

DEPARTMENT OF ENGINEERING MANAGEMENT

**The Risk Constrained Cash-in-Transit Vehicle Routing  
Problem with Time Windows**

**Luca Talarico, Kenneth Sørensen & Johan Springael**

**UNIVERSITY OF ANTWERP**  
**Faculty of Applied Economics**



Stadscampus  
Prinsstraat 13, B.226  
BE-2000 Antwerpen  
Tel. +32 (0)3 265 40 32  
Fax +32 (0)3 265 47 99  
[www.ua.ac.be/tew](http://www.ua.ac.be/tew)

# **FACULTY OF APPLIED ECONOMICS**

DEPARTMENT OF ENGINEERING MANAGEMENT

## **The Risk Constrained Cash-in-Transit Vehicle Routing Problem with Time Windows**

**Luca Talarico, Kenneth Sørensen & Johan Springael**

RESEARCH PAPER 2013-012  
JULY 2013

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium  
Research Administration – room B.226  
phone: (32) 3 265 40 32  
fax: (32) 3 265 47 99  
e-mail: [joeri.nys@ua.ac.be](mailto:joeri.nys@ua.ac.be)

**The papers can be also found at our website:**  
[www.ua.ac.be/tew](http://www.ua.ac.be/tew) (research > working papers) &  
[www.repec.org/](http://www.repec.org/) (Research papers in economics - REPEC)

**D/2013/1169/012**

# The Risk Constrained Cash-in-Transit Vehicle Routing Problem with Time Windows

L. Talarico\*, K. Sörensen, J. Springael

University of Antwerp Operations Research Group ANT/OR

Prinsstraat 13, 2000 Antwerp, Belgium

July 2013

This paper proposes a variant of the Vehicle Routing Problem in which a particular kind of risk constraint has been introduced to model the problem of routing vehicles in the cash-in-transit industry and a hard time window constraint, with no waiting times, delimits the customer's visit within a specified time interval. Two metaheuristic algorithms have been developed to cope with medium and large instances of the problem. In a computational experiment, the best parameter settings for each algorithm are determined. The resulting metaheuristics are compared, in their best possible setting, solving some benchmark instances for the capacitated vehicle routing problem with time windows.

**Key words:** Vehicle routing; Risk; Security; Cash-in-Transit; Metaheuristic.

## 1 Introduction

Vehicle routing problems (VRP) have been widely studied by a large number of researchers during the last years. Different variants of the VRP have been modelled to represent many real-life problems (e.g., waste collection, post distributions, dangerous materials transportation). An interesting field of application, which has not received much attention so far, concerns the issue of *security* during the transportation of cash or valuable goods.

The cash-in-transit (CIT) industry groups transportation companies that deal with the physical transfer of banknotes, coins and items of value from one location to another. In general the transfer of cash and valuables happens between customers (e.g., jewellery stores, shopping centres, retail stores, casinos, and other locations where large amounts

---

\*Corresponding author: University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium, Tel:+32 3 265 40 17, E-mail: luca.talarico@ua.ac.be

of cash or valuables are present) and one or more cash deposits or banks. It is clear that, as a consequence of the nature of the transported goods, CIT companies are constantly exposed to real risks such as robbery, armed assault and so on.

During the last decades a lot of efforts have been made to prevent attacks on CIT vehicles. Substantial investments have been made by CIT companies in vehicles, equipment, infrastructure and technology (e.g., armoured vehicles, weapons on board, on-board drop safes and interlocking doors, active vehicle tracking). However a careful planning of the activities is necessary to reduce the risk even further.

An approach suggested in the literature is to reduce the risk of being attacked by building routes that are “unpredictable” for criminals. In so-called “peripatetic” routing problems [1–3], customers are visited several times, but the use of the same arc twice is explicitly forbidden by using time windows.

In previous work of Talarico et al. [4] the Risk-constrained Cash-in-Transit Vehicle Routing Problem (RCTVRP for short) was introduced. The RCTVRP is a variant of the VRP problem where the total risk that each vehicle incurs during its route, is limited by a pre-specified *risk threshold*. Using a specific risk index that is proportional both to the amount of cash being carried and the time or distance covered by the vehicle carrying the cash, it is possible to measure and limit the global risk faced by each vehicle along its route.

In this paper an additional real-life constraint is introduced in the RCTVRP. In particular we define a hard time window constraint for which no waiting times are allowed. This corresponds to the way CIT companies work in reality: CIT companies have to visit customers within their time window and for security reasons, they never wait at the customer location until the opening of its time window. The model can also be used for other sectors such as the chemical industry for the picking up of dangerous substances by defining an appropriate risk index.

The main contributions of this article are fourfold. (1) A new variant of the RCTVRP is proposed, introducing a hard time window constraint for which no waiting times are allowed. This new problem is called the Risk constrained Cash-in-Transit Vehicle Routing Problem with Time Windows (RCTVRPTW). (2) A mathematical formulation of the RCTVRPTW is presented. (3) Two efficient metaheuristic algorithms to solve medium and large instances of the RCTVRPTW are introduced. (4) A new set of benchmark instances, derived from the instances for the capacitated VRP problem with time windows, has been defined and it can be used as a library of instances for the RCTVRPTW (the instances are available at <http://antor.ua.ac.be/downloads>).

The remainder of the paper is organized as follows. In Section 2 the literature on vehicle routing with time window constraints and in risk-prone situations is surveyed. Section 3 outlines the problem description and a mathematical formulation for the RCTVRPTW is presented. In Section 4 the different components of the metaheuristic solution strategies are described. In Section 5, the metaheuristics are tested and computational results are

reported. Section 6 presents some conclusions and suggestions for future research as well as some possible extensions of the problem.

## 2 Literature

In general the risk, as defined by the Center for Chemical Process Safety [5], can be seen as a cardinal measure of potential economic loss, human injury or environmental damage in terms of both incident probability and magnitude of the loss, injury, or damage. Thus the *risk* associated with a specific (unwanted) event can be expressed by the product of two factors: the likelihood that the event will occur ( $p_{\text{event}}$ ) and its consequences ( $C_{\text{event}}$ ):

$$R_{\text{event}} = p_{\text{event}} \cdot C_{\text{event}} \quad (1)$$

In Russo and Rindone [6] risk is defined in terms of three main components: (a) the occurrence of an event (expressed as the probability or the frequency of a specific unwanted event happening); (b) the vulnerability defined as a measure of the susceptibility of the objects to be protected to unwanted events; (c) the exposure that represents a weighted value of people, goods and infrastructure affected during and after the event.

In Talarico et al. [4], the general formulation of the risk (in formula (1)) has been translated for the CIT sector. In particular the unwanted event has been associated to the loss of cash or valuables after a robbery and the probability of this event (e.g., the probability of being robbed) has been related to the exposure of the vehicle along the route.

The concept of risk has been widely studied in the context of the shortest path problem, whereas it has only received limited attention in the field of vehicle routing problems. In the domain of hazardous material transportation, many works add a risk index in the formulation of the basic shortest path problem, in order to find a “safe” and shortest path from an origin to a destination. In general the shortest path problem and all the derived problems (e.g.,  $k$  shortest paths) are aimed at finding the path with the minimum cost between a given origin and destination, passing through some intermediate nodes of the network.

An early contribution to define the risk on a path from an origin to a destination node, is due to Pijawka et al. [7]. In this paper, the concept of “vulnerability assessment” is introduced and used to evaluate transportation risks and capabilities to respond to and mitigate the hazards. A risk score for individual routes is presented, reflecting the interaction of four variables: (a) the number of hazardous events that have occurred on the route, (b) the hazardous materials accident probability, (c) the population-at-risk and potential hazard rating and (d) the volume of hazardous materials by class. Always considering a shortest path problem, in Scanlon and Cantilli [8] the risk is managed by

evaluating the consequences of an accident for the community traversed by the path. Abkowitz and Cheng [9] present a mathematical model for the bi-objective problem in which both the path cost and the risk of the path are considered while transporting hazardous materials.

For many routing application (e.g., routing of tankers which must serve a set of petrol stations) a vehicle or a fleet of vehicles is used to serve all the customers in the network. Moreover each customer must be visited exactly once. This problem is known in the literature as the vehicle routing problem (VRP). The VRP was proposed for the first time by Dantzig and Ramser [10] and due to the important real-life applications in the fields of transportation, distribution and logistics, it has attracted the attention of the scientific community during the last decades.

The aim of the VRP problem is to determine  $N$  vehicle routes, where a route is a tour that begins at the depot, traverses a subset of the customers in a specified sequence and returns to the depot. Each customer must be assigned to exactly one of the  $N$  vehicle routes and the total size of deliveries assigned to each vehicle must not exceed the vehicle capacity.

One of the early contribution considering a vehicle routing problem instead of a shortest path problem is due to Tarantilis and Kiranoudis [11]. In particular the paper focuses on mitigating the population exposure to the risk of accidents by solving a variant of the vehicle routing problem. The goal is to select routes not close to aggregate populations points in order to reduce the population exposed to risk. In Zografos and Androutsopoulos [12] an heuristic for solving the bi-objective vehicle routing problem with time windows, where both risk and travel cost need to be minimized, is presented. In this paper the risk index is defined as the product between the hazardous materials accident probability and the population that is contained in a given distance from the roadway segment.

Time windows in vehicle routing problems occur naturally in many real-life situations such as postal deliveries, bank deliveries, school bus routes, etc. The importance of time window constraints is related to the specific application. For instance in cases like postal deliveries, a violation of the time windows may cause only customer dissatisfaction with a relatively small loss of profit, whereas late delivery of raw material for a factory operating according to the just-in-time principles, may cause disturbances in the production and consequently lead to significant losses (see Kontoravdis and Bard [13]).

A brief description of the most common time window constraint applied to routing problems is given in Solomon [14]. The service at a customer  $i$ , involving pickup and/or delivery of goods, can begin at time  $b_i$ , within a time window  $[e_i, l_i]$ . The value  $e_i$  represents the earliest time whereas  $l_i$  is the latest time that customer  $i$  will permit the start of service. The time needed to serve customer  $i$  is denoted by  $s_i$ . Hence, if a vehicle travels directly from customer  $i$  to customer  $j$  and arrives too early at  $j$ , it will wait, that is,  $b_j = \max(e_j, b_i + s_i + d_{ij})$ , where  $d_{ij}$ , is the direct travel time between  $i$  and  $j$ . Note that the times  $b_i$ , for  $i \in N$ , at which services begin are decision variables.

In general it is possible to distinguish between *soft time window* constraints in which the customer's time window may be violated by paying an appropriate penalty (see, e.g., Ando and Taniguchi [15]) and *hard time window* constraints where no violation of the time windows is allowed (see e.g., Hu et al. [16]). In the case of hard time windows, if the vehicle arrives at customer  $i$  before  $b_i$  it will wait until the opening of its time window. It is also possible to impose a restriction on the amount of waiting time incurred by any vehicle.

In [17], a parameter  $W_{MAX}$  which restricts the waiting time at any customer is defined. In the CIT sector in the majority of the cases it is necessary to cope with time window constraints. In fact each customer can be visited for picking up money only within a predefined time interval, i.e. only within the normal working hours and only if the strongbox is open. In addition, some strongboxes used for instance by shopping malls, jewellery stores, etc. are equipped with a security mechanism, preventing any manual opening. Moreover, due to safety aspects, which play a crucial role in the CIT sector, it is necessary to avoid any kind of waiting time.

### 3 Problem description

In this section the RCTVRPTW is formally defined and a MIP formulation is developed. The goal of the problem is to determine  $K$  routes, a route being a tour that begins at the depot, traverses a subset of the customers in a specified sequence and returns to the depot. Each customer must be assigned to exactly one of the  $K$  routes, and must be visited within its hard time window. No waiting times are permitted. The total risk for each route must not exceed the risk threshold  $T$ . The routes should be chosen to minimize the total travel cost.

Assuming that a vehicle only picks up cash and never drops it off along its route, a risk index  $R_j^r$  can be defined for each customer  $j$  visited along the route  $r$  as follows (see also Talarico et al. [4]):

$$R_j^r = R_i^r + P_i^r \cdot d_{ij} \quad (2)$$

where  $P_i^r$  represents the amount of money on board of the vehicle when it arrives at customer  $i$  along the route  $r$  and  $d_{ij}$  is the distance between two consecutive customers  $i$  and  $j$ .  $P_i^r$  is defined for each customer in route  $r$  and it is obtained thus summing the amount of cash picked up by the vehicle at each customer visited along route  $r$  from the depot to the customer  $i$ . It is worth observing that in our formulation we used  $d_{ij}$  expressed as the distance between two consecutive customers  $i$  and  $j$ , but it could represent any other measure, not necessary related to the length of the roadway segment which connect  $i$  and  $j$ . For instances  $d_{ij}$  can be used to measure the probability of an accident on a specific roadway segment given its characteristics such as lane with,

number of lanes etc. (see Sattayaprasert and Pradhananga [18] and Milovanović et al. [19] for further details).

The risk index, defined in (2), is a cumulative and increasing measure of the risk incurred by the vehicle while it travels along its route. Because of this fact, the *global route risk* of a route  $r$ , denoted by  $GR^r$ , is the risk incurred by the vehicle upon its return to the depot. In order to generate a solution  $x$  composed by safe routes the following constraint is imposed:

$$GR^r \leq T \quad \forall r \in x \quad (3)$$

Unlike in the traditional VRP, it is the risk constraint and not the capacity of the vehicle that prohibits visiting all the customers in a single tour. Furthermore, the global route risk of a route  $r$  (e.g.,  $(A, B, C, D)$ ) is in general not the same as the global route risk of the reversed route  $\bar{r}$  (e.g.,  $(D, C, B, A)$ ).

In the remainder of the article we suppose that it is possible to delay the departure from the depot (instead of having a departure time equal to 0) in order to arrive at a customer when its time window is open. Given a set of feasible routes, the maximum delay ( $D_{MAX}^r$ ) of the departure time from the depot, that does not violate the time windows of customer  $i$ , visited at time  $b_i$  along route  $r$ , is given by the following formula:

$$D_{MAX}^r = \max\{0, \min_{\forall i \in r} \{l_i - b_i\}\} \quad \forall r \quad (4)$$

The value  $l_i - b_i$  represents the freedom to delay the beginning of the service at customer  $i$  before the closing of its time window.

If the vehicle arrives at customer  $j$ , along the arc  $(i, j)$ , before  $e_j$ , the minimum required departure delay from the depot (called  $D_j^r$ ) to make the visit of  $j$  feasible, is computed as follows:

$$D_j^r = \max\{0, e_j - (b_i + s_i + d_{ij})\} \quad \forall i, j, r \quad (5)$$

The visit of customer  $j$  is feasible, and thus the departure delay from the depot applicable, only if  $D_j^r \leq D_{MAX}^r$ . The value  $D_{MAX}^r$  should be computed considering the nodes already visited from the depot until node  $j$  along route  $r$ .

In Figure 1 we show an example of the computation of the departure delay from the depot to visit the customers within their time windows. The numbers on the arcs represent the travel times between customers, the numbers on the nodes represent the time window  $[e_i, l_i]$  and the service duration  $s_i$ , respectively. In particular the visit of the node  $B$  is feasible only if a departure delay of 1 unit of time is applied. In fact  $D_j^r = \max\{0, 11 - (5 + 1 + 4)\} = 1$  and  $D_j^r \leq D_{MAX}^r = 2$ .

After the insertion of the node  $B$  in the route, the value of  $D_{MAX}^r$  is updated considering the departure delay applied to the depot. In fact, if the vehicle starts his route at time 1 the customer  $A$  will be visited at time 6 and the customer  $B$  will be visited at time 11 and thus  $D_{MAX}^r = 1$ . This means that the departure from the depot should be delayed for an additional 1 unit of time so as to avoid the violation of the time windows for the nodes in the route  $r$ . In the next section the mathematical formulation of the problem is presented.

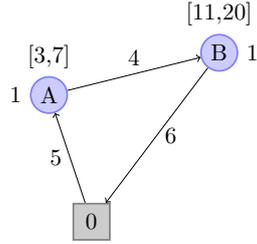


Figure 1: Example for the computation of the departure delay

### 3.1 Mathematical formulation

For the sake of simplicity the depot has been divided in two dummy nodes called  $O$  (origin) from which the vehicle's routes depart and  $D$  (destination) at which the vehicle's routes end. Formally, the RCTVRPTW is defined on a directed and complete graph  $G = (V, A)$  with a set of vertices  $V = \{O, D\} \cup N$ , where  $N$  corresponds to the set of customers  $N = \{1, \dots, n\}$ . The set of direct arcs is given by  $A = \{N \times N\} \cup \{O \times N\} \cup \{N \times D\}$  where  $N \times N$  represents the distance between customers,  $O \times N$  represents the distance between the depot and the customers and  $N \times D$  is the distance between the customers and the depot.

Each customer  $i \in N$  has a non-negative demand  $p_i$  which represents the cash to be picked up by the vehicle during its visit. A non-negative distance (or travel time)  $d_{ij}$  is associated with each arc  $(i, j) \in A$ .

In addition each customer  $i \in N$  must be visited within a specified time interval characterized by the earliest time  $e_i$ , and the latest time  $l_i$ . The variable  $b_i$  represents the start of the service at a customer  $i$  such that  $e_i \leq b_i \leq l_i$ . The duration of the service at node  $i$ , expressed in units of time, is represented by  $s_i$ .

All vehicles start from the depot empty ( $p_O = 0$ ) and perform a single route, visiting a sequence of customers before returning to the depot, collecting the cash (demand) in each node. At the start of their tour, each vehicle's *risk index* is equal to zero. A vehicle traveling between nodes  $i$  and  $j$  increases its risk index by a value equal to the product

of the amount of cash it carries when it left node  $i$  and the distance (or travel time) between  $i$  and  $j$ .

In the formulation, three sets of labels are used for each customer: (1)  $P_i^r$  is equal to the cash carried by the vehicle when it arrives at customer  $i$  along route  $r$ ; (2)  $R_i^r$  is the value of the risk index when the vehicle arrives in node  $i$  along route  $r$ ; (3)  $b_i^r$  represent the arrival time at node  $i$  along route  $r$ .

Note that, for a given customer  $i$ , all but one of its labels will be zero, since each customer is only visited once and is therefore associated to a single route  $r$ .

The boolean decision variable  $x_{ij}^r$  is equal to 1 if arc  $(i, j) \in A$  is used by the vehicle in route  $r$  and 0 otherwise. The number of routes is determined as part of the optimization problem, and is at most equal to  $n$ . A MIP formulation of the RCTVRPTW is shown below.

$$\min \sum_{r \in N} \sum_{(i,j) \in A} d_{ij} x_{ij}^r \quad (6a)$$

s.t.

$$\sum_{j \in N} x_{Oj}^r = \sum_{i \in N} x_{iD}^r \quad \forall r \in N \quad (6b)$$

$$\sum_{j \in N} x_{Oj}^1 = 1 \quad (6c)$$

$$\sum_{i \in N} x_{iD}^r \geq \sum_{j \in N} x_{Oj}^{r+1} \quad \forall r \in N \setminus \{n\} \quad (6d)$$

$$\sum_{r \in N} \sum_{(i,j) \in A} x_{ij}^r = 1 \quad (6e)$$

$$\sum_{h \in N} x_{hj}^r - \sum_{k \in N} x_{jk}^r = 0 \quad \forall j \in N; \forall r \in N \quad (6f)$$

$$P_O^r = 0 \quad \forall r \in N \quad (6g)$$

$$x_{ij}^r = 1 \Rightarrow P_i^r + p_j = P_j^r \quad \forall (i, j) \in A; \forall r \in N \quad (6h)$$

$$R_O^r = 0 \quad \forall r \in N \quad (6i)$$

$$x_{ij}^r = 1 \Rightarrow R_i^r + P_i^r d_{ij} = R_j^r \quad \forall (i, j) \in A; \forall r \in N \quad (6j)$$

$$0 \leq R_D^r \leq T \quad \forall r \in N \quad (6k)$$

$$x_{ij}^r = 1 \Rightarrow b_i^r + s_i + d_{ij} = b_j^r \quad \forall (i, j) \in A; \forall r \in N \quad (6l)$$

$$e_i \leq b_i^r \leq l_i \quad \forall i \in N; \forall r \in N \quad (6m)$$

$$P_i^r, R_i^r, b_i^r \geq 0 \quad \forall i \in V; \forall r \in N \quad (6n)$$

$$x_{ij}^r \in \{0, 1\} \quad \forall (i, j) \in A; \forall r \in N \quad (6o)$$

The objective function (6a) minimizes the total distance traveled by all vehicles combined. Constraint (6b) imposes that each route starts at depot  $O$  and ends at the depot  $D$ . Constraint (6c) states that the first route ( $r = 1$ ) starts at the depot  $O$ . Constraint (6d) enforces that route  $r + 1$  cannot exist unless route  $r$  also exists (for  $r > 1$ ), which forces routes to be numbered consecutively. Constraint (6e) ensures that each customer is visited exactly once. Constraint (6f) imposes that in route  $r$ , the vehicle can leave node  $j$  only if it has previously entered it.

Constraints (6g)–(6h) are used to define the cumulative demand from the depot to each node  $i$  along route  $r$ . Constraints (6i)–(6k) ensure that the global route risk is at most equal to the risk threshold  $T$ . Finally, constraints (6l)–(6m) are used to handle time windows of service at the customers. Subtours are eliminated by constraints (6g)–(6h) as well as by (6i)–(6k) and by (6l)–(6m). Constraints (6n)–(6o) define the domains of the decision variables.

## 4 Solution approaches

In order to solve the RCTVRPTW in a reasonable amount of time and to obtain near-optimal solutions, two metaheuristics have been developed. The first one, called m-grC-VNS (Multi-start Greedy Randomized Constructive heuristic plus Variable Neighborhood Search), can be classified as a GRASP metaheuristic (see Feo and Resende [20]). The second solution approach has been named p-grC-VNS (Perturb-and-improve Greedy Randomized Constructive heuristic plus Variable Neighborhood Search) and can be classified in the framework of the iterated local search metaheuristic (see Lourenço et al. [21]).

Both metaheuristics (m-grC-VNS and p-grC-VNS) create an initial solution using a constructive heuristic and improve those solutions using a variable neighborhood search strategy (VNS block). The metaheuristics differ in their diversification strategy, which is either a multi-start heuristic (in m-grC-VNS) or a perturb-and-improve heuristic (or perturbation used in p-grC-VNS). The multi-start and perturbation structures are abbreviated as m- and p-, respectively. In the following section the three macro components of the solution approaches are presented and more details are explained.

### 4.1 Greedy randomized constructive heuristic

A constructive heuristic (denoted using the abbreviation grC) combined with a greedy randomized mechanism for the selection of the unvisited nodes has been developed to generate a feasible initial solution for the RCTVRPTW. In the literature, greedy randomized constructive heuristics have been successfully used to solve a variety of related problems not only in the routing domain (see e.g., Kontoravdis and Bard [13]), but also for machine scheduling (see e.g., Feo et al. [22]) and set covering problems (see e.g., Feo and Resende [23]).

The constructive heuristic described in this paper works as follows:

- For the generation of a new route, a *total score* ( $S_{tot}$ ) is assigned to each unvisited node in set  $U$ . Then, the nodes in  $U$  are ordered following a decreasing order of the *total score*. A restricted candidate list (*RCL*) of the first  $\alpha$  nodes is created. A random node from the *RCL* is selected as first node of the route. Then the selected node is removed from set  $U$  and saved as current node.
- New unvisited nodes in  $U$  are selected and added to the route as long as the risk threshold is not exceeded. The selection of the next node to add works as follow:
  - (a) An eligible list  $L_e$  is created by inserting all nodes that can be reached, starting from the current node, within their time windows and without violating the risk constraint. Then, the nodes in  $L_e$  are ordered in ascending order of distance from the current node.
  - (b) A restricted candidate list ( $RCL_e$ ) is created that consists of the first  $\alpha$  nodes of  $L_e$ . If  $RCL_e$  is empty a new route is created otherwise a random node from  $RCL_e$  is selected, added to the current route, removed from set  $U$  and saved as the current node.

The *total score*  $S_{tot}$  assigned to each node is calculated as the sum of three partial scores: (1)  $S_p$ , which represent the partial score associated to each node looking at its demand. (2)  $S_e$ , which represent the partial score associated to each node looking at the earliest time of its time window. (3)  $S_d$ , which represent the partial score associated to each node looking at its distance from the depot.

Each partial score is given by the relative position of the node in the ranking list of the unvisited nodes according to three different criteria: decreasing distance from the depot (for  $S_d$ ); increasing demand value (for  $S_p$ ) and increasing earliest start (for  $S_e$ ).

The partial scores are combined together following formula in (7) where  $\gamma_p, \gamma_e, \gamma_d \in \mathbb{R}$  represent three coefficients used to weigh the partial scores.

$$S_{tot} = \gamma_p \cdot S_p + \gamma_e \cdot S_e + \gamma_d \cdot S_d \quad (7)$$

The coefficients can assume values  $0 \leq \gamma_p, \gamma_e, \gamma_d \leq 1$  with the following constraint:

$$\gamma_p + \gamma_e + \gamma_d = 1 \quad (8)$$

## 4.2 VNS block

Starting from an initial solution both the m-grC-VNS and the p-grC-VNS heuristics use a Variable Neighborhood Search strategy to improve the current solution and find local optima. The VNS is composed of six of the most common local search operators for vehicle routing problems [24], modified for the RCTVRPTW. The local search operators used in the paper can be divided in two categories:

- *Intra Route Local Search Operators* which attempt to improve a single route. In particular we used the *Internal Relocate Operator* and the *Internal Or-Opt Operator* shown in Figure 2.
- *Inter Route Local Search Operators* which change more than one route simultaneously. Our VNS block implements the *External Exchange Operator*, the *External Relocate Operator*, the *External Cross-Exchange Operator* and the *External Two-Opt Operator* shown in Figure 3.

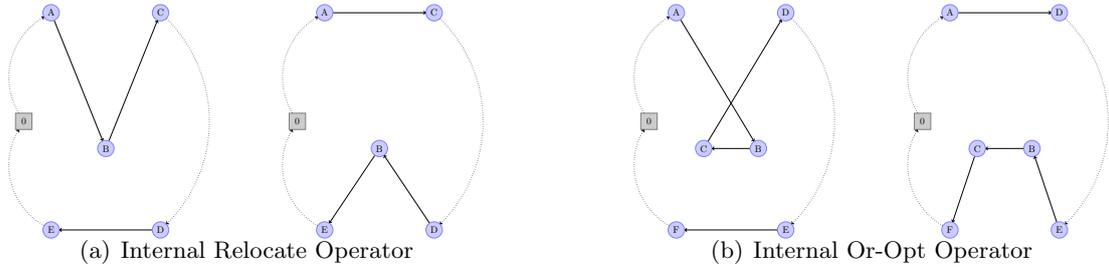


Figure 2: Intra Route Local Search Operators

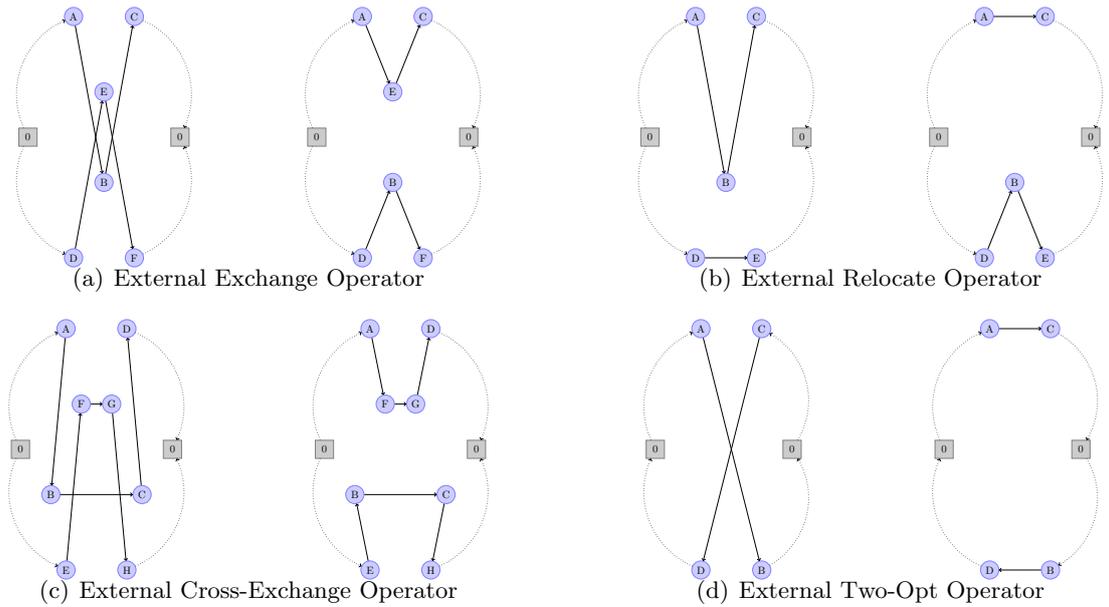


Figure 3: Inter Route Local Search Operators

The local search operators apply variants of the calculations of Talarico et al. [4] to efficiently determine whether a move results in an improved solution and what the new global risk indices are for the different routes affected by the move. These allow the feasibility checks for the risk constraint to be done in  $O(1)$ . Each local search operator uses a first improvement descent strategy.

As soon as a local search operator finds and applies a move to improve the current solution, the VNS block is restarted from the new current solution. The VNS block stops when the current solution cannot be further improved by any of the local search operators. The structure of the VNS block is described in Figure 4.

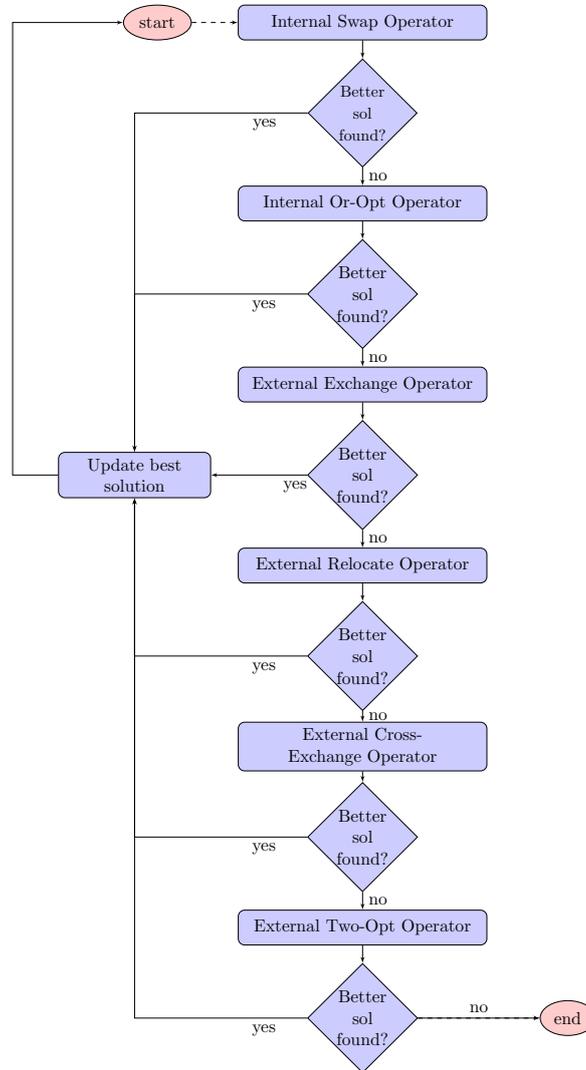


Figure 4: VNS block

### 4.3 Diversification structure

The m-grC-VNS and p-grC-VNS are different in their structure and the mechanism used to escape from local optima (diversification mechanism).

In m-grC-VNS a multi-start diversification strategy is used, that repeats both the constructive heuristic and the VNS block until a stopping condition (maximum number of iterations  $\rho$ ) is met.

In the p-grC-VNS metaheuristic a perturb-and-improve (or perturbation) heuristic structure is used. The perturb-and-improve structure uses the constructive heuristic only once and restarts the VNS block from a perturbed solution. During the perturbation heuristic a destroy-and-repair operator is used.

- *Destroy phase*: a random route from the solution  $x^*$  is selected. All the customers are removed from this route and inserted in a list of unvisited nodes ( $\mathcal{L}$ ). This step is repeated  $\xi \cdot m$  times where  $m$  represents the number of routes in  $x^*$  and  $\xi$  is the percentage number of routes to be destroyed from  $x^*$ .
- *Repair phase*: a new current solution  $x$  is constructed starting from the non destroyed routes of  $x^*$  and adding new routes which contain the nodes in  $\mathcal{L}$ . These new routes are generated applying the constructive heuristic described before replacing the set of unvisited nodes ( $U$ ) with  $\mathcal{L}$ .

After the application of the destroy-and-repair operator the new solution is saved as current solution and is improved using the VNS block. The p-grC-VNS metaheuristic terminates when a maximum number of iterations ( $\varphi$ ) is reached. A general overview of the multi-start and the perturbation heuristics is shown in Figure 5.

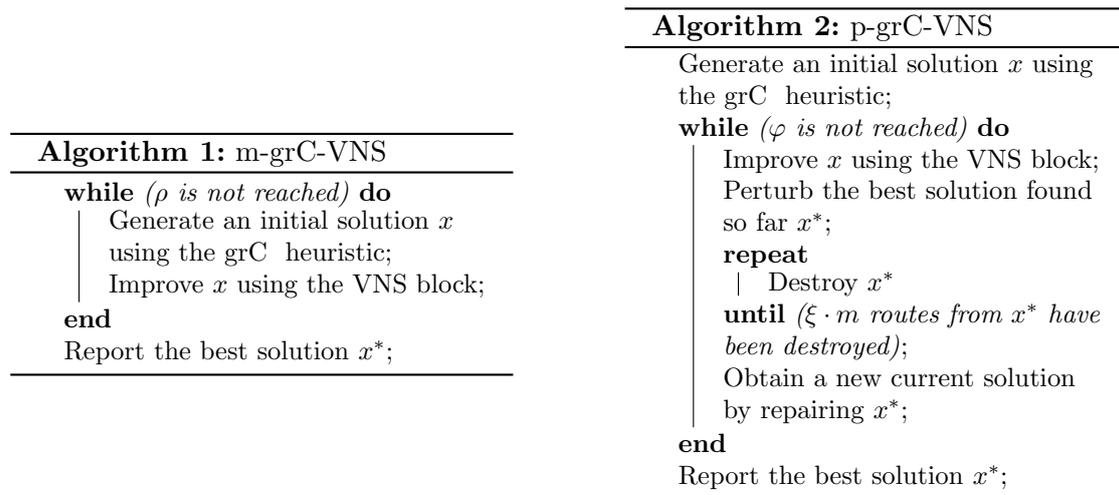


Figure 5: Multi-start versus Perturbation

## 5 Experiments

Since the RCTVRPTW has not been studied before, no test instances are available in the literature. For this reason we used 24 benchmark instances for the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). We adapted these instances to the RCTVRPTW by replacing the capacity constraint with the risk constraint (see section 5.1).

The metaheuristics have been coded in Java and an experimental analysis has been carried out in two sequential steps. (a) In a first stage, the internal parameters of each metaheuristic (see paragraph 5.2) are calibrated in such a way that each metaheuristic generates the best possible results when averaged over all possible parameter combinations. (b) In a second stage, once the optimal values of the parameters for each metaheuristic have been determined, the two calibrated metaheuristics are compared (see paragraph 5.3). This comparison is done by solving all the instances contained in the test set.

### 5.1 Test instances

A set of 24 instances taken from the available benchmark instances for the CVRPTW have been used in our computational experiments. In particular we used medium and large instances available in the literature from Cordeau et al. [25] and Homberger and Gehring [26]<sup>1</sup> ranging from 100 to 600 nodes. The characteristic of the instances are reported in Table 1.

Each CVRPTW instance contains the following values: the maximum number of nodes; the maximum number of vehicles; the capacity of each vehicle; the node coordinates; the time window and demand of each node. It is worth noticing that in the RCTVRPTW it is not the capacity of the vehicle, but the risk threshold that restricts the length of each route.

To convert an instance for the CVRPTW to an instance for the RCTVRPTW, we maintained unchanged the principal instance features (i.e. time windows, nodes coordinates, max number of nodes), with the exception of the demand vector and vehicle capacity. In fact the demand associated to each node in the RCTVRPTW contributes to determine the global route risk that need to be less than the risk threshold which, in turn, replaces the max capacity of the vehicle.

Looking at the best known solution ( $\bar{x}^*$ ) for a given CVRPTW instance, we associated an increasing value of demand to the nodes contained in each route  $r$  of  $\bar{x}^*$ . In other words we assigned a demand equal to 1 to the first node visited in each route contained in  $\bar{x}^*$ , the value 2 to the second node, 3 to the third node and so on. Using the new demand vector, we defined the global route risk values for all the routes contained in  $\bar{x}^*$ .

---

<sup>1</sup>The instances are available at <http://neo.lcc.uma.es/vrp>.

Table 1: CVRPTW instances

Author	Name	# Nodes	Best CVRPTW Solution	Known
<i>Cordeau</i>	c101	100	828.94	
	c201	100	591.56	
	r101	100	1650.8	
	r201	100	1253.26	
	rc101	100	1708.8	
	rc201	100	1406.94	
<i>Hombberger</i>	C1_2.1	200	2704.57	
	C2_2.1	200	1931.44	
	R1_2.1	200	4784.11	
	R2_2.1	200	4483.16	
	C1_2.2	200	2917.89	
	RC2_2.3	200	2601.87	
	C1_4.1	400	7152.06	
	C2_4.1	400	4116.14	
	C1_4.2	400	7695.55	
	C1_4.3	400	7092.6	
	R1_4.3	400	7898.63	
	RC1_4.3	400	7579.9	
	C1_6.1	600	14095.64	
	C2_6.1	600	7774.16	
	R1_6.2	600	19092.54	
	C1_6.3	600	13777.81	
	R1_6.3	600	17229.91	
	RC1_6.3	600	15425.99	

We set the risk threshold  $T$  equal to the maximum value of the global route risk as in formula (9). We repeated the same procedure for all the test instances.

$$T = \max GR^r \quad \forall r \in \bar{x}^* \quad (9)$$

In this way we have the guarantee that the best solutions for the CVRPTW instances remain feasible solutions for the RCTVRPTW.

Moreover, since the nodes contained in the routes of the best know CVRPTW solution present (after the changes) an increasing demand value, it will be likely improbable to reduce the cost of the solution by doing intra-route changes, without violating the risk threshold.

For this reason, although we do not have any guarantee on the optimality of the solutions for the RCTVRPTW, the best known CVRPTW solutions are very likely to be excellent solutions for the RCTVRPTW.

The new set of instances is available at the following web address: <http://antor.ua.ac.be/downloads>.

## 5.2 Parameter configuration

In order to calibrate the metaheuristics, a subset of the test instances (*c101*, *c1\_2\_1*, *c1\_4\_2* and *C1\_6\_3*) has been solved using m-grC-VNS and p-grC-VNS in a full factorial experiment, testing all the combinations of the heuristic parameters. A brief description of the heuristic parameters, as well as the tested values and the number of tested values is given in Table 2.

Table 2: Heuristic parameters

Parameter	Description	Metaheuristic	Values	#
$\rho$	Number of restart (multi-start heuristic)	m-grC-VNS	10	1
$\varphi$	Number of perturbation (perturb-and-improve heuristic)	m-grC-VNS	10	1
$\xi$	Maximum percentage number of routes in best solution found so far to be destroyed	p-grC-VNS	5,10,...,95,100%	20
$\alpha$	Size of the restricted candidate lists	p-grC-VNS	1,2,3,4,5,6,7,8	8
$\gamma_d$	Coefficient used to weight the score for the distance	m-grC-VNS p-grC-VNS	0,0.1,0.2,...,0.9,1	11
$\gamma_p$	Coefficient used to weight the score for the demand	m-grC-VNS p-grC-VNS	0,0.1,0.2,...,0.9,1	11
$\gamma_e$	Coefficient used to weight the score for the time windows	m-grC-VNS p-grC-VNS	0,0.1,0.2,...,0.9,1	11

Analysing the results obtained over all the possible parameters the optimal configuration of the heuristic parameters has been determined. In the following graphs we show the relations between the average gap from the best found solutions (expressed in percentage) and each heuristic parameter. In particular we report:

- the relationship between the parameter  $\xi$  and the average % gap from the best found solutions (see Figure 6);
- the relation between the parameter  $\alpha$  and the % gap from the best found solutions (see Figure 7).

In Table 3 the best settings for the heuristic parameters are reported. If we increase the number of iterations  $\rho$  or  $\varphi$ , the quality of the solutions (lower values of the objective functions) can only improve, but at the expense of increasing the computing time. In the second phase of the experiments, described in paragraph 5.3, m-grC-VNS and p-grC-VNS have been tested using different values for the parameters  $\rho$  and  $\varphi$  in order to analyse the relation between the quality of the solution and the computational time.

## 5.3 Computational results

The computational experiments were performed using a machine with an Intel core i7-2760QM 2.40GHz processor with 4GB RAM. Both the m-grC-VNS and p-grC-VNS are

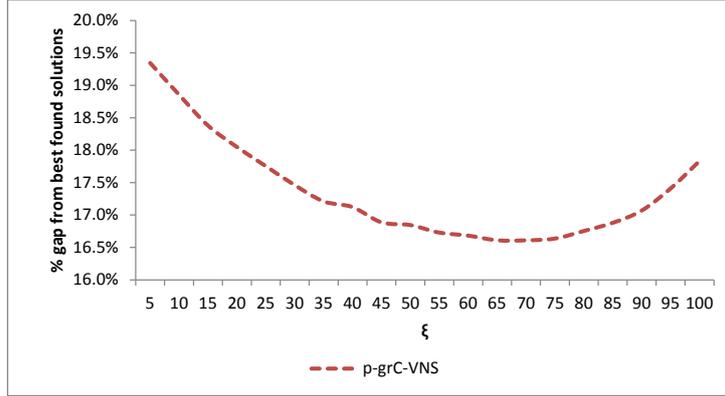


Figure 6: Relationship between  $\xi$  and the average % gap from the best found solutions

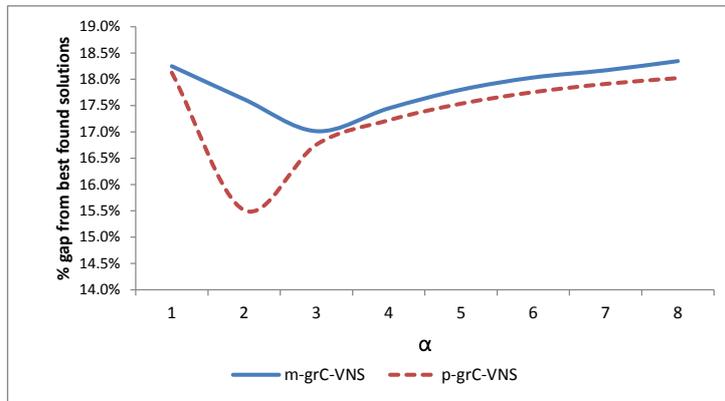


Figure 7: Relation between  $\alpha$  and average % gap from the best found solutions

Table 3: Optimal settings for the two metaheuristics

Parameter	m-grC-VNS	p-grC-VNS
$\xi$	-	75%
$\alpha$	3	2
$\gamma_d$	0,8	0.8
$\gamma_p$	0,1	0.1
$\gamma_e$	0,1	0.1

tested on all instances.

We performed 3 different *experimental tests* with  $\rho$  (number of restart used in the m-grC-VNS) and  $\varphi$  (number of perturbation used in p-grC-VNS) both equal to 10, 100 and 1000. Larger values of  $\rho$  and  $\varphi$  mainly increase the robustness of the algorithms by decreasing the average cost of the solutions obtained at the expenses of an higher computational time. For each instance 30 runs have been executed. For each *experimental test* we report: (1) the percentage gap between the best solutions obtained after 30 runs and

the best known solutions of the CVRPTW instances (column *% BestGap* in Table 4), which can be related to the capacity of the metaheuristics to find good solutions; (2) the percentage gap between the average cost of the solutions obtained after 30 runs and the best known solutions of the CVRPTW instances (column *% AvgGap* in Table 4), which can be related to the robustness of the metaheuristics, and (3) the average computing time in seconds needed to solve the 24 test instances (column *AvgTime* in Table 4).

For sake of readability we reported the aggregated results obtained by each metaheuristic.

Table 4: Computational Results

$\rho - \varrho$	Metaheuristic	% BestGap	% AvgGap	AvgTime
10	m-grC-VNS	11.8%	19.4%	37.3
	p-grC-VNS	1.3 %	5.1%	24.9
100	m-grC-VNS	8.5%	13.6%	351.7
	p-grC-VNS	-0.8%	2.0%	271.5
1000	m-grC-VNS	6.0%	10.0%	3405.2
	p-grC-VNS	-2.1%	-0.1%	2492.5

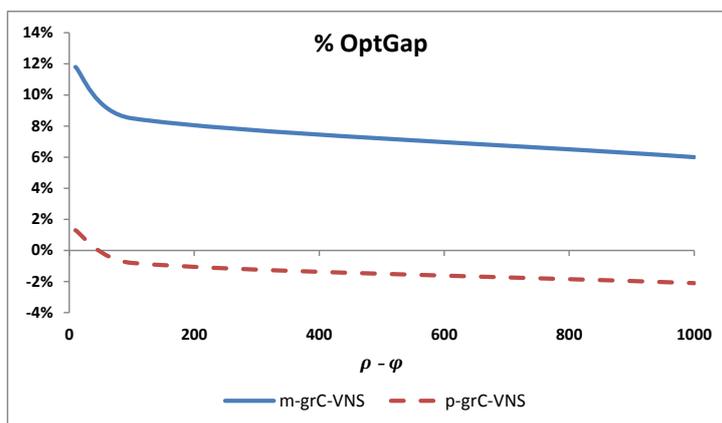


Figure 8: Relationship between  $\rho - \varphi$  and the percentage gap between the best solutions obtained after 30 runs and the best known solutions of the CVRPTW instances

In the Figures 8 and 9, the m-grC-VNS and p-grC-VNS metaheuristics are compared. The p-grC-VNS performs better than m-grC-VNS from all points of view. In fact p-grC-VNS was able to obtain better solutions than m-grC-VNS with better robustness. Considering the computational time, p-grC-VNS was on average 28% faster than m-grC-VNS. This result is most likely due to the perturbation heuristic which retains part of the routes contained in the best solution found so far without generating a new initial solution from scratch.

As expected, if the values of  $\rho$  and  $\varphi$  increase, both metaheuristics are able to find better solutions with a better robustness. However, as the values of  $\rho$  and  $\varphi$  grow by a factor of

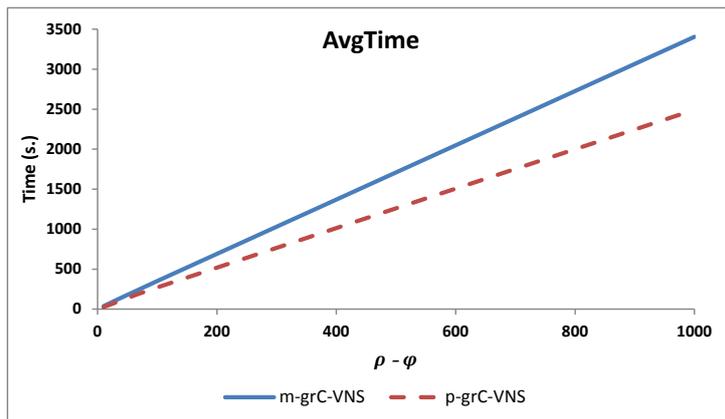


Figure 9: Relationship between  $\rho - \varphi$  and the computational time

10, the computational time increases by approximately by factor of 9. Observing the graph in Figure 8 it seems that the capacity to find better solutions reaches a certain maximum for values of  $\rho$  (or  $\varrho$ ) greater than 1000.

In other words, setting the values of  $\rho$  and  $\varphi$  greater than 1000, the computational time will increase without significant enhancements in the quality of the solutions.

## 6 Conclusions

In this paper we have presented a new variant of the VRP problem useful in the cash-in-transit industry. Starting from the RCTVRP defined in [4] we introduced a hard time window constraint where no waiting time at the customer's location is allowed. We developed two metaheuristics to solve the RCTVRPTW and we named them m-grC-VNS and p-grC-VNS respectively. Both metaheuristics are composed of (1) a constructive heuristic with a greedy random procedure to select the unvisited nodes and (2) a VNS block to improve the current solution. The metaheuristics use two different heuristic structures to escape from local optima: a multi-start structure for the m-grC-VNS and a perturb-and-improve structure for the p-grC-VNS.

We used some benchmark instances for the capacitated vehicle routing problem with time windows. Due to the risk constraint, which is present in the RCTVRPTW, we adapted the instances by varying the demand vector and defining a risk threshold that maintains the feasibility of the best known optimal solutions for the CVRPTW instances. In so doing we compared the solutions obtained using the m-grC-VNS and p-grC-VNS with effective solutions in term of minimization of the travel distance. We first performed a full factorial experiment to find the optimal configuration of the parameters used by each metaheuristic. Then we used both m-grC-VNS and p-grC-VNS in their optimal configuration to solve 24 test instances.

In general both metaheuristics obtained encouraging results. Since the risk constraint replaced the capacity constraint, for some test instances, we obtained solutions with a lower cost than the best optimal solutions for the CVRPTW. In particular, using the p-grC-VNS metaheuristic we found solutions with a percentage gap from the best known solutions for the CVRPTW instances equal to -2.1% in reasonable computational time (on average 1.7 minutes to solve each instance in the test set which contains 24 medium and large VRP instances between 100 to 600 nodes). It is also worth observing that the p-grC-VNS outperformed the m-grC-VNS for all the measured indices (computational time, robustness and quality of the solutions).

Future research can be aimed at handling the risk as an additional objective to be minimized and not only as a constraint. An interesting work concerns the development of an algorithm to generate alternative and safe routes. Moreover, the problem can be extended in several ways, taking into consideration route length restrictions, precedence relations and delivery of money to customers.

## References

- [1] R. Wolfer Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computer and Operation Research*, 30:1269–1287, 2003.
- [2] J. Krarup. The peripatetic salesman and some related unsolved problems. In B. Roy (Ed.), editor, *Combinatorial programming: methods and applications*, page 173–178, Reidel: Dordrecht, The Netherlands, 1975.
- [3] S.U. Ngueveu, C. Prins, and R. Wolfer Calvo. A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics*, 10:253–266, 2010.
- [4] L. Talarico, K. Sörensen, and J. Springael. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *University of Antwerp, Faculty of Applied Economics Working Papers*, (5), 2013.
- [5] CCPS Center for Chemical Process Safety. *Guidelines for chemical transportation safety, security, and risk management*. American Institute of Chemical Engineers, Hoboken, New Jersey, 2008.
- [6] F. Russo and C. Rindone. Planning in road evacuation: classification of exogenous activities. In Pratelli and Brebbia C.A., editors, *Urban Transport XVII Urban Transport and the Environment in the 21st Century*, volume 116 of *WIT Transactions on the Built Environment*, pages 639–651, Southampton, 2011. WIT Press.
- [7] K. D. Pijawka, S. Foote, and A. Soesila. Risk assessment of transporting hazardous materials: Route analysis and hazard management. In Transportation Research Board, editor, *Transportation Research Record 1020*, National Research Council, Washington, D.C., 1985.

- [8] R. D. Scanlon and E. J. Cantilli. Assessing the risk and safety in the transportation of hazardous materials transportation. In Transportation Research Board, editor, *Research Record 1020*, National Research Council, Washington, D.C., 1985.
- [9] M. Abkowitz and P. Cheng. Developing a risk-cost framework for routing truck movements of hazardous materials. *Accident Analysis and Prevention*, pages 39–51, 1988.
- [10] G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.
- [11] C.D. Tarantilis and C.T. Kiranoudis. Using the vehicle routing problem for the transportation of hazardous materials. *Operational Research. An International Journal*, 1(1):67–78, 2001.
- [12] K.C. Zografos and K.N. Androutsopoulos. A heuristic algorithm for solving hazardous materials distribution problems. *European Journal of Operational Research*, 152:507–519, 2004.
- [13] G. Kontoravdis and J.F. Bard. A grasp for the vehicle routing problem with time windows. *ORSA Journal on Computing Winter*, 7:10–23, 1995.
- [14] M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.
- [15] N. Ando and E. Taniguchi. Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, 6:293–311, 2006.
- [16] X. Hu, M. Huang, and A.Z. Zeng. An intelligent solution system for a vehicle routing in urban distribution. *International Journal of Innovative Computing, Information and Control*, 3:189–198, 2007.
- [17] N. Balakrishnan. Simple heuristics for the vehicle routing problem with soft time windows. *Journal of the Operational Research Society*, 44:279–287, 1993.
- [18] P. Hanaoka S. Sattayaprasert, W. Taneerananon and R. Pradhananga. Creating a risk-based network for hazmat logistics by route prioritization with ahp: - case study: Gasoline logistics in rayong, - thailand. *International Association of Traffic and Safety Sciences*, 32(1):74–87, 2008.
- [19] B. Milovanović, V. D. Jovanović, P. Živanović, and S. Žeželj. Methodology for establishing the routes for transportation of dangerous goods on the basis of the risk level - case study: City of belgrade. *Scientific Research and Essays*, 7(1):38–50, 2012.
- [20] T.A. Feo and M. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, (6):109–133, 1995.

- [21] H.R. Lourenço, O. Martin, and T. Stützle. *Iterated Local Search: Framework and Applications*, volume Handbook of Metaheuristics 2nd. Edition of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, 2010.
- [22] T.A. Feo, J.F. Bard, and K. Venkatraman. A *GRASP* for a difficult single machine scheduling problem. *Computer and Operations Research*, 18(8):635–643, 1991.
- [23] T.A. Feo and M. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operational research letters*, (8):67–71, 1989.
- [24] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, (39): 104–118, 2005.
- [25] J.F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. *Vrp with time windows*. SIAM monograph on discrete mathematics and applications, Philadelphia, 2002.
- [26] J. Homberger and H. Gehring. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *Information Systems and Operational Research, special issue: Metaheuristics for location and routing problems*, pages 297–318, 1999.