

MASTER

A Heuristic Approach for the VRPTW using dual information of its LP formulation

van de Westelaken, Max

Award date:
2020

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain



Department of Computer Science
& Department of Industrial Engineering and Innovation Sciences
Information Systems Research Group (IE&IS)

A Heuristic Approach for the VRPTW using dual information of its LP formulation

Master Thesis

Max van de Westelaken

Supervisors:

dr. M. Firat (IE&IS)
dr. R. Medeiros de Carvalho (M&CS)
dr.ir. C. Hurkens (M&CS)

Version 1.1

Eindhoven, November 2020

Abstract

This thesis deals with the Vehicle Routing Problem with Time Windows (VRPTW), one of the most applicable theoretical problems in current times. With it being an NP-Hard problem, we have to look at heuristics to achieve a suitable solution in an acceptable time frame.

We introduce a duality analysis of a master Linear Programming (MLP) formulation of the VRPTW. With it, we can access information on the customers and routes of the system and which to prioritize. By relaxing the the MLP formulation, we show through the dual analysis that there is a competition between customers to be serviced by a certain vehicle. Based on this, we can choose to group customer to a vehicle or introduce a new one, which services customers that are not being served enough in the current solution.

Based on these insights, a heuristic has been developed with three phases: preprocessing, warm-up and saturation. Route centers are selected to represent a vehicle, and from there routes are constructed. We identify three main components of the heuristic: route construction, finding a new route center and customer grouping. By balancing these three components, the heuristic comes to a full integer solution.

Based on a computational study, we show that the heuristic works well in many cases, but improvements still have to be made further. The project will continue on after this thesis to be submitted to a leading journal.

Preface

The following work is the culmination of many months of research, coding, testing and frustrations. Because of these things, I am extremely proud of the final result that you are about to read.

This research topic has been running for some time longer than this master thesis shows, and can be considered as being very involved. From the mathematics behind it to the actual digging through code, it has taken considerable time to get to the point we are at the moment. The theoretical backbone of the heuristic is not only complex, it is also vital in understand why exactly the heuristic is important.

Preliminary work has been done on this topic before, including the first version of the code that was used for this research. Over the last months, both myself and dr. Firat have worked hard to bring it to a working point, where results are put out in reasonable time.

The results of the study have been written in the form of a paper. After submitting this piece to the university, it will be reworked to shape it into a form for submission to a leading journal in the area of logistics and operations management. Because of that, at some points the explanations can be brief. I welcome anyone who wants to know more to dive deeper into the methodology and setup of the heuristic. The mathematics involved are by no means simple, but they are not as difficult as one might think at first glance.

I would like to thank dr. Firat for all the help and guidance he has given during the course of this project. In a time of uncertainty and adjusting to a new form of life, I could always fall back on his experience and know-how. Even when time was very limited, dr. Firat was always able to carve out a slot so we could talk in more detail about the code or the report. For that, my deepest thanks and appreciation.

I also want to thank dr. Medeiros de Carvalho and dr.ir. Hurkens, for their speed in reading the thesis and their extensive questions during the thesis defense.

A quick and essential thanks also goes to the many people on StackOverflow, who have almost always had the problems I had, but were smart enough to ask for answers on the internet.

A final thanks goes out to all my friends and family, whose continuous support has made this project much more enjoyable.

A Heuristic Approach for the VRPTW using dual information of its LP formulation

Max van de Westelaken^a, Murat Firat^b

^a*School of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, the Netherlands*

^b*School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands*

Abstract

This paper introduces a duality analysis of a master Linear Programming (MLP) formulation of the Vehicle Routing Problem with Time Windows (VRPTW). The considered MLP model is the slightly modified version of the relaxation of the Dantzig-Wolfe decomposition by expressing a VRPTW solution as a non-negative convex combination of constructed routes. The MLP model is basically the so-called reformulation of the VRPTW used in many Branch-and-Price (BP) algorithms. Our dual analysis shows that a pricing competition occurs in the dual model and the dual values of decision variables can guide us in making certain decisions like customer grouping and introducing a new vehicle to an existing (incomplete) solution. By using our dual interpretation, we propose a heuristic algorithm that greedily constructs a routing plan by iteratively solving the MLP model as a central optimization mechanism. The objects to select in the MLP model are routes that are constructed by using a Dynamic Programming (DP) based method. We keep the total number of routes bounded by a constant number, hence the size of the MLP model is fixed. A complete routing plan, i.e. an integer solution to the MLP model, is obtained by making the aforementioned decisions. We provide further details of the algorithm and show its efficiency by means of a computational study.

Keywords: Integer Linear Programming, Vehicle Routing Problem, Capacity Management

1. Introduction

The Vehicle Routing Problem with Time Windows (VRPTW) is one of the basic benchmark problems in optimization. It is a generalization of the Vehicle Routing Problem (VRP) that was firstly introduced by Dan59. The NP-hardness of the VRPTW is rooted in the Traveling Salesman Problem (TSP). Many exact solution methods to the VRPTW use a reformulation with a set packing structure and employ a Column Generation (CG) method in a Branch-and-Bound search, for example Desrochers92. Although remarkable progress is obtained in the size of solved instances, exact algorithms suffer from long running times in solving real-life instances. The main issue is that the pricing sub-problem is NP-hard and requires either a long running time or large amounts of memory to solve optimally. Some researchers worked on developing heuristics based on Branch-and-Bound methods, whose termination criterion is either time or solution quality or a mix of both.

The basic idea of our approach is to express the customer visits as decision variables and thus not enforcing the requirement that every customer should be visited by exactly one vehicle. Initially, we have an incomplete VRPTW solution when the master Linear Programming (MLP) model is solved, based on finding one or more "route centers", which are the foundations of paths and represent a vehicle. Then, we greedily assign certain customers to those route centers and reach a complete routing solution. The decision of assigning customers to route centers is based on the evidence obtained from the solutions of the MLP model and its dual. Relaxing the requirement of visiting customers exactly once is not a new idea, for example Kohl97 and Kalle06 worked on exact algorithms based on Lagrangian relaxation in which the relaxed constraints of the VRPTW are the aforementioned ones.

In this work, the proposed heuristic approach makes use of the information given by LP-duality in order to find good-quality solutions to the VRPTW. Our approach has important similarities to exact algorithms. It uses the master LP model with set packing structure that is used by many Branch-and-Price

algorithms. Our algorithm does not backtrack during its course, and it makes greedy decisions in this sense. As customers are assigned to used vehicles, the path sets are revised in order to update the information obtained from dual solutions.

35 *Our contribution.* The contribution of this paper is two-fold. Firstly, we provide a dual interpretation of the MLP formulation of the VRPTW. Our dual interpretation shows that competition happens among customers in the dual model, and the values of dual decision variables provide us with some information about that customer competition. As the second contribution of
40 this paper, we propose a heuristic approach to solve the VRPTW that makes use of the information provided by our dual interpretation. Our computational experimentation shows that our heuristic is promising in finding good quality solutions for benchmark cases.

The paper is organized as follows. First, we briefly describe and categorize
45 the VRPTW in Section 2. Related work in the literature is outlined in Section 3. We will take a deeper look at the theoretical backbone in Section 4. Section 5 presents our proposed heuristic algorithm to the VRPTW by first outlining its important properties in an overview, then expanding on each part. Computational results are reported in Section 6. Finally, conclusions and possible
50 research directions are discussed in Section 7.

2. Preliminaries

This section briefly describes the VRPTW, and defines several concepts that are necessary for a formal description of our heuristic method.

2.1. Problem description

55 An instance of the VRPTW consists of a set $N = \{0, 1, \dots, n\}$ of locations on a plane, where 0 is the depot and others are customer locations, a set V of homogeneous vehicles of capacity $Q \in \mathbb{Z}_+$. Every customer $i \in N \setminus \{0\}$, also denoted by N' , requires a service of length $sv_i \in \mathbb{Z}_+$ time units for a demand of

amount $q_i \in \mathbb{Q}$. The service at customer i can only start in time interval $[e_i, l_i]$
60 where $e_i, l_i \in \mathbb{Z}_+$ are called earliest time (or release date), and latest arrival
time (or due date) respectively. Hence, arriving earlier than e_i requires waiting
till e_i , but later than l_i implies violation of the feasibility. The planning horizon
of the problem is defined by the time window of the depot and it is denoted
by $[e_0, l_0]$. The distance between customers i and i' is denoted by $d_{i,i'}$, and
65 is equal to the Euclidean distance of the arc $[i, i'] \in N \times N$ on a plane where
customer locations are specified as x - and y -coordinates. We assume that a
unit distance is traveled in a unit time, i.e. the distance of an arc is equal to
the travel time on it.

PROBLEM: VEHICLE ROUTING PROBLEM WITH TIME WINDOWS
70 (VRPTW)

INSTANCE AND FEASIBILITY:

Set N of customers with demands and time windows and the depot,
set V of homogenous vehicles with capacities.

A feasible route of a vehicle is a sequence of visited customers such
75 that total demand does not exceed the vehicle capacity, and that
every visited customer is serviced within its time window, and de-
pot departure and depot arrival stay within the planning horizon.
A feasible routing solution is a set of feasible routes such that ev-
ery customer is serviced exactly by one vehicle, and the number of
80 used vehicles does not exceed the number of available vehicles in the
depot.

QUESTION: Does there exist an routing plan with number of vehicles
less than k and for k vehicles with a smaller total travel distance less
than D ?

85 *2.2. Preprocessing*

Given an instance of the VRPTW, we conduct some preprocessing steps. Firstly, we adapt the time windows as follows

$$e_i = \max\{e_i, e_0 + d_{0,i}\}, l_i = \min\{l_i, l_0 - d_{i,0}\}, \quad i \in N \quad (1)$$

Having found adapted time windows, incoming and outgoing arcs around customers are ranked with respect to their adapted lengths. Incoming and outgoing arc lists of customers are non-decreasingly ordered and the indices of arcs in these lists become their ranking. Let $start_i$ denote the start time of the service of customer i . It follows that $e_i \leq start_i \leq l_i$.

2.2.1. Simple lower bounds on the number of used vehicles

Let L_{Veh} denote the lower bound on the number of vehicles in all feasible routing solutions for a given VRPTW instance.

Using total demand. Trivially, We can find minimum number of vehicles to serve all customers by the total customer demand as

$$L_{Veh} \geq L_P = \left\lceil \frac{\sum_{i \in N'} q_i}{Q} \right\rceil \quad (2)$$

Using incompatible customers. Another way to get a lower bound on the number of vehicles is by using incompatible customers.

(Incompatible customers) Two customers that cannot be served in a route without violating temporal constraints or vehicle capacity are called incompatible.

A customer set in which every pair of customers is incompatible also gives us a lower bound on the number of vehicles in a feasible routing solution. The maximum cardinality of aforementioned customer set can be found by solving the maximum independent set problem in customer network. Unfortunately, this problem is NP-Hard in the strong sense. Therefore, we settle on a heuristic for finding a maximal independent set. In this heuristic, an independent set

of customers is constructed greedily. Having chosen a customer to add to the
 110 independent set, all customers in the network that are connected by an arc to
 the chosen customer are deleted. This continues until no customer is left to
 choose. The decision of selecting the first customer to start the independent set
 is made by checking the number of connected customers in the network. By
 taking this, we eliminate as many candidates as possible.

115 The process is visualized in Figure 1. Nodes represent customers, and an arc
 represents the fact that two customers can exist in one route together. Firstly,
 in Figure 1a, an initial route center (RC) is found. This customer is the one with
 the most connections to other customers in the network. Then all its neighbours
 are eliminated from the set in Figure 1b. This process repeats in Figure 1c and
 120 1d, until we arrive in the final solution in Figure 1e.

Let IND denote the set of maximal independent set found by using the
 heuristic described above. In this case, $IND \subset N'$. Then, we define L_{Veh} as
 follows

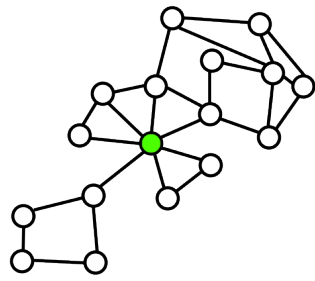
$$L_{Veh} = \max\{L_P, |IND|\} \quad (3)$$

2.3. Constructing the Routes

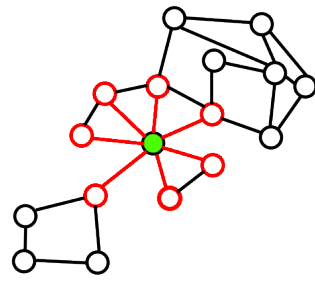
125 *Route centers.* We distinguish some customers as the ones defining routes,
 meaning that each of such customers have sets of routes passing through them.
 Eventually, this corresponds to dedicating one vehicle to every route center in
 our solutions.

(Route Center) The building block of routes and the representation of a
 130 vehicle in the solution. Each route must contain exactly one route center.

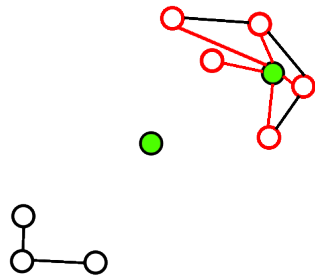
Let $RC \subseteq N'$ denote the set of route centers. In our algorithm RC is
 initialized with the customers in the independent set solution, IND . Then,
 some other customers may be added and we will refer to this extension of RC
 as “route center selection”.



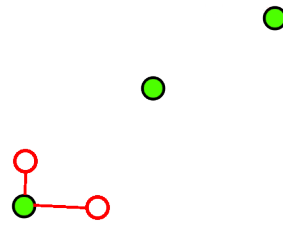
(a) Finding first RC



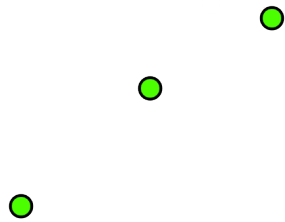
(b) Eliminating neighbours



(c) Second RC



(d) Third RC



(e) Final Independent Set

Figure 1: Finding Independent Customers

135 *Routes.* A sequence $r = (r_1, r_2, \dots, r_{l(r)})$ of customers with subset C_r of visited customers, is called a “route”, if r satisfies the following feasibility conditions:

- Every customer $i \in N$ is visited by r at most once, i.e. $|C_r| = l(r)$
- Route r has exactly one route center, i.e. $|C_r \cap RC| = 1$
- Every customer in r is serviced within its time window.
- 140 • Total demand does not exceed vehicle capacity, i.e. $\sum_{i=1}^{l(r)} q_{r_i} \leq Q$.

3. Related work

There is extensive literature for the VRPTW that contains a wide range of algorithms like priority rule based simple heuristics, (adaptive) large neighborhood search, and exact algorithms. A good summary of the literature of the VRPTW until the 1990s can be found in the review of Des88. In this survey, 145 the authors mention that the literature lacks (at that time) an exact approach to the VRPTW, and few years later Desrochers92 proposed one of the first exact solution methods to the VRPTW. The authors’ method solves a reformulation of the VRPTW with a set packing structure and employs the CG technique 150 in order to do bounding in a Branch-and-Bound search. The corresponding sub-problem amounts to finding the shortest path in a modified network with time windows and capacity constraints, and it is solved by using a Dynamic Programming method. The largest instance size that is solved optimally was with 14 customers at that time, and the results of Desrochers92 showed that a 155 high ratio of 25-customer instances are solved optimally within 10 minutes by using the computation power of the 1990s.

Some years later, Fisher97 proposed two optimization algorithms to the VRPTW, namely a Lagrangian Relaxation/Variable splitting approach and a K-tree approach. In the former, two sub-problems (a semi-assignment problem 160 and shortest path problem with time windows and capacity constraints) are solved. The authors report that 100-customer benchmark instances with clustered and randomly located customers solve to optimality with varying solution

times between 10 and 70 minutes. The conclusion is that both optimization algorithms perform best especially on the instances with clustered customers.

165 Fixing decision variables in a VRPTW formulation is a general trick researchers have used. For example the authors of Cacchiani14 propose a heuristic approach to the Periodic Vehicle Routing Problem (PVRP). The proposed algorithm solves a master LP model by fixing binary variables to 1 whose solution values are 1, and fixing the value of the variable that has the highest fractional
170 value. After the fixing, the authors find new columns by taking into account the changed dual values of the master LP model. Another example can be found in the work by Desaulniers20. They work in branch-price-and-cut algorithms for the VRPTW, where it is possible to fix a variable to zero according to the dual solution. It deals with a set of arcs, from which one can be eliminated to fix
175 the variables. In their paper, they extend this to two-arc sequences, which are much more difficult to completely eliminate. The proposed technique indirectly removes them from the sequences in the network, leading to good results on benchmark instances.

In another paper, Huang11 introduce binary variables to allow not visit-
180 ing/outsourcing some customers, and minimize the weighted sum of these variables in the objective function. The authors propose a Lagrangian heuristic to the Vehicle Routing Problems with the Private Fleet and the Common Carrier (VRPPC). Another paper by Schneider14 deals with the VRPTW with recharging stations, where the vehicles have to recharge after a certain distance
185 has been used. They show that a hybrid solution of neighborhood search and tabu search can lead to good solutions.

The relaxation of constraints is not a new concept. As far back as the 1990's, Kohl97 showed that Lagrangian relaxation can be used on the VRPTW. The key here was finding the Lagrangian multipliers to come to a good solution.
190 Similarly, a paper by Baldacci11 looked at relaxation of the routes themselves, using dual ascent heuristics found in the literature. The new techniques allow for a reduced set partitioning formulation and quickly and efficiently solve many bookmark instances. Recently, Liu19 has shown that for the VRPTW with

load-dependent costs, there is a possibility of relaxing the constraint that every
195 customer has to be visited by a vehicle. Instead, they introduce 'virtual' vehicles.
Impressive results were yielded.

A paper that deals with the VRPTW is the paper by Bianchessi2019, which
adds as a constraint that multiple vehicles can service one customer. In the
paper, a new method is proposed which is a tailored version of a branch-and-
200 price-and-cut algorithm. Integer solutions are built in a relaxed model. A sparse
subnetwork is created, in which all time-window-feasible routes are enumerated,
cutting out all infeasible solutions. Experimentation has shown that this ap-
proach can solve some previously unsolved instances.

In the paper of Gunluk06, the multi-depot VRPTW is studied. The au-
205 thors propose a so-called Fix-Price Heuristic that works in a similar manner of
our heuristic algorithm. In their follow-on fixing procedure, the columns with
solution values not smaller than 0.95 are fixed to 1. Then, all columns are up-
dated in order respect the fixing decisions. Next, the LP model is solved, and
fixing decisions are made as long as variables with convenient solution values
210 are found. When no variable with desired solution value is found, the threshold
value is decreased to 0.85. The procedure is terminated if no variables having
solution values greater than or equal to the reduced threshold value. Besides
this fixing procedures, the proposed heuristic approach of Gunluk06 has other
components to solve the studied problem efficiently.

215 One of the recent works on the VRPTW is conducted by Nagata09. The
authors propose a sophisticated approach for reducing the number of routes,
and it is based on the ejection pool that is combined with a concept reminiscent
of the Guided Local Search. The benchmark instances described by GeHo01
are used in experimentation of the proposed approach. By limiting the solution
220 time to multiples of 10 minutes (maximum 5 hours), the authors were able to
find new best known solutions for several instance sizes between 400 and 1000
customers. To the best of our knowledge, Nagata09 have currently the best
solutions for the instances of large size in the literature.

There are also more sophisticated methods that have been used for solv-

225 ing the VRPTW. For example, Xu15 implements a combination of a genetic
algorithm and a particle swarm optimization for an extremely complex system
to solve the VRPTW. Experimental results show that the algorithm is very
efficient and competitive, but also obtains optimal solutions. Kalina15 imple-
230 mented a method based on agent negotiation, where the problem is decomposed
for the different vehicles who are treated as actors. It was an improvement on
existing agent-based simulations at the time. An interesting paper is submitted
by Ukhan20, in which the authors argue for a system that manages to choose
the correct implementation for the VRPTW out of a list of candidates. They
show that the speed and quality of results can be dependant on the complexity
235 of the problem, as well as the problem size. Because of that, certain solution
methods are more or less suitable. This paper shows that the need for ICT
and decisions on the correct method are vital. Having more proper methods to
choose from can help a lot.

We refer to surveys by BraGen05a, and BraGen05b for more recent exact
240 heuristic algorithms for the VRPTW, as well as the paper by Costa2019 for a
very recent survey on exact branch-and-price algorithms for the generic VRP.

4. Theoretical Background

In this section, we will look more closely at the theoretical framework on
which we will base the heuristic. We will take a proper look at the primal-dual
245 method and its implications, showing how we can get information from the dual
model. This section will be highly theoretical and mathematical, beginning
with a formulation of a linear program to solve the VRPTW. From there, we
formulate the dual of this program. These models are nudged to optimality
using the complimentary slackness theorem and then restricted, after which an
250 improvement step is detailed. Finally, we will look at how to interpret these
values, then at an illustrative example.

Our master LP model allows us to start with a partially feasible solution that
serves a subset of customers initially. It contains customer assignment variables

to decide which customers are to be served by available vehicles introduced
 255 so far. The objective has as its primary goal “maximizing” the number of
 selected customers and secondary “minimizing” the total distance traveled in
 the selected paths. A big coefficient M is used to denote the hierarchy in two
 aforementioned goals. Note that our master LP finds a routing plan for a given
 number of vehicles, that is $|RC|$. Next, we give the formulation of our master
 260 LP model.

Table 1 explains the parameters and decision variables. The formulation of
 the master IP model is given in (4)-(7).

Table 1: Sets, parameters, and variables

Sets	
N'	set customer locations, $N' = N \setminus \{0\}$
\mathcal{R}	set of all constructed routes,
Parameters	
M	the objective coefficient of primary goal variables
c_r	cost (traveled distance) of route $r \in \mathcal{R}$
δ_r^i	indicates if route $r \in \mathcal{R}$ visits customer $i \in N'$
Decision Variables	
x_r	selection variable of path $r \in \mathcal{R}$
y_i	selection variable of customer $i \in N'$,

$$(SP) \quad \text{Max} \quad M \left(\sum_{i \in N'} y_i \right) - \sum_{r \in \mathcal{R}} c_r x_r \quad (4)$$

subject to:

$$\sum_{r \in \mathcal{R}} \delta_r^i x_r - y_i = 0, \quad i \in N' \quad (5)$$

$$y_i \in \{0, 1\}, \quad i \in N' \quad (6)$$

$$x_r \in \{0, 1\}, \quad r \in \mathcal{R} \quad (7)$$

265 Constraint (5) couples selections of a customer and the routes visiting that customer. Note that in standard IP reformulations in the literature, for example Desrochers92, all y_c variables are fixed to the value 1 as the right hand side of (5). Next, we relax all binary variables in (6)-(7) and we obtain the master LP model as

$$(P) \quad \text{Max} \quad M \left(\sum_{i \in N'} y_i \right) - \sum_{r \in \mathcal{R}} c_r x_r \quad (8)$$

subject to:

$$\sum_{r \in \mathcal{R}} \delta_r^i x_r - y_i = 0, \quad i \in N' \quad (9)$$

$$y_i \leq 1, \quad i \in N' \quad (10)$$

$$y_i \geq 0, \quad i \in N' \quad (11)$$

$$x_r \geq 0, \quad r \in \mathcal{R} \quad (12)$$

270 In the following section we use the primal-dual method to analyze how an optimal solution in the dual model is obtained, which will enable us to interpret the values of dual variables in optimal solutions.

4.1. Dual analysis via primal-dual method

275 In this section, we show how the dual of the master LP model in (8)-(12) is solved optimally by incorporating the primal-dual method. After the explanations, numerical examples will also be given in the end of this section. The primal-dual method was proposed by Dan56, and it has been used to design approximation algorithms for many problems in graph theory. In mathematical programming, it is known that many ideas of the exact algorithms to
280 a number of network design problems are implicit in the primal-dual algorithms. Interested readers are referred to Goemans96 for an extensive analysis of the primal-dual method in network design problems.

To start our analysis, we give the dual of our master LP model by letting
 285 $\lambda_i, \gamma_i, \kappa_i$ be the dual variables corresponding to the constraints (9)-(12).

$$(D) \quad \text{Min} \quad \sum_{i \in N'} \gamma_i \quad (13)$$

subject to:

$$(\gamma_i + \kappa_i) - \lambda_i \geq M, \quad i \in N' \quad (14)$$

$$\sum_{i \in C_r} \lambda_i \geq -c_r, \quad r \in \mathcal{R} \quad (15)$$

$$\gamma_i \geq 0, \quad i \in N' \quad (16)$$

$$\kappa_i \leq 0, \quad i \in N' \quad (17)$$

Complementary slackness condition.. By the Complementary Slackness (CS) theorem, given primal and dual feasible solutions $(y, x; \lambda, \gamma)$ are optimal if and only if the following equalities are satisfied

$$x_r \left(\sum_{i \in C_r} \lambda_i + c_r \right) = 0, \quad r \in \mathcal{R} \quad (18)$$

$$y_i (\gamma_i + \kappa_i - \lambda_i - M) = 0, \quad i \in N' \quad (19)$$

$$(1 - y_i) \gamma_i = 0, \quad i \in N' \quad (20)$$

$$(y_i) \kappa_i = 0, \quad i \in N' \quad (21)$$

290 For the detailed analysis of the CS conditions and an extensive analysis of the linear optimization, we refer to the the book of BertTsit97. In the primal-dual method, a given dual feasible solution is improved towards the optimal solution by using the “restricted primal” model, which minimizes the violations from the CS conditions. The basic idea is that the satisfaction of CS conditions
 295 is greedily increased till full satisfaction is reached. In order give the formal

definition of the restricted primal model, we need to define several sets related to CS conditions as

$$K = \{r \in \mathcal{R} \mid \sum_{i \in C_r} \lambda_i + c_r = 0\} \quad (22)$$

$$J = \{i \in N' \mid \gamma_i + \kappa_i - \lambda_i = M\} \quad (23)$$

$$I = \{i \in N' \mid \gamma_i = 0\} \quad (24)$$

$$L = \{i \in N' \mid \kappa_i = 0\} \quad (25)$$

The set K is said to contain all routes in price balance in the dual solution. Note that only routes in K can have positive x_r values by CS condition (18).
 300 Similarly, only customers in J can have positive selection values. Finally, we define the slack variable s_i for the customers not in the set I to quantify the violation from the CS condition. The violation of CS condition (19) due to customers not in set J is simply the value of y_i .

Restricted primal model. For a given a dual feasible solution (λ, γ) with sets
 305 K, J, I ; we can formulate a restricted primal problem that minimizes the violation of CS conditions as

$$(RP) \quad \text{Min} \quad z_{RP} = \sum_{i \notin I} s_i + \sum_{r \notin K} x_r + \sum_{i \notin (J \cup L)} y_i \quad (26)$$

subject to:

$$\sum_{r \in \mathcal{R}} \delta_r^i x_r - y_i = 0, \quad i \in N' \quad (27)$$

$$y_i \leq 1, \quad i \in I \quad (28)$$

$$y_i \geq 0, \quad i \in N' \quad (29)$$

$$y_i + s_i = 1, \quad i \notin I \quad (30)$$

$$x_r \geq 0, \quad r \in \mathcal{R} \quad (31)$$

$$s_i \geq 0, \quad i \notin I \quad (32)$$

By constraints (28) and (29), customers in I can have any y value, and the violation of CS condition (20) of those not in I amounts to the value of slack
 310 variable s in constraints (30). From this, we can see there are two possibilities:

Case $z_{RP}^ = 0$:* Master LP is solved to optimality, i.e. all CS conditions are satisfied.

Case $z_{RP}^ > 0$:* Dual feasible solution $(\lambda, \gamma, \kappa)$ is improved to another dual feasible solution $(\lambda'', \gamma'', \kappa'')$ with a smaller objective value. To explain how this
 315 improvement is achieved, we first need to consider the dual of the (RP) model.

$$(DRP) \quad \text{Max} \sum_{i \in N'} \gamma'_i \quad (33)$$

subject to:

$$\gamma'_i + \kappa'_i - \lambda'_i \leq 1, \quad i \notin (J \cup L) \quad (34)$$

$$\gamma'_i + \kappa'_i - \lambda'_i \leq 0, \quad i \in (J \cup L) \quad (35)$$

$$\sum_{i \in C_r} \lambda'_i \leq 1, \quad r \notin K \quad (36)$$

$$\sum_{i \in C_r} \lambda'_i \leq 0, \quad r \in K \quad (37)$$

$$\gamma'_i \leq 1, \quad i \notin I \quad (38)$$

$$\kappa'_i \geq 0, \quad i \in I \quad (39)$$

$$\gamma'_i \leq 0, \quad i \in I \quad (40)$$

For the sake of simplicity in the notation, decision variables γ' are used in the (DRP) model. They are different from γ dual variables in the dual of the MLP model. In the case of $z_{RP}^* > 0$, we have $\sum_{i \in N'} \gamma'_i = z_{RP}^* > 0$ by strong
 320 duality. The improved dual solution is found as

$$(\lambda'', \gamma'', \kappa'') = (\lambda, \gamma, \kappa) - \Delta(\lambda', \gamma', \kappa') \quad (41)$$

where $\Delta > 0$ is called the dual improvement step value. Then we have

$$\sum_{i \in N'} \gamma''_i = \sum_{i \in N'} \gamma_i - \Delta \sum_{i \in N'} \gamma'_i < \sum_{i \in N'} \gamma_i \quad (42)$$

Preserving dual feasibility gives us the maximum value of Δ . This is done by checking constraints (14)-(16) as

$$\gamma''_i + \kappa''_i - \lambda''_i \geq M \Rightarrow \Delta \leq \min_{i \notin (J \cup L), \gamma'_i + \kappa'_i - \lambda'_i > 0} \left\{ \frac{\gamma_i + \kappa_i - \lambda_i - M}{\gamma'_i + \kappa'_i - \lambda'_i} \right\} \quad (43)$$

$$\sum_{r \in C_r} (\lambda_i - \Delta \lambda'_i) + c_r \geq 0 \Rightarrow \Delta \leq \min_{r \notin K, \sum_{i \in C_r} \lambda'_i > 0} \left\{ \frac{\sum_{i \in C_r} \lambda_i + c_r}{\sum_{i \in C_r} \lambda'_i} \right\} \quad (44)$$

$$\gamma''_i = \gamma_i - \Delta \gamma'_i \geq 0 \Rightarrow \Delta \leq \min_{i \notin I, \gamma'_i > 0} \left\{ \frac{\gamma_i}{\gamma'_i} \right\} \quad (45)$$

Sets J, K, I and L are updated according to solution values. This dual
 325 improvement procedure is repeated, by iteratively solving the (*DRP*) model, till
 we obtain objective value $\sum_{i \in N'} \gamma'_i = 0$ that implies the optimal dual solution
 is obtained.

4.2. Interpreting dual problem.

In the dual model, γ (κ) variables represent positive (negative) budget. If one
 330 of γ and κ variables is non-zero, then the other must be zero by CS conditions
 (18)-(21). Hence, $(\gamma + \kappa)$ is the “budget” of customers in the dual solution.
 Dual improvement step Δ is the “price” paid/collected in a dual improvement
 iteration. In dual iterations there are some price-collecting customers and some
 price-paying customers. The decision variable γ' (κ') in the (*DRP*) model tells
 335 us if a customer is a price collector (payer). The value of variable γ'_i (κ'_i) is the
 portion of the collected (paid) price for customer i .

A route constrains the total value of the budgets of visited customers. Dual
 constraints (15) enforce that the total budget of customers visited by route
 $r \in \mathcal{R}$ can have the smallest value $-c_r$, and routes with a minimum total
 340 budget are said to be in price balance. Once a route reaches price balance, it
 is added to set K , and the total budget should not change in the next dual
 iteration (constraint (37)). A route may leave price balance by increasing its
 total customer budget. In a dual iteration, constraints (36) enforce that routes
 not in price balance can have at most one dual improvement price drop in total
 345 budget. A *necessary condition* for a route to be selected in the optimal primal
 solution is to be in price balance. In light of our dual understanding, we have
 the following observation.

Table 2: Dual pricing rounds of the illustrative example

Instance Data						
$N' = \{i_1, i_2, i_3, i_4, i_5\}, RC = \{i_2, i_4\}, \mathcal{R}'_{i_2} = \{r_1, r_2, r_3\}, \mathcal{R}'_{i_4} = \{r_4\}$						
$r_1 = (i_1, i_2, i_3), r_2 = (i_2, i_3), r_3 = (i_2, i_5), r_4 = (i_4, i_5)$						
$c_{r_1} = 75, c_{r_2} = 50, c_{r_3} = 80, c_{r_4} = 60$						
Dual Pricing Solution						
Initial dual feasible solution: $\gamma_{i_k} = 1.5M, \kappa_{i_k} = 0, \lambda_{i_k} = 0.5M, i_k \in N'$						
Dual iteration	Payers	Collectors	Δ	K	I	$\sum \gamma'$
1	i_3, i_5	i_2	$M + 50$	\emptyset	\emptyset	$6.5M - 50$
2	i_2, i_4	i_5	$M + 50$	$\{r_2\}$	\emptyset	$5.5M - 100$
3	i_3, i_5	i_2	10	$\{r_2, r_4\}$	\emptyset	$5.5M - 110$
4	i_1, i_3, i_5	i_2, i_4	$0.5M - 60$	$\{r_2, r_4\}$	$\{i_3\}$	$5M - 50$
5	i_1, i_3	i_2	85	$\{r_1, r_2, r_4\}$	$\{i_3\}$	$5M - 135$
Final budgets: $\gamma_{i_1} = M - 25, \gamma_{i_2} = 2M - 50, \gamma_{i_3} = 0, \gamma_{i_4} = M - 110, \gamma_{i_5} = M + 50$						

A route center with a high γ value experiences high competition among customers to get serviced by one of its routes in the primal solution. Hence, the route center with smallest γ value in the dual solution is the most convenient for making customer grouping decisions.

In the remainder of this section we give an illustrative example in order to show how dual pricing rounds occur, and present computational results of some initial master LP model due to Solomon benchmark instances.

4.3. An illustrative example

Let us consider a simple problem instance that is given in Table 2. Clearly the only feasible, hence optimal, solution to the corresponding master LP model is $x_{r_1}^* = x_{r_4}^* = 1$, and $x_{r_2}^* = x_{r_3}^* = 0$ with objective value $5M - 135$. Table 2 also shows the dual pricing solution after five dual iterations. Initially all customers have a budget of amount $\gamma = 1.5M$ which results in the dual objective value $7.5M$.

In the first iteration, customers i_5 and i_3 compete for being serviced in a route of route center i_2 . In the second iteration, route center i_4 involves a competition with route center i_2 to service customer i_5 in its route. The

Data: VRPTW Instance (Section 2.1), route sets $\mathcal{R}'_{rc}, rc \in RC$

Result: Optimal dual solution with pricing rounds

Initial dual feasible solution: $\gamma_c^0 = (1 + \theta)M, \lambda_c^0 = \theta M, \kappa_c^0 = 0$;

Set: index $i = 0$, initialize dual improvement step $\Delta = \theta$;

while $\Delta > 0$ **do**

- Construct/update DRP_i model as in (33)-(40) with sets K_i, J_i, I_i, L_i ;
- Solve DRP_i model;
- if** *Objective value of DRP_i is zero* **then**
- break ;
- Determine largest value of Δ by (43) and (45);
- Update $\gamma_c^{i+1}, \lambda_c^{i+1}, \kappa_c^{i+1}$ as in (41) or customers in DRP_i solution;
- Find sets $K_{i+1}, J_{i+1}, I_{i+1}, L_{i+1}$;
- $i++$;

Output optimal dual solution $(\gamma^*, \lambda^*, \kappa^*)$;

Algorithm 1: Finding the optimal solution to a VRPTW problem

365 competition in the first iteration repeated with a different price in third iteration.
 In the last two iterations, the prices are only for appearing in routes and price values tend to decrease.

4.4. Solomon benchmark instances.

370 Having shown how the dual improvements occur, an iterative algorithm to solve the dual of the MLP model is implemented. As Algorithm 1 shows, we start with a dual feasible solution in which every customer has an initial budget slightly exceeding the coefficient M . As long as the optimal objective value of the (DRP) model is significantly greater than zero and the dual improvement step is

375 positive, the current dual feasible solution is improved. In every dual iteration, sets K, J, I and L are updated. The implementation is coded in Java and the (DRP) model is solved by using CPLEX 12.6.1. The optimal dual solutions found after dual improvement/pricing rounds are verified by the dual solution of the MLP model that has also been found using CPLEX 12.6.1. Numerical

380 results and the number of dual rounds for corresponding instances are given in Table 3. We do not report the solution times, since our main concern here is

Table 3: Computational results of primal-dual method

Ins.	$ RC ^*$	z_D^*	#Rounds
C101	10	99171.06	58
C102	8	87245.36	131
C103	7	77336.33	68
C104	4	48652.58	27
C105	5	55488.42	29
C106	4	45130.29	31
C107	1	11853.27	69
C108	1	11853.27	60
C109	1	11867.77	39
R101	17	91559.96	328
R102	16	92059.64	222
R103	13	90859.44	278
R104	7	74808.94	164
R105	7	61133.50	76
R106	6	60989.03	65
R107	5	57724.92	87
R108	5	59987.25	70
R109	4	43079.24	21
R110	3	35690.76	20
R111	5	57181.50	61
* Initial set of route centers			

to show the equality of the solutions values found by directly solving the MLP model and by solving its dual with pricing rounds.

385 5. Methodology

In this section we explain a heuristic algorithm to solve the VRPTW. Note that this heuristic is a special way of using the dual understanding of master LP models as seen in Section 4. The heuristic algorithm utilizes the dual understanding that was the topic of the previous section. The algorithm has three
390 phases; Pre-processing, a warm-up phase, and a saturated phase. Before going

into details, we shall provide a schematic overview of the heuristic in Section 5.1. After that, we will dive into the details of route construction in Section 5.2, how the information from the dual is retrieved in Section 5.3, how to find a new route center in Section 5.4, and finally the process of grouping a customer
395 with a route center in Section 5.5.

5.1. An overview of the approach

As can be seen in Figure 2, there are three phases in the heuristic. In the beginning, there is the preprocessing phase. The heuristic algorithm starts with initializing the set of route centers which is done by finding a maximal
400 independent set of (incompatible) customers, as was detailed in Section 2.2. After that, the route centers are initialized in the code. This phase is relatively short and we will not take a deeper look into it.

The second phase is the Warm-Up phase. Routes are constructed using the found route centers as will be detailed in Section 5.2. Immediately after that,
405 the master LP is solved and checked using Gurobi 9.1. the primal and dual solutions of the MLP are ready to proceed further and provide information for the rest of the heuristic. This is the first iteration of the LP, as detailed in Section 5.3.

If we already have a proper solution, we can simply end. If this is not the
410 case, we have to at least get $|RC|$ to L_{Veh} as was found in Section 2.2. We go to find a new route center, as is detailed in Section 5.4, which also includes a new iteration of the linear program to verify which candidate to pick. Afterwards, the heuristic loops back to create full routes for the new route center and solve the LP again. We call this phase 'Warm-Up' because we want to get to the
415 situation where $|RC| = L_{Veh}$.

After that comes the 'Saturated' phase, which is called such because we have reached the lower bound on vehicles. Therefore, the system has to use more sophisticated decision making. There is a large loop in this phase, where route construction is followed by either finding a new route center, customer
420 grouping, or both.

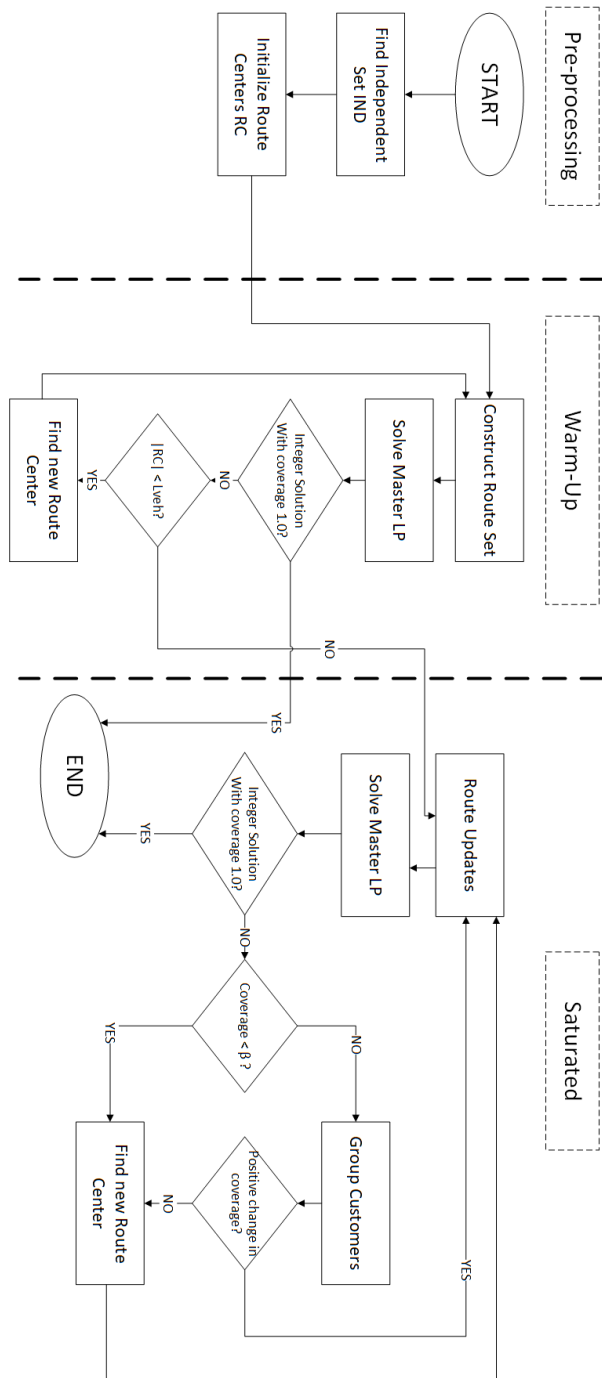


Figure 2: Overview of Heuristic Approach

If the system has found an integer solution with coverage 1.0, we can safely end. If not, we check whether the coverage is below a certain threshold β . In our implementation, we use 0.875 for β . Preliminary testing results of the heuristic algorithm showed that the results do not change significantly for β values from
425 0.5 to 0.90. If the coverage is high enough, we go into customer grouping as described in Section 5.5. If not, we need to find another route center.

There is a possibility that customer grouping causes the coverage to go down or make it stagnate. If this is the case, we need a new route center as well. The dual solution of the master LP model is used in both finding new route centers
430 and in customer grouping.

Complexity and Time. As seen in the section above, there are four points at which we can do an LP iteration: after the route construction in the Warm-Up and Saturated phases, and during the finding of a new route center in these two phases as well. It can be seen that the number of LP iterations, and thus the
435 time required, heavily depends on the size of IND . If this is large and close to L_{Veh} , less route centers have to be found and so less LP iterations need to be done. Complex problems will require more iterations to solve. For example, looking forward to Section 6, c101 only needs one LP iteration to solve. The quicker we can get out of the Warm-Up phase, the quicker we will be done with
440 the program.

Since there are a lot of variables, they are compiled in Table 4 for easy lookup.

5.2. Route construction.

445 In the following section, we will look more closely at how exactly the routes are constructed for the linear program. As was detailed in Section 5, these routes one half of the objective value. At any time, we are trying to minimize the total length of all selected routes. To do that, we try to make the routes as well as possible.

Sets	
N'	Set of customer locations, $N' = N \setminus \{0\}$
R	Set of all routes
$R_{rc,i}$	Set of routes going through route center rc , of length i
K_{rc}	Set of price-balanced paths going through route center rc
Customer $c \in N'$	
sv_c	Service time
q_c	Demand
e_c	Earliest arrival time
l_c	Latest arrival time
$start_c$	Start time
c_{comp}	Number of competitors
Route $r \in R$	
$l(r)$	Length
C_r	Set of customers
r_i	Customer at position for $i \in 1, \dots, l(r)$
r_{qual}	Quality
r_{gb}	Beginning of grouped part
r_{ge}	Ending of grouped part
Route center rc	
rc_{cp}	Length of central path
rc_{gr}	Number of grouped customers
Misc.	
Q	Capacity of a vehicle/route
$d_{i,j}$	Euclidian distance between customers i and j
L	Size of route set
β	Coverage threshold
ϵ	Lag

Table 4: Reference Table for customers, routes and sets

450 Our route construction method has certain similarities to the one proposed
 by Kok10. In their proposed DP heuristic, the number of states at every iteration
 is bounded, and the expansion of states is limited by a constant number.
 However, in their case, they are looking at a situation with driving hour regulations.
 In our heuristic, we put a bound on both the total number of states
 455 (routes), as well as the number of routes at every iteration of the route construction.
 We will take into account the time windows of customers, as per the VRPTW.

To do that, we propose a greedy method of route construction. Starting
 from the building block of only having the route center, step by step we extend
 460 the routes by adding one customer at a time. Because we are using a greedy
 method, we end up with good routes of a smaller length as well. This means
 that while the linear program is being solved, it also has a chance to take these
 good shorter routes into account. That adds a layer of flexibility to the method,
 which wouldn't be there if we only considered longer routes.

465 In the ILP reformulation we used within our algorithm, the routes are
 grouped for every route center in RC . Let \mathcal{R}_{rc} denote the routes set of route
 center rc . The routes in \mathcal{R}_{rc} are constructed greedily such that $\mathcal{R}_{rc} = \bigcup_i \mathcal{R}_{rc,i}$
 where i denotes the length of routes in subset $\mathcal{R}_{rc,i}$. It is important to mention
 that the subsets are fixed size, i.e. $|\mathcal{R}_{rc,i}| = L$ where L a predetermined size
 470 depending on the dedicated memory.

Central Paths, Grouped Paths and Full Paths. There are two phases in route
 construction, based on the customer grouping. In the first phase, the *central
 paths* are extended to *grouped paths*, by doing extensions with only the grouped
 customers. More information on the grouping can be found in Section 5.5.

475 (Central Path) A sequence of customers in a route that contains only a
 specific number of “fixed” customers is called a central path of that route.

The central paths are chosen from the paths that were created during route
 construction. The paths that are chosen all come from the grouped paths:

(Grouped Path) A sequence of customers in a route that contains all “fixed”
480 customers is called a grouped path of that route.

Grouped paths are created during the route construction phase, and consist only of grouped customers. Central paths are chosen during the customer grouping, and are the foundation of the route construction, reducing the number of possibilities and thus allocating more memory to the creation of new paths.
485 Grouped paths are then extended into full paths. It can be the case that a grouped path is already a full path. If this is the case, all possible customers have been grouped.

The length of a central path of a route center is denoted by rc_{cp} . The number of grouped customers of a route center is denoted by rc_{gr} . Initially, $rc_{cp} = 1$
490 and $rc_{gr} = 0$. In the second phase, the grouped paths are extended to full paths, which also include customers that are not grouped with the route center. So, for a route center $rc \in RC$, $\mathcal{R}_{rc,i}$ where $rc_{cp} \leq i \leq rc_{gr}$ only uses grouped customers. For $i > rc_{gr}$, all customers are considered for extensions.

Next, we will take a look at exactly how the extensions work in the heuristic.
495 Extensions are done in both directions, so forward and backward, in that order.

Forward Extensions. The procedure for finding forward extensions can be found in Algorithm 2. For $\mathcal{R}_{rc,l}$, we firstly take the routes of $\mathcal{R}_{rc,l-1}$ and extend them based on the pseudocode. Line 4 checks if the customer is already in the route. Line 5 checks that the earliest time we can make it to the customer is at most
500 the latest arrival time of this customer, i.e. we do not violate time windows. Line 6 checks the capacity of the vehicle is not exceeded. Lines 7 and 8 deal with making it back to the depot and having enough time to start from the depot, respectively.

Data: route center rc , length l , Route set $\mathcal{R}_{rc,l-1}$, Customers N'

Result: Route set $R_{rc,l}$

```

1 Create empty route set  $R_{rc,l}$ ;
2 forall  $r \in R_{rc,l-1}$  do
3     forall  $c \in N'$  do
4         if  $c \in C_r$  or  $c = rc$  then Continue;
5         if  $e_{r_{l(r)}} + sv_{r_{l(r)}} + d_{r_{l(r)},c} > l_c$  then Continue;
6         if  $\sum_{i=1}^{l(r)} q_{r_i} + q_c > Q$  then Continue ;
7         if  $e_{r_{l(r)}} + sv_{r_{l(r)}} + d_{r_{l(r)},c} + sv_c + d_{c,0} > l_0$  then Continue;
8         if  $\min(l_c - d_{r_{l(r)},c} - sv_c - \sum_{i=1}^{l(r)} d_{r_i,r_{i+1}}, l_{r_{l(r)}}) - d_{0,r_1} \leq e_0$  then Continue;
9         Extension is considered feasible;
10        Determine quality value  $r_{qual}$ ;
11        if  $|R_{rc,l}| < L$  then
12            | Insert  $r$  into  $R_{rc,l}$ ;
13        else if  $r_{qual} > \text{lowest quality found in } R_{rc,l}$  then
14            | Insert  $r$  into  $R_{rc,l}$ , kick out route with lowest quality;
15        else Continue;
16 Output  $R_{rc,l}$ ;
```

Algorithm 2: Extending routes forward

505 All feasible extensions are given a quality value, as is seen on line 10. This quality value of route r with route center rc is measured in two parts:

- $\sum_{i=1}^{r_l-1} d_{r_i,r_{i+1}}$. The total traveled distance of the extension. This is looked at normalized against other found extensions. We want to minimize this, to be as efficient as possible.
- 510 • $\frac{l_0 - (e_{r_{l(r)}} + sv_{r_{l(r)}} + d_{r_{l(r)},c} + sv_c)}{l_0 - (l_{rc} + sv_{rc})}$. The time spent in the route, referenced against the time that is left to get to the depot from the route center. In other words, how efficiently you are using your time.

The quality can also be considered as a detour cost, or how efficient it is to visit that specific customer.

515 As mentioned in Section 2.3, we only allow L routes of length i for a certain route center. To determine if a route is selected, we look at the quality as detailed above. If $|R_{rc,i}| < L$ so far, we add the new route to $R_{rc,i}$. If $|R_{rc,i}| = L$, we check where to insert the new route based on the quality it has. If it has a

lower quality than the last route in $R_{rc,i}$, we do not put it in the route set. If
520 it is inserted, we kick out the route with the lowest quality. This is represented
in Algorithm 2 by lines 11-15.

Backward Extensions. For backward extensions, we consider a very similar
setup as for the forward extensions, so we will not go into it in detail. It is
depicted in Algorithm 3 and works very similar.

525 The quality of a backwards extension is made similarly to that of a forward
extension. For backwards extensions, we consider the same set $R_{rc,l}$ as for the
forward extensions. The set will thus end up as a good mixture between these
two methods.

It can be the case that we create an extension that is already existing in
530 $R_{rc,l}$. If this happens, we filter out the newly created one and just continue
with the heuristic. This filtering is done by checking whether two routes are the
same based on the arrival, ending, and sum of IDs of all customers in the route.

Data: route center rc , length l , Route set $\mathcal{R}_{rc,l-1}$, Customers N'
Result: Route set $R_{rc,l}$

- 1 Get route set $R_{rc,l}$, which is filled with forward routes.
- 2 **forall** $r \in \mathcal{R}_{rc,l-1}$ **do**
- 3 **forall** $c \in N'$ **do**
- 4 **if** $c \in C_r$ **or** $c = rc$ **then** Continue;
- 5 **if** $l_{r_1} - d_{c,r_1} \leq e_c + sv_c$ **then** Continue;
- 6 **if** $\sum_{i=1}^{l(r)} q_{r_i} + q_c > Q$ **then** Continue ;
- 7 **if** $\min(l_{r_1} - d_{c,r_1} - sv_c, l_c) - d_{0,c} \leq l_0$ **then** Continue;
- 8 Extension is considered *feasible*;
- 9 Determine quality value r_{qual} ;
- 10 **if** $|R_{rc,l}| < L$ **then**
- 11 | Insert r into $R_{rc,l}$;
- 12 **else if** $r_{qual} > \text{lowest quality found in } R_{rc,l}$ **then**
- 13 | Insert r into $R_{rc,l}$, kick out route with lowest quality;
- 14 **else** Continue;
- 15 Output $R_{rc,l}$;

Algorithm 3: Extending routes backward

Insertions. Insertions are different from extensions in the way they approach
 535 previous routes and how to check for them. Here, we will describe why these
 insertions are necessary and how they function.

Insertions are closely linked to the customer grouping as described in Section
 5.5. In previous iterations of the heuristic, it was observed that the customer
 grouping is not flawless. Specifically, what seemed to happen was that some
 540 customers in important routes would be "skipped" for customer grouping. This
 caused the heuristic to not come to a complete integer solution. One way to
 ensure feasible solutions would be found, is by using *insertions* into existing
 routes.

Before doing the insertions themselves, we have to take a look at which
 545 customers we are actually going to consider. To do this, there is a simple
 heuristic to find candidates. If we are looking to create insertions of length l ,
 we look into the route set $R_{r_c, l-1}$. We iterate over all customers $c \in N'$ and
 then routes $r \in R_{r_c, l-1}$. For every position in between two customers r_i, r_{i+1} ,
 we then check if there exists an arc between r_i and c and an arc between
 550 c and r_{i+1} . This would mean there is actually space to insert this customer.
 Then, we check if the added distance of inserting this customer would not be
 too much, compared to the rest of the route. If this is the case, we add the
 customer to the set *Insertioncustomers*. It can be seen that if a customer is
 in *Insertioncustomers*, it can be inserted in at least one path r of length $l - 1$.

555 Finding the insertions themselves works very similarly. For every customer
 $c \in \text{Insertioncustomers}$, we look at every route r in $R_{r_c, l-1}$, and we check
 every position $i = 1, 2, \dots, l(r)$. We consider an insertion in between customers
 r_i and r_{i+1} *feasible* if:

- There exists an arc from customer r_i to c .
- 560 • There exists an arc from c to customer r_{i+1} .
- There is enough time after r_i to insert a customer.
- $\sum_{i=1}^{l(r)} q_{r_i} + q_c \leq Q$. The added capacity can still fit in the vehicle.

- $c \notin C_r$. The customer is not already in the route.
- c is not already grouped with a route center.
- 565 • $e_{r_i} + sv_{r_i} + d_{r_i,c} < l_c$. The arrival at customer c does not exceed its time window.
- Coming back to the path at r_{i+1} is possible within the timeframe.
- The quality is satisfied.

If an insertion is feasible, it is added to $R_{rc,l}$.

570 Already, it can be seen that insertions would take a long time if we allow them at every length l . Therefore, after testing, it was deemed necessary only in one situation: the route construction iteration *after* all grouped customers have been used.

Route Set. The route set $\mathcal{R}_{rc,2}$ is fully enumerated in order to be able to get dual solutions with as much as information possible. As mentioned before, route sets 575 $\mathcal{R}_{rc,i}$ for $i \geq 3$ may be huge in size in general. Hence we require that $|\mathcal{R}'_{rc,i}| \leq L$ for $i \geq 3$ where L is a constant number, based on the memory that has been used in the route construction so far. For route centers with a short expected route length, this means more memory is allocated to each possible length. So 580 \mathcal{R}'_{rc} becomes a fixed-size route set of the route center rc which results in a fixed-size MLP model containing the selection variables of routes in sets $\mathcal{R}'_{rc,i}$, for all $i \geq 1$, and $rc \in RC$.

5.3. Solving the Master LP

In this section, a brief overview is given on what happens after the routes 585 have been constructed. We will not go into detail on the exact workings of the program, but will mention the information that will be used by the customer grouping and new route center finding.

Information. Before the LP has been solved, we retrieve two pieces of information. Firstly, we set the γ values of all customers to the ones found by Gurobi. Subsequently, the y_i values are recorded as well for later use. Secondly, the price-balanced paths are determined.

(Price-balanced path) A path with a reduced cost of 0. Only price-balanced paths can be selected by the LP.

As seen in Section 4, we call the set of all price-balanced paths K .

Solving the LP. The master LP is solved after route construction has finished. In this way, the system can immediately use the most recent information available.

The LP iteration first has to be set up by the system. We make use of Gurobi v9.1. The primal is created and initialized, after which the customers and their constraints are added to it. From there, Gurobi solved the LP iteration with the primary goal of maximizing the coverage, and the secondary goal of minimizing the route length as discussed in Section 4.

5.4. Route Center Selection

In this section, we go over the procedure for selecting a new route center. As seen in Figure 2, this happens in two phases of the program. First we look at finding the candidates, then at the actual selection. Note that during the selection, we have another LP iteration to find the best suitable new route center.

Finding Candidates. The procedure of selecting a new route center begins with determining route center candidates.

(Route Center Candidate) A customer $c \notin RC$ that has evidence of being a good route center, based on dual values.

For each existing route center, a network is created, similarly to the one used in customer grouping in Section 5.5. This network only counts the number of

615 appearances of each customer c in the price-balanced routes of the route center.

So, for customer c , that would be $\sum_{r \in K_{rc}} \delta_r^c$.

From there, the number of competitors is determined for each c .

(Competitor) A customer $c' \in N'$, $c' \neq c$, $c' \notin RC$ which never appears in K_c . The number of competitors of c is denoted by c_{comp} .

620 c_{comp} is found by going through all routes $r \in K_c$, and checking which customers appear. The interpretation is that, the more competitors there are, the fewer different customers c sees. If a customer does not appear in many different routes of a route center, it means it is more likely to not belong to that route center.

625 Let $CAND$ demote the set of route center candidates. The customer list is sorted on the competitive value, calculated as:

$$competitiveValue = \frac{c_{comp}}{\max_{c' \in N', c' \neq c, c' \notin RC} c'_{comp}} \quad (46)$$

If there is room in $CAND$ and c is not close to any $c' \in CAND$, c is added to $CAND$.

Selecting a new Route Center. After the candidates have been selected and put
630 into $CAND$, we start route construction once more. This time, we include every $cand \in CAND$ as a candidate route center in the LP, with each $cand$ being given a fraction of the memory for a normal route center. Routes are created for these $cand$ based on the methodology found in Section 5.2.

635 Based on this, we have found a new possible solution. We have to bound the number of candidates that can be selected in the MLP model. To do that, we add a constraint:

$$\sum_{r \in \mathcal{R}_c: c \in CAND} x_r \leq 1 \quad (47)$$

Per iteration of finding a new route center, we set the limit of selection to 1. This is so that we don't add unnecessary route centers. The candidate that is chosen is the one with the highest primal selection value. Ties are broken by

640 the size of K_c . Further ties are broken by the γ^* values of customers. It can be seen that this is considered as another iteration of the linear program.

Once a new route center has been found, the constructed routes of all route centers and candidates are removed. The candidates are removed from *CAND*. Afterwards, the route sets R_{rc} for every route center is constructed once more
645 and the LP is solved.

5.5. Customer grouping

In this section, we will look at the customer grouping procedure. As seen in Figure 2, this happens during the Saturated phase of the program.

In Section 3, we mention several studies using variable fixing. The main
650 advantage of variable fixing is to reduce the problem size and focus the heuristic's efforts on finding the solution. In this work, we exploit the structure of the dual model as explained in Section 4. The main idea of customer grouping is to assign a customer to a route center, such that all routes of the route center have to include that customer. By doing that, the number of possible routes gets
655 restricted and the number of variables for each route centers is reduced. The decision on which customer to group is based on the information from the dual.

The main motivation for doing customer grouping is that, by reducing the number of variables, we are able to reduce the problem size and be more focused on the solution. We use the information from the dual, and introduce the
660 concept of grouping scores. After that, we look at how exactly the central paths are being set up for use in route construction.

Grouping Scores. To confidently make a grouping decision, we need evidence that grouping a certain customer will actually help us in finding the optimal solution. This evidence comes in the form of the grouping scores:

665 (Grouping Score) A quantified value of the confidence of assigning a customer to a route center, based on the dual information.

In this section, we will explain how exactly these scores are calculated and what impacts these scores have on grouping decisions.

First, we must find a base on which to build the grouping scores. To do
670 that, we introduce the concept of a network for a route center rc . In Section
4, we introduced the set K , which contains all price-balanced paths. Let K_c be
the set of price-balanced paths that go through customer $c \in N'$, and K_{rc} be
the price-balanced paths that include route center $rc \in RC$. Then, we find the
network as:

675 vspace2mm (Network) Representation of K_{rc} , where scores and tallies of
all customer in N' are being recorded.

For the customer grouping, we consider a special kind of this network, called
a weighted dual network. In a weighted dual network, we specifically go through
the routes to find the dual scores. The heuristic goes through all $r \in K_{rc}$ one
680 by one.

One of the aspects of the grouping score has to do with the relative closeness
to the grouped section of a route. The grouped section is the subsection of
the route that contains only grouped customers. This grouped section has a
beginning index r_{gb} and an end index r_{ge} . It holds that $r_1 \leq r_{gb} \leq r_{ge} \leq r_l$.
685 We call the closeness to the grouped section of a route of customer i : $r_{i_{cl}}$. The
scores are determined as:

- For $r_i < r_{gb}$, $r_{i_{cl}} = r_{gb}/(r_{gb} - r_i)$. Here, r_{gb} is the length of the tail on the
front.
- For $r_{gb} \leq r_i \leq r_{ge}$, $r_{i_{cl}} = 0$.
- 690 • For $r_{ge} > r_i$, $r_{i_{cl}} = (l - r_{ge})/(1 + r_i - r_{ge})$. Here, $(l - r_{ge})$ is the length of
the tail at the end of the route.

As can be seen, the closer the customer is to the grouped section of a route, the
smaller the score. Therefore, while the heuristic algorithm calculates scores for
all customer-route center pairings, the customers visited right before and after
695 route centers have the highest probability of having these good scores.

After the network is created, every customer c has three items recorded:

- $\sum_{r \in K_{rc}} \delta_r^c$. The number of appearances of this customer in the price-balanced paths.
- $\sum_{r \in K_{rc}} r_{ccl}$. The cumulative closeness of this customer to the grouped sections of the routes.
- $\sum_{r \in K_{rc}} x_r$. The primal selection values of the routes.

700

Based on the network, scores can be calculated for the route center. One score is determined for every customer that has been seen by the route center at least once. There are three components to the score:

705

1. The *stability* of the route center, which is the γ value relative to all other route centers' γ . What this component measures is how 'confident' the LP is in assigning customers to this route center overall.
2. The *primal values* of the customer, as determined by the networking strategy above in $\sum_{r \in K_{rc}} x_r$. If a visit is strongly selected in the primal solution, that provides evidence to the claim that grouping is a good idea.
3. The *dual competition* of the customer, which is the relative closeness $\sum_{r \in K_{rc}} r_{ccl}$ as described above. This is rooted in the dual values.

710

Each of these components has a weight associated with them as well. All of these weights are also determined by the total coverage achieved so far. For example, the stability of the route center is important if the coverage is very low, since we will have to find new route centers soon. Similarly, the primal values are more important when the coverage is high, since there will be a lot of competition. Note that only a customer with good scores in all three of these components is considered to have a strong grouping score.

715

720

After this process is repeated for every route center, all the customer-route center pairings are sorted based on their score. Those customers with a score higher than the threshold α will be grouped. In our implementation, $\alpha = 0.7$. Every grouped customer is then assigned to its route center.

725

Central Route Setting. From this point on, we consider the setting of central paths. As discussed in Section 5.2, central paths are the building blocks of

routes, from which the rest follows. One could ask why we need central paths in the first place. After all, if you start with creating routes using only grouped customers, the variables are sufficiently bounded. While that is true, the reason we want central routes is twofold.

730 Firstly, customer grouping works best when we have a lot of customers grouped with one route center. However, what happens then is that the route construction does not gain much from having the grouped customer. There would be many possible routes of a short length. Making paths using 3 grouped customers is quick; making them using 30 could take a long time.

735 Secondly, if we manage to set central paths of a longer length, this frees up memory for the rest of the route construction. Recall that L is the number of paths that can exist for routes of length l . L is based on the allocated memory for the route center, divided in such a way to let routes flourish. If we are able to set the central paths, our building block, to become longer, the memory that
740 has been freed up and can be used for the rest of the route set. Therefore, setting the central path lets us explore more possibilities, while not increasing the total size of the route set.

Now that we know why, let us look at how exactly we determine the central paths for a route center. After customer grouping has been finished, we go
745 over all route centers $rc \in RC$ in the system. As we saw before, the number of grouped customers for a route center is rc_{gr} . If new customers have been grouped with this route center, we need to consider if we want to set new central paths as well.

After experimentation, the result of this is the concept of lag. Lag is the
750 amount of difference between the length of the central path and the total number of grouped customers for this route center. In other words, $rc_{gr} - rc_{cp} = \epsilon$, where ϵ is the lag between the two. The advantage of using this notion of lag is that we restrict the number of possible routes of a shorter length, but allow more flexibility in routes of a longer length.

755 We check whether we have exceeded ϵ after customer grouping. If this is not the case, we can simply return to the route construction. If ϵ is indeed

exceeded, we go to select the new route centers. We first determine how much longer the central paths will be. This is equal to $rc_{gr} - \epsilon = l$. We take the paths of this length l and of them, set the top k as central. In our implementation, k
760 $= 5$. The paths are retrieved from the previous iteration of route construction. Once we find the new central paths, we can start the next iteration of route construction using these as the building block. These central paths are added to the LP as well, even if they are not price-balanced.

6. Computational results

765 We implemented the proposed heuristic algorithm in Java coding environment. During the course of the algorithm, all MLP models are solved by using Gurobi 9.1. The results that are presented in this section are obtained by using a personal computer with Intel i5 3.1 GHz Processor, and 16GB capacity of RAM.

770 100-customer instances of Sol87 are used in our experimentation as benchmark instances. Table 5 gives preprocessing values of the instances. For example, C101 has initially the number of route centers equal to the lower bound on the number of vehicles. In instances C107-C109 and C204-C208 there is only one route center in the beginning. The high number of MLP iterations for these
775 instances are due to runs of the model to find new route centers.

Table 6 shows the results found by our implementation. In a group of instances, the columns show the names of instances, solution properties of our heuristic; first number of vehicles and second traveled distance, and the percentage gap between the best known solution in travel distances. Finally, we
780 list number of MLP solving iterations for the instances in the last column of an instance.

The results of the C10X instances show that, while they can take a bit of time, all except one solve to optimality. For C20X, one instance clearly stands out. C204 optimally only has three vehicles, and so the optimality gap here is
785 considered to be very large. For the other instances, the biggest gap found was

Table 5: Maximal Independent Sets, lower bound of number of vehicles

Instance	$ IND $	L_{veh}	Instance	$ IND $	L_{veh}
C101	10	10	C201	2	3
C102	8	10	C202	2	3
C103	7	10	C203	2	3
C104	4	10	C204	1	3
C105	5	10	C205	1	3
C106	4	10	C206	1	3
C107	1	10	C207	1	3
C108	1	10	C208	1	3
C109	1	10			

3.7% for C203 and 4.7% for C205, and all others are at 1.0% or below. For C202, C203 and C204, the running times are quite large. Many iterations of customer grouping had to be performed in these instances before getting to a solution with complete coverage. They get faster over time as central paths get longer and more variables are fixed.

7. Conclusions and further research

This paper has two main contributions. Firstly, it provides a dual analysis of a master LP formulation of the VRPTW by using the primal-dual method. Secondly, a heuristic algorithm of the VRPTW is proposed by using the understanding from the dual analysis. In the dual analysis it is observed that the dual solution is the final state of a pricing competition among customers to appear in primal route of route centers.

The most popular route center has a high dual decision variable, which means that the primal solution of that route center is not reliable to make customer grouping decisions. On the other hand, the existence of desperate customers in the dual solution, i.e. customers with highly negative dual variables, provides

Table 6: 100-customer Solomon benchmark instances

	LP Heuristic		Time*	Iter		LP Heuristic		Time*	Iter
Ins.	V./D.	d(%)			Ins.	V./D.	d(%)		
C101	10/828.94	0.0	2.14	1	C201	3/591.56	0.0	8.21	3
C102	10/828.94	0.0	9.43	5	C202	3/591.56	0.0	106.46	10
C103	10/828.94	0.0	15.40	7	C203	3/614.75	3.7	177.81	19
C104	10/826.83	0.24	22.62	13	C204	4/771.	N/A	104.23	23
C105	10/828.94	0.0	12.76	13	C205	3/616.24	4.7	23.58	7
C106	10/828.94	0.0	10.79	13	C206	3/592.05	0.5	11.21	5
C107	10/828.94	0.0	11.90	19	C207	3/594.11	1.0	15.876	6
C108	10/828.94	0.0	13.70	19	C208	3/593.82	0.8	10.96	5
C109	10/828.94	0.0	17.00	19					

*in seconds.

us an evidence for increasing the number of used vehicles. We underline the fact
805 that all conclusions we present in this paper depend on the quality/size of the
fixed-size route sets.

The proposed heuristic algorithm uses a master LP model of the VRPTW
as a central optimization mechanism. It finds a complete routing solution by
making customer grouping decisions by checking the evidence by not only using
810 the primal selection values, but also the dual solution properties of the master
LP model. To the best of our knowledge, this work is the first one incorporating
the properties of the dual solution in solving routing problems. We believe that
several issues in the proposed heuristic algorithm can be improved, leading to
better results. Firstly the speed needs to be looked at, for example my limiting
815 the number of times route construction is done. Limiting the number of LP
iterations could work in our favor as well, though we should be careful with this
since it could influence the grouping decisions being made.

The master LP model under consideration of this paper can be used for sev-
eral extensions of the VRPTW that involve real-life aspects like time-dependent
820 travel times, stochastic travel times, and driving time regulations. These aspects
of the problem will then be considered in the route construction routine of the

heuristic.

Finally, our dual analysis can be used to analyze other Operations Research problems that can be formulated as a master LP model. Then the dual interpretation will be useful in understanding the underlying processes resulting in the dual solution. This may give an opportunity to develop similar heuristic approaches for these problems.

8. Bibliography

References

- [1] Baldacci, R., Mingozzi, A. Roberti, R. 2011, "New route relaxation and pricing strategies for the vehicle routing problem", *Operations research*, vol. 59, no. 5, pp. 1269-1283.
- [2] Bertsimas, D., Tsitsiklis, J., 1997, "Introduction to Linear Optimization", *Athena Scientific, Belmont, Massachusetts*.
- [3] Bianchessi, N., Irnich, S., "Branch-and-Cut for the Split Delivery Vehicle Routing Problem with Time Windows", *Transportation Science* Vol 53. Issue 2, pp. 179-192
- [4] Bräysy, O., Gendreau, M., 2005, "Vehicle routing problem with time windows, Part I: Route construction and local search algorithms", *Transportation Science* Vol.39, pp. 104-118.
- [5] Bräysy, O., Gendreau, M., 2005, "Vehicle routing problem with time windows, Part II: Metaheuristics", *Transportation Science* Vol.39, pp. 119-139.
- [6] Cacchiani V.,Hemmelmayr, V.C., Tricoire, F., 2014. "A set-covering based heuristic algorithm for the periodic vehicle routing problem", *Discrete Applied Mathematics*, 163, pp. 53-64.
- [7] Costa, L., Contardo, C., Desaulniers, G., "Exact Branch-Price-and-Cut Algorithms for Vehicle Routing", *Transportation Science* Vol 54, Issue 4

- [8] Dantzig, George Bernard, Ramser, John Hubert, 1959. "The Truck Dispatching Problem", *Management Science*, 6, pp. 80-91.
- 850 [9] Dantzig, G., B., Ford, L., R., Fulkerson, D., R., 1956. "A primal-dual algorithm for linear programs", *Linear Inequalities and Related Systems*, Editors: H.W. Kuhn and A.W. Tucker, Princeton University Press, Princeton, NJ, pp. 171-181.
- [10] Desaulniers, G., Gschwind, T., Irnich, S., 2020, "Variable Fixing for Two-Arc Sequences in Branch-Price-and-Cut Algorithms on Path-Based Models", *Transportation Science* Vol. 54, Issue 5, pp. 54
- 855 [11] Desrochers, M., Lenstra, J.,K., Savelsbergh, W.,P., 1988. "Vehicle Routing With Time Windows: Optimization and Approximation, *Vehicle Routing: Methods and Studies*, pp. 65-84.
- [12] Desrochers M. , Desrosiers J. , Solomon M., 1992. "A new optimization algorithm for the vehicle routing problem with time windows., *Operations Research*, pp. 342-354.
- 860 [13] Fisher, M., L., Jornsten, K.,O., Madsen, O.,B.,G., 1997. "Vehicle routing with time windows: Two optimization algorithms, *Operations Research*, pp. 488-492.
- 865 [14] Gehring, H., Homberger, J., 2001, "A Parallel Two-phase Metaheuristic for Routing Problems with Time Windows", *Asia-Pacific Journal of Operational Research*, Vol. 18, pp. 35-47.
- [15] Goemans, M., X., Williamson, D. P., 1996, "The primal-dual method for approximation algorithms and its application to network design problems", *In D. S. Hochbaum, editor, Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, 1996.
- 870 [16] Günlük, O., Kimbrel, T., Ladanyi, L., Schieber, B., Sorkin, G. B., 2006, "Vehicle Routing and Staffing for Sedan Service", *Transportation Science*, Vol. 40, pp. 313-326.
- 875

- [17] Huang, K., Hsu, C-P., 2011, "A Lagrangian Heuristic for the Vehicle Routing Problems with the Private Fleet and the Common Carrier", *Journal of the Eastern Asia Society for Transportation Studies*, Vol. 9, pp. 644-659.
- [18] Kallehauge, B., Larsen, J., Madsen, Oli B.G., 2006, "Lagrangian duality applied to the vehicle routing problem with time windows", *Computers & Operations Research*, Vol.33, pp. 1464–1487.
- [19] Kalina, P., Vokřínek, J. Mařík, V. 2015, "Agents toward vehicle routing problem with time windows", *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 19, no. 1, pp. 3-17.
- [20] Kohl, N., Madsen, Oli B.G., 1997, "An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation", *Computers & Operations Research*, Vol.45, pp. 395–406.
- [21] Kok, A.L, Meyer, C. M., Kopfer, H., Schutten, J. M. J., 2010, "A Dynamic Programming Heuristic for the Vehicle Routing Problem with Time Windows and European Community Social Legislation", *Transportation Science*, Vol.44, pp. 442-454 .
- [22] Laporte, G., 1992, "The Vehicle Routing Problem: An overview of exact and approximate algorithms", *European Journal of Operational Research* Vol.59, pp. 345-358.
- [23] Liu, R. Jiang, Z. 2019, "A constraint relaxation-based algorithm for the load-dependent vehicle routing problem with time windows", *Flexible Services and Manufacturing Journal*, vol. 31, no. 2, pp. 331-353.
- [24] Nagata, Y., Bräysy, O., 2009, "A powerful route minimization heuristic for the vehicle routing problem with time windows", *Operations Research Letters* Vol.37, pp. 333–338.
- [25] Schneider, M., Stenger, A., Goeke, D., 2014, "The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations", *Transportation Science*, Col. 48, Issue 4, pp. 500-520

- [26] Solomon, M., 1987. "Algorithms for the vehicle routing and scheduling
905 problem with time window constraints", *Operations Research*, 35, pp. 254-
265.
- [27] Ukhan, N., Kondratenko, G., Sidenko, I. Kondratenko, Y. 2020, "ICT for
planning and optimization of transport routes with time windows", CEUR
Workshop Proceedings, pp. 365.
- 910 [28] Xu, S., Liu, J., Zhang, F., Wang, L. Sun, L. 2015, "A Combination of ge-
netic algorithm and particle swarm optimization for vehicle routing problem
with time windows", *Sensors (Switzerland)*, vol. 15, no. 9, pp. 21033-21053.