The secure time-dependent vehicle routing problem with uncertain demands

Citation for published version (APA):

Allahyari, S., Yaghoubi, S., & van Woensel, T. (2021). The secure time-dependent vehicle routing problem with uncertain demands. *Computers & Operations Research*, 131, [105253].
<https://doi.org/10.1016/j.cor.2021.105253>

Document license:
TAVERNE

DOI:
[10.1016/j.cor.2021.105253](https://doi.org/10.1016/j.cor.2021.105253)

Document status and date:
Published: 01/07/2021

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

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
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

The secure time-dependent vehicle routing problem with uncertain demands

Somayeh Allahyari^a, Saeed Yaghoubi^{a,*}, Tom Van Woensel^b

^a School of Industrial Engineering, Iran University of Science and Technology, 16846-13114 Tehran, Iran

^b School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands

ARTICLE INFO

Article history:

Received 27 February 2020

Revised 28 December 2020

Accepted 14 February 2021

Available online 19 February 2021

Keywords:

Combinatorial optimization

Vehicle routing problem

Transportation of valuables

GRASP

Iterated local search

ABSTRACT

This paper addresses the transportation of valuable goods in which security carriers are interested to optimize the operating costs and the vehicle routes' security. In real operations, route planning and scheduling are normally performed manually based on individual experience. Hereupon, an optimizing approach should be developed to target the robbery risk reduction and the distribution network' operational costs. Facing the dangerous nature of operations, some considerations play an important role, including demand fluctuations and traffic congestion of the urban environment. To handle these issues, we respectively benefit from the robust optimization theory and considering time-dependent travel speeds with satisfying the "first-in-first-out" property. This paper proposes a rich vehicle routing problem denoted by the secure time-dependent vehicle routing problem with time windows including pickup and delivery with uncertain demands (S-TD-VRPTWPD-UD). A mathematical formulation and an efficient solution approach combining the greedy randomized adaptive search procedure (GRASP) and the iterated local search (ILS) are developed to minimize the predictability of route plans using an integrated risk index, besides the travel costs. Extensive computational experiments on this problem are performed to analyze the impact of the demand uncertainty and the speed time-dependency and to show the efficiency of the GRASP $\times$ ILS implementation. The results show the significant improvements due to the time-dependency as well as the extra cost of protecting the model against the worst-case scenario of demand requests by deriving the robust counterpart of the S-TD-VRPTWPD-UD.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

1.1. Motivations of the problem

Considering security in vehicle routing and scheduling operations have outstandingly attracted the interest of security carriers in the transportation of valuables/cash. They intend to optimize operating costs while ensuring security during transportation between bank branches, automated teller machines, jewelries, shopping centers, etc. Due to the dangerous nature of these operations, particular equipments are used by carriers and cash-in-transit companies, however, a robbery always remains inevitable with unwanted consequences. One category of these outcomes is predictable proportional to the loss of valuables being carried. On the other category, we face unpredictable consequences maybe with heavy additional costs related to damaged vehicles or treat-

ment of harmed staff or users of the road environment. For some statistics concerning the importance of the cash in transit (CIT) sector, the interested readers are referred to Kempton Express (2017) and Talarico et al. (2017).

According to past records, one of the main reasons for the robbery, which we focus, can be seen in the lack of investigating security issues in route planning and scheduling (Yan et al., 2012). In such a problem, this paper follows the risk index proposed by Allahyari et al., 2021 integrating several aspects of secure routing where the efficiency of incorporating risk components is demonstrated by analyzing a real case. The introduced problem lies in one of many real-life applications of the vehicle routing problem (VRP). VRP as one of the most widely studied problems in the field of combinatorial optimization has drawn considerable attention of both academics and practitioners from 1959 (Dantzig et al., 1959) onwards. During more than a half-century of enriching VRP, many efforts have been devoted to developing several variants of this problem by considering different types of objectives and constraints (Toth and Vigo, 2014) such as (1) pickup and delivery types of demand requests and (2) time windows to visit customers. Both

* Corresponding author.

E-mail addresses: s.allahyari@ind.iust.ac.ir (S. Allahyari), yaghoubi@iust.ac.ir (S. Yaghoubi), t.v.woensel@tue.nl (T. Van Woensel).

mentioned features which we consider, are necessary to better fit the real-life application of the introduced problem.

Security carriers in real operations plan the vehicle routes based on average travel times and estimated delivery/pickup requests at the beginning of the day. Traffic fluctuations are always experienced by all residents in the urban environment due to daily or periodically rush hours, road accidents or weather conditions, etc. This event may result in some adverse consequences including the quality of committed services to customers by variations in trip duration. To cope with this problem, we benefit from time-dependent travel speeds satisfying the “first-in-first-out” property.

On the other hand, due to the dangerous nature of the valuable transportation, vehicle routes should be designed robust addressing the variability in demands. Most of the works to represent uncertainty in the VRP lie in the category of stochastic programming where the strong assumptions about distribution of the uncertain parameter or definition of scenarios are needed. Since these approaches deal with minimization of the expected value, the resulting solution that is sensitive to real data may not be feasible for some realizations. Moreover, finding the actual probability distribution of the uncertain parameter may be hard due to insufficient historical data or complexity of the numerous scenarios. Robust optimization as a rapidly developing methodology handles the optimization problems affected by non-stochastic “uncertain-but bounded” data realizations. To formulate the bounded demand uncertainty, we choose the best solution among those “immunized” against data uncertainty by optimizing the worst-case value among the data uncertainty set using the robust optimization methodology introduced by [Soyster \(1973\)](#). In other words, this paper intends to help decision-makers in finding a set of least-cost routes that remain feasible for all possible realizations of the uncertain customer demands. The resultant solution determines an optimal *a priori* route which is immune for every demand realization in a given bounded uncertainty set, i.e. box uncertainty set.

1.2. Recent studies in cash/valuables conveyance routing problem

To increase security in route planning, some approaches have been proposed in the literature. The first effort dates back to 1995 when the *m*-peripatetic salesman problem (*m*-PSP) was introduced by [Krarup \(1995\)](#). This problem aims to solve a traveling salesman problem for *m* periods where each arc is forbidden to reuse in the periods. The *m*-peripatetic vehicle routing problem (*m*-PVRP) by extending the *m*-PSP, considers a fleet of capacitated vehicles located at a central depot ([Ngueveu et al., 2009](#)). Spreading the arrival times is another approach to making routes less predictable proposed by [Michallet et al. \(2014\)](#) and [Calvo and Cordone \(2003\)](#). Another way can be designing dissimilar routes by defining a similarity index based on some features of the arcs shared between the solution pairs (see, [Talarico et al., 2015](#)). In the other work, [Talarico et al. \(2015\)](#) measured the global risk of each route by proposing a specific risk index in a cumulative and linear. Their index is proportional to the amount of cash being carried (as the consequences of an incident) and the traveled distance (as the probability of that incident). As stated before, we follow the risk index proposed by [Allahyari et al., 2021](#) integrating several aspects of secure routing.

A survey of literature shows few works in the VRP studies with security issues dealing with uncertainty (see, e.g., [Yan et al., 2014](#)). Regardless of the works investigating the security issues, several VRP variants study the fields of the robust optimization and the time-dependent routing, separately. Since this paper deals with the both fields to handle the two main considerations in handling uncertain demands and time-dependent travel speeds, we review them in the following.

The time-dependent vehicle routing problem was first put forward by [Malandraki and Daskin \(1992\)](#). The authors used a travel time step function which was criticized for the sake of not respecting the first-in-first-out (FIFO) principle. To solve this problem, later works such as [Donati et al. \(2008\)](#) and [Figliozzi \(2012\)](#) considered a step function for speed distribution along with the time intervals which is first proposed by [Ichoua et al. \(2003\)](#). To handle traffic congestion, [Van Woensel et al. \(2008\)](#) introduced an innovative modeling scheme to capture travel times. Alongside proposing heuristic solution methods (see, [Balseiro et al., 2011](#); [Harwood, 2013](#)), [Dabia et al. \(2013\)](#) presented a branch-and-price algorithm to solve this problem. The interested reader is referred to [Gendreau et al. \(2015\)](#) for a complete review of travel time modeling, applications, and solution methods.

Moreover, there are some VRP studies applying the robust convex optimization methodology to handle the deep uncertainty in demand requests (see, e.g., [Bertsimas and Simchi-Levi, 1996](#); [Sungur et al., 2008](#); [Gounaris et al., 2013](#)). In this methodology, a given bounded uncertainty set is assumed for each uncertain parameter which can be box ([Sungur et al., 2008](#)), ellipsoidal ([Ben-Tal and Nemirovski, 1998](#)), polyhedral or combination of them ([Bertsimas and Sim, 2004](#)), where conservativity decreases from box to polyhedral models to allow a trade-off between robustness and performance. For the review of these models and recent developments in this field, we refer to [Ben-Tal et al., \(2009\)](#); [Li et al., \(2011\)](#); [Gabrel et al., \(2014\)](#).

1.3. Main contributions of the paper

Following two Sections 1.1 and 1.2, the problem addressed in this paper proposes a rich variant of VRP which follows security and efficiency in routing plans for serving customers with pickup and delivery requests in a pre-defined time window. To do so, total imposed transportation risk via an integrated index, in addition to the cost of routing is minimized. For the problem, we consider two main realistic considerations including uncertain demands and time-dependent travel speeds handled by the robust optimization and the time-dependent routing, respectively. To the best of our knowledge, this VRP variant has not been previously addressed by the literature. We call this problem the secure time-dependent vehicle routing problem with no-wait time windows including pickup and delivery with uncertain demands (S-TD-VRPTWPD-UD). For the S-TD-VRPTWPD-UD, we formulate a mathematical formulation and develop an efficient solution approach combining the greedy randomized adaptive search procedure (GRASP) and the iterated local search (ILS). Extensive computational experiments on this problem are performed to analyze the impact of uncertainty in demands and time-dependency in travel speeds and to show the efficiency of the GRASP $\times$ ILS implementation.

The remainder of the paper is organized as follows. In Section 2, we provide the formal description of the S-TD-VRPTWPD-UD where a mixed-integer programming model is developed. Details of the proposed hybrid metaheuristic algorithm are outlined in Section 3. In Section 4, we report the results of computational tests of the proposed algorithms on the generated S-TD-VRPTWPD-UD instances and some of its variants. Finally, Section 5 presents some conclusions of the research.

2. Problem description

This section presents a formal definition of the S-TD-VRPTWPD-UD. Consider a complete directed time-dependent network $G = (\mathcal{N}, \mathcal{A}, \mathcal{T}, \mathcal{P})$. The node set $\mathcal{N} = \{0, 1, \dots, n+1\}$ includes node 0 and its dummy node $n+1$ to indicate the depot from which

all routes start and end respectively, and n remaining nodes denoted by set $\mathcal{N}_c$ indicating the customers. To consider different values of valuable items, both delivery and pickup demand $d_i \geq 0$ and $p_i \geq 0$ are given for each customer $i \in \mathcal{N}_c$ that should be routed in a pre-specified hard time interval $[a_i, b_i]$ by exactly one vehicle with a service time e_i . Because of the riskiness of the operations, waiting at each customer to open the related time window is prohibited. The set of arcs are defined as $\mathcal{A} = \{(i, j) | i, j \in \{(\{0\} \times \mathcal{N}_c) \cup (\mathcal{N}_c \times \mathcal{N}_c) \cup (\mathcal{N}_c \times \{n+1\})\}, i \neq j\}$ where a non-negative routing cost of arc (i, j) is c_{ij} . Also, we consider a set $\mathcal{K} = \{1, 2, \dots, \kappa\}$ for homogeneous vehicles available with a fixed operating cost $\mathcal{F}$ and a capacity $\mathcal{C}$ equivalent to the maximum worth of valuables. Each vehicle is used for at most one route.

The risk function, used in this paper, considers the occurrence of an attack both on the arc and at the node during the transportation. This index integrates four security components using the additivity feature that is proved by [Erkut and Ingolfsson \(2005\)](#) as follows: (a) minimizing the product of the probability of an incident occurrence and the consequences of that incident which are proportional to the travel time of the road segment passed by the vehicle and the number of valuables being carried, respectively, (b) minimizing the reused arcs proportional to their travel times during the realized planning horizon with the aim of reducing similarity, (c) spreading the arrival times at each customer within their time windows during the realized planning horizon with the aim of decreasing regularity, (d) minimizing the overall travel time of each route with the aim of diminishing the probability of robbery. At the start of the trip of all vehicles, the risk index is equal to zero and increases node by node during their tour according to the weighted summation of four mentioned criteria, i.e. terms 2–5 in relation (26), where $\alpha_1, \alpha_2, \alpha_3$ and α_4 denote the related weights, respectively. More precisely, this index at each node j on vehicle k (r_j^k) after traversing node i is equal to terms 1–5 in relation (26). Corresponding to the risk function, we consider the set $\mathcal{B}$ for the representation of b realized days before the day we intend to plan.

Accordingly, let two parameter sets y_j^b and x_{ij}^b denote respectively the arrival time at customer $i \in \mathcal{N}_c$ on the realized day $b \in \mathcal{B}$ and whether arc $(i, j) \in \mathcal{A}$ is traversed on the realized day $b \in \mathcal{B}$ or not.

The aim of S-TD-VRPTWPD-UD is to route at most $|\mathcal{K}|$ vehicles starting their routes from a central depot for serving all customers under both capacity and time window restrictions by minimizing the total cost including the cost of routing, the cost of using the vehicles as well as the total imposed risk, taking into account traffic congestion and uncertain demands. Section 2.1 proposes the time-dependent version of the problem with deterministic demands followed by Section 2.2 to present the robust counterpart of that to consider uncertainty in demands.

2.1. Time-dependent model with deterministic demands

Let $\mathcal{T} = \{t_0, t_0 + \gamma, \dots, t_0 + H\gamma\}$ be the set of discrete times in time horizon T to represent the time-dependent network including H time slots $T_h = [t_h, t_{h+1}]$. Accordingly, the set $\mathcal{S} = \{s_{ij}^h\}$ gives the time-dependent travel speeds for each arc $(i, j) \in \mathcal{A}$ through the time slot T_h assumed to be a step function. To satisfy the FIFO property, we assume a piecewise linear function $\tau_{ij}(t)$ for time-dependent travel times corresponding to each departure time

$t \in T$ and arc $(i, j) \in \mathcal{A}$. To model time-dependency, let two nodes i and j spaced out by a distance of $\mathcal{L}_{ij}$ where the arrival time at them places in time slots T_{h_a} and T_{h_b} , respectively with $h_a \in \{0, \dots, H-1\}$ and $h_b \in \{h_a, \dots, H-1\}$. Travel time for each arc (i, j) is calculated by summing up the travel times of all its sections that is obtained based on the cross-boundary of successive time slots at different travel speed as the form of the conditional function (1).

$$\tau_{ij}(y_i^k(t)) = \begin{cases} \mathcal{L}_{ij}/s_{ij}^{h_a} & \text{if } h_a = h_b, \\ y_j^k(t) - y_i^k(t) & \text{if } h_b = h_a + 1, \\ (t_{h_a+1} - y_i^k(t)) + (h_b - h_a - 1)\gamma + (y_j^k(t) - t_{h_b}) & \text{if } h_b > h_a + 1. \end{cases} \quad (1)$$

Where continuous variable $y_j^k(t)$ denotes the time at which service is started at vertex $j \in \mathcal{N}$ by vehicle $k \in \mathcal{K}$ as follows.

$$y_j^k(t) = \begin{cases} \mathcal{L}_{ij}/s_{ij}^{h_a} + y_i^k(t) & \text{if } \mathcal{L}_{ij}/s_{ij}^{h_a} < t_{h_a} - y_i^k(t), \\ (\mathcal{L}_{ij} - L^{h-1})/s_{ij}^{h_a} + t_h & \text{if } (\mathcal{L}_{ij} - L^{h-1})/s_{ij}^{h_a} < t_{h+1} - t_h, \text{ for } h \in \{h_a + 1, \dots, h_b\}. \end{cases} \quad (2)$$

where L^h is calculated as $s_{ij}^{h_a}(t_{h_a+1} - t)$ for $h = h_a$ and $L^{h-1} + s_{ij}^{h_a}(t_{h+1} - t_h)$ for $h \in \{h_a + 1, \dots, h_b - 1\}$.

The binary variable x_{ij}^k is equal to 1 iff arc $(i, j) \in \mathcal{A}$ is traversed by vehicle $k \in \mathcal{K}$. Let l_j^k represents the load of vehicle $k \in \mathcal{K}$ just after meeting vertex $j \in \mathcal{N}$. Moreover, continuous variable r_j^k denotes the risk index value of the vehicle $k \in \mathcal{K}$ at vertex $j \in \mathcal{N}$. It is noted that M is a large positive constant.

Here, the mathematical model of the time-dependent problem with deterministic demands is presented.

$$\text{Minimize } \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N}_c} \mathcal{F} x_{0j}^k + \sum_{k \in \mathcal{K}} \sum_{i, j \in \mathcal{N}_c} c_{ij} x_{ij}^k + \sum_{k \in \mathcal{K}} R_k \quad (3)$$

Subject to:

$$r_j^k \leq R_k \quad \forall j \in \mathcal{N}, k \in \mathcal{K}, \quad (4)$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{0\}} x_{ij}^k = 1 \quad \forall i \in \mathcal{N}_c, \quad (5)$$

$$\sum_{j \in \mathcal{N}_c} x_{0j}^k = \sum_{j \in \mathcal{N}_c} x_{j(n+1)}^k \quad \forall k \in \mathcal{K}, \quad (6)$$

$$\sum_{j \in \mathcal{N} \setminus \{0\}} x_{ij}^k = \sum_{j \in \mathcal{N} \setminus \{n+1\}} x_{ji}^k \quad \forall i \in \mathcal{N}_c, k \in \mathcal{K}, \quad (7)$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N}_c} x_{0j}^k \leq |\mathcal{K}|, \quad (8)$$

$$\sum_{j \in \mathcal{N}_c} x_{0j}^1 = 1, \quad (9)$$

$$\sum_{i \in \mathcal{N}_c} x_{i(n+1)}^k \geq \sum_{j \in \mathcal{N}_c} x_{0j}^{k+1} \quad \forall k \in \mathcal{K} \setminus \{\kappa\}, \quad (10)$$

$$y_i^k(t) + e_i + \tau_{ij}(y_i^k(t)) \leq y_j^k(t) + M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, \quad (11)$$

$$y_i^k(t) + e_i + \tau_{ij}(y_i^k(t)) \geq y_j^k(t) - M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, \quad (12)$$

$$a_j \leq y_j^k(t) \leq b_j \quad \forall j \in \mathcal{N}, k \in \mathcal{K}, \quad (13)$$

$$l_i^k + (p_j - d_j) \leq l_j^k + M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, \quad (14)$$

$$r_0^k \geq \sum_{i \in \mathcal{N}_c} \sum_{j' \in \mathcal{N} \setminus \{0\}} d_{ij'} x_{ij'}^k \quad \forall k \in \mathcal{K}, \quad (15)$$

$$0 \leq l_i^k \leq \mathcal{C} \quad \forall i \in \mathcal{N}, k \in \mathcal{K}, \quad (16)$$

$$\begin{aligned} r_i^k + \alpha_1 \tau_{ij}(y_i^k(t)) l_i^k + \alpha_2 \sum_{b \in \mathcal{B}} \tau_{ij}(y_i^k(t)) x_{ij}^k x_{ij}^b \\ - \alpha_3 \sum_{b \in \mathcal{B}} (2w_j^b - y_j^k(t) - 2y_j^b v_j^b + y_j^b)_{j \neq n+1} \\ + \alpha_4 y_j^k(t) \leq r_j^k + M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, \end{aligned} \quad (17)$$

$$w_j^b \leq M v_j^b \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c, \quad (18)$$

$$w_j^b \leq \sum_{k \in \mathcal{K}} y_j^k(t) \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c, \quad (19)$$

$$\sum_{k \in \mathcal{K}} y_j^k(t) - M(1 - v_j^b) \leq w_j^b \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c, \quad (20)$$

$$\sum_{k \in \mathcal{K}} y_j^k(t) - y_j^b \leq M v_j^b \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c, \quad (21)$$

$$y_j^b - \sum_{k \in \mathcal{K}} y_j^k(t) \leq M(1 - v_j^b) \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c, \quad (22)$$

$$r_0^k = 0 \quad \forall k \in \mathcal{K}, \quad (23)$$

$$r_j^k \leq M \sum_{i \in \mathcal{N} \setminus \{n+1\}} x_{ij}^k \quad \forall j \in \mathcal{N} \setminus \{0\}, k \in \mathcal{K}, \quad (24)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, \quad (25)$$

$$v_j^b \in \{0, 1\} \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c, \quad (26)$$

$$w_j^b \geq 0 \quad \forall b \in \mathcal{B}, j \in \mathcal{N}_c. \quad (27)$$

The objective function (3) minimizes the carrier's operating costs and the sum of imposed risk. The last term is the linear form of $\sum_{k \in \mathcal{K}} \max_{j \in \mathcal{N}} r_j^k$ using constraints (4). Relations (5)–(7) impose the degree constraints. While constraints (5) ensure that each customer $i \in \mathcal{N}_c$ is routed exactly once, constraints (6) and (7) guarantee the equality of entering and exiting arcs to each node i belonging to the customer set and the depot, respectively. Constraint (8) imposes the limited number of homogenous vehicles followed by constraints (9) and (10) to use the routes in an ascending order of their index. The arrival times to each node are imposed by constraints (11) and (12) which also forbid the forming of subtours. Constraints (13) consider the time windows restriction at the customer nodes. These time related constraints also forbid the waiting time and delayed or early service. Correspondingly, constraints (14)–(16) are also applied to make the load value of vehicle k just after visiting the node j .

The risk value at each node are defined by the converted constraints (17)–(22) where the original form of these constraints are given in constraints (28). The reason of the conversion is to get the linearized version of the nonlinear constraints (28) in the situation that the travel time of each arc is a constant parameter. Recall that the risk index at each node j on vehicle k (r_j^k) after traversing node i is equal to terms 1 – 5 in the relation (28). The four included components (i.e., terms 2 – 5) in a weighted summation mode follow (1) minimized valuables' volume being carried over the traveled distance, (2) irregular arrival times at each cus-

tomers, (3) different passed road segments and (4) minimized arrival times to each customer, respectively. Regarding the linearization, we innovatively rewrite constraints (28) by defining the new binary variables v_j^b , $\forall b \in \mathcal{B}, j \in \mathcal{N}_c$ as the three constraint sets (29) and (21)–(22). Now, the constraints (29) are transformed to the constraints (18)–(20), where continuous variables w_j^b , $\forall b \in \mathcal{B}, j \in \mathcal{N}_c$ are defined as the product of two sets of variables $\sum_{k \in \mathcal{K}} y_j^k$ and v_j^b .

$$\begin{aligned} r_i^k + \alpha_1 \tau_{ij}(y_i^k(t)) l_i^k + \alpha_2 \sum_{b \in \mathcal{B}} \tau_{ij}(y_i^k(t)) x_{ij}^k x_{ij}^b - \alpha_3 \sum_{b \in \mathcal{B}} |y_j^k(t) - y_j^b|_{j \neq n+1} \\ + \alpha_4 y_j^k(t) \leq r_j^k + M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, \end{aligned} \quad (28)$$

$$\begin{aligned} r_i^k + \alpha_1 \tau_{ij}(y_i^k(t)) l_i^k + \alpha_2 \sum_{b \in \mathcal{B}} \tau_{ij}(y_i^k(t)) x_{ij}^k x_{ij}^b - \alpha_3 \sum_{b \in \mathcal{B}} (2v_j^b - 1)(y_j^k(t) - y_j^b)_{j \neq n+1} \\ + \alpha_4 y_j^k(t) \leq r_j^k + M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}. \end{aligned} \quad (29)$$

The initial value of the risk index for all routes is considered zero in constraints (23). Constraints (24) show the risk of each node that just is related to its visiting vehicle, can get value. Finally, constraints (25)–(27) define the variables.

2.2. Robust optimization model to formulate uncertain demands

The aim of this section is to model finding a set of least-cost routes that remain feasible for all possible realizations of the uncertain customer demands. To do so, we apply the robust optimization to handle uncertainty in demand requests (pickup and delivery). To define a robust solution, both feasibility robustness and optimality robustness are considered. The former implies that the solution remains feasible for possible values of data in uncertain space, while the latter follows to minimize unpleasant deviation from the optimal value for possible values of data in uncertain space. Based on the degree of these criteria, we face three categories of robust programming approaches including hard worst case (HWC), soft worst case, and realistic robust programming. The first approach in which we are interested to provide maximum immunity against uncertainty, draws the attention of risk-averse decision-makers who perceive the terrible outcomes of small perturbations. The remaining approaches are a more flexible version than the first one in terms of feasibility robustness and cost-benefit logic, respectively.

Let (A) $\mathcal{M} \text{ minimize } cx + d : ax \leq b$ be a general deterministic linear optimization model. Suppose c, d, a, b are subject to uncertainty, varying in a specified closed bounded box. Based on the set-induced robust optimization (Ben-Tal and Nemirovski, 1998), we require to know a bounded set to which the uncertain parameters belong, without needing to find the probability distribution of them. Accordingly, (B) $\mathcal{M} \text{ minimize } cx + d : ax \leq b, \forall c, d, a, b \in U$ is the uncertain model including a set of linear optimization problems where c, d, a, b are considered to vary in a given close bounded box (Ben-Tal and Nemirovski, 1998; Ben-Tal et al., 2009). The general form of this box uncertainty set is displayed as $U_{Box} = \{\xi \in \mathbb{R}^{n_s} : |\xi_j - \bar{\xi}_j| \leq \mu \hat{\xi}_j, j = 1, \dots, n_s\}$, where $\xi_j, \bar{\xi}_j, \mu$ and $\hat{\xi}_j$ are j th parameter of n -dimension vector ξ , the nominal value of ξ_j , the uncertainty level and the uncertainty scale, respectively.

To obtain the robust counterpart of this problem, we have (C) $\mathcal{M}inimize\{\mathcal{O}(x) = \sup_{(c,d,a,b \in U_{Box})} cx + d : ax \leq b, \forall c, d, a, b \in U_{Box}\}$. The optimal objective solution of problem (C) is not ever worse than $\mathcal{O}(x^*)$ where x^* is the optimal solution of model (B). To have the tractable version of model (C), we should use the extreme points of U_{Box} , but not with the routine approach available in the literature (e.g., Sungur et al., 2008) to develop the HWC robust optimization. Sungur et al. (2008) modeled HWC for the capacitated VRP with uncertainty in demands. To do so, the authors considered maximum value for the delivery demands to be feasible for all scenarios where the same value would be realized. Considering maximum value for the delivery demands to realize in our problem, could not result in the best objective value for the worst case and also even a feasible solution in some situations. To illustrate this issue, let us have a complete four-node graph in Fig. 1. Numbers on the arcs and nodes show respectively the length (travel time) of arcs and the box uncertainty sets for pickup and delivery demands of nodes where request pattern is unknown and varies between two upper and lower bounds. Regarding the introduced set $\mathcal{B}$, we assume the only realized day be route (0–3–2–1–0). Time windows and alpha values are set $[0, 40]$ for all nodes and $\{1, 4, 3, 1\}$, respectively.

Regarding the concept of HWC robust optimization, the maximum value for pickup and delivery demands is considered to remain feasible in all scenarios. The request amounts, which customers will take in reality, have an uncertain nature. In this uncertain space, we face with four possible cases in terms of the extreme points where the nodes take: (a) their upper bounds of both pickup and delivery requests, (b) lower bounds of delivery and upper bounds of pickup, (c) upper bounds of delivery and lower bounds of pickup and (d) lower bounds of both pickup and delivery.

Fig. 1. A four-node example.

Table 1
Results of four cases on the extreme points of demands.

Solution		Case (a)			Case (b)			Case (c)			Case (d)		
		Load	Risk	OFV	Load	Risk	OFV	Load	Risk	OFV	Load	Risk	OFV
Route 0–1–2–3–0	0–1	67	620	630	67	620	630	67	620	630	67	620	630
	1–2	66	1300	1320	73	1370	1390	54	1180	1200	61	1250	1270
	2–3	61	1880	1910	72	2060	2090	45	1600	1630	56	1780	1810
	3–0	44	2360	2400	65	2750	2790	25	1890	1930	46	2280	2320
Route 0–3–2–1–0	0–1	67	720	730	67	720	730	67	720	730	67	720	730
	1–2	50	1280	1300	60	1380	1400	47	1250	1270	57	1350	1370
	2–3	45	1800	1830	59	2040	2070	38	1700	1730	52	1940	1970
	3–0	44	2320	2360	65	2770	2810	25	2030	2070	46	2480	2520

If we consider the classic objective function (e.g., Sungur et al., 2008), i.e., $\mathcal{M}inimize \sum_{k \in \mathcal{K}} \sum_{i,j \in \mathcal{N}} c_{ij} x_{ij}^k$, all aforementioned cases result in two optimal solutions with the same objective 40 (i.e., 0–1–2–3–0 and 0–3–2–1–0). This conclusion does not hold in our problem with the objective function $\mathcal{M}inimize \sum_{k \in \mathcal{K}} \sum_{i,j \in \mathcal{N}} c_{ij} x_{ij}^k + \sum_{k \in \mathcal{K}} R_k$ (see, Table 1). In Table 1, three columns are reported for each case including the load of vehicle on each arc, the risk value at the end point of each arc and total objective function. According to the results, case b with solution (0–1–2–3–0) results in the worst case with cost 2790. In other words, we face the cost between 1930 and 2790 with the selection of route (0–1–2–3–0), where this cost for route (0–3–2–1–0) puts in range of [2070, 2810]. Based on the realized scenarios' results, not only unlike the classical robust routing model, the objective function values of the two routes are different (i.e., route (0–3–2–1–0) is optimal for case a and route (0–1–2–3–0) is optimal for the other cases), but also depending on the vehicle's capacity, the classical robust model's solution can be impossible. This example clearly showed the inefficiency of the classical robust models (e.g., Sungur et al., 2008) for our problem. To develop the HWC robust model, we reformulate the constraint sets (14)–(15) as the constraint sets (30)–(31) where the bar ($\bar{\cdot}$) and hat ($\hat{\cdot}$) accents distinguish the nominal value and the uncertain part of the uncertain parameters p_j and d_j assumed to vary in a specified closed bounded box $[\bar{p}_j - \hat{p}_j, \bar{p}_j + \hat{p}_j]$ and $[\bar{d}_j - \hat{d}_j, \bar{d}_j + \hat{d}_j]$, respectively.

$$l_i^k + (\bar{p}_j + \hat{p}_j - \bar{d}_j + \hat{d}_j) \leq l_j^k + M(1 - x_{ij}^k) \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, \quad (30)$$

$$l_0^k \geq \sum_{i \in \mathcal{N}^c} \sum_{j' \in \mathcal{N} \setminus \{0\}} (\bar{d}_i + \hat{d}_i) x_{ij'}^k \quad \forall k \in \mathcal{K}. \quad (31)$$

3. Solution approach

As with most routing problems, directly solving the proposed mathematical model is difficult, except for small instances (see also the computational results reported in Section 4). To cope with this challenge, a meta-heuristic algorithm is proposed for the S-TD-VRPTWPD-UD combining iterated local search (ILS) and greedy randomized adaptive search procedure (GRASP), called the GRASP $\times$ ILS (see, Algorithm 1). The GRASP heuristic framework as a multi-start algorithm was first put forward by Feo et al. (1995) where two phases are run at each iteration including construction and local search. The former procedure builds an initial feasible solution while the latter tries to improve that solution. On the other hand, the ILS in each iteration applies local search and shaking procedures to intensify and diversify the solution space, respectively (Lourenço et al., 2003).

Algorithm 1: The framework of the GRASP $\times$ ILS.

Input:
 φ_1 : number of GRASP iterations;
 φ_2 : number of ILS iterations;
 φ_3 : number of iterations of each segment;
 $\mathcal{R}$: a set of removal operators;
Output: $bstSol$;
 $bstSol := \emptyset$;
 $Cost(bstSol) := +\infty$;
 $GRASP_{cnt} := 0$;
while ($GRASP_{cnt} \leq \varphi_1$) **do**
 $GRASP_{cnt} := GRASP_{cnt} + 1$;
 $Sol_{tmp} := \text{Construction}()$;
 $Sol_{bst} := Sol_{crnt} := Sol_{tmp} := \text{Local search}(Sol_{tmp})$;
 $ILS_{cnt} := 0$;
 while ($ILS_{cnt} \leq \varphi_2$) **do**
 $ILS_{cnt} := ILS_{cnt} + 1$;
 $Sol_{tmp} := \text{Shaking}(Sol_{crnt})\{$
 $q_{dst} := U(q_{min}, q_{max})$;
 $Destroy := \text{Choose a removal operator } j \in \mathcal{R} \text{ with probability } \mathcal{P}_j^t$;
 $Sol_{tmp} := Destroy(Sol_{crnt})$;
 $Sol_{tmp} := RandInsertion(Sol_{tmp})$;
 $Sol_{tmp} := \text{Local search}(Sol_{tmp})$;
 if ($Cost(Sol_{tmp}) < Cost(Sol_{crnt})$) **then**
 $Sol_{crnt} := Sol_{tmp}$;
 if ($Cost(Sol_{crnt}) < Cost(Sol_{bst})$) **then**
 $Sol_{bst} := Sol_{crnt}$;
 end
 end
 else
 $rand := \text{Select randomly from } \{0,1\}$;
 if $rand = 0$ **then**
 $Sol_{crnt} := Sol_{tmp}$;
 end
 end
 For each operator $j \in \mathcal{R}$ update weights $\mathcal{P}_j^{t+1}$ using the *adaptive weight adjustment* procedure every φ_3 iterations;
 end
 if ($Cost(Sol_{bst}) < Cost(bstSol)$) **then**
 $bstSol := Sol_{bst}$;
 end
end
Return $bstSol$.

The GRASP $\times$ ILS consists of φ_1 iterations of the GRASP algorithm, at each iteration of which a construction algorithm generates an initial feasible solution followed by the ILS procedure with φ_2 iterations to possibly obtain an improved solution. Given the current solution Sol_{crnt} , each iteration of the ILS algorithm contains (1) a shaking procedure working based on some extraction procedures and a random insertion operator and (2) a local search procedure. To choose the extraction operator to remove q_{dst} nodes, we apply a *roulette wheel* mechanism working based on the weights of the operators. These weights are dynamically updated after every φ_3 iterations using the *adaptive weight adjustment* procedure depending on their performance throughout the search in the diversification direction. If solution Sol_{tmp} obtained by utilizing the mentioned procedures improves the current solution Sol_{crnt} , the algorithm accepts that as a new current solution. Otherwise, it diversifies the search space randomly. Finally, the algorithm returns the best solution ($bstSol$) by finding the best solution for each iteration of the GRASP algorithm (Sol_{bst}). A similar combination of the GRASP and ILS algorithms for vehicle routing has been introduced by Allahyari et al. (2015) and Prins (2009). In the fol-

lowing, we present the details of each procedure in which a *departure time optimization algorithm* is embedded to optimize the departure time of each vehicle from the depot. To better search the solution space and mitigate the infeasibilities during the search, the waiting times are allowed with a big penalty constant in the objective function. Notably, solutions with waiting time are not considered as infeasible solutions during the search, unless in the final solution.

3.1. Construction procedure

To quickly obtain an initial feasible solution, we use the results of Cordeau et al. (2002). The authors analyzed and compared some of the well-known classical heuristics according to four attributes including accuracy, speed, simplicity, and flexibility. Based on their comparison, Clark and Wright method (Clarke and Wright, 1964) has the distinct advantage of being very quick and simple to implement. The method acts based on initially constructing back and forth routes for all customer nodes and gradually merging them using a saving criterion (Toth and Vigo, 2014). For the initialization

step, we use this method followed by the ILS procedure on the framework of $GRASP \times ILS$ to cover the accuracy and flexibility attributes. To make the randomization feature of the GRASP algorithm, the one-point routes are merged at random order. During the implementation, we impose the time window and capacity restrictions of the S-TD-VRPTWPD-UD.

3.2. Local searches

Most of the metaheuristic algorithms for the combinatorial optimization problems include, or are based upon, a local search heuristic (see, [Aarts and Lenstra, 1997](#) for an overview). In VRP, these procedures attempt to improve any feasible solution by exchanging a sequence of arcs or nodes within or between the vehicle routes. To intensify the solution space, we develop the local search phase described in this section. This phase is composed of some *inter-route* and *intra-route* neighborhood structures executed in a random order, every time the local search phase is called. The search continues until no improvement is found by applying any neighborhoods. The *intra-route* procedures operate on each vehicle route separately, while the *inter-route* ones deal with all routes at the same time. For both of two categories, we have developed *NodeRelocation*, *SequenceRelocation*, *SequenceSwap* and *Crossover* described in the following.

- **NodeRelocation:** The aim of this procedure is to improve the solution cost by reassigning each of the nodes. More precisely, a routed node i is extracted from its original position and inserted into its best position. The procedure is applied to all customer nodes in random order and stops when there is no improvement by the relocating of them.
- **SequenceRelocation:** This algorithm looks for the situations in which relocating a string of consecutive nodes into the new position would result in an improvement in the solution cost. We face two possible ways for the relocation of the sequence considering both the original and reversed order of that sequence. In the case of improving the solution cost, the move is implemented. The procedure is applied to all possible sequences of customers with at most three consecutive customers or less, in a random order, until no improvement could be achieved.
- **SequenceSwap:** Another approach to improve the solution cost could be exchanging all possible node sequences in the solution. We have restricted the length of sequences to at most three consecutive customers for the *inter-route* version and to one for the *intra-route* version. Suppose the *inter-route* version of this operator for a random pair of routes which is shown in [Fig. 2](#). Starting with two random strings $i_1 - i_2 - i_3$ and $j_1 - j_2$, the *SequenceSwap* procedure checks the four possible ways of exchanging presented in [Fig. 3](#). The best move improving the total cost of the solution is accepted. The procedure is applied to all pairs of routes and all possible strings with the mentioned restriction on the number of the included nodes, where both routes and strings are chosen randomly. The procedure continues until no improvement could be achieved.
- **Crossover:** This procedure substitutes two non-consecutive arcs with two new arcs to search for the improvement in the solution cost. All possible ways of reconnecting the generated partial routes considering the reversal of the routes are

Fig. 2. A pair of routes.

Fig. 3. Exchange two sequences on a pair of routes.

investigated and the best one improving the solution cost is implemented. Consider again the random pair of routes shown in [Fig. 2](#) for the *inter-route* version of this operator. [Fig. 4](#) presents the eight considered ways to substitute two random arcs (i_1, i_2) and (j_1, j_2) for this example. This algorithm is run for all pairs of routes and their arcs, where both routes and arcs are chosen randomly until no improvement could be obtained.

3.3. Shaking procedure

As a diversification mechanism, we have designed a shaking procedure that iteratively relocates a given percentage of nodes (say q_{dst}) in the solution. More precisely, a restricted solution $\mathcal{S}'$ is built on the current solution $\mathcal{S}^{ol_{cmt}}$ by extracting q_{dst} nodes from its original position, where q_{dst} is randomly generated in the interval $[q_{min}, q_{max}]$. The extraction is performed node-by-node with employing some removal procedures instead of a random way, inspired by the adaptive large neighborhood search (ALNS) method ([Ropke and Pisinger, 2006](#)). To choose the extraction operator $j \in \mathcal{R}$, we apply a *roulette wheel* mechanism working based on the weights of the operators. After every φ_3 iterations, these weights are dynamically updated using the *adaptive weight adjustment* procedure depending on their performance throughout the search in the diversification direction. Given the restricted solution $\mathcal{S}'$, we insert q_{dst} nodes randomly while the feasibility is maintained. This process continues until all the removed nodes are routed. The mentioned procedures are described in the following.

- **Adaptive weight adjustment procedure:** The choice of the removal operator $r \in \mathcal{R}$ is regulated by a *roulette wheel* mechanism, where initially all operators have equal chances. After every φ_3 iteration, the selection probabilities of each operator $r \in \mathcal{R}$ are updated via $\mathcal{P}_r^{t+1} = \mathcal{P}_r^t(1 - \lambda) + \pi_r \lambda / \omega_r$, where λ is the roulette wheel parameter called reaction factor, π_r is the score of operator r and ω_r refers to the number of times of applying operator r thorough the last φ_3 iterations. In the case of finding a new best solution, the score of the related operator is increased by σ_1 . If the solution is better than the current solution and not the best one, the score is augmented by σ_2 . In case of having the solution worse than the current solution but accepted, the score is increased by σ_3 . It is worth mentioning that the parameter setting in [Section 4.2](#) regulates these scores in the diversification direction in accordance with the nature of the addressed shaking procedure.

- **Removal procedures**

1. **Random removal:** This operator removes q_{dst} nodes from the solution at random.

Fig. 4. Crossovers on a pair of routes.

2. **Worst cost removal:** This operator iteratively removes q_{dst} highest cost nodes. This cost is calculated as the difference between the values of the objective function (3) where the node is in the solution and is not.
3. **Worst time removal:** Removing the requests with the highest waits or delays in service start times seems to be helpful followed iteratively by this operator until q_{dst} nodes are extracted. In S-TD-VRPTWPD-UD, waiting has been forbidden but to better search the solution space, it is allowed during the search with a big penalty to the objective function.
4. **Worst demand removal:** This operator searches the possibility of moving the pickup nodes toward the end of their routes to reduce the risk index and improve the feasibility from the capacity viewpoint. In this description, a node with a pickup amount greater than its delivery one is called the pickup node and vice versa for the delivery node. Therefore, the algorithm iteratively removes the pickup node with the highest positive summation of the net delivery demand of the next delivery nodes, i.e. $i^* = \operatorname{argmax}_{j \in \tilde{A}} \{(d_j - p_j), 0\}$, where $\tilde{A}$ is the set of nodes after the pickup node i .
5. **Shaw removal:** It seems easier to shuffle the related nodes in a predefined way where the idea of extracting similar nodes was first put forward by Shaw (1998). By randomly removing the first node, this operator iteratively extracts $q_{dst} - 1$ nodes via a relatedness measure including the similarity of the nodes with the previously removed node in four attributes including (1) distance, (2) start time of time window, (3) arrival time and (4) net delivery (pickup) demand of the delivery (pickup) nodes, where $\psi_1 - \psi_4$ are the weights of associated terms.

3.4. Departure time optimization

Sometimes the problem with zero departure time from the origin may not be feasible where the waiting time is not allowed. On the other hand, the delayed departure time may improve the solution quality due to the combination of the components at risk function (17). Also, it may provide better opportunities during the search. To handle such issues, we use the *departure time optimization algorithm* (DOA) finding the optimal departure time for a given vehicle route (Allahyari et al., 2021). This function increases the departure time of each vehicle from the origin so as to maximize the objective function by changing the two last components of risk definition as well as by eliminating disallowed waiting times.

At the beginning of the algorithm for each route, the minimum freedom (say χ) is calculated as the difference between the time to reach the customers and the end of their related time window

including waiting time. The value χ , splitting into $v (= 5)$ equal intervals, is added to the time to reach the nodes on the route, respectively from the beginning. This increase at a node with waiting time may lead to reducing this waiting and therefore the arrival times related to the next nodes (even perhaps with waiting) are not affected. In such a situation, the possibility of augmenting the departure time from the beginning is checked while the space for the subsequent intervals remains. Finally, the best departure time associated with the v intervals is chosen at minimized both risk index and waiting time.

4. Computational results

In this section, we present extensive computational experiments to evaluate the performance of the *GRASP × ILS* implementation with and without demand uncertainty and time dependency. More precisely, the characteristics of the generated data instances are explained in Section 4.1, followed by describing the parameters setting in Section 4.2. We provide several numerical experiments to analyze the solutions with time-dependent travel speeds in comparison with the solutions under constant speed in Section 4.3. Regarding the robust model, Section 4.4 presents the effect of the proposed robust model under different uncertainty levels versus the deterministic model. In Section 4.5, we finally report the results of the model S-TD-VRPTWPD-UD where both time-dependent travel speeds and uncertain demands are simultaneously considered. To show the efficiency of the *GRASP × ILS*, some experiments are also presented in Section 4.6 using the original Solomon's VRPTW dataset, related to the special case of our challenging problem.

Our algorithms have been implemented in C++ and tested on a server with 4.2 gigahertz speed and 4 gigabyte RAM. It is noted that the best and average solution values are reported over five runs and all presented calculation times are in seconds.

4.1. Test problems

In this section, we first describe the generation of the test instances based on the most well-known VRPTW benchmark Solomon (1987). To cover the various geographical locations of the nodes, we selected two classes R1 and RC1 including 20 instances. These sets consist of randomly distributed nodes and a combination of random and clustered nodes, respectively. All instances contain 100 customers and 25 capacitated vehicles, each of which has coordinates x and y , a delivery demand, a time window, and a service time.

The tailored features for S-TD-VRPTWPD-UD added to the Solomon dataset are as follows: (1) the usage cost of vehicles, (2) the pickup demand of nodes, (3) the number of realized days, (4) the

Table 2
Travel speed matrix for Scenario 1 and 2.

Scenario 1		T					
		1	2	3	4	5	6
c	1	0.58	0.68	0.81	0.68	0.58	0.51
	2	0.87	1.02	1.22	1.02	0.87	0.76
	3	1.27	1.49	1.78	1.49	1.27	1.11
Scenario 2		T					
		1	2	3	4	5	6
c	1	0.36	0.48	0.72	0.48	0.36	0.29
	2	0.72	0.95	1.43	0.95	0.72	0.57
	3	1.34	1.78	2.67	1.78	1.34	1.07

time to visit customers on each realized day, (5) the status of being traversed for each arc on realized days, (6) coefficients embedded in the risk function, i.e. $\alpha_1 - \alpha_4$, (7) time dependency settings and (8) uncertainty set of the pickup and delivery demands. For each instance, the cost of homogenous vehicles is generated as continuous values uniformly distributed in [20, 30]. The pickup demands are set randomly in the range of the delivery demands. We assume the value 2 for the number of realized days (i.e. $|\mathcal{B}|$). Two next items (4 and 5) are produced by running the *construction* procedure for $|\mathcal{B}|$ times. To set the coefficients of the risk function, i.e. $\alpha_1 - \alpha_4$, we have evaluated different values based on the expert opinion on the solutions over several runs of some random test data. The result is 0.007, 7, 0.01 and 0.07 for $\alpha_1 - \alpha_4$, respectively.

To calculate the travel times, we design a $c \times T$ time-dependent travel speed matrix for two different degrees of time-dependency in the form of two scenarios where Scenario 2 has a higher degree of time-dependence. The arcs are randomly assigned to three different categories ($c \in \{1, 2, 3\}$), where the speed increasingly changes from category 1 to 3. All instances use the same category matrix. For each category of arcs, we use the meaningful speed distribution with morning and evening congested time slots. The travel speed matrix with 3 categories and 6 equal time slots are presented for the two Scenarios 1 and 2 in Table 2. The travel speeds in the middle of the day, i.e. third slot, are divided by the values of 1.2, 1.4, 1.6 for Scenario 1 and 1.5, 2, 2.5 for Scenario 2 to obtain the speeds toward the beginning and the end slots, respectively. To have the same “average” difficulty with the original dataset, the average speed has been set to 1 in each matrix. Both scenarios are applied to each instance. It is noted that the Solomon instances are tightly constrained in the hard time windows mode. Therefore, a small decrease in travel speeds makes the problem infeasible by violating some time windows. To handle this problem, the categories of a few arcs connected to the depot are increased to the higher level.

4.2. Parameters setting

This section involves the fine-tuning of the user-controlled parameters by specifying the main effects of the objective function mean values using the Minitab software (see, <https://support.minitab.com/>). By defining some levels for the parameters, the aim is to determine which level of each parameter effectively minimizes the response value, i.e. here, the objective function of the S-TD-VRPTWPD-UD. Accordingly, we designed Taguchi tests including 27 cases for 7 parameters at the three considered levels stated in Table 3 instead of 3^7 tests (see, <https://support.minitab.com/>). To perform the Taguchi tests of Table 4, we randomly chose a tuning set consisting of 5 instances from our dataset, the results of which are reported in this table. The main effects plot of Fig. 5 examines

differences between level means for 7 parameters by graphing the mean of the objective function for each parameter level connected by a line. Corresponding to this figure, the chosen levels are asterisked in Table 3.

4.3. Experiments on the time-dependent situation

This section evaluates the effect of the time-dependency in travel speeds on the solutions of $GRASP \times ILS$ for the S-TD-VRPTWPD-UD dataset, where the demand requests take their nominal values (i.e., under deterministic situation). To do so, we use two designed Scenarios 1 and 2 in Table 2. Table 5 presents the solutions obtained with time-dependent travel speeds for Scenarios 1 and 2 and the scenario with constant speed 1.0 over all problems (recall the average speed in each matrix has been set to 1.0). For each instance in the constant speed situation, we report the objective value and the number of used vehicles in the columns labeled by “OFV” and “#Vhl”, respectively. For each instance in Scenarios 1 and 2, three numbers are reported including (1) the percentage deviation of the objective value from the objective value in the constant speed situation, (2) the percentage deviation of the total travel time from the total travel time in the constant speed situation and (3) the number of used vehicles, presented in the columns “Gap”, “GapTime” and “#Vhl”, respectively. The row “Overall” contains the averages taken over the entire set of R1 and RC1 test problems.

The results show the use of time-dependent travel speeds considerably improves the objective value over all instances by adding the degree of time-dependency except for r103 and r104. This improvement is also observed on average for the two next reported numbers. More precisely, the use of the time-dependent travel speeds of Scenario 1 results in 9.49%, 6.67% and 2.47% improvements in comparison with the constant speed situation for set R1 in the objective value, total travel time and the number of used vehicles, respectively. For set RC1, these numbers in the same order are 9.04%, 9.81% and 4.79% improvements in comparison with the constant speed situation. Using the higher degree of the time-dependency of Scenario 2, we observe these numbers in the same order as 11.02%, 8.97% and 3.13% for set R1 and 10.90%, 12.42% and 6.31% for set RC1 which show more improvements than the Scenario 1.

4.4. Experiments on the robust situation

To assess the performance of the proposed robust model, we consider three different uncertainty levels, i.e. $\mu = 0.0, 0.1, 0.2$, where the corresponding uncertainty set is defined as $[\bar{\xi} - \mu\hat{\xi}, \bar{\xi} + \mu\hat{\xi}]$ for each uncertain parameter ξ . As mentioned before, $\bar{\xi}$ and $\hat{\xi}$ denote the nominal value of ξ and the uncertainty

Table 3
Parameters.

Parameter	φ_1			φ_2			q_{min}, q_{max}			φ_3		
Level	1	2	3	1	2	3	1	2	3	1	2	3
Value	40	45 [*]	50	100	120	140 [*]	4,20	5, 24 [*]	6,26	75 [*]	320	450

Parameter	λ			$\sigma_1, \sigma_2, \sigma_3$			$\psi_1, \psi_2, \psi_3, \psi_4$		
Level	1	2	3	1	2	3	1	2	3
Value	0.07 [*]	0.10	0.13	3,2,1	1, 2, 3 [*]	3,1,1	1,1,3,3	3,2,1,3	1, 1, 1, 1 [*]

Table 4
Designed tests for Taguchi analysis.

Test	φ_1	φ_2	q	φ_3	λ	σ	ψ	r104	r110	r106	rc103	rc108
1	1	1	1	1	1	1	1	3373.92	3432.37	4064.80	4401.21	3790.83
2	1	1	1	1	2	2	2	3373.86	3428.82	4080.19	4394.90	3776.39
3	1	1	1	1	3	3	3	3380.65	3437.03	4096.13	4392.50	3790.62
4	1	2	2	2	1	1	1	3370.61	3443.43	4091.49	4368.96	3783.66
5	1	2	2	2	2	2	2	3362.21	3418.74	4087.69	4412.40	3789.45
6	1	2	2	2	3	3	3	3374.15	3442.52	4076.47	4386.76	3793.29
7	1	3	3	3	1	1	1	3372.87	3436.81	4094.44	4409.79	3789.23
8	1	3	3	3	2	2	2	3370.42	3436.47	4083.07	4410.22	3795.23
9	1	3	3	3	3	3	3	3358.81	3426.65	4080.64	4397.55	3780.74
10	2	1	2	3	1	2	3	3364.23	3421.29	4092.44	4393.22	3782.47
11	2	1	2	3	2	3	1	3375.24	3420.14	4089.01	4432.50	3778.19
12	2	1	2	3	3	1	2	3368.60	3428.27	4081.69	4390.03	3782.03
13	2	2	3	1	1	2	3	3372.93	3422.80	4078.89	4348.76	3776.71
14	2	2	3	1	2	3	1	3368.42	3421.77	4100.01	4404.10	3793.89
15	2	2	3	1	3	1	2	3364.26	3430.94	4080.45	4423.58	3800.80
16	2	3	1	2	1	2	3	3365.32	3434.37	4079.55	4379.95	3797.55
17	2	3	1	2	2	3	1	3369.33	3428.14	4086.09	4417.36	3780.47
18	2	3	1	2	3	1	2	3367.83	3421.60	4083.79	4339.34	3788.75
19	3	1	3	2	1	3	2	3372.80	3424.65	4104.55	4424.56	3792.82
20	3	1	3	2	2	1	3	3346.13	3421.20	4101.21	4405.14	3796.66
21	3	1	3	2	3	2	1	3375.81	3440.19	4074.82	4405.25	3793.39
22	3	2	1	3	1	3	2	3369.96	3430.13	4075.61	4402.73	3777.28
23	3	2	1	3	2	1	3	3373.73	3422.40	4087.93	4394.83	3793.04
24	3	2	1	3	3	2	1	3373.83	3434.69	4078.36	4417.36	3791.13
25	3	3	2	1	1	3	2	3367.55	3439.21	4066.91	4391.16	3797.90
26	3	3	2	1	2	1	3	3369.71	3414.92	4081.47	4419.63	3790.86
27	3	3	2	1	3	2	1	3366.15	3403.15	4076.59	4397.58	3794.24

Fig. 5. Main effects plot for means.

Table 5
Results under time-dependent travel speeds of two Scenarios 1 and 2 and under constant speed.

Instance	Constant speed		Scenario 1			Scenario 2		
	OFV	#Vhl	Gap	GapTime	#Vhl	Gap	GapTime	#Vhl
r101	12472.40	24	-11.61	-3.41	24	-14.32	-11.21	22
r102	6963.75	20	-24.22	-4.42	21	-27.02	-5.76	20
r103	4813.64	16	-18.40	-8.14	16	-16.59	-7.92	16
r104	3513.72	13	-12.05	-7.49	13	-11.61	-7.85	13
r105	6652.30	19	-13.02	-7.58	18	-17.20	-15.97	16
r106	3902.71	15	-3.62	0.58	16	-4.52	0.70	16
r107	3380.27	14	-3.24	-8.99	13	-4.00	-9.49	13
r108	3014.38	12	-5.27	-9.36	11	-6.97	-10.72	12
r109	3520.38	16	-6.15	-3.26	15	-8.09	-8.42	14
r110	3351.19	16	-5.30	-8.18	15	-7.39	-11.23	15
r111	3320.10	14	-5.54	-7.67	13	-6.67	-7.61	14
r112	3000.41	14	-5.47	-12.17	13	-7.85	-12.18	14
Overall	4825.44	16.00	-9.49	-6.67	15.67	-11.02	-8.97	15.42
rc101	7897.78	20	-13.14	-7.03	19	-13.55	-11.21	18
rc102	5064.09	18	-5.66	-9.08	16	-6.94	-11.53	16
rc103	4173.78	14	-5.37	-8.08	14	-6.74	-9.72	13
rc104	3636.23	13	-6.86	-11.10	13	-8.99	-11.68	13
rc105	7390.08	18	-19.40	-11.75	17	-20.66	-16.41	16
rc106	4717.30	18	-8.19	-15.04	15	-11.05	-14.90	16
rc107	3996.94	14	-7.03	-8.78	14	-8.85	-12.78	14
rc108	3661.32	14	-6.70	-7.60	14	-10.40	-11.10	14
Overall	5067.19	16.13	-9.04	-9.81	15.25	-10.90	-12.42	15.00

Table 6
Results under the different uncertainty levels of ($\mu =$) 0.0, 0.1 and 0.2.

Instance	Deterministic ($\mu = 0.0$)		$\mu = 0.1$		$\mu = 0.2$	
	OFV	#Vhl	Gap	#Vhl	Gap	#Vhl
r101	12472.40	24	1.85	25	3.19	25
r102	6963.75	20	1.94	21	4.66	21
r103	4813.64	16	2.53	17	7.18	16
r104	3513.72	13	3.23	13	9.25	15
r105	6652.30	19	2.26	19	5.58	20
r106	3902.71	15	4.90	16	10.73	15
r107	3380.27	14	4.28	14	10.45	14
r108	3014.38	12	4.33	13	10.87	13
r109	3520.38	16	3.85	16	9.26	16
r110	3351.19	16	3.66	16	10.04	17
r111	3320.10	14	3.96	14	9.99	15
r112	3000.41	14	3.83	14	9.56	15
Overall	4825.44	16.00	3.44	16.50	8.46	16.83
rc101	7897.78	20	2.96	21	6.76	20
rc102	5064.09	18	4.72	18	10.35	18
rc103	4173.78	14	6.91	14	13.83	17
rc104	3636.23	13	6.90	13	14.93	15
rc105	7390.08	18	3.69	19	6.50	20
rc106	4717.30	18	5.28	18	10.47	18
rc107	3996.94	14	6.66	15	14.37	18
rc108	3661.32	14	6.66	15	12.99	16
Overall	5067.19	16.13	5.48	16.63	11.29	17.75

scale, respectively. A particular case of interest for the uncertainty scale, which we have used, is $\hat{\xi} = \bar{\xi}$, where box contains ξ whose relative deviation from the nominal value is of size up to μ .

Table 6 quantifies the relative extra cost of the robust solution to protect against the worst-case scenario in comparison with the cost of the deterministic solution. This cost is given by $100 * (z_r - z_d) / z_d$ in the columns labeled by “Gap”, where z_r and z_d are the objective value of the robust approach with worst-case demand and the objective value of the deterministic one, both

found by the $GRASP \times ILS$. The columns “OFV” and “#Vhl” report the objective value under no time-dependence situation and the corresponding number of used vehicles, respectively. By increasing the uncertainty level from zero to 0.1, we can protect the solution against the worst-case scenario of the demand requests with 3.44% and 5.48% increase in objective value on average for R1 and RC1, respectively. Similarly, these numbers are obtained as 8.46% and 11.29% in the situation we raise the uncertainty level from zero to 0.2.

Table 7

Results under the simultaneous time-dependent travel speeds and uncertain demands on the S-TD-VRPTWPD-UD dataset.

Instance	Constant speed			Scenario 1			Scenario 2		
	$\mu = 0.0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.0$	$\mu = 0.1$	$\mu = 0.2$
r101	12472.40	12703.00	12870.30	11023.90	11179.50	11396.50	10686.40	10833.20	11032.80
r102	6963.75	7111.88	7301.32	5277.44	5429.31	5609.73	5082.18	5231.38	5394.12
r103	4813.64	4935.53	5159.39	3927.77	4043.38	4205.56	4015.13	4156.29	4323.50
r104	3513.72	3635.20	3846.91	3090.42	3194.38	3366.25	3105.65	3192.63	3357.92
r105	6652.30	6802.69	7023.28	5786.23	5867.68	6075.73	5507.81	5609.11	5786.74
r106	3902.71	4093.79	4321.38	3761.59	3884.12	4069.45	3726.35	3850.75	4024.03
r107	3380.27	3524.83	3733.39	3270.90	3378.65	3576.02	3245.17	3349.67	3520.25
r108	3014.38	3145.00	3342.09	2855.44	2960.95	3122.89	2804.32	2887.34	3053.76
r109	3520.38	3658.29	3849.00	3303.86	3396.95	3582.14	3235.54	3330.63	3496.67
r110	3351.19	3473.80	3687.81	3173.57	3264.36	3410.45	3103.47	3193.08	3328.33
r111	3320.10	3459.20	3660.06	3136.09	3253.08	3424.99	3098.73	3207.59	3355.97
r112	3000.41	3115.19	3287.28	2836.15	2930.95	3075.85	2764.82	2847.06	2966.51
Overall	4825.44	4971.53	5173.52	4286.95	4398.61	4576.30	4197.96	4307.39	4470.05
rc101	7897.78	8131.33	8431.89	6860.08	7078.03	7310.87	6827.62	6996.78	7216.04
rc102	5064.09	5303.18	5587.98	4777.37	4971.81	5224.39	4712.51	4957.86	5203.67
rc103	4173.78	4462.11	4751.00	3949.83	4128.55	4327.67	3892.66	4089.36	4284.41
rc104	3636.23	3890.51	4182.69	3386.91	3577.07	3812.21	3309.43	3469.90	3688.20
rc105	7390.08	7662.88	7870.09	5956.37	6111.94	6386.24	5863.48	6035.93	6304.41
rc106	4717.30	4966.49	5211.42	4330.79	4509.07	4758.74	4195.82	4368.86	4551.70
rc107	3996.94	4262.97	4571.20	3716.07	3900.90	4155.06	3643.10	3811.16	4012.68
rc108	3661.32	3905.17	4137.07	3416.08	3584.16	3777.64	3280.45	3409.95	3585.76
Overall	5067.19	5323.08	5592.92	4549.19	4732.69	4969.10	4465.63	4642.48	4855.86

4.5. Experiments on the S-TD-VRPTWPD-UD

Table 7 presents the detailed results of the S-TD-VRPTWPD-UD dataset by simultaneously considering of the proposed time-dependency in travel speeds and uncertainty in demands which were discussed in Sections 4.3 and 4.4. Similar to these sections, we use (1) two designed Scenarios 1 and 2 for the time-dependent travel speeds and (2) uncertainty levels $\mu = 0.1$ and 0.2 for the robust model, against the constant speed and the certain demands ($\mu = 0.0$), respectively. The reported numbers in the table show the objective value of each instance. The results are similar to Sections 4.3 and 4.4 regarding two proposed features of the time-dependency and the demand uncertainty, respectively. More precisely, using Scenarios 1 and 2 yields at least 9.31% and 10.97% improvements averagely in comparison with the constant speed situation, respectively. Moreover, protecting the time-dependent model against the worst-case scenario of demand requests leads to at most 3.50% and 8.45% extra cost averagely in comparison with the cost of the deterministic model under the uncertainty levels of 0.1 and 0.2, respectively.

4.6. Benchmarking the GRASP×ILS implementation

In order to assess the GRASP × ILS implementation, this section provides some experiments. We benchmark the implementation using the original Solomon's VRPTW dataset (Solomon, 1987). To do so, we relax some features of the S-TD-VRPTWPD-UD including the risk elements, the time-dependency, the demand

uncertainty, and the pickup option in addition to allowing waiting times.

Table 8 presents the results of the GRASP × ILS implementation in comparison with some state-of-the-art solution techniques for the three classes R1, RC1, and C1 of Solomon's dataset. For each method reported in columns two to eleven, "Bst" and "#Vhl" represent the best-obtained solution and the corresponding number of used vehicles, respectively. The five last columns represent (1) the best number of used vehicles, (2) the related solution value, (3) the running time in seconds, (4) the percentage deviation of this solution from the best solution among the mentioned state-of-the-art solution techniques that have the same number of used vehicles and (5) the average percentage deviation of the comparison performed in the previous item, respectively. The row "Overall" contains averages taken over the entire set of R1, RC1, and C1 test problems.

The negative gaps reported in the two last columns confirm the effectiveness of GRASP × ILS in fair comparison with other heuristics that have the same number of used vehicles. More precisely, we observe the three gaps -0.63% , -0.14% , and 0.00% as the gap with the best solution and the gaps -2.43% , -2.10% , and -0.16% as the average gap, averaged for respectively R1, RC1, and C1 sets. The GRASP × ILS dominates Ghannadpour et al. (2014) in all instances (for C1 is equal), from both total traveled distance and the number of used vehicles viewpoints, other than one instance, i.e. rc107. This is also averagely for Ursani et al. (2011) over both sets R1 and RC1, Yu et al. (2011) over both sets R1 and C1 as well as Hong (2012) over set C1, all from both objectives points of view.

Table 8
Results of benchmarking the GRASP $\times$ ILS.

Instance	Cordeau & Maischberger ^a (Cordeau and Maischberger, 2012)		Hong ^b (Hong, 2012)		Ursani et al. ^c (Ursani et al., 2011)		Ghannadpour et al. ^d (Ghannadpour et al., 2014)		Yu et al. ^e (Yu et al., 2011)		GRASP $\times$ ILS				
	#Vhl	Bst	#Vhl	Bst	#Vhl	Bst	#Vhl	Bst	#Vhl	Bst	#Vhl	Bst	Time	Gap	Avg. gap
r101	19	1650.80	18	1612.29	20	1646.90	19	1677.00	19	1655.03	19	1650.80	301.27	0.00	-0.61
r102	17	1486.12	16	1473.41	18	1474.28	17	1566.80	18	1491.18	17	1487.31	362.71	0.08	-2.50
							18	1504.10			18	1472.81		-0.10	-1.14
r103	13	1294.23	12	1279.37	15	1222.68	14	1287.00	14	1243.22	14	1213.62	422.83	-2.38	-4.04
r104	10	981.20	10	1025.47	11	989.53	10	974.24	10	982.01	10	987.71	480.93	1.38	-0.26
											11	978.65		-1.10	-1.10
r105	14	1377.11	14	1377.95	16	1382.78	15	1424.60	16	1380.44	14	1384.89	376.57	0.56	0.53
							16	1382.50			15	1360.78		-4.48	-4.48
r106	12	1252.62	12	1276.48	13	1250.11	13	1269.00	13	1265.36	13	1239.98	440.70	-0.81	-1.70
r107	10	1104.66	11	1153.86	12	1083.42	11	1108.80	11	1100.25	11	1079.22	452.70	-1.91	-3.68
r108	9	963.99	10	990.82	10	952.44	10	971.91	9	958.66	10	939.90	495.06	-1.32	-3.25
r109	11	1194.73	12	1179.73	13	1160.69	12	1222.70	12	1101.99	12	1160.08	422.79	5.27	-0.51
r110	10	1118.84	11	1113.10	12	1080.69	12	1156.50	12	1119.53	12	1077.97	453.78	-0.25	-3.58
r111	10	1096.73	11	1155.39	12	1057.64	11	1111.90	12	1091.11	11	1067.60	460.21	-3.98	-5.79
r112	9	989.27	10	981.46	10	965.00	10	977.00	10	974.73	10	960.52	502.68	-0.46	-1.44
Overall	12.00	1209.19	12.25	1218.28	13.50	1188.85	12.83	1259.58	13.00	1196.96	12.75	1187.47	431.02	-0.63	-2.43
rc101	14	1696.95	15	1671.54	16	1660.55	15	1690.60	15	1650.14	15	1642.86	344.96	-0.44	-1.66
rc102	12	1554.75	13	1447.14	15	1494.92	15	1493.20	13	1514.85	13	1505.49	383.18	4.03	1.71
											14	1473.37		-1.33	-1.38
rc103	11	1261.67	11	1313.79	12	1276.05	12	1331.80	11	1277.11	12	1279.15	406.35	0.24	-1.86
rc104	10	1135.48	11	1163.54	10	1151.63	11	1156.10	10	1159.37	10	1143.28	441.71	0.69	-0.48
rc105	13	1633.72	13	1502.48	16	1556.21	15	1611.50	15	1617.88	15	1528.22	367.33	-5.17	-5.35
rc106	11	1424.73	12	1406.25	14	1402.25	13	1437.60	13	1387.63	13	1383.57	382.65	-0.29	-2.03
rc107	11	1232.20	11	1278.96	12	1212.83	11	1222.10	11	1280.01	11	1239.81	404.65	1.45	-1.03
rc108	10	1147.69	11	1172.83	11	1133.25	11	1156.20	11	1157.44	11	1128.42	433.68	-0.43	-2.28
Overall	11.50	1385.90	12.13	1369.57	13.25	1360.96	12.88	1387.39	12.38	1380.55	12.50	1356.35	395.56	-0.14	-2.10
c101	10	828.94	10	828.94	10	828.94	10	828.94	10	828.93	10	828.94	401.07	0.00	0.00
c102	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94	436.30	0.00	0.00
c103	10	828.06	10	839.37	10	828.06	10	828.06	10	828.06	10	828.06	458.50	0.00	-0.34
c104	10	824.78	10	838.98	10	824.78	10	824.78	10	828.20	10	824.78	473.28	0.00	-0.53
c105	10	828.94	10	828.94	10	828.94	10	828.94	10	828.90	10	828.94	406.44	0.00	0.00
c106	10	828.94	10	842.10	10	828.94	10	828.94	10	828.94	10	828.94	427.50	0.00	-0.39
c107	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94	419.64	0.00	0.00
c108	10	828.94	10	832.74	10	828.94	10	828.94	10	830.94	10	828.94	447.13	0.00	-0.17
c109	10	828.94	10	828.94	10	828.94	10	828.94	10	829.22	10	828.94	450.12	0.00	-0.01
Overall	10.00	828.38	10.00	833.10	10.00	828.38	10.00	828.38	10.00	829.01	10.00	828.38	435.55	0.00	-0.16

^a For the sequential algorithm, on a Intel Xeon CPU X 7 350 (2.93 GHz); for the parallel algorithm on a Linux cluster with 128 nodes and infiniband interconnections, each node being equipped with a dual Intel Xeon CPU E5 472 (3.0 GHz). Run time has not been reported.

^b On a notebook computer with Intel Core 2 Duo CPU T8300 (2.40 GHz) and 2046 MB RAM in c# programming language.

^c On a PC (3.00 GHz) running under the Unix operating system. Run time has not been reported.

^d On a PC with Core 2 Duo CPU (3.00 GHz) and 2.9 GB RAM. Run time has not been reported.

^e On a PC (1.00 GHz) equipped with 512 MB of RAM in C++ programming language. Run time has not been reported.

5. Conclusions

In this paper, we introduced an extension of the well-known VRP, namely S-TD-VRPTWPD-UD, interested by security carriers in the transportation of valuable goods. The aim of S-TD-VRPTWPD-UD is to optimize the operating costs and the security of vehicle routes to mechanize the manual work of real operations in an accurate way with more realistic assumptions. Hereupon, we considered demand fluctuations in the nature of the problem as well as traffic congestion in urban environments. To do so, these considerations respectively benefited from the robust optimization theory and considering the time-dependent travel speeds with fulfilling the “first-in-first-out” feature. A mathematical formulation and an efficient solution approach combining the GRASP and ILS were developed to minimize the travel costs and the predictability of route plans using an integrated risk index. We performed several experiments to evaluate the model in a time-dependent and an uncertain environment. The results indicated that the time-dependent model leads to the significant improvements against

the model with constant travel speeds. By deriving the robust counterpart of the S-TD-VRPTWPD-UD, we calculated the extra cost of protecting the time-dependent model against the worst-case scenario of demand requests in comparison to the cost of the deterministic model under some uncertainty levels. Finally, the efficiency of the GRASP $\times$ ILS was shown on the VRPTW Solomon's dataset regarding some of the state-of-the-art solution techniques. As a future research, we suggest analyzing the solutions when the risk index is constrained with a given threshold instead of considering it as a part of the objective function.

CRediT authorship contribution statement

Somayeh Allahyari: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing. **Saeed Yaghoubi:** Writing - review & editing, Resources, Project administration. **Tom Van Woensel:** Supervision, Conceptualization, Methodology, Writing - review & editing.

References

- Aarts, E.E.H., Lenstra, J.K., 1997. Local search in combinatorial optimization. Princeton University Press.
- Allahyari, S., Salari, M., Vigo, D., 2015. A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem. *Eur. J. Oper. Res.* 242 (3), 756–768.
- Allahyari, S., Yaghoubi, S., Van Woensel, T., 2021. A novel risk perspective on location-routing planning: An application in cash transportation. Working paper.
- Balseiro, S.R., Loiseau, I., Ramonet, J., 2011. An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers Oper. Res.* 38 (6), 954–966.
- Ben-Tal, A., Nemirovski, A., 1998. Robust convex optimization. *Math. Oper. Res.* 23 (4), 769–805.
- Ben-Tal, A., El Ghaoui, L., Nemirovski, A., 2009. Robust optimization, Vol. 28. Princeton University Press.
- Bertsimas, D.J., Simchi-Levi, D., 1996. A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Oper. Res.* 44 (2), 286–304.
- Bertsimas, D., Sim, M., 2004. The price of robustness. *Oper. Res.* 52 (1), 35–53.
- Calvo, R.W., Cordone, R., 2003. A heuristic approach to the overnight security service problem. *Computers Oper. Res.* 30 (9), 1269–1287.
- Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12 (4), 568–581.
- Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F., 2002. A guide to vehicle routing heuristics. *J. Oper. Res. Soc.* 53 (5), 512–522.
- Cordeau, J.F., Maischberger, M., 2012. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers Oper. Res.* 39 (9), 2033–2050.
- Dabia, S., Ropke, S., Van Woensel, T., De Kok, T., 2013. Branch and price for the time-dependent vehicle routing problem with time windows. *Transp. Sci.* 47 (3), 380–396.
- Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. *Manage. Sci.* 6, 80–91.
- Donati, A.V., Montemanni, R., Casagrande, N., Rizzoli, A.E., Gambardella, L.M., 2008. Time dependent vehicle routing problem with a multi ant colony system. *Eur. J. Oper. Res.* 185 (3), 1174–1191.
- Erkut, E., Ingolfsson, A., 2005. Transport risk models for hazardous materials: revisited. *Oper. Res. Lett.* 33 (1), 81–89.
- Gabrel, V., Murat, C., Thiele, A., 2014. Recent advances in robust optimization: An overview. *Eur. J. Oper. Res.* 235 (3), 471–483.
- Gendreau, M., Ghiani, G., Guerriero, E., 2015. Time-dependent routing problems: A review. *Computers Oper. Res.* 64, 189–197.
- Gounaris, C.E., Wiesemann, W., Floudas, C.A., 2013. The robust capacitated vehicle routing problem under demand uncertainty. *Oper. Res.* 61 (3), 677–693.
- Harwood, K.G., 2013. Investigation into heuristic methods of solving time variant Vehicle Routing Problems (Doctoral dissertation). Cardiff University.
- Hong, L., 2012. An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers Oper. Res.* 39 (2), 151–163.
- Feo, T.A., Resende, M.G., 1995. Greedy randomized adaptive search procedures. *J. Global Optim.* 6, 109–133.
- Figliozzi, M.A., 2012. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transp. Res. Part E: Logistics Transp. Rev.* 48 (3), 616–636.
- Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R., Ghoseiri, K., 2014. A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Appl. Soft Comput.* 14, 504–527.
- Ichoua, S., Gendreau, M., Potvin, J.Y., 2003. Vehicle dispatching with time-dependent travel times. *Eur. J. Oper. Res.* 144 (2), 379–396.
- Kempton Express, (2017, September). 49% increase in cash in transit attacks in 2017. <https://kemptonexpress.co.za/156546/49-increase-in-cash-in-transit-attacks-in-2017/>.
- Krarup, J., 1995. The peripatetic salesman and some related unsolved problems. In: *Combinatorial programming: methods and applications*. Springer, Dordrecht, pp. 173–178.
- Li, Z., Ding, R., Floudas, C.A., 2011. A comparative theoretical and computational study on robust counterpart optimization: Robust linear optimization and robust mixed integer linear optimization. *Ind. Eng. Chem. Res.* 50 (18), 10567–10603.
- Malandraki, C., Daskin, M.S., 1992. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transp. Sci.* 26 (3), 185–200.
- Lourenço, H.R., Martin, O., Stützle, T., 2003. Iterated local search. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, vol. 57. Kluwer Academic Publishers, pp. 321–353.
- Michallet, J., Prins, C., Amodeo, L., Yalaoui, F., Vitry, G., 2014. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers Oper. Res.* 41, 196–207.
- Ngueveu, S.U., Prins, C., Calvo, R.W., 2009. A hybrid tabu search for the m-peripatetic vehicle routing problem. In: *Matheuristics*. Springer, Boston, MA, pp. 253–266.
- Prins, C., 2009. A GRASP × evolutionary local search hybrid for the vehicle routing problem. *Bio-inspired Algorithms for the Vehicle Routing Problem*. Springer, pp. 35–53.
- Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* 40 (4), 455–472.
- Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: *International conference on principles and practice of constraint programming*. Springer, Berlin, Heidelberg, pp. 417–431.
- Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35 (2), 254–265. VRP web <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/>.
- Soyster, A.L., 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.* 21 (5), 1154–1157.
- Sungur, I., Ordnez, F., Dessouky, M., 2008. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IEE Trans.* 40 (5), 509–523.
- Talarico, L., Sörensen, K., Springael, J., 2015. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *Eur. J. Oper. Res.* 244 (2), 457–470.
- Talarico, L., Sörensen, K., Springael, J., 2015. The k-dissimilar vehicle routing problem. *Eur. J. Oper. Res.* 244 (1), 129–140.
- Talarico, L., Sörensen, K., Springael, J., 2017. A biobjective decision model to increase security and reduce travel costs in the cash-in-transit sector. *Int. Trans. Oper. Res.* 24 (1–2), 59–76.
- Toth, P., Vigo, D., 2014. *Vehicle Routing: Problems, Methods, and Applications*, Vol. 18. SIAM, Philadelphia.
- Ursani, Z., Essam, D., Cornforth, D., Stocker, R., 2011. Localized genetic algorithm for vehicle routing problem with time windows. *Appl. Soft Comput.* 11 (8), 5375–5390.
- Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N., 2008. Vehicle routing with dynamic travel times: A queueing approach. *Eur. J. Oper. Res.* 186 (3), 990–1007.
- Yan, S., Wang, S.S., Chang, Y.H., 2014. Cash transportation vehicle routing and scheduling under stochastic travel times. *Eng. Optim.* 46 (3), 289–307.
- Yan, S., Wang, S.S., Wu, M.W., 2012. A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem. *Computers Ind. Eng.* 63 (2), 464–473.
- Yu, B., Yang, Z.Z., Yao, B.Z., 2011. A hybrid algorithm for vehicle routing problem with time windows. *Expert Syst. Appl.* 38 (1), 435–441.