Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 2655–2669
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.23209.2493

# Developing a green vehicle routing problem model with time windows and simultaneous pickup and delivery under demand uncertainty: Minimizing fuel consumption

Mohammad Bagher Fakhrzad*, Seyed Masoud Hoseini Shorshani, Hasan Hosseininasab, Ali Mostafaeipour

*Department of Industrial Engineering, Yazd University, Yazd, Iran*

*(Communicated by Ali Jabbari)*

**Abstract**

The vehicle routing problem has attracted much attention in the recent decade. Considering the real-world constraints, many extensions have been developed. This paper develops a new model for the green vehicle routing problem with simultaneous pickup and delivery under demand uncertainty. Due to the problem's complexity, the standard solvers are only able to solve small-scale instances. To solve the large-scale problems, a two-stage algorithm based on the modified AVNS is proposed. Extensive computational experiments are conducted using modified versions of Solomon's benchmark instances to show the performance of the algorithm. The results affirm that the two-stage algorithm is capable of generating optimal solutions for small-size instances and the planned routes generated for large-size instances were significantly more robust against the increase of uncertainty parameters.

Keywords: Vehicle routing problem, time window, demand uncertainty, simultaneous pickup and delivery
2020 MSC: 68T37

## 1 Introduction

In the optimization area, the term vehicle routing problem (VRP) refers to a well-known scheduling model that aims at finding the best routes for distributor vehicle(s) in which the overall cost of distribution is minimized [12]. Considering the wide-ranged applications, many efforts have been done in the VRP area to take the real-world assumptions into account. Among the proposed models, the vehicle routing problem with time windows (VRPTW) and its extension, pickup, and delivery problem (PDP) have attracted much attention from researchers during the past decade. According to the best knowledge of the authors, [28] and [15] conducted the earliest studies and put forward in this area. PDP can be grouped into four main classes: (1) vehicle routing problem with clustered backhauls (VRPCB) (2) vehicle routing problem with mixed linehaul and backhauls (VRPMB) (3) vehicle routing problem with divisible delivery and pick-up (VRPDDP) (4) vehicle routing problem with simultaneous delivery and pick-up (VRPSDP) [26]. Vehicle routing problem with simultaneous pickup and delivery under time windows limitation (VRPSPDTW) introduced by Ai and Kachitvichyanukul [2] and Wang and Chen [33] that is each customer should be visited within its time windows. In classical formulation, customers' demand and pickup are known and deterministic. However, such

*Corresponding author
*Email addresses:* mfakhrzad@yazd.ac.ir (Mohammad Bagher Fakhrzad), hosseinimasoud@stu.yazd.ac.ir (Seyed Masoud Hoseini Shorshani), hhn@yazd.ac.ir (Hasan Hosseininasab), mostafaei@yazd.ac.ir (Ali Mostafaeipour)

*Received:* February 2021 *Accepted:* April 2021

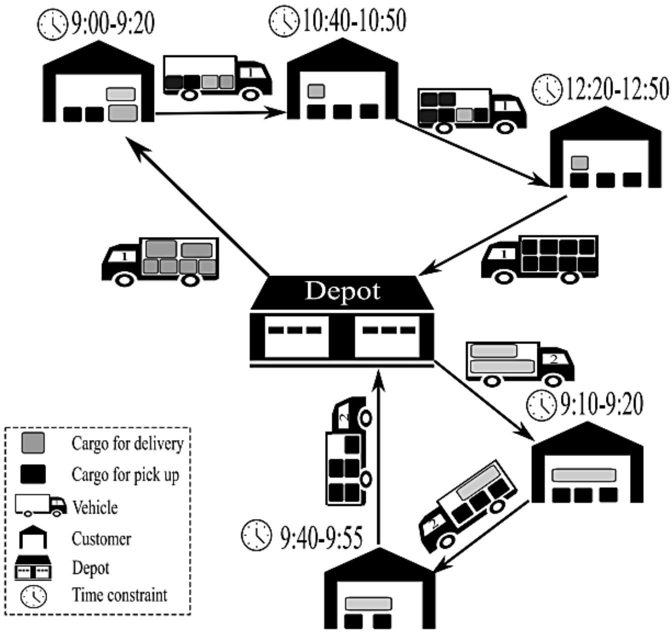


Figure 1: An example of the VRPSPDTW

an assumption is obviously violated in the real-case studies. For instance, a number of waiting students in a station is not decisive [10]. As another example, healthcare logistics medication demand for suppliers is not certain [8].

## 2 Literature review

Generally speaking, VRPTW includes two modes, (1) soft time windows in which vehicles are allowed to visit customers out of the predefined windows enduring a penalty, and (2) hard time windows in that vehicles must visit customers within the time windows. For example, postal deliveries, security patrol service, urban newspaper distribution, grocery delivery, bank deliveries, school bus routing, and industrial refuse collection have hard time windows constraints [32]. VRPTW have been studied by the researchers during the past decades [19, 14, 5, 31, 24, 11]. An extended version of VRPTW, say VRP with simultaneous pickup and delivery was firstly introduced by [23]. This problem is generally defined as a pickup and delivery problem, which has been applied in many fields, such as the grocery delivery system, parcel delivery, and home health care services. If the delivery and pickup services are demanded in the logistics system, we call this pickup and delivery problem. When the customers have pickup requests and delivery requests simultaneously, the variant pickup and delivery problem become a vehicle routing problem with simultaneous pickup–delivery (VRPSPD) and if the customers have time windows constraints, the variant VRPSPD is defined as a vehicle routing problem with simultaneous pickup–delivery with time window (VRPSPDTW).

Increasing the traffic congestion and negative environmental impact of transportation motivated the researchers to develop economic models in the VRP area. The term green vehicle routing problem (GVRPs) is a major key to reducing hazardous effects of transportation, such as air pollution, greenhouse gas (GHG) emissions, noise, and so on [23]. Moghdani et al. [25] classified GVRPs into eight groups considering the objective of the problems:

- Green vehicle routing problem
- Pollution-routing problem
- Green heterogeneous vehicle routing problem
- Energy minimizing vehicle routing problem
- Time-dependent vehicle routing problem
- Fuel consumption in vehicle routing problem

Table 1: Outline of different green VRP classes

| Variants of GRVP | Objective | Data/Case study | Uncertainty | Solution approach | CO2 calculation method |
|---|---|---|---|---|---|
| Green VRP | Single | Case study | Deterministic | Metaheuristic | Factor |
| Heterogeneous | Multiple | Experimental | Nondeterministic | Software | Fuel consumption model |
| Pollution-Routing | Many obj | Theoretical | - | Exact solver | Calculate emissions from road transportation |
| Energy Minimization | - | - | - | Exact | Fuel consumption model |
| Time-Dependent | - | - | - | Heuristic | Calculate transportation emissions/energy consumption |
| Fuel Consumption | - | - | - | Hybrid | National atmospheric emissions inventory |
| Electric VRP | - | - | - | - | Comprehensive modal emission model |
| Other | - | - | - | - | Vehicle specific power |

- Electric vehicle routing problem

A rich literature can be found in each class in terms of model development, solution approach, and also real-world case studies. Table 1 shows a summary review of different classes. Figure 2 demonstrates different cost functions employed to develop green vehicle routing models. From the solution approach viewpoint, different kinds of algorithms are adapted to deal with the complexity of GVRP models. In this regard, a simple classification can be inspired by the literature including exact, heuristic, metaheuristic, and hybrid algorithms. Figure 3 visualizes some of the related approaches.

Related research will be reviewed to provide an outlined survey and determine the contribution of the current study. Erdoğan and Miller-Hooks [13] presented a mathematical model regarding the location of the alternative fueling station in the VRP model. They used two construction heuristics to obtain a feasible solution regarding customer and station location simultaneously, so as to minimize the possibility of running out of vehicle fuel. Latterly, Schneider et al. [27] developed this model by presenting VRP with intermediate stops and engaged an adaptive variable neighborhood search algorithm to solve the problem. Bruglieri et al. [9] presented a more realistic model of GVRP containing a new formulation to investigate a reduction in the alternative fuel station (AFS). For better performance in the computation process, the cost of inserting any halt between each pair of customers was pre-measured and their model included a pre-computation of AFS. Leggieri and Haouari [20] presented non-linear formulation consumption constraints of GVRP, using a reformulation linearization technique in which the pre-processing computation is performed to reduce the number of variables and constraints. Ashish and Pishvaee [4] attempted to assess different aspects of performing alternative fuels in VRP by measuring the effects of different pollutants (i.e., NOx, HC, and CO) on human health and the ecosystem through various mathematical models based on Well-to-Wheel and Tank-to-Wheel analyses. Our review of the relevant literature only reveals a limited number of papers that deal with GVRP. However, many papers can be found in the literature which is related to our research. Despite the rich literature in this area, many aspects of GVRP models have not been developed by researchers. Inspiring the fact, the current research aimed at developing a green vehicle routing problem model with time windows and simultaneous pickup and delivery under demand uncertainty to minimize fuel consumption. Therefore, to provide a comprehensible description of the study, the next sections are organized as follows: Section 3 introduces a base model of the research known as vehicle routing problem with simultaneous pickup and delivery with time windows (VRPSPDTW) in both deterministic and stochastic contexts. Section 4 is assigned to explain the details of the proposed model based on a robust formulation and demand uncertainty set. Section 5 presents a two-stage algorithm for solving the new model. In section 5, numerical experiments will be conducted and the results will be discussed. Finally, section 6 conclude the research.

## 3 Vehicle routing problem with simultaneous pickup and delivery

### 3.1 Deterministic demand

In general, The VRPSPDTW is defined on a complete graph $G = (N_0, i)$ with $N_0 = \{0, 1, \ldots, n+1\}$ as the nodes set and $E = \{(i,j) \mid i,j \in N_0, i \neq j\}$ as the arc set. According to this notation, $N = \{1, 3, \ldots, n\}$ denotes

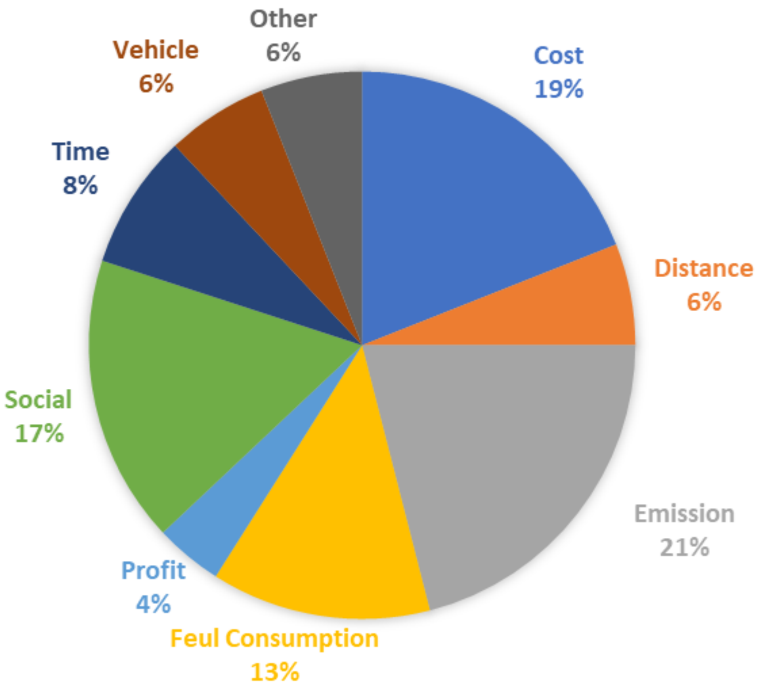


Figure 2: Percentage of using different cost functions in multi-objective models

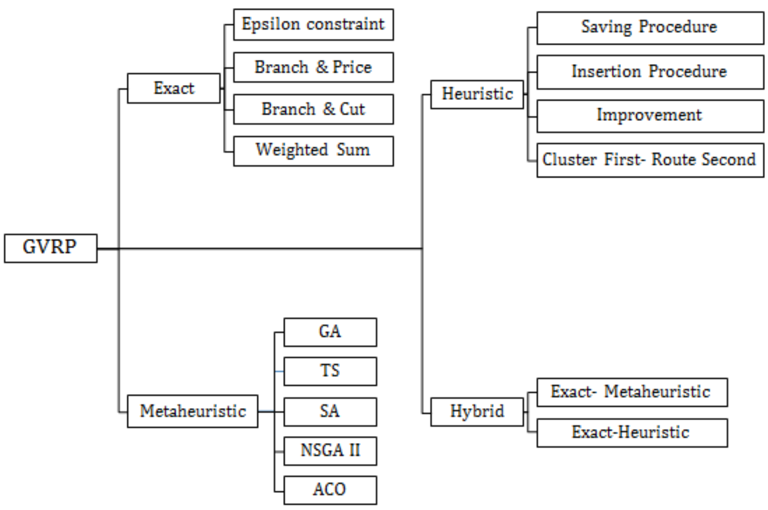


Figure 3: Classification of algorithms employed in GVRP models

actual customers and dummy nodes 0 and N+1 are vehicle depots. For each customer $i \in N, D_i and P_i$ stands for the deterministic demand and pickup, respectively. All nodes, except the depot, have positive demand and pickup. The set $K = \{1, 2, \ldots, m\}$ denotes the fleet of vehicles with the same capacity $Q$. All of the vehicles should start from the depot and come back to the depot after visiting all assigned customers. Each customer has a hard time window $\{a_i, b_i\}$; i.e., the arrival time of each vehicle to customer $i$ must be within the time window interval. The required time to serve each customer is defined as $t_i$. The distance between the paired customers $i$ and $j$ is shown as $d_i j$. According to [28], the speed of the vehicles is 1, that is, the travel time between customers $i$ and $j$ is the same as the distance between this pair of customers. Each route starts at the depot, visits a number of customers (at most once), and then returns to the depot. In this paper, we used mixed integer programming (MIP) for the formulation of VRPSPDTW. The decision variables are defined as follows:

$x_{ij}^k, i \in N_0, j \in N_0, k \in K$: if the vehicle $k$ goes directly from node $i$ to node $j$ equals to 1; otherwise 0.

$L_{1k}$: Load amount of vehicle $k \in K$ when exiting the depot.

$L_j$: Load amount of vehicle when leaving the customer $j \in N$.

$y_{ik}$: arrival time of vehicle $k$ at node $i \in N$.

$$\min \sum_{j \in N_0} \sum_{k \in K} x_{0j}^k \tag{3.1}$$

$$\min \sum_{(i,j) \in N_0} \sum_{k \in K} d_{ij} x_{ij}^k \tag{3.2}$$

$$\text{such that} \quad \sum_{i \in N_0} \sum_{k \in K} x_{ij}^k = 1, \quad \text{for all } j \in \mathbb{N} \tag{3.3}$$

$$\sum_{i \in N_0} x_{ir}^k - \sum_{j \in N_0} x_{rj}^k = 0, \quad \text{for all } k \in K, r \in N_0 \tag{3.4}$$

$$\sum_{j \in N} x_{0j}^k = 1, \quad \text{for all } k \in K \tag{3.5}$$

$$\sum_{i \in N} x_{i0}^k = 1, \quad \text{for all } k \in K \tag{3.6}$$

$$L_k = \sum_{i \in N_0} \sum_{j \in N} D_j x_{ij}^k, \quad \text{for all } k \in K \tag{3.7}$$

$$L_j \geq L_k - D_j + P_j - M(1 - x_{0j}^k), \quad \text{for all } j \in N, k \in K \tag{3.8}$$

$$L_j \geq L_i - D_j + P_j - M(1 - \sum_{k \in K} x_{ij}^k), \quad \text{for all } i, j \in N, \quad i \neq j \tag{3.9}$$

$$L_{1k} \leq Q, \quad \text{for all } k \in K \tag{3.10}$$

$$L_j \leq Q + M(1 - \sum_{i \in N_0} x_{ij}^k), \quad \text{for all } j \in N, k \in K \tag{3.11}$$

$$y_{ik} - y_{jk} + t_i + t_{ij} \leq M(1 - \sum_{k \in K} x_{ij}^k), \quad \text{for all } i, j \in N_0, k \in K \tag{3.12}$$

$$a_i \leq y_{jk} \leq b_i, \quad \text{for all } i, j \in N_0, k \in K \tag{3.13}$$

$$x_{ij}^k \in \{0, 1\}, \quad \text{for all } i, j \in N_0, k \in K. \tag{3.14}$$

In this model, Equations 3.1 and 3.2 are respectively the objective functions for minimizing dispatching cost and total traveling cost. Constraint 3.3 implies that each customer is visited once and only once by just one vehicle. Constraint 3.4 ensures that any vehicle that visits a customer must immediately leave it after the completion of its service. Constraints 3.5 and 3.6 ensure that all vehicles must choose the central depot as the starting and the ending point; i.e., each vehicle must start from the central depot and return to it at the end. Constraint 3.7 is the initial vehicle load. Constraint 3.8 is the vehicle loads after the first customer and constraint 3.9 is the vehicle loads after the subsequent customers. Constraints 3.10 and 3.11 are vehicle capacity constraints. Constraints 3.12 and 3.13 ensure the feasibility of the time schedule. The above model is for the case when the demand is deterministic and known in advance. However, in many applications, the demand may be unknown, which invalidates this model. Thus, a model should be developed to deal with demand uncertainty. Such a model is given in Section 3.

### 3.2 Uncertain demand

According to Bertsimas and Brown [6], robust optimization has two scenario-based and interval-based approaches. In the interval-based approach, an uncertain set is defined and all of its extreme points are named uncertain points. Then, for each uncertain parameter, an interval is defined and the worst case of that parameter is used in the optimization problem. In a real VRP model, when a parameter is non-deterministic, the model must be formulated under uncertainty. Hu et al. [17] formulated the VRPTW under uncertain demand and travel time and proposed a two-stage heuristic algorithm to tackle the NP-Hardness of the model and obtain high-quality solutions. Inspired by their work and using the studies in [6, 7], this paper formulates the VRPSPDTW with stochastic demand. The demand uncertainty is denoted here by $U_(q^k)$. Accordingly, the multidimensional demand uncertainty can be shown as follows:

$$U_q = X_{k\in K} U_q^k \tag{3.15}$$

$$U_q^k = \left\{ \tilde{q} \in R^{|N^k|} \mid \tilde{q}_i = \bar{q}_i + \alpha_i \hat{q}_i, \sum_{i\in N^k} |\alpha_i| \le \Gamma_q^k, |\alpha_i| \le 1, \Gamma_q^k = \lceil \theta_q |N^k| \rceil, \forall i \in N^k \right\}. \tag{3.16}$$

Eq. 3.15 indicates that the demand uncertainty set $U_q$ is the Cartesian product of the demand uncertainty set $U_q^k$ for each vehicle in equation 3.16. $N^k$ represents the set of customers on the route of vehicle $k, \bar{q}_i$ stands for the nominal value of uncertain demand $\tilde{q}_i, \hat{q}_i$ denotes the maximum deviation from $\bar{q}_i$ for $i \in N^k, \alpha_i$ is the auxiliary variable, and $\gamma_q^k$ controls the level of uncertain demand. Moreover, $\theta q \in [0,1]$ is the demand budget coefficient: if $\theta q = 0$, there is no demand uncertainty, and if $\theta q = 1$, each customer demand can take a value within the interval $[\bar{q}_i - \hat{q}_i, \tilde{q}_i + \hat{q}_i]$. To incorporate demand uncertainty into the proposed model, it is possible to alter each decision variable $L_j$ in the deterministic model by $L_j$ (q) [17, 1]. Accordingly, the proposed VRPSPDTW can be formulated as follows:

$$Min \sum_{j\in N_0} \sum_{k\in K} x_{0j}^k \tag{3.17}$$

$$Min \sum_{(i,j)\in N_0} \sum_{k\in K} d_{ij} x_{ij}^k \tag{3.18}$$

s.t:

$$\sum_{i\in N_0} \sum_{k\in K} x_{ij}^k = 1, \quad \forall j \in N \tag{3.19}$$

$$\sum_{i\in N_0} x_{ir}^k - \sum_{j\in N_0} x_{rj}^k = 0, \quad \forall k \in K, \quad \forall r \in N_0 \tag{3.20}$$

$$\sum_{j\in N} x_{0j}^k = 1, \quad \forall k \in K \tag{3.21}$$

$$\sum_{i\in N} x_{i0}^k = 1, \quad \forall k \in K \tag{3.22}$$

$$L_{0k}(q) = \sum_{i\in N_0} \sum_{j\in N} D_j x_{ij}^k, \quad \forall k \in K, \quad q \in ext\{U_q\} \tag{3.23}$$

$$L_j(q) \ge L_{0k} - D_j + P_j - M(1 - x_{0j}^k), \quad \forall j \in N, \quad \forall k \in K, \quad q \in ext\{U_q\} \tag{3.24}$$

$$L_j(q) \ge L_i - D_j + P_j - M(1 - \sum_{k\in K} x_{ij}^k), \quad \forall i \in N, \quad \forall j \in N, \quad i \ne j, \quad q \in ext\{U_q\} \tag{3.25}$$

$$L_{1k}(q) \le Q, \quad \forall k \in K, \quad q \in ext\{U_q\} \tag{3.26}$$

$$L_j(q) \le Q + M(1 - \sum_{i\in N_0} x_{ij}^k), \quad \forall j \in N, \quad \forall k \in K, \quad q \in ext\{U_q\} \tag{3.27}$$

$$y_{ik} - y_{jk} + t_i + t_{ij} \le M(1 - \sum_{k\in K} x_{ij}^k), \quad \forall i \in N_0, \quad \forall j \in N_0, \quad \forall k \in K \tag{3.28}$$

Table 2: ...

| Parameter | Description |
|---|---|
| $c_d$ | fix cost for utilizing each vehicle |
| $c_t$ | travel cost per distance unit |
| $\varphi_{ij}^k$ | Average acceleration of the vehicle k in arc $(i, j)$ |
| $g$ | gravity |
| $\delta_{ij}$ | angle between the road connecting node $i$ to node $j$ |
| $C_{rr}$ | Rolling resistance |
| $W_k$ | Weight of vehicle $k$ |
| $A_k$ | Front area of vehicle $k$ |
| $\rho_k$ | Air density |
| $\nu_{ij}^k$ | Average speed of vehicle k in arc $(i, j)$ |

$$a_i \leq y_{jk} \leq b_i, \quad \forall i \in N_0, \quad \forall j \in N_0, \quad \forall k \in K \tag{3.29}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i \in N_0, \quad \forall j \in N_0, \quad \forall k \in K \tag{3.30}$$

Constraints 3.7 to 3.11 are modified for each extreme point of the set $U_q(q \in ext\{U_q\})$, as demonstrated in relations 3.23 to 3.27.

## 4 Green VRPSPDTW with demand uncertainty

Considering the notations illustrated at previous sections and incorporating the green parameters as below the related green VRPSPDTW with demand uncertainty can be extracted as follows:

$$minz_1 = \sum_{k \in K} (c_d \sum_{j \in N_0} x_{0j}^k + c_t \sum_{i \in N_0} \sum_{j \in N_0} d_{ij} x_{ij}^k) \tag{4.1}$$

$$minz_1 = \sum_{q \in extU_q} \sum_{k \in K} \sum_{i \in N_0} \sum_{j \in N_0} (\varphi_k + g. \sin \delta_{ij} + g.Crr \cos \delta_{ij})(W_k + L_j(q)) d_{ij} x_{ij}^k + \sum_{k \in K} \sum_{i \in N_0} \sum_{j \in N_0} 0.5 c_d A_k \rho d_{ij} x_{ij}^k \tag{4.2}$$

s.t:

$$\sum_{i \in N_0} \sum_{k \in K} x_{ij}^k = 1, \quad \forall k \in K, \forall j \in N_0 \tag{4.3}$$

$$\sum_{i \in N_0} x_{ir}^k - \sum_{j \in N_0} x_{rj}^k = 0, \quad \forall k \in K, \forall r \in N_0 \tag{4.4}$$

$$\sum_{j \in N_0} x_{0j}^k = 1, \quad \forall k \in K \tag{4.5}$$

$$\sum_{i \in N_0} x_{i0}^k = 1, \quad \forall k \in K \tag{4.6}$$

$$L_{0k}(q) = \sum_{i \in N_0} \sum_{j \in N_0} D_j x_{jk}, \quad \forall k \in K, q \in extU_q \tag{4.7}$$

$$L_j(q) \geq L_{0k}(q) - D_j + P_j - M(1 - x_{0j}^k), \quad \forall j \in N, \forall k \in K, q \in extU_q \tag{4.8}$$

$$L_j(q) \geq L_i(q) - D_j + P_j - M(1 - \sum_{k \in K} x_{ij}^k), \quad \forall i \in N, j \in N, i \neq j, q \in extU_q \tag{4.9}$$

$$L_{1k}(q) \leq Q, \quad \forall k \in K, q \in extU_q \tag{4.10}$$

$$L_j(q) \leq Q + M(1 - \sum_{i \in N_0} x_{ij}^k, \quad \forall j \in N, \forall k \in K, q \in extU_q \tag{4.11}$$

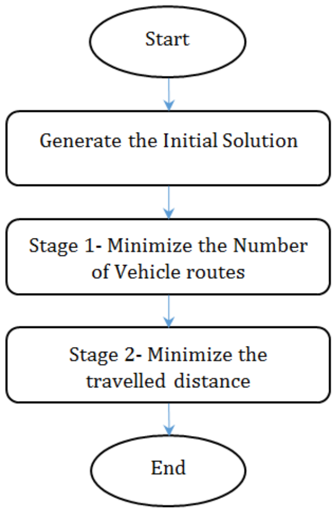


Figure 4: Framework of the two-stage algorithm

$$y_{ik} - yjk + ti + (d_{ij}\ u^k_{ij}) \leq M(1 - \sum_{k\ inK} x^k_{ij}, \quad \forall i \in N_0, \forall j \in N_0, \forall k \in K \tag{4.12}$$

$$a_i \leq y_{jk} \leq b_i, \quad \forall i \in N_0, \forall j \in N_0, \forall k \in K \tag{4.13}$$

$$x^k_{ij} \in \{0, 1\}, \quad \forall i \in N_0, \forall j \in N_0, \forall k \in K \tag{4.14}$$

## 5 Solution approach

This section describes the steps of the solution algorithm to be employed for examining the extended model. Many efficient algorithms have been developed to solve NP-hard problems and obtain reliable solutions [22, 3]. Among the existing algorithms, Hu et al. [17] proposed a two-stage method for achieving efficient solutions. In this paper, the proposed algorithm is based on a two-stage scheme, where the first stage minimizes the number of vehicles and the second one aims at minimizing the total traveled distance. The framework of this new algorithm is illustrated in Figure 4.

### 5.1 Initial solution

The first step of the two-stage algorithm is to generate an initial feasible solution, for which all time windows and customer constraints are considered. Since the quality of the initial solution can affect the output of the algorithm, the sequential insertion heuristic algorithm by Joubert and Claasen is employed to guarantee the feasibility and quality of solutions [18]. In their study, a time windows compatibility matrix (TWC) is adopted for the initial and extended routes holding the time window constraints.

### 5.2 Minimize the number of vehicles

In the first stage, using an adaptive variable neighborhood search (AVNS), the employed algorithm attempts to minimize the number of vehicles [29]. The pseudo-code for the first stage is illustrated in Figure 5.

Compute the total slack of each route.

Find and remove the route with the minimum number of customers. In the case of ties, a route is randomly selected.

Assign customers of the removed route to the route with the maximum slack. In the case of ties, a route is randomly selected.

Execute the AVNS algorithm to obtain a feasible solution.

If no feasible route is obtained, return to the last feasible route; otherwise, return to the obtained route.

According to this Pseudo-code, calculating the floats and running the AVNS algorithm are two major tasks of stage 1.

1. *Compute the total slackness of each route.*
2. *Find and remove the route with minimum number of customers. In the case of ties, a route is randomly selected.*
3. *Assign customers of removed route to a route with maximum slackness. In the case of ties, a route is randomly selected.*
4. *Execute adaptive variable neighborhood search (AVNS) to obtain a feasible solution.*
5. *If no feasible routes are obtained, return the last feasible routes; otherwise return the obtained routes.*

Figure 5: Pseudo-code for stage 1

#### 5.2.1 Route slack

The slack of each route is equivalent to the slack of each vehicle and demonstrates the total amount of time a vehicle can be delayed without violating the time windows. This measure can be calculated as follows:

$$Slack_k = \sum_{i \in r_i} max\{0, b_i - y_{ik}\} \tag{5.1}$$

where $r_i$ denotes route $i$.

#### 5.2.2 AVNS algorithm

As a metaheuristic algorithm, the variable neighborhood search (VNS) algorithm can be applied to a wide range of combined optimization problems [16]. This algorithm begins with an initial solution and a set of neighborhood structures. Then, the major solution loop is replicated until the stopping condition is reached. As mentioned in the earlier part of section **??**, the AVNS algorithm was initially introduced by Stenger et al. (2013). AVNS algorithm has two basic advantages. First, several routes are selected and their node's order can be adjusted in the shaking phase of an iteration such that it yields the minimum probability of exiting the local optimum. Secondly, it can be adopted on the recent search performance. The most basic phases of the AVNS algorithm are local search and shaking phases. To implement the proposed AVNS algorithm, two sets of operators known as Lin and Kernighan (LK) heuristics and cross-exchange operator should be defined [21, 30].

**Lin and Kernighan (LK) heuristics**

LK heuristics belongs to the class of local optimization algorithms and has been defined for the exchange of the edges of a route with those of another route. For a possible route, the algorithm repeats the exchanges many times so that the length of the tour is reduced. Since the complexity of finding $K-opt$ exchanges increases rapidly for larger $K$, usually $2-opt$ and $3-opt$ are used in heuristics. Starting with the longest edge in the route, the edges are iteratively removed and added, such that the pair of the removed and added edges shares an endpoint, fulfills the partial gain criterion, and results in a feasible tour if the tour is closed. With the partial gain criterion, only those pairs of edges are considered for the removal and insertion that the overall sequence of exchanges results in an overall improvement. By restricting the search to promising options for larger $K$, this criterion reduces the neighborhood drastically. If, for instance, the sequential removal and insertion of two pairs of edges result in no positive gain, LK does not continue to remove and add more pairs. As soon as an improving move is found with the closure of the tour, the move is executed and LK is restarted. The heuristic stops if for each initially removed edge no improving move can be found [3]. An example is provided in Figure 6. Starting with the removal of edge (1,2), edge (2,5) starting from one of the endpoints is added, such that $d_12 > d_25$. Because of the positive gain, the process continues by removing an edge incident to node 5 and adding an edge, e.g., (4,5) and (4,6). If $d_12 + d_45 > d_25 + d_46$, the moves can be continued by removing edge (5,6) and adding edge (1,5), to obtain a $3-opt$ move.

**Cross exchange operator**

Cross exchange operator (CE) is a local search operator aims to exchange two sub-routes $\hat{r}_i$ and $\hat{r}_j$ of two different routes $r_i$ and $r_j$, respectively [30]. Such an exchange is shown in Figure 7. Cross exchange is performed in two stages: 1) identifying the initial point of the two sub-routes, and 2) estimating the appropriate length for each of the sub-routes. To be more detailed, let $(i_\nu, i_{(\nu+1)})$ be a candidate edge to be removed from route $r_i$. For node $i$, a node in a different route (which belongs to the $T$ closest neighbors of node $i$) is randomly selected. Let node $j_\tau$ be such a neighbor. If edge $(i_\nu, j_\tau)$ is added as a new edge, one of the edges $(j_\tau, j_{(\tau-1)})$ or $(j_\tau, j_{(\tau+1)})$ must be eliminated. Finally, the incident node of the removed edge $j_{(\tau+1)} or j_{(\tau-1)}$ is randomly selected and reconnected to $i_{(\nu+1)}$.

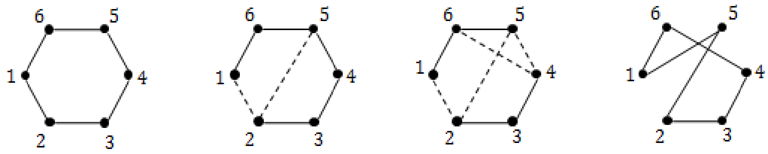

Figure 6: The sequential removal and insertion in LK nodes

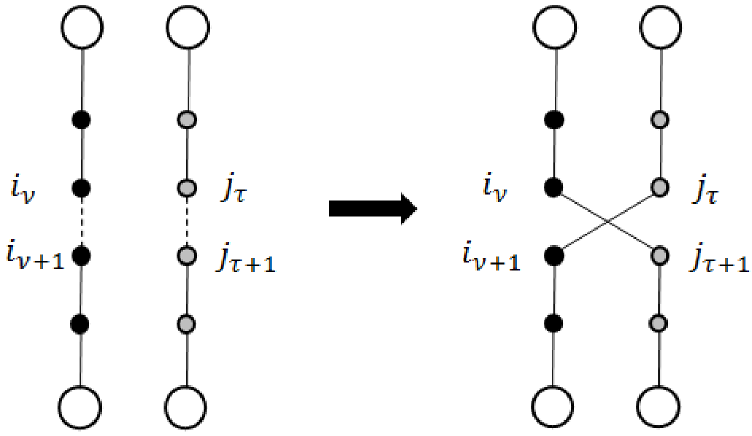

Figure 7: Illustration of a CE operator for two routes

### 5.3 Minimize the traveled distance

After minimizing the number of vehicles in the previous step of the algorithm, the last feasible solution must be shaken such that the routes are improved and the minimum traveled distance is achieved. In this regard, the AVNS algorithm has six local search operators as well as a scoring scheme to improve the feasible solution in terms of the total traveled distance: intra-route swap, intra-route reinsert, intra-route $2-opt$, inter-router swap, inter-router reinsert, and inter-route $2-opt$. The graphic illustration of these six neighborhood operators is shown in Figure 8. In each iteration of the AVNS algorithm, six loops are considered, where, in each loop, one of the operators is randomly selected and employed to improve the current solution. The operators are selected based on the roulette wheel scheme with an equal chance for each operator. As soon as an improvement is observed in a loop, the corresponding operator receives a score to be selected with more chance. According to [17], if a new overall best solution is achieved after applying a selection method, then a score of nine is added to the method. Hereby, the more effective operator will have more chances to be selected.

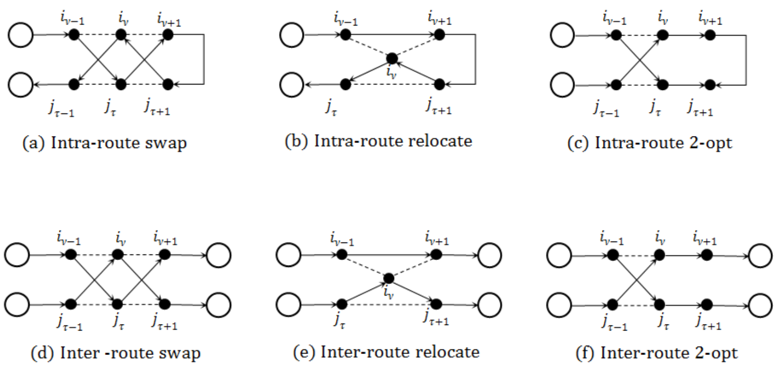


Figure 8: Six neighborhood operators

Table 3: Results for R1 instance

| Instance | Proposed Alg. | | | Modified PSO (Norouzi et al. 2017) | | |
|---|---|---|---|---|---|---|
| | #Vehicles | Travelled Distance | Obj. fun | #Vehicles | Travelled Distance | Obj. fun |
| R101 | 21 | 1722 | 7413 | 21 | 1770 | 7512 |
| R102 | 20 | 1492 | 7406 | 21 | 1607 | 7449 |
| R103 | 14 | 1295 | 7105 | 14 | 1313 | 7139 |
| R104 | 11 | 1083 | 7005 | 11 | 1125 | 7060 |
| R105 | 17 | 1463 | 7109 | 18 | 1513 | 7122 |
| R106 | 13 | 1412 | 7078 | 14 | 1429 | 7136 |
| R107 | 12 | 1255 | 7023 | 12 | 1284 | 7111 |
| R108 | 11 | 1128 | 7214 | 11 | 1147 | 7314 |
| R109 | 13 | 1289 | 7000 | 16 | 1321 | 7072 |
| R110 | 12 | 1156 | 6978 | 14 | 1242 | 7028 |
| R111 | 11 | 1316 | 7123 | 12 | 1373 | 7158 |
| R112 | 11 | 1131 | 6989 | 13 | 1195 | 7015 |
| **Average** | **13.8** | **1313** | **7120** | **14.5** | **1328** | **7176** |

### 5.4 The termination criterion

The algorithm is stopped when 200 successive iterations show no improvement or it reaches 5000 iterations.

## 6 Numerical experiments

In this section, the details of the numerical experiments to test the proposed VRPSPDTW under demand uncertainty are described. The test instances were derived from the well-known Solomon's instances [15]. The instances are grouped into six datasets $R1, R2, C1, C2, RC1, andRC2$. Each dataset contains between eight to twelve problems each with 100 customers. The six problem types are categorized as follows:

$C$: with clustered customers whose time windows were generated based on a known solution;
$R$: with customer locations generated uniformly over a square;
$RC$: with a combination of randomly placed and clustered customers.
Where,
Type 1 has narrow time windows and small vehicle capacity, and
Type 2 has large time windows and large vehicle capacity.

In this paper, datasets $R$ and $RC$ are used in the experiments. To generate pickup data, the method proposed by Angelelli and Mansini (2002) is employed. They modified Solomon's instances and calculated the pickup amount $p_i$ corresponding to the delivery amount $D_i$ as $P_i = (1 - \psi)D_i$ if $i$ is even and $P_i = (1 + \psi)D_i$ if $C$ is odd. In all calculations, $\psi$ is considered 0.2. Each nominal value $\bar{q}_i$ was assumed to be equal to the corresponding customer demand. In addition, we assumed that the maximum demand deviation $\hat{q}_i$ was $0.2\bar{q}_i$. We assumed that $\theta_q = 0.2$ for all data.

To evaluate the performance of the proposed algorithm the introduced model is also solved by the modified PSO algorithm developed by [23]. The parameters $c_d$ and $c_t$ are randomly drawn from interval [0.35-0.65]. The corresponding speed of vehicles is selected from the interval [65-90] and randomly assigned to roads. Objective functions are combined with weights of 0.6 and 0.4, respectively.

The first experiment was conducted on R1 instances, which contain randomly distributed customers with narrow time windows, short scheduling horizons, and small-capacity vehicles. Table 3 summarizes the results, where solution approaches can be compared in terms of the objective function. In addition, the increases in the number of vehicles and total traveled distance are outlined in Table 3. It is obvious that in some instances, the number of vehicles has not increased for the proposed two-stage algorithm. In all cases, the two-stage algorithm outperforms the modified version of PSO.

For R2 instances, whose results have been summarized in Table 4, it can be concluded that when the customers have loose time windows fewer vehicles are needed to establish feasible tours than for R1 problems.

Table 4: Results for R2 instance

| Instance | Proposed Alg. | | | Modified PSO (Norouzi et al. 2017) | | |
|---|---|---|---|---|---|---|
| | **#Vehicles** | **Travelled Distance** | **Obj. fun** | **#Vehicles** | **Travelled Distance** | **Obj. fun** |
| R201 | 5 | 1652 | 7488 | 7 | 1679 | 7537 |
| R202 | 4 | 1588 | 7576 | 4 | 1652 | 7579 |
| R203 | 3 | 1339 | 7255 | 3 | 1542 | 7374 |
| R204 | 3 | 1156 | 7104 | 3 | 1179 | 7115 |
| R205 | 3 | 1693 | 7295 | 4 | 1763 | 7296 |
| R206 | 3 | 1519 | 7202 | 3 | 1554 | 7313 |
| R207 | 3 | 1239 | 7195 | 3 | 1328 | 7222 |
| R208 | 3 | 1050 | 7413 | 3 | 1094 | 7504 |
| R209 | 3 | 1572 | 7195 | 3 | 1645 | 7210 |
| R210 | 3 | 1468 | 7051 | 4 | 1507 | 7083 |
| R211 | 3 | 1308 | 7251 | 3 | 1308 | 7345 |
| **Average** | **3.27** | **1417** | **7275** | **3.63** | **1436** | **7325** |

Table 5: Results for RC1 instance

| Instance | Proposed Alg. | | | Modified PSO (Norouzi et al. 2017) | | |
|---|---|---|---|---|---|---|
| | **#Vehicles** | **Travelled Distance** | **Obj. fun** | **#Vehicles** | **Travelled Distance** | **Obj. fun** |
| R201 | 14 | 1696 | 7498 | 16 | 1821 | 7733 |
| R202 | 12 | 1554 | 7599 | 14 | 1602 | 7741 |
| R203 | 11 | 1261 | 7381 | 13 | 1433 | 7550 |
| R204 | 10 | 1164 | 7176 | 11 | 1242 | 7340 |
| R205 | 14 | 1548 | 7422 | 16 | 1655 | 7593 |
| R206 | 11 | 1424 | 7340 | 13 | 1479 | 7564 |
| R207 | 11 | 1232 | 7295 | 12 | 1386 | 7537 |
| R208 | 10 | 1139 | 7485 | 12 | 1328 | 7659 |
| **Average** | **11.63** | **1,377** | | **13.38** | **1,493** | |

Similar results are illustrated in Table 5 for RC datasets

Since each robust solution may be affected by uncertain parameters, the proposed model is solved using different uncertainty parameter $\theta_q$. As discussed in Section 3, $\theta_q \in [0,1]$ is the demand budget coefficient, where if $\theta_q = 0$, there is no demand uncertainty, and if $\theta_q = 1$, the demand can take a value within the interval $[\bar{q}_i - \hat{q}_i, \bar{q}_i + \hat{q}_i]$. To conduct such an experiment, $\theta_q$ is considered to be equal to $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Table 6 shows these results. It is obvious that increasing the uncertainty parameter leads to greater number of vehicles and largest travelled distance. However, when the number of vehicles increases by one, an immediate drop is seen in travelled distance.

## 7 Conclusions

In this paper, a robust version of the green vehicle routing problem with simultaneous pick-up and delivery under demand uncertainty was proposed. Due to the complexity of the problem, the adapted robust model was only able to handle small-sized instances using standard solvers. To solve large-sized instances, a two-stage algorithm consisting of cross exchange and adaptive variable neighborhood search operators was designed. Extensive computational experiments were conducted using modified versions of Solomon's benchmark instances. The results showed that the two-stage algorithm was able to produce acceptable solutions for large-sized instances. The robustness of the solutions was examined by increasing the uncertainty parameter, where, a slight increase was seen during the increase of the parameter. Comparing the results versus the modified version of the PSO algorithm affirmed the superiority of the robust method in dealing with benchmark datasets.

Table 6: Results for datasets with different uncertainty parameters

| **Instance** | | **Uncertainty Parameter** | | | | | |
|---|---|---|---|---|---|---|---|
| | | $\theta_q = 0$ | $\theta_q = 0.2$ | $\theta_q = 0.4$ | $\theta_q = 0.6$ | $\theta_q = 0.8$ | $\theta_q = 1$ |
| R101 | #Vehicles | 21 | 21 | 22 | 22 | 22 | 23 |
| | Distance | 1722.1 | 1770.8 | 1659.5 | 1712.4 | 1745.1 | 1779.8 |
| R103 | #Vehicles | 14 | 14 | 14 | 14 | 15 | 15 |
| | Distance | 1295.4 | 1313.9 | 1329.1 | 1348.0 | 1309.2 | 1331.0 |
| R105 | #Vehicles | 17 | 18 | 18 | 18 | 18 | 19 |
| | Distance | 1463.6 | 1513.9 | 1521.19 | 1526.8 | 1549.4 | 1551.0 |
| R107 | #Vehicles | 12 | 12 | 13 | 14 | 14 | 14 |
| | Distance | 1255.2 | 1284.2 | 1243.7 | 1245.1 | 1249.1 | 1258.9 |
| R109 | #Vehicles | 13 | 16 | 15 | 16 | 16 | 17 |
| | Distance | 1289.7 | 1322.0 | 1330.1 | 1324.8 | 1340.0 | 1239.4 |
| R111 | #Vehicles | 11 | 12 | 12 | 13 | 13 | 13 |
| | Distance | 1316.1 | 1373.7 | 1387.7 | 1370.4 | 1390.1 | 1397.9 |
| R202 | #Vehicles | 4 | 4 | 4 | 5 | 5 | 5 |
| | Distance | 1588.1 | 1652.5 | 1623.2 | 1653.3 | 1653.3 | 1653.3 |
| R204 | #Vehicles | 3 | 3 | 4 | 4 | 5 | 5 |
| | Distance | 1156.1 | 1179.9 | 1169.1 | 1190.4 | 1192.3 | 1192.1 |
| R206 | #Vehicles | 3 | 3 | 4 | 5 | 5 | 5 |
| | Distance | 1519.8 | 1554.5 | 1554.5 | 1561.4 | 1561.4 | 1561.4 |
| R208 | #Vehicles | 3 | 3 | 4 | 4 | 5 | 5 |
| | Distance | 1050.3 | 1094.1 | 1085.1 | 1085.1 | 1085.1 | 1085.1 |
| R210 | #Vehicles | 3 | 4 | 4 | 4 | 4 | 4 |
| | Distance | 1468.6 | 1507.1 | 1507.1 | 1507.1 | 1507.1 | 1507.1 |
| RC101 | #Vehicles | 14 | 16 | 16 | 16 | 16 | 17 |
| | Distance | 1696.9 | 1821.5 | 1832.0 | 1840.1 | 1857.9 | 1830.0 |
| RC103 | #Vehicles | 11 | 13 | 13 | 13 | 14 | 14 |
| | Distance | 1261.7 | 1433.4 | 1450.6 | 1455.9 | 1455.9 | 1469.1 |
| RC105 | #Vehicles | 14 | 16 | 16 | 16 | 16 | 16 |
| | Distance | 1548.4 | 1655.2 | 1663.7 | 1671.0 | 1674.5 | 1674.5 |
| RC107 | #Vehicles | 11 | 12 | 12 | 13 | 13 | 13 |
| | Distance | 1232.3 | 1386.6 | 1394.0 | 1389.0 | 1394.4 | 1394.4 |

## References


[1] A. Agra, M. Christiansen, R. Figueiredo, L.M. Hvattum, M. Poss, and C. Requejo, *The robust vehicle routing problem with time windows*, Comput. Oper. Res. **40** (2013), no. 3, 856–866.

[2] T.J. Ai and V. Kachitvichyanukul, *A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery*, Comput. Oper. Res. **36** (2009), no. 5, 1693–1702.

[3] F. Arnold and K. Sörensen, *Knowledge-guided local search for the vehicle routing problem*, Comput. Oper. Res. **105** (2019), 32–46.

[4] H. Ashtineh and M.S. Pishvaee, *Alternative fuel vehicle-routing problem: A life cycle analysis of transportation fuels*, J. Cleaner Prod. **219** (2019), 166–182.

[5] R. Baldacci, A. Mingozzi, and R. Roberti, *Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints*, Eur. J.f Oper. Res. **218** (2012), no. 1, 1–6.

[6] D. Bertsimas and D.B. Brown, *Constructing uncertainty sets for robust linear optimization*, Oper. Res. **57** (2009), no. 6, 1483–1495.

[7] D. Bertsimas and M. Sim, *The price of robustness*, Oper. Res.earch **52** (2004), no. 1, 35–53.

[8] Benjamin Biesinger, Bin Hu, and Günther Raidl, *An integer l-shaped method for the generalized vehicle routing problem with stochastic demands*, Electronic Notes in Discrete Mathematics **52** (2016), 245–252.

[9] M.aurizio Bruglieri, S. Mancini, F. Pezzella, and O. Pisacane, *A new mathematical programming model for the green vehicle routing problem*, Electronic Notes Discrete Math. **55** (2016), 89–92.

[10] H. Caceres, R. Batta, and Q. He, *School bus routing with stochastic demand and duration constraints*, Transport. Sci. **51** (2017), no. 4, 1349–1364.

[11] W.C. Chiang and C.Y. Cheng, *Considering the performance bonus balance in the vehicle routing problem with soft time windows*, Procedia Manufactur. **11** (2017), 2156–2163.

[12] G.B. Dantzig and J. Ramser, *The truck dispatching problem*, Manag. Sci. **6** (1959), no. 1, 80–91.

[13] S. Erdoğan and E. Miller-Hooks, *A green vehicle routing problem*, Transport. Res. Part E: Logistics Transport. Rev. **48** (2012), no. 1, 100–114.

[14] M. Gendreau, O. Jabali, and W. Rei, *50th anniversary invited article—future research directions in stochastic vehicle routing*, Transport. Sci. **50** (2016), no. 4, 1163–1173.

[15] B.L. Golden and A.A. Assad, *Vehicle routing with time-window constraints*, Amer. J. Math. Manag. Sci. **6** (1986), no. 3-4, 251–260.

[16] P. Hansen, N. Mladenović, R. Todosijević, and S. Hanafi, *Variable neighborhood search: basics and variants*, Eur. J. Comput. Optim. **5** (2017), no. 3, 423–454.

[17] C. Hu, J. Lu, X. Liu, and G. Zhang, *Robust vehicle routing problem with hard time windows under demand and travel time uncertainty*, Comput. Oper. Res. **94** (2018), 139–153.

[18] J.W. Joubert and S.J. Claasen, *A sequential insertion heuristic for the initial solution to a constrained vehicle routing problem*, ORiON **22** (2006), no. 1, 105–116.

[19] B. Kallehauge, *Formulations and exact algorithms for the vehicle routing problem with time windows*, Comput. Oper. Res. **35** (2008), no. 7, 2307–2330.

[20] V. Leggieri and M. Haouari, *A practical solution approach for the green vehicle routing problem*, Transport. Res. Part E: Logistics Transport. Rev. **104** (2017), 97–112.

[21] S. Lin and B.W. Kernighan, *An effective heuristic algorithm for the traveling-salesman problem*, Oper. Res. **21** (1973), no. 2, 498–516.

[22] R. Liu, X. Xie, V. Augusto, and C. Rodriguez, *Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care*, Eur. J. Oper. Res. **230** (2013), no. 3, 475–486.

[23] H. Min, *The multiple vehicle routing problem with simultaneous delivery and pick-up points*, Transport. Res. Part

A: Gen. **23** (1989), no. 5, 377–386.

[24] D.M. Miranda and S.V. Conceição, *The vehicle routing problem with hard time windows and stochastic travel and service time*, Expert Syst. Appl. **64** (2016), 104–116.

[25] R. Moghdani, K. Salimifard, E. Demir, and A. Benyettou, *The green vehicle routing problem: A systematic literature review*, J. Cleaner Prod. **279** (2021), 123691.

[26] S.N. Parragh, K.F. Doerner, and R.F. Hartl, *A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations*, J. Betriebswirtschaft **58** (2006), 81–117.

[27] M. Schneider, A. Stenger, and J. Hof, *An adaptive vns algorithm for vehicle routing problems with intermediate stops*, Or Spectrum **37** (2015), no. 2, 353–387.

[28] M.M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Oper. Res. **35** (1987), no. 2, 254–265.

[29] A. Stenger, D. Vigo, S. Enz, and M. Schwind, *An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping*, Transport. Sci. **47** (2013), no. 1, 64–80.

[30] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, *A tabu search heuristic for the vehicle routing problem with soft time windows*, Transport. Sci. **31** (1997), no. 2, 170–186.

[31] D. Taş, O. Jabali, and T. Van Woensel, *A vehicle routing problem with flexible time windows*, Comput. Oper. Res. **52** (2014), 39–54.

[32] Paolo Toth and Daniele Vigo, *Vehicle routing: problems, methods, and applications*, SIAM, 2014.

[33] H.-F. Wang and Y.-Y. Chen, *A genetic algorithm for the simultaneous delivery and pickup problems with time window*, Comput. Ind. Engin. **62** (2012), no. 1, 84–95.