand and the total capacity of the fleet ascribed to perform the transportation task. Figure 6a shows the correlation between the set's tightness and optimization achieved upon considering the maximum execution time. The distance savings achieved with the highest execution time were also compared to the systems' size, and the result is shown in Fig. 6b. The result demonstrates that GLS optimization results achieved with increased execution time correlate poorly with the set's tightness. However, it is important to highlight that the tightness interval in Augerat instances is low (0.82-0.94), which can explain the low correlation between the parameters. On the other hand, improvement in distance optimization seems more related to the size of the system considered.

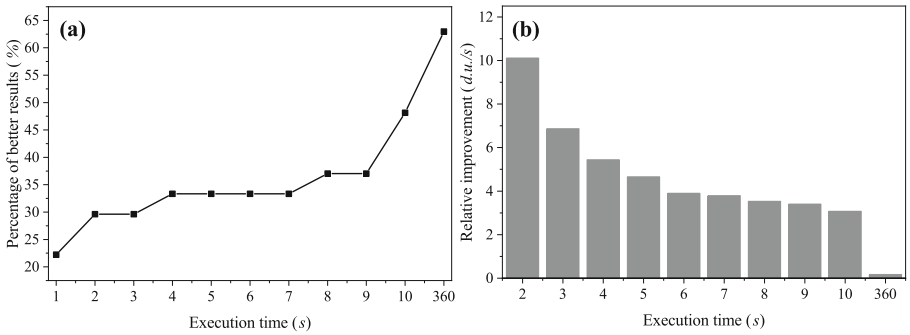

Fig. 5. (a) Percentage of better results achieved with GLS according to the execution time and (b)optimization performance by increasing execution time from standard value (1 s). Time interval in the X-axis is not proportional to the actual time interval.

For instance, GLS algorithms can achieve higher distance savings increasing the execution time in bigger systems.

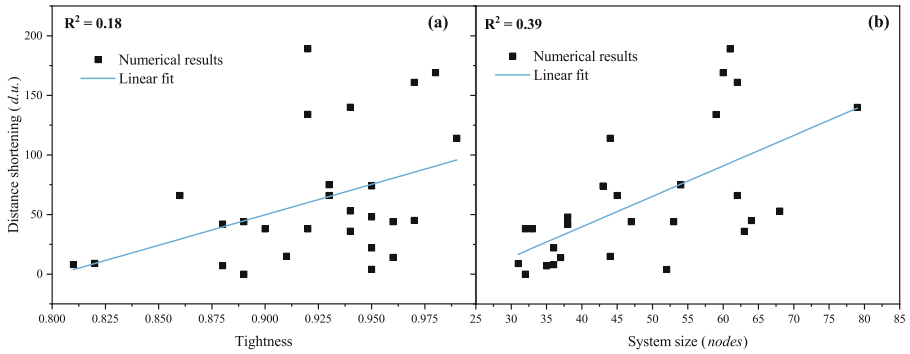

Fig. 6. Correlation between distance shortening and (a) tightness, (b) system size

4.3 Environmental Analysis in Distance Savings

Most trucks and vehicles used in the fleets are moved by petroleum-based fuels responsible for emitting gaseous fractions of environmentally harmful compounds, such as SO_x, CO_x, and NO_x. In Europe, the fleet of trucks represents around 4% of the total number of vehicles on roads, but they are responsible for 25% of the total CO_2 emission. Several studies have been devoted to analyzing the investments required to reduce emissions by trucks [4,23,24]. One alternative is route optimization since traveling shorter distances will emit fewer gases into the atmosphere. Recent studies and reports have also demonstrated that

urban delivery trucks with a 4×2 axle configuration (4-UD) emitted, on average, 307 gCO_2/t-km. For instance, considering the distance optimization achieved upon increasing the execution time would return CO_2 emissions shown in Fig. 7. For this calculation, *km* was considered as distance unit, *kg* as capacity and demand units.

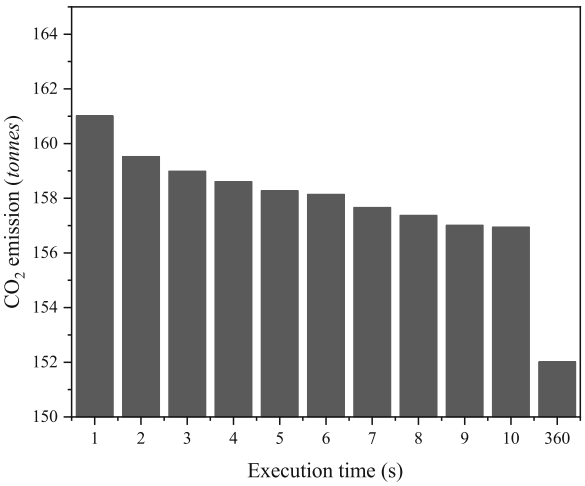

Fig. 7. CO_2 emission estimate according to the algorithm's execution time. Time interval in the X-axis is not proportional to the actual time interval.

The result shown in Fig. 7 has the same tendency observed for optimized distances shown in Fig. 4, as expected. Representing the results in terms of CO_2 emission is fundamental to increasing awareness of how powerful route optimization can be to spare the environment from more pollution. For instance, in a real scenario, increasing the execution time from 1 to 10 s could result in up to 29.91 tonnes less CO_2 being thrown into the atmosphere. The value increases even more, considering the highest execution time used here (66.45 tonnes). On the other hand, it is also important to remember that the optimal execution time must be defined to meet the application scenario of the algorithm.

5 Conclusions and Future Work

In this work, GLS available in open-source OR-Tools have demonstrated the capacity to solve the Augerat instances considering different execution times. The algorithm has demonstrated a significant increase in distance savings upon increasing the execution time. Most importantly, the results shown here regarding the algorithm's performance in solving the Augerat instances have proven the versatility of the algorithm. For instance, increasing the execution time from 1 to 10 s would not affect the viability of employing the tool in a real scenario

of dynamic route planning. One example could be the dynamic route planning in waste collection, for which the route can be planned during truck stops to perform the collection service. In this regard, the execution time of 10 s would not delay the collection task and could significantly improve the result considering the standard execution time. Furthermore, if the main purpose is distance shortening, the algorithm can also be tuned by increasing the execution time to return more improvements in distance shortening. We anticipate the tool being studied for a diverse range of services in the transportation sector.

References

1. Abdelatti, M.F., Sodhi, M.S.: An improved GPU-accelerated heuristic technique applied to the capacitated vehicle routing problem. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pp. 663–671 (2020). https://doi.org/10.1145/3377930.3390159
2. Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A., Rinaldi, G.: Computational results with a branch and cut code for the capacitated vehicle routing problem (1995)
3. Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I.: The vehicle routing problem: state of the art classification and review. Comput. Indust. Eng. **99**, 300–313 (2016). https://doi.org/10.1016/j.cie.2015.12.007
4. Breed, A.K., Speth, D., Plötz, P.: CO_2 fleet regulation and the future market diffusion of zero-emission trucks in Europe. Energy Policy **159**, 112640 (2021). https://doi.org/10.1016/j.enpol.2021.112640
5. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manage. Sci. **6**(1), 80–91 (1959). https://doi.org/10.1287/mnsc.6.1.80
6. Desaulniers, G., Errico, F., Irnich, S., Schneider, M.: Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. **64**(6), 1388–1405 (2016). https://doi.org/10.1287/opre.2016.1535
7. Dhanya, K.M., Kanmani, S., Hanitha, G., Abirami, S.: Hybrid crow search-ant colony optimization algorithm for capacitated vehicle routing problem. In: Zelinka, I., Senkerik, R., Panda, G., Lekshmi Kanthan, P.S. (eds.) ICSCS 2018. CCIS, vol. 837, pp. 46–52. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-1936-5_5
8. Dohn, A., Rasmussen, M.S., Larsen, J.: The vehicle routing problem with time windows and temporal dependencies. Networks **58**(4), 273–289 (2011). https://doi.org/10.1002/net.20472
9. Dubois, F., Renaud-Goud, P., Stolf, P.: Capacitated vehicle routing problem under deadlines: an application to flooding crisis. IEEE Access **10**, 45629–45642 (2022). https://doi.org/10.1109/ACCESS.2022.3170446
10. Haghani, A., Jung, S.: A dynamic vehicle routing problem with time-dependent travel times. Comput. Oper. Res. **32**(11), 2959–2986 (2005). https://doi.org/10.1016/j.cor.2004.04.013
11. Hannan, M., Akhtar, M., Begum, R., Basri, H., Hussain, A., Scavino, E.: Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm. Waste Manag. **71**, 31–41 (2018). https://doi.org/10.1016/j.wasman.2017.10.019. https://www.sciencedirect.com/science/article/pii/S0956053X17307675

12. Jia, Y.H., Mei, Y., Zhang, M.: Confidence-based ant colony optimization for capacitated electric vehicle routing problem with comparison of different encoding schemes. IEEE Trans. Evol. Comput. **26**(6), 1394–1408 (2022). https://doi.org/10.1109/TEVC.2022.3144142
13. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. Eur. J. Oper. Res. **189**(2), 293–309 (2008). https://doi.org/10.1016/j.ejor.2007.05.055
14. Mehmood, T.: Does information technology competencies and fleet management practices lead to effective service delivery? Empirical evidence from e-commerce industry. Int. J. Technol. Innov. Manag. (IJTIM) **1**(2), 14–41 (2021). https://doi.org/10.54489/ijtim.v1i2.26
15. Mohammed, M.A., et al.: Solving vehicle routing problem by using improved k-nearest neighbor algorithm for best solution. J. Comput. Sci. **21**, 232–240 (2017). https://doi.org/10.1016/j.jocs.2017.04.012
16. Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G.: The electric vehicle routing problem with nonlinear charging function. Transport. Res. Part B: Methodol. **103**, 87–110 (2017). https://doi.org/10.1016/j.trb.2017.02.004
17. Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. Eur. J. Oper. Res. **225**(1), 1–11 (2013). https://doi.org/10.1016/j.ejor.2012.08.015
18. Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Comput. Oper. Res. **34**(8), 2403–2435 (2007). https://doi.org/10.1016/j.cor.2005.09.012
19. Praveen, V., Keerthika, P., Sivapriya, G., Sarankumar, A., Bhasker, B.: Vehicle routing optimization problem: a study on capacitated vehicle routing problem. Mater. Today: Proceed. **64**, 670–674 (2022). https://doi.org/10.1016/j.matpr.2022.05.185
20. Silva, A.S., et al.: Solving a capacitated waste collection problem using an open-source tool. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Garau, C. (eds.) Computational Science and Its Applications - ICCSA 2022 Workshops. ICCSA 2022. Lecture Notes in Computer Science, vol. 13378, pp. 140–156. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10562-3_11
21. Silva, A.S., et al.: Dynamic urban solid waste management system for smart cities. In: Simos, D.E., Rasskazova, V.A., Archetti, F., Kotsireas, I.S., Pardalos, P.M. (eds.) Learning and Intelligent Optimization. LION 2022. Lecture Notes in Computer Science, vol. 13621, pp. 178–190. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-24866-5_14
22. Silva, A.S., et al.: Capacitated waste collection problem solution using an open-source tool. Computers **12**(1), 15 (2023). https://doi.org/10.3390/computers12010015. https://www.mdpi.com/2073-431X/12/1/15
23. Siskos, P., Moysoglou, Y.: Assessing the impacts of setting CO_2 emission targets on truck manufacturers: a model implementation and application for the EU. Transport. Res. Part A: Policy Pract. **125**, 123–138 (2019). https://doi.org/10.1016/j.tra.2019.05.010
24. Tian, J., Yang, D., Zhang, H., Liu, L.: Classification method of energy efficiency and CO_2 emission intensity of commercial trucks in china's road transport. Procedia Eng. **137**, 75–84 (2016). https://doi.org/10.1016/j.proeng.2016.01.236. Green Intelligent Transportation System and Safety
25. Toth, P., Vigo, D.: The vehicle routing problem. SIAM (2002). https://doi.org/10.1137/1.9780898718515

26. Vidal, T., Laporte, G., Matl, P.: A concise guide to existing and emerging vehicle routing problem variants. Eur. J. Oper. Res. **286**(2), 401–416 (2020). https://doi.org/10.1016/j.ejor.2019.10.010
27. Voudouris, C., Tsang, E.P., Alsheddy, A.: Guided local search. In: Gendreau, M., Potvin, JY. (eds.) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 146, pp. 321–361. Springer, MA (2010). https://doi.org/10.1007/978-1-4419-1665-5_11