Solving Capacitated Time-Dependent Vehicle Routing Problem

Filip Ribić, Tomislav Erdelić, Tonči Carić, Martina Erdelić
Faculty of Transport and Traffic Sciences, Department of Intelligent Transport Systems
University of Zagreb
Vukelićeva 4, 10000, Zagreb, Croatia
{fribic, terdelic, tcaric, merdelic}@fpz.unizg.hr

Abstract. *A vehicle routing problem aims to determine a set of vehicle routes to deliver goods to the customers. Most of the variants of the problem consider static traffic conditions, while in the real world, travel times depend on the departure time. In this paper, a capacitated time-dependent vehicle routing problem is observed, which considers time-dependent travel times between the customers, and delivery with vehicles that have limited load capacity. To solve the problem, an iterative local search metaheuristic is applied that couples the ruin-recreate principle with the common local search procedures. Several scenarios were developed to analyze the impact of time-dependent travel times on the solution quality in terms of the total travel time and customer configuration.*

Keywords. time-dependent vehicle routing problem, total distance traveled, total time, heuristics

1 Introduction

Delivering goods to customers is a complex logistic process. In it is important to deliver the goods on time and meanwhile to reduce the total delivery costs in terms of total travel time and total traveled distance (Toth and Vigo, 2001). The key question in such process is to determine which routes to choose and how different route choices affect the change in total travel time and total distance traveled. Vehicles used in such processes are often limited by the load capacity in terms of vehicle cargo volume or maximum vehicle load-bearing. Another important aspect of the delivery are traffic conditions, or more precisely time spent on the route. Due to the traffic conditions, route travel time depends on the departure time, especially in large urban centres where traffic congestions occur typically in rush periods (Erdelić et al., 2021). As traffic congestions significantly increase the travel time, the total route travel time can change significantly, and thus increase the total delivery costs (up to 3%), (Kellner, 2016).

In the literature, the well-known problem of delivering goods is modelled as NP-hard Vehicle Routing Problem (VRP), (Toth and Vigo, 2001). The goal of the VRP is to determine the least-cost delivery routes from a warehouse to customers, subject to side constraints (Laporte, 1992). These side constraints often resemble real-life limitations of the problem: limited vehicle load capacity (Capacitated VRP, CVRP), pickup and delivery of goods (VRP with Pickup and Delivery, VRPPD), customer delivery time windows (VRP with Time Windows, VRPTW), heterogeneous fleet, etc., (Erdelić and Carić, 2019). Driven by the impact of traffic congestions on routing decisions, a Time-Dependent VRPW (TD-VRPTW) was formulated (Figliozzi, 2012), which considers time-dependent travel times on the road network. The delivery period is discretized into smaller time periods, usually representing rush and non-rush periods. In non-rush periods typically, the free flow speed is considered, while in the rush periods, the speed significantly decreases. These time periods and speeds are often determined by knowledge. On the other hand, there are research papers that present the exact time periods and speed values derived from large historical vehicle tracking data (Erdelić et al., 2021).

In this paper, the Time-Dependent VRP (TD-VRP) is observed, which can be considered as a subproblem of both TD-VRPTW and CVRP. In this problem, vehicles are limited by their load capacity, travel at time-dependent travel times and do not have to visit a customer within its time window (only the warehouse working times are respected). As VRP is an NP-hard problem, the TD-VRP is also NP-hard. This means that exact procedures are only capable of solving problems with a relatively low number of customers (up to 50 customers), and most often, the metaheuristics and heuristics are applied to solve larger problems (Vidal et al., 2013). Over the years, a vast number of heuristic and metaheuristic procedures have been applied to efficiently solve the VRP problems (Erdelić and Carić, 2019). Usually, the best results are achieved with a metaheuristic procedure that guides the search, such as tabu search, simulated annealing, large neighborhood search, variable neighborhood search, genetic algorithm, ant colony, iterative local search (Vidal et al., 2013). The problem-specific heuristic procedures that intensify the search are used within the metaheuristic procedure, such as local search operators (relocate, exchange, 2-Opt, etc.), nearest neighbor heuristic, Clark-and-Wright savings method, etc., (Erdelić and Carić, 2019).

Compared to the related research papers of Figliozzi (2012); Carić and Fosin (2020) and Rožić et al. (2015), this paper is one of the few that considers the TD-VRP problem without time windows and its influence on the total route duration and customer configuration; thus, investigating the real-life problems in which customers do not have time windows, and can be visited at any time during the day. The applied Iterative Local Search (ILS) heuristic to solve the problem has not yet been applied on such problem. Additionally, we investigated the impact of different speed values in different time periods on solution quality in the form of several specific scenarios.

The rest of the paper is organized as follows. The description of the problem, together with the test instances, are presented in Section 2. The solution method used to solve the problem is described in Section 3 together with the heuristic for the initial solution, Local Search (LS) operators and ILS. The results of the paper are presented in Section 4. Finally, the conclusion of the paper is given in Section 5.

2 Problem description

In this section, a TD-VRP problem modelled as a Mixed Integer Program (MIP) is described. Additionally, an example of test instances used to test the solution method is presented.

2.1 MIP model

TD-VRP is formulated as a MIP model by Equations 1-7 with the goal of minimizing the overall total travel time. Let $V = \{1, \dots, N\}$ be a set of geographically scattered customers that need to be served. Vertices 0 and $N + 1$ denote the warehouse instances, and every route begins at vertex 0, and ends at vertex $N + 1$ ($V_{0,N+1} = V \cup \{0\} \cup \{N + 1\}$). Graph G is defined as $G = (V_{0,N+1}, A)$, where A is the set of arcs $A = \{(i, j)|i, j \in V_{0,N+1}, i \neq j\}$. The binary variable $x_{ij} \in \{0, 1\}$ (Equation 1) is equal to 1 if arc (i, j) is traversed in the solution, and 0 otherwise. Each vehicle has a load capacity C. Each customer i has a service time s_i and load demand q_i. The warehouse has a working time $[0, l_0]$. Additionally, two more decision variables for customers ($i \in V_{0,N+1}$) are used: τ_i - begin time of service and u_i - remaining load capacity. The arc value $t_{ij}(x)$ represents the time function that determines time needed to traverse the arc with the departure time x. The objective function is given by Equation 2 as the sum of travel times per arcs.

Equations 3 and 4 ensure the arc connectivity of customers, meaning that each customer can have only one entry and one exit arc which results in the constraint that each customer has to be visited only once. Equation 5 ensures travel time feasibility of arcs between customers i and customer j. If arc (i, j) is traversed, then the begin time at customer j, τ_j, has to be equal to the sum of begin time at customer i, τ_i, time-dependent travel time between i and j (depends on the departure time) $t_{ij}(\tau_i + s_i)$, and service time of customer i, s_i. Equation 6 ensure arcs load flow, in similar way as equation 5. Equation 7 ensures that the leaving warehouse instance has a remaining load capacity equal to the vehicle load capacity C.

$$x_{ij} \in \{0, 1\},\ \forall i \in V_0, j \in V_{N+1}, i \neq j \tag{1}$$

$$\min \sum_{i \in V_0} \sum_{j \in V_{N+1}, i \neq j} t_{ij}(\tau_i + s_i)x_{ij} \tag{2}$$

$$\sum_{j \in V_{N+1}, i \neq j} x_{ij} = 1,\ i \in V \tag{3}$$

$$\sum_{i \in V_{N+1}, i \neq j} x_{ji} - \sum_{i \in V_0, i \neq j} x_{ij} = 0,\ j \in V \tag{4}$$

$$\begin{aligned} \tau_i + (t_{ij}(\tau_i + s_i) + s_i)x_{ij} - l_0 \cdot (1 - x_{ij}) = \tau_j, \\ \forall i \in V_0, \forall j \in V_{N+1}, i \neq j \end{aligned} \tag{5}$$

$$\begin{aligned} 0 \leq u_j \leq u_i - x_{ij}(q_i + C) + C, \\ \forall i \in V_0, \forall j \in V_{N+1}, i \neq j \end{aligned} \tag{6}$$

$$u_0 = C \tag{7}$$

The time of one working day is discretized in different time periods in order to take into account traffic congestions that occur in the so-called rush hours. This means that the function of travel time on arc $t_{ij}(\tau_i + s_i)$ is discretized into a finite number of time intervals $k \in \{T_0, \dots, T_K\}$, and to each time period a static speed v_k is assigned. The problem considers that the distance on arc d_{ij} does not change, and the travel time on arc in time period k can be computed as $t_{ij}(k) = d_{ij}/v_k$. The departure time customers i expressed as $\tau_i + s_i$ is used to determine the period k in which the vehicle starts to travel from customer i to customer j.

2.2 Data

To test their methods and compare them to other methods, researchers typically solve the benchmark instances. In this paper, one instance from the well-known Solomon VRPTW test instances is used (Solomon, 1987). Each customer in instance has a location $C(x, y)$ in the Cartesian coordinate system, service time and demand. The warehouse working time consist of opening (0) and closing time (l_0). The distance between the customers is computed by Euclidean

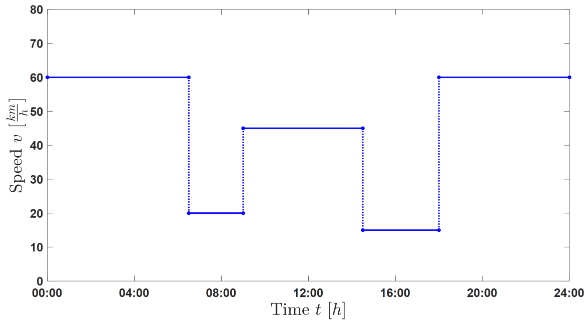


Figure 1: Speeds

distance given by Equation 8. The warehouse working time $[0, l_0]$ is discretized into five-time buckets of equal duration: $[0, 0.2l_0\rangle$, $[0.2l_0, 0.4l_0\rangle$, $[0.4l_0, 0.6l_0\rangle$, $[0.6l_0, 0.8l_0\rangle$, and $[0.8l_0, l_0\rangle$. In periods of congestion, speeds are reduced in order to better represent the actual state of the traffic system, while in other intervals, speeds are closer to the free flow speed. The example of speeds per discretized intervals for one path in one working day is presented in Figure 1. In total, five time periods are observed on the x-axis with corresponding speed values on the y-axis. As it can be seen, during the night and evening, a free-flow speed with the value of 60 km/h is present, while during the morning and afternoon rush hours, the speed drops to 20 km/h, and 15 km/h respectively. Between the rush hour periods, the speed is somewhere in between with the value of 45 km/h.

Used instance contains a subset of 25 customers presented in Figure 2. The customers are presented with black circles, where the size of the circle represents the customer demand, the higher the demand is, the larger the circle is. The warehouse is presented as a red circle. The name of the instance $C101$ indicates that customers are groupped in clusters.

$$d(C_1, C_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (8)$$

The values of distances and travel times on arcs (paths) between customers are commonly stored in 2D matrices, representing the shortest path between each

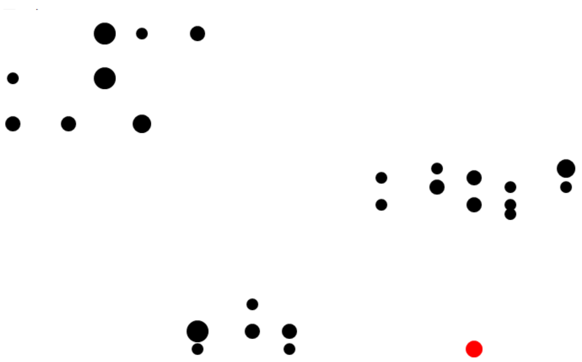


Figure 2: Instance C101_25

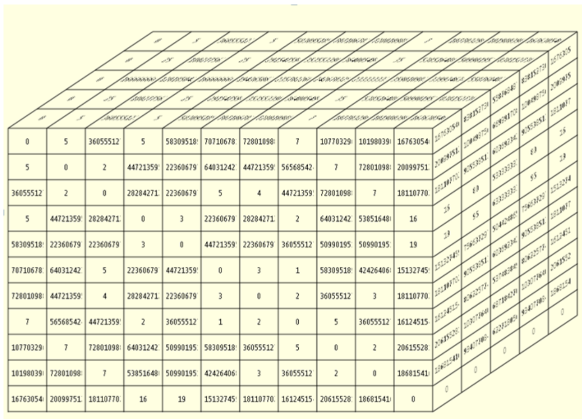


Figure 3: 3D travel time matrix

customer pair (Erdelić et al., 2021). Such matrices are computed in a preprocessing step to reduce the computation time, as computing the shortest path between customer pairs is time-consuming. Here, as the Euclidean distance was used, there is no need for the computation of the matrices, but for the sake of the complicity, the matrices will be used. Including time-dependent travel times results in the 3D travel time matrix, where the third dimension represents the number of discretized time periods. The example of a 3D travel time matrix is presented in Figure 3. For example, a value of five in the first row and second column represents a travel time of five between the warehouse and the first customer in first time period.

3 Methodology

The methodology section consists of three parts: (i) creation of the initial solution, (ii) LS procedure, and (iii) ILS procedure. All applied heuristic methods search in the feasible solution space, which is constrained by the vehicle load capacity and by the total route duration due to the warehouse working hours.

3.1 Initial solution heuristic

The heuristic used for the creation of the initial solution is based on the time-oriented nearest neighbour heuristic (Solomon, 1987). The heuristic in each iteration selects the customer that is the closest to the current customer in terms of the travel time.

The procedure for creating an initial solution is presented by the Algorithm 1. The input to the algorithm is a list of all customers V. First, all customers, except the warehouse, are added to the list of unvisited customers. Next, a new vehicle route is opened, and a warehouse is added to it. Further on, the algorithm searches through all unvisited customers for the unvisited customer who has the lowest arrival time and can be added to the vehicle route, respecting the load capacity of the vehicle and total route duration (enough

Algorithm 1 Heuristic for the creation of the initial solution

Input: List of customers V

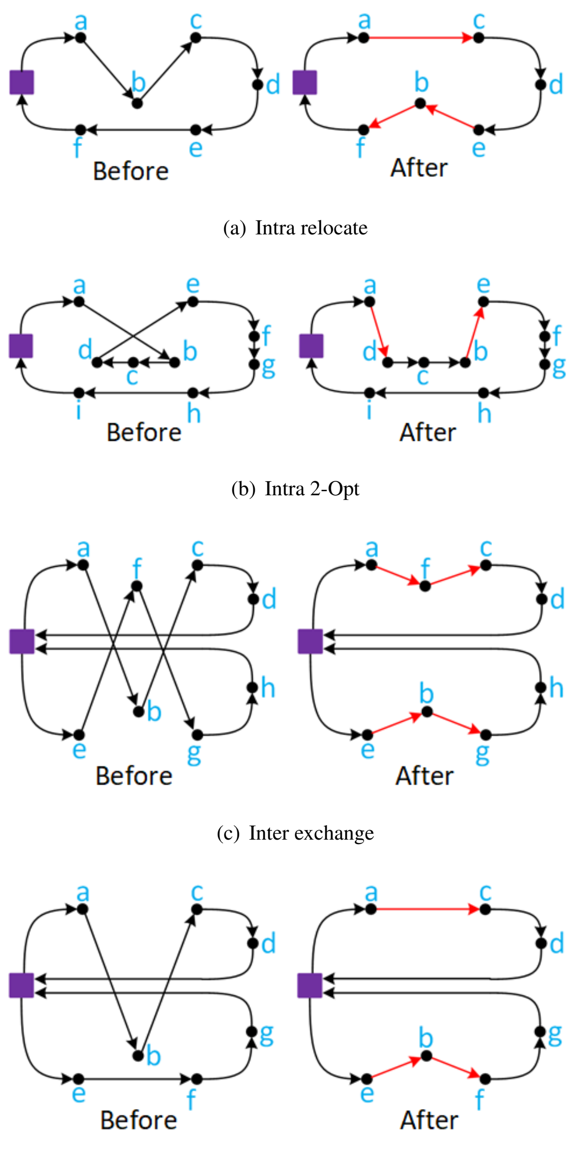
Add all customers V, except warehouse, to the list of unvisited customers
Open a new route, add warehouse to it and set the warehouse as the current customer
while there are no unvisited customers **do**
  Find the timely closest unvisited customer c_{clos} to the current customer
  if load capacity and the total duration of the vehicle route is not violated by adding a customer c_{clos} **then**
    Add customer c_{clos} to the current vehicle route
    Remove customer c_{clos} from the list of unvisited customers
    Reduce the available load capacity of current vehicle
  else
    Close the current vehicle route by adding a warehouse at the end
    Open a new route, add warehouse to it and set the warehouse as the current customer
  end if
end while

time to return to the depot) is not exceeded. In case in which the observed customer exceeds the capacity of the vehicle or the total route duration, the vehicle is returned to the warehouse, and a new vehicle is opened. This means that the warehouse is added after the last visited customer and a new route with a warehouse is opened. By adding a customer to the vehicle, the capacity of the vehicle increases by it's demand. If a customer can be added to the vehicle route, it is removed from the list of unvisited customers and saved to the list of visited customers of the current vehicle. The process is repeated until all customers are visited. The final result of the initial solution is the number of vehicles needed to visit all customers, the configuration of customers per vehicle routes with the total traveled time and the total traveled distance.

3.2 LS

The created initial VRP solution is often 20-30% far from the optimal solution to the problem (Erdelić and Carić, 2019). By searching the neighborhood of the current (initial) solution, a better solution can be found, commonly called local optima. The local optima is obtained by operators that calculate the difference in total travel time made by vehicles after switching customers among themselves or by inserting and removing customers from certain vehicle routes, but only if:

(i) a feasible solution regarding the load capacity and warehouse working hours is produced,



(a) Intra relocate

(b) Intra 2-Opt

(c) Inter exchange

(d) Inter relocate

Figure 4: LS operators

(ii) saving in the total travel travel time objective is achieved.

Depending whether the changes are performed on a single route or between multiple routes intra and inter LS operators are used. In total, four LS operators are applied, presented in Figure 4. After applying each of the operator on the current solution, the local optima in search composite neighborhood is found.

3.2.1 Intra operators

Intra operators are used to change the order of customers within the same route. The vehicle capacity limit with the intra operators does not need to be checked due to the rotation of the customer within the same route, so the capacity requirement remains unchanged.

Intra relocate is an operator that moves a customer from one position in a route to another position within the same route. As presented in Figure 4(a) three arcs in the same route are removed, and three new red ones are inserted. Time savings is defined as the difference between the total travel time before the change and after the change.

Intra 2-Opt operator is used to remove the arcs intersections within the route. As shown in Figure 4(b) two arcs that intersect are removed, and two new red ones are inserted, resulting in a decrease of the total travel time. Additionally, the part of the route between the removed arcs needs to be reverted (instead of $b \rightarrow c \rightarrow d$, the reversed order is $d \rightarrow c \rightarrow b$).

3.2.2 Inter operators

Inter operators are used to move customers between different routes in the solution. It is mandatory to check the vehicle capacity limit and total route duration with the inter operators.

Inter exchange operator replaces positions of two customers in different vehicle routes. Replacement is only performed if the vehicle capacity limit is not exceeded and time savings are achieved. As shown in Figure 4(c) two customers are exchanged with in total four arcs removed and four red ones inserted (two per route).

Inter relocate operator switches customers from one position in the route to another position in another route. As shown in Figure 4(d) one customer is removed from one route to another route, meanwhile removing three arcs and inserting three red ones.

3.2.3 Search strategy

LS operators are called in the following order: *inter exchange*, *inter relocate*, *intra 2-Opt* and *intra relocate*. Such order is selected to first improve the solution by moving customers between different routes and then to improve the customer configuration within each route. This makes sense, as there is no point in the route improvement with intra operator if a particular customer will be removed from the route with inter operator as it leads to an overall better solution. The operators are applied in such order, as long as there was at least one improvement by either of the operators. This means that the loop for the LS procedure repeats until there is no improvement. Within a search of a single operator, a *best-of* technique is used, meaning that the best move from all the possible moves found by a particular operator is conducted.

3.3 ILS

LS operators explore the narrow solution space around the current solution, which often tends to lead to the same local optima. To escape the local optima and search in the other parts of the solution space, the ruin-recreate principle to diversify the search is applied. Here, a random removal and insertion of customers in the solution is used. The description of ILS metaheuristic based on the ruin-recreate principle is given by Algorithm 2. The input to the algorithm is the currently best solution s, achieved either by the initial construction heuristics or LS procedure. In each iteration, n_{EJ} percent of customers are randomly selected from the solution and removed. Random removed customers are tried to be reinserted at some positions in vehicle routes. When inserting customers in vehicles, it is mandatory to check the capacity limitations of the vehicle and total route duration. After the evaluation of insertions, if either constraint is violated, the insertion of customers in the solution will not be performed, and the procedure for insertion will repeat itself to at maximum n_{INS} iterations. If a feasible solution is found, a LS is performed in an attempt to find a better solution. During the algorithm execution, two solution instances are tracked: current best solution s and temporal solution s'. The ruin-recreate principle is always performed on the current best solution. To prevent the data loss regarding the configuration of current best solution s in case when the ILS was not able to find a better solution than the best so far, at the start of each iteration, the best solution s is copied to s'. In the case when the total travel time of solution s' is better than the total travel time of s, the configuration of s' is copied to s. The whole procedure of removal, insertion and improvement is repeated for n_{ILS} iterations.

Algorithm 2 ILS

Input: Best solution s

while repeat n_{ILS} iterations **do**
  Copy solution s to s'
  Randomly remove n_{EJ} percent of customers from the solution s'
  while repeat n_{INS} iterations **do**
    Determine random positions of customer insertions in vehicles' routes in solution s'
    if solution s' would be feasible **then**
      Perform determined insertions on solution s'
      Perform LS on s'
      if solutions s' is better than s **then**
        Copy solution s' to s
      end if
      Break loop
    end if
  end while
end while

Due to the random selection of customers' removals and insertions after each restart of the program, there will be a different solution that may be better than the first solution obtained. It is important to note that this procedure searches the solution space beyond the local optima in order to find a better solution that may then be a globally optimal solution, but this is not guaranteed.

4 Results

In this section, the proposed methodology is evaluated on the test instance described in Section 2.2, and the analysis of time-dependent speeds is conducted. The parameters used for solving the problems are the following: $n_{ILS} = 30$, $n_{EJ} = 10$, $n_{INS} = 10\%$. The ILS is implemented as a single-thread code in the C# programming language. All tests were performed on a machine with Intel(R) Core(TM) i5-9300H CPU (2.24 Ghz) and 8 GB of RAM.

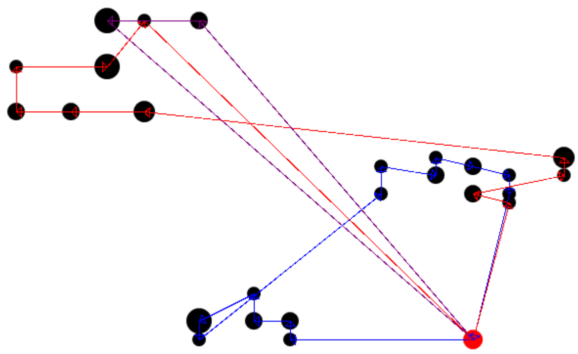
4.1 Evaluation of a search process

To solve the observed instance of the VRP, the discretization of time intervals and speeds is given by Table 1 (Scenario A). The warehouse working hours are set between 0 and $l_0' = l_0/4$. The closing time is shortened compared to the l_0 value from the Solomon instance (Solomon, 1987) because the closing time l_0, originally used for the problem with time windows, is too large for the observed TD-VRP problem where time windows are not included. This means that all deliveries in TD-VRP routes are always done by the time l_0'.

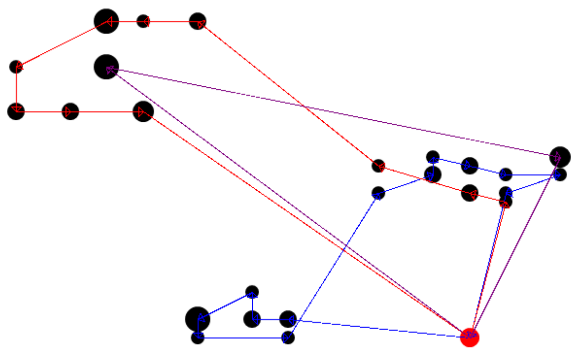
To show how the ILS works, the examples of solutions for scenario A, are presented in Figure 5. The metrics used to evaluate the solution are the Number of Vehicles (NV) in the solution, cumulative Total Travel Time (TTT) of all vehicles in the solution and cumulative Total Traveled Distance (TTD) of all vehicles in the solution. First, the initial solution is created with the time-oriented nearest neighbor heuristic (Figure 5(a)), resulting in NV=3, TTT=617.14, and TTD=266.29 (no real-world units are used). The values of TTT and TTD were decreased by a single application of the LS procedure (Figure 5(b)) to TTT=481.37 and TTD=256.53, with the same number of vehicles. The ILS strategy further decreased the values to TTT=433.88 and TTD=215.17 (Figure 5(c)). The results of the scenario A indicate that the proposed procedure is able to efficiently solve the problem and decrease TTT and TTD values compared to their initial values.

Period k	1	2	3	4	5
Time T_k	$0.05l_0'$	$0.15l_0'$	$0.2l_0'$	$0.4l_0'$	l_0'
Speed v_k	1.0	0.2	0.3	0.2	1.0

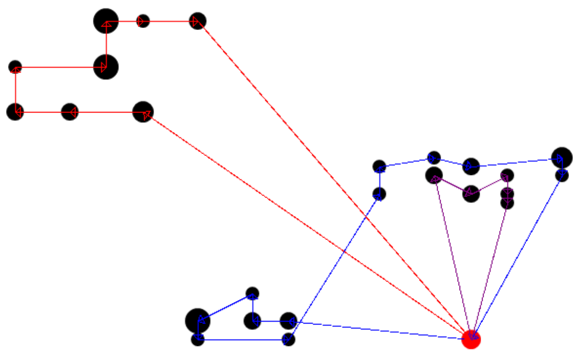
Table 1: Speed values in scenario A



(a) Initial solution - NV:3, TTT: 617.14, TTD: 266.29



(b) LS - NV:3, TTT: 481.37, TTD: 256.53



(c) ILS - NV:3, TTT: 433.88, TTD: 215.17

Figure 5: Example of search stages

4.2 Effect of time-dependent speeds

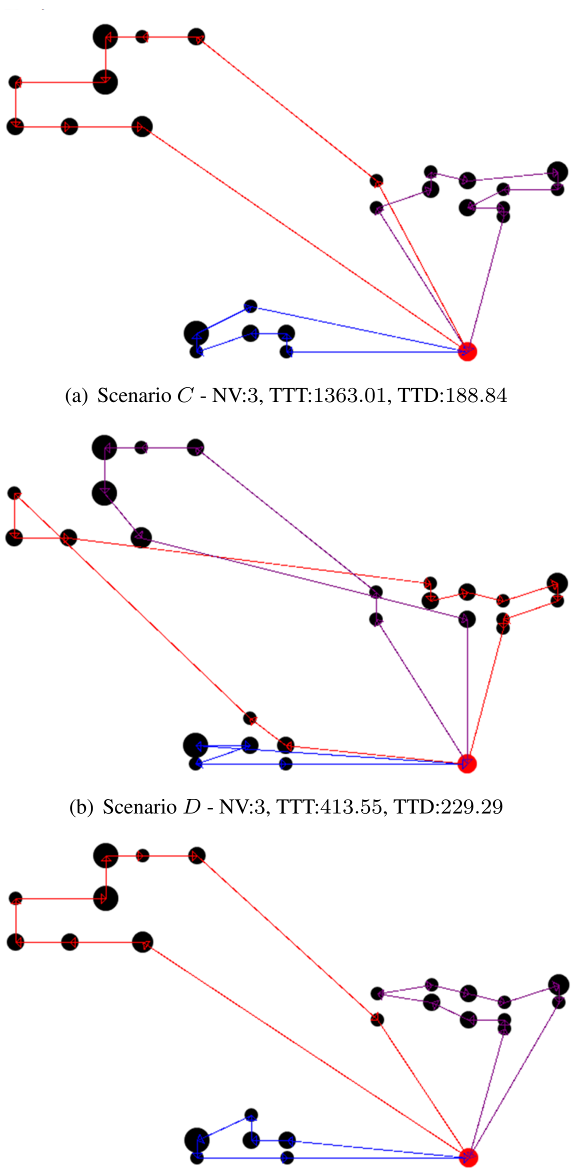
To investigate the effect of time-dependent speeds on the solution quality and customer configuration, another scenario B is observed. The time periods and speeds in scenario B are set in such way that they represent static traffic conditions, so the values of speeds are all equal to 1 in all time periods. In this case, the ILS resulted in NV=3, TTT=205.34, and TTD=205.34. As speed values are equal to 1, the TTT and TTD have the same values. Compared to scenario A, which uses higher speeds, the TTT in scenario B reduced by 52.67%, while the TTD reduced by 4.57%. As it can be seen, lower speeds during the mid-day time period primarily affect the TTT and not the TTD. Therefore, the knowledge of time-dependent travel times has a signif-

Period k	1	2	3	4	5
Time T_k	$0.05l'_0$	$0.15l'_0$	$0.2l'_0$	$0.4l'_0$	l'_0
Speed v_k (C)	0.1	0.2	0.2	0.2	0.1
Speed v_k (D)	0.9	0.1	0.7	0.1	0.9
Speed v_k (E)	0.8	0.6	0.7	0.6	0.8

Table 2: Speed values in scenarios C, D and E

icant impact on the delivery duration and can lead to an increase in the transportation cost in terms of driver salary and penalties for late arrivals, especially in time-precise deliveries.

Additional three scenarios are tested: constant congestion (scenario C), rush hours (scenario D) and light congestion (scenario E), with speed values presented in Table 2. The same time intervals are kept as in scenario A. The results are presented in Figure 6, and are the following: (i) scenario C - NV=3, TTT=1363.01, and TTD=188.84 (Figure 6(a)); (ii) scenario D - NV=3, TTT=413.55, and TTD=229.29 (Figure 6(b)); and (iii) scenario E - NV=3, TTT=280.57, and TTD=188.53 (Figure 6(c)). Again the TTD is slightly affected by the different speeds while the TTT is highly affected, especially in scenario C where the travel times increased six times compared to the scenario B, which considers static traffic conditions. As we can see on the figure 6 the configuration of customers per vehicle routes also changes, with the significant change in the scenario D.



(a) Scenario C - NV:3, TTT:1363.01, TTD:188.84

(b) Scenario D - NV:3, TTT:413.55, TTD:229.29

(c) Scenario E - NV:3, TTT:280.57, TTD:188.53

Figure 6: Results for scenarios C, D and E

5 Conclusion

In this paper, a variant of the VRP called capacitated TD-VRP is considered. TD-VRP takes into account time-dependent traffic conditions in the form of time-dependent travel times and vehicles with a limited load capacity. To solve the problem, an ILS metaheuristic is used that combines the ruin-recreate principle to diversify the search and LS procedure to intensify the search. The initial solution, used as input to the metaheuristic, is created using a time-oriented nearest neighbor heuristic. The results on one instance containing 25 customers show that the proposed metaheuristic is able to efficiently solve such problems. The conducted analysis regarding the different speed values in different periods of the delivery time (horizon) showed that time-dependent speeds have little effect on the total distance traveled, while they have a significant effect on the total travel time, especially in heavily-congested scenarios.

For future research, the idea is to investigate the metaheuristic behaviour on instances with several thousand customers, as well as the improvement of the total travel time computation regarding the transitions between two neighboring time periods.

Acknowledgments

The research has been partially supported by Croatian Science Foundation under project IP-2018-01-8323 and project KK.01.1.1.01.0009 (DATACROSS) within the activities of the Centre of Research Excellence for Data Science and Cooperative Systems supported by the Ministry of Science and Education of the Republic of Croatia. Authors are also very grateful to the Faculty of Transport and Traffic Sciences, University of Zagreb for conference publication grant.

References

Tonči Carić and Juraj Fosin. Using congestion zones for solving the time dependent vehicle routing problem. *PROMET - Traffic&Transportation*, 32(1):25–38, 2020. doi: 10.7307/ptt.v32i1.3296. URL `https://traffic.fpz.hr/index.php/PROMTT/article/view/3296`.

Tomislav Erdelić and Tonči Carić. A survey on the electric vehicle routing problem: Variants and solution approaches. *Journal of Advanced Transportation*, 2019:1–48, May 2019. doi: 10.1155/2019/5075671. URL `https://www.hindawi.com/journals/jat/2019/5075671/`.

Tomislav Erdelić, Tonči Carić, Martina Erdelić, Leo Tišljarić, Ana Turković, and Niko Jelušić. Estimating congestion zones and travel time indexes based on the floating car data. *Computers, Environment and Urban Systems*, 87:101604, 2021. ISSN 0198-9715. doi: https://doi.org/10.1016/j.compenvurbsys.2021.101604.

Andres Miguel Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):616–636, 2012. ISSN 13665545. doi: 10.1016/j.tre.2011.11.006.

Florian Kellner. Insights into the effect of traffic congestion on distribution network characteristics a numerical analysis based on navigation service data. *International Journal of Logistics Research and Applications*, 19(5):395–423, 2016. ISSN 1469848X. doi: 10.1080/13675567.2015.1094043.

Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59 (3):345 – 358, 1992. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(92)90192-C. URL `http://www.sciencedirect.com/science/article/pii/037722179290192C`.

Lovro Rožić, Juraj Fosin, and Tonči Carić. Solving the time dependent vehicle routing problem using real-world speed profiles. *Central European Conference on Information and Intelligent Systems, 2015*, 2015.

Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987. doi: 10.1287/opre.35.2.254. URL `https://doi.org/10.1287/opre.35.2.254`.

Paolo Toth and Daniele Vigo, editors. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. ISBN 0-89871-498-2.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1 – 21, 2013. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2013.02.053. URL `http://www.sciencedirect.com/science/article/pii/S0377221713002026`.