Discrete Optimization

Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem

CrossMark

Luca Talarico*, Kenneth Sörensen, Johan Springael

University of Antwerp Operations Research Group ANT/OR, Prinsstraat 13, 2000 Antwerp, Belgium

ARTICLE INFO

ABSTRACT

This paper proposes a variant of the well-known capacitated vehicle routing problem that models the routing of vehicles in the cash-in-transit industry by introducing a risk constraint. In the *Risk-constrained Cash-in-Transit Vehicle Routing Problem* (RCTVRP), the risk of being robbed, which is assumed to be proportional both to the amount of cash being carried and the time or the distance covered by the vehicle carrying the cash, is limited by a risk threshold.

A library containing two sets of instances for the RCTVRP, some with known optimal solution, is generated. A mathematical formulation is developed and small instances of the problem are solved by using IBM CPLEX.

Four constructive heuristics as well as a local search block composed of six local search operators are developed and combined using two different metaheuristic structures: a multistart heuristic and a perturb-and-improve structure. In a statistical experiment, the best parameter settings for each component are determined, and the resulting heuristic configurations are compared in their best possible setting. The resulting metaheuristics are able to obtain solutions of excellent quality in very limited computing times.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Over the last decades, vehicle routing problems have drawn the interest of a large number of researchers, and several variants have been developed to model different real-life situations. 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 between banks, large retailers, shopping centres, ATMs, jewellers, casinos, and other locations where large amounts of cash or valuables are present.

The cash-in-transit (CIT) industry groups transportation companies that deal with the physical transfer of banknotes, coins and items of value. In general the transfer of cash and valuables happens between customers (typically retail and/or financial organisations) and one or more cash deposits or banks. It is clear that, as a consequence of the nature of the transported goods, resisting crime is a significant challenge, and CIT companies are constantly exposed to risks such as robberies.

In order to estimate the importance of the CIT sector we note that in the United Kingdom alone, more than £500 billion are transported each year (£1.4 billion per day). Some robberies manage to attract a large share of media attention. In February 2013 in Foggia (Italy)

300 kilogram of gold was robbed. In the same month and year in Brussels (Belgium) a robbery of €50 million happened, and in March 2013 in Varese (Italy) a criminal organization carried out a robbery of €10 million. Money stolen in CIT attacks represents a major source of funding for organized crime. The latest statistics from the British Security Industry Association show that attacks against CIT couriers remain a serious and growing problem throughout the world (British Security Industry Association, 2013).

During the last decades, in an effort to reduce the incidence of robberies, CIT firms have heavily invested in better vehicles, equipment, infrastructure, and technologies (e.g., armoured vehicles, weapons on board, on-board drop safes and interlocking doors, active vehicle tracking). However, no level of security measures can completely prevent robberies from happening (Erasmus, 2012).

According to Smith and Louis (2010); Yan, Wang, and Wu (2012), one of the main reasons robberies are so prevalent is a lack of analysis of security issues in the route planning phase. The authors warn that a careful planning of the cash-in-transit activities is generally advisable to reduce the risk of being robbed. For this reason, a recent study of Yan et al. (2012) proposes a model to formulate more flexible routing and scheduling practices that incorporates a new concept of similarity for routing and scheduling solutions considering both time and space measures to reduce the risk of robbery.

Another 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 (Krarup, 1975;

---

* Corresponding author. Tel.:+32 3 265 40 17.
  E-mail address: luca.talarico@uantwerpen.be, talaricos@hotmail.it (L. Talarico).

Ngueveu, Prins, & Wolfler Calvo, 2010a, 2010b), customers are visited several times within a planning horizon, but the use of the same arc twice is explicitly forbidden. In Wolfer Calvo and Cordone (2003) the "unpredictability" is ensured by introducing time windows with a minimum and maximum time lag between two consecutive visits of the same customer. In this way it is possible to generate a wide variety of solutions, as required for security reasons. A similar concept is followed by Michallet, Prins, Amodeo, Yalaoui, and Vitry (2014); Michallet, Prins, Amodeo, Yalui, and Vitry (2011) where regularity (in terms of time at which the visit of the customer happens) is avoided, by managing time windows in which each customer can be visited.

Except for some practical situations that explicitly require customer visits at regular intervals, the moments in time at which customers are visited are generally variable in the CIT sector depending on the amount of money that needs to be deposited or picked up, which is seldom regular. Therefore, CIT firms define their routing plan on a daily basis, depending on the customers that need to be visited. Routing plans in the CIT sector should both be safe and efficient, while taking into account two critical issues: minimization of the travelled cost/time, as well as limiting the exposure of the transported goods to robbery. The Risk-constrained Cash-in-Transit Vehicle Routing Problem (RCTVRP), that is developed in this paper, attempts to achieve this.

Different from the existing approaches known in the literature, we limit the total risk that any vehicle may incur during its operations to a pre-specified *risk threshold*. To this end, a risk index is defined to measure the exposure of the vehicle, while it is outside of the depot. To the best of our knowledge, this particular way to handle security in a vehicle routing context is new. Moreover, it is complementary to methods that achieve security through unpredictability, and can potentially be combined with them.

Informally, the RCTVRP is defined as follows. Given a depot, as well as a set of customers each with a given "demand", corresponding to a sum of cash that needs to be picked up, the objective of the problem is to define a set of routes, one for each vehicle. Each vehicle leaves from a depot, visits a set of customers picking up cash and returns to the same depot at the end of its route. On each arc it travels, a vehicle incurs a certain amount of risk that is proportional to the both time or distance travelled and the amount of cash carried on that arc. The total risk incurred by a vehicle is the sum of the individual risks incurred on each arc. For each vehicle route, this total risk should be at most equal to a predefined risk threshold that can be quantified by the CIT firm depending on a series of factors such as the amount of money to be transported, the characteristic of the network, and the company's attitude to risk. In a preliminary analytical stage several scenarios presenting different risk thresholds (see e.g. Section 5) could be evaluated and the most suitable risk threshold can be adopted by the CIT company to generate routes that are both efficient and safe.

The main focus of this work is the development of a decision model, together with the description of the solution approaches. The major contributions of this paper are fourfold.

(1) A new variant of the vehicle routing problem, the Risk-constrained Cash-in-Transit Vehicle Routing Problem (RCTVRP) is introduced. The main distinguishing feature of this problem is the risk constraint that is used to limit the risk each vehicle runs.
(2) A library containing two sets of problem instances for the RCTVRP, some with known optimal solution, is generated.
(3) A mathematical formulation of the RCTVRP is developed and optimal solutions for small problem instances are found by using the IBM CPLEX solver.
(4) Efficient metaheuristic approaches to solve small, medium and large instances of the RCTVRP are presented, tuned using a statistical experiment, and then compared.

The remainder of the paper is organized as follows. In Section 2 the literature on vehicle routing in risk-prone situations is surveyed, and the concept of *risk constraint* is introduced. Section 3 outlines a mathematical formulation for the RCTVRP, while the different components of the metaheuristic approaches are described in Section 4. In Section 5, the algorithms are tested and computational results are reported. Section 6 presents some conclusions and suggestions for future research. A description of the sets containing the test instances, including the procedure that was used to generate them, can be found in Appendix A.

2. Risk constraint

Besides the "peripatetic" vehicle routing problems discussed earlier, the concept of risk has only received limited attention in the context of vehicle routing problems for the CIT sector, whereas it has been thoroughly analysed in the literature on transportation of chemical and hazardous materials (*hazmat*).

In the hazmat transportation literature many models have been developed in which a risk function is defined for each road section and safe routes are selected looking at the minimization of the operating costs. The risk function is in general based on the substance being transported, but also on the road characteristics (e.g., tunnels, road condition, light, traffic). See for example Androutsopoulos and Zografos (2012); Bianco, Caramia, Giordani, and Piccialli (2013); Reniers et al. (2010); Van Raemdonck, Macharis, and Mairesse (2013).

As defined by the Center for Chemical Process Safety (2008), risk can be seen as an index of potential economic loss, human injury, or environmental damage, that is measured in terms of both the incident probability and the magnitude of the loss, injury, or damage. The *risk* associated with a specific (unwanted) event can be expressed as the product of two factors: the likelihood that the event will occur (p_{event}) and its consequences (C_{event}). A risk therefore is an index of the "expected consequence" of the unwanted event.

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

In case of hazmat, an undesirable event is an accident that results in the release of hazardous substances with severe consequences on the population in the neighbourhood of the incident. The consequences of the event depend on several factors such as the substance carried, the size of the population living near the accident, etc. The probability of an accident occurring depends on the type of substance transported and on the road characteristics such as lane width, number of lanes, etc. (see Milovanović, Jovanović, Živanović, & Žeželj, 2012 for further details). Another study also considers the influences of weather conditions on the accident probability (Akgün, Parekh, Batta, & Rump, 2007).

The definition of risk in Eq. (1) has many desirable features (additivity, linearity) that facilitate the solution process. As mentioned in Díaz-Ovalle, Vázquez-Román, De Lira-Flores, and Mannan (2013), the risk estimation always refers to specific scenarios and the attitude to risk of various decision-makers may differ. However, a comparison between the risk measures, that have been modelled in the field of hazmat transportation, as well as the possible attitudes towards risk, is beyond the scope of this paper. For a more elaborate discussion on risk measures, the reader is referred to Erkut and Ingolfsson (2005). In the remainder of our work we suppose that the decision maker has a risk-neutral attitude.

In the CIT sector, differently from the hazmat transportation, the goods being transported are not dangerous, but a robbery might generate two different types of unwanted consequences. The first type consists of the *foreseeable consequences* that are mainly linked to the loss of the cash/valuables being transported. The second type includes the *unforeseeable consequences* that are related to the criminal activity itself. An armed assault, e.g., might result in CIT personnel or third persons being seriously harmed. Therefore unforeseeable consequences
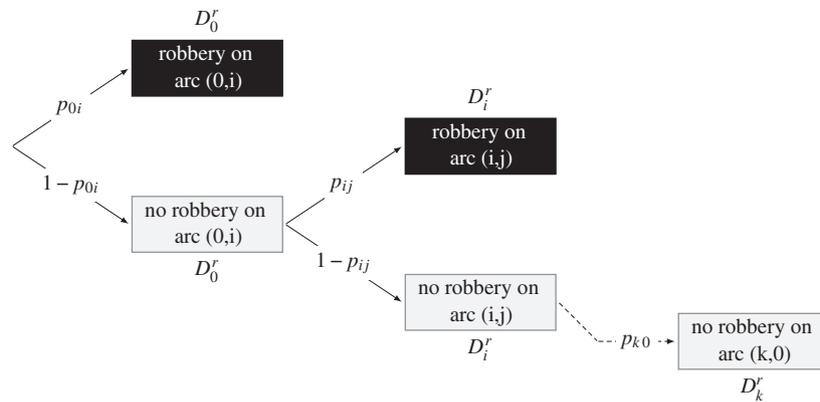
Fig. 1. Partial probability tree showing the risk along route *r*.

involve large additional costs that, in some cases, might be significant, e.g., costs related to damaged vehicles and/or equipment, costs of policing, ambulance and hospital treatments, the opportunity cost as a result of inactive CIT personnel. Sometimes, even the road infrastructure can be damaged during a robbery resulting in additional costs for the users of the road network. Both foreseeable and unforeseeable consequences, can be mitigated by taking specific insurances against these events, but clearly there is no pay-off against human tragedy and trauma.

CIT firms can impact the *foreseeable consequences* by selecting appropriate cash collection routes aimed at minimizing the values of the transported goods and thus mitigating the consequences of a robbery. The *unforeseeable consequences* on the other hand are unrelated to the amount of cash/valuables being transported. Moreover, they are not under the direct control of CIT companies and since they cannot be quantified a priori, we only focus on the *foreseeable consequences* of a robbery. Therefore, in the remainder of the paper, the perspective of a CIT company is analysed, addressing only the financial loss of the transported goods.

2.1. Measuring risk in the RCTVRP

In Russo and Rindone (2011) 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* as a measure of the susceptibility of the objects to be protected from unwanted events; (c) the *exposure* that represents a weighted value of people, goods and infrastructure affected during and after the event.

Following this structure, the risk of being robbed along an edge (i, j) in route *r* can be computed as follows:

(1) the occurrence of a robbery p_{ij} expressed as the probability of a robbery happening. It can be influenced by several factors such as the road characteristics (number of lanes, type of road segment, traffic condition), the weather conditions, the time at which the robbery happens (e.g., day-time or night-time);

(2) the vulnerability v_{ij} defined as a measure of the probability that the robbery succeeds given that it occurs. It might depend on several factors such as the modus operandi of the criminals, the type of vehicle (e.g., heavy or light armoured), the weapons on board, the crew preparedness, the police promptness and so on;

(3) the consequence of the robbery D_i which quantifies the loss of cash/valuables transported by the vehicle until node *i* in case a robbery happens on edge (i, j) along route *r*.

Therefore supposing that a robbery might happen only on an edge (i, j), the risk faced by the vehicle before arriving at node *j* is given by

the following formula:

$$R_j^r = p_{ij} \cdot v_{ij} \cdot D_i^r \tag{2}$$

We can express the risk for an entire vehicle route starting from Eq. (2). Since a route *r* is a sequence of arcs, we can view the travel of the vehicle along *r* as a probabilistic experiment, as shown in Fig. 1. Erkut and Ingolfsson (2005) demonstrate the additivity of the risk index by using a similar approach. In particular the risk associated with *r* can be expressed as follows:

$$p_{0i} \cdot v_{0i} \cdot D_0^r + (1 - p_{0i}) \cdot p_{ij} \cdot v_{ij} \cdot D_i^r + \cdots$$
$$+ (1 - p_{0i}) \cdot (1 - p_{ij}) \cdot (\ldots) \cdot p_{k0} \cdot v_{k0} \cdot D_k^r \tag{3}$$

Supposing that the probability of being robbed more than once along route *r* can be neglected, Eq. (3) can be simplified and rewritten as the sum of the expected consequence of the robbery on each arc (i, j) contained in *r*:

$$\sum_{(i,j) \in r} p_{ij} \cdot v_{ij} \cdot D_i^r \tag{4}$$

According to Smith and Louis (2010), most CIT offenders are professional armed robbers. Compared with amateur offenders, professional criminals are prepared to use weapons. Moreover, they present a higher level of motivation with a greater tendency to gather critical information about the targeted companies, plan the offence and adopt effective attack strategies (Pillay, 2008). As a result they have a greater success rate compared to other categories of robbers.

Two factors may influence the likelihood of an attack: (a) the amount of transported cash and (b) the characteristics of the arcs. Supposing that the robbers do not have an accomplice inside the CIT company, it is safe to assume that robbers do not know the value of the goods transported on any given edge. On long arcs, we can assume that robbers have more opportunities to attack, due to the larger amount of time spent by the CIT vehicle on these arcs. Moreover, long arcs present a larger number of attack points, escape hatches or points where criminals might hide their vehicles/equipment.

Due to the lack of real data and for the sake of simplicity two basic assumptions are made:

(1) the probability of an attack (p_{ij}) is proportional to the length (c_{ij}) of the edge traversed by the vehicle; and

(2) the vulnerability factor (v_{ij}) is assumed to be constant for all edges and equal to the risk rate computed by using the past records of successful attacks (from the criminal perspective) over the total attempted attacks (Veroni & Bourguet, 2008). Therefore, in the remainder of the paper the vulnerability factor can be omitted.

We are aware that these assumptions are not always realistic. However, a different measure than the edge length can trivially be
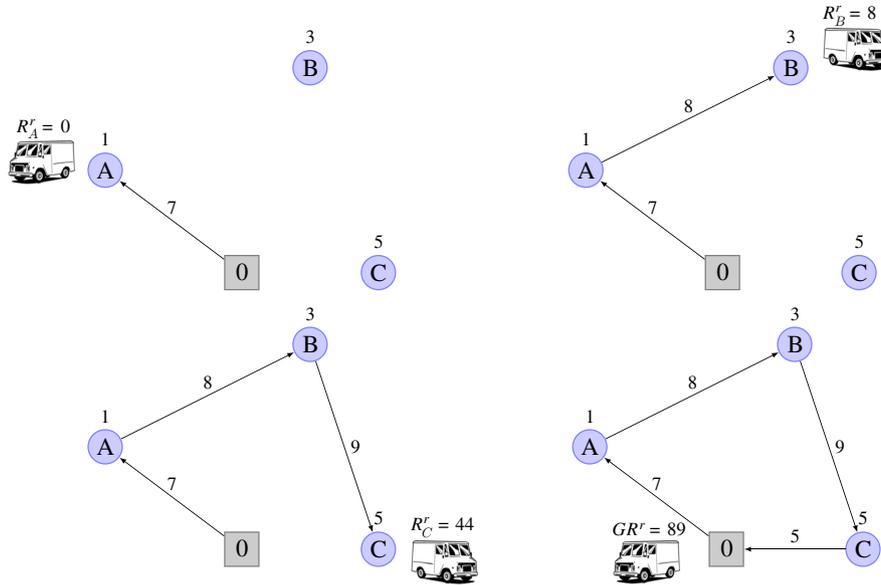
Fig. 2. Illustrative example.

employed in the model instead of p_{ij}. This paper uses the edge length c_{ij} as a proxy of p_{ij} for reasons of simplicity: the edge length is a given parameter of any vehicle routing problem, and, moreover, no (realistic) data is available on edge risks for any set of vehicle routing instances in the literature. The algorithms developed in Section 4, however, do not depend in any way on the chosen measure for p_{ij}. Further studies about an alternative risk index and investigations about the complexity of the resulting problem are left for future work.

In the remainder of the paper, it is assumed that the vehicle is empty when leaving the depot, after which it retrieves funds from the customers visited on its route, before returning to the depot where a secure safe awaits. The vehicle is supposed to only *pick up* cash along its route and never to drop it off, before the depot is reached. Under these assumptions, to determine the route risk, a risk index R_i^r is computed for each node i in route r in a recursive manner starting from the depot 0 using Eq. (5), where D_i^r is the amount of money on board of the vehicle when it leaves node i along route r and c_{ij} is the length of arc (i, j) contained in r.

$$R_j^r = R_i^r + D_i^r \cdot c_{ij} \qquad (5)$$

The risk index of a route is a cumulative (increasing) measure of the risk incurred by a vehicle while it travels along its route. The *global route risk* of route r, denoted as GR^r, is the risk incurred by a vehicle upon its return to the depot. In line with real-life CIT applications: (i) the higher the money transported, the higher the risk; (ii) the longer the vehicle route, the higher the risk.

In the RCTVRP, the global route risk of every route is limited to a certain maximum value, called the *risk threshold T*. The latter can be used by a CIT firm to limit the risk along its routes, and as a consequence, it could be a critical factor to be considered in order to negotiate fair insurance policies. Actually, insurance companies compute premiums on the basis of the risk exposure of a CIT firm which is strictly related to several factors such as: the investment in security, training of personnel, risk management activities, reputation of the CIT company and loss records (Turner, 2008). Since premiums are very closely linked to the risk exposure and CIT firms are under pressure to cut their operative costs (Turner, 2009), a possible path towards the reduction of insurance premiums lies in the risk reductions efforts aimed at generating relatively safe routes (Turner & Edward, 2005). This goal could be achieved by using risk thresholds or by using any other technique aimed at lowering the risk profile of a CIT firm.

2.2. Illustrative example

The example in Fig. 2 may clarify the calculation of the global route risk. The numbers on the arcs represent their length, while the values on the nodes denote the demand associated to each customer. It is assumed that a vehicle visits nodes A, B, and C and then returns to the depot in a route r. The risk index at node A (top left side of Fig. 2) is zero because the vehicle travels empty along arc $(0, A)$ ($R_A^r = d_0 \cdot c_{0A} = 0 \cdot 7 = 0$). At node A the vehicle collects one unit of money and continues to customer B. The risk index at node B is therefore equal to $R_B^r = R_A^r + (d_0 + d_A) \cdot c_{AB} = 1 \cdot 8 = 8$ (top right side of Fig. 2). Similarly, the risk index at node C is equal to $R_C^r = R_B^r + (d_0 + d_A + d_B) \cdot c_{BC} = 8 + (1 + 3) \cdot 9 = 44$ (bottom left side of Fig. 2). The global route risk when the vehicle returns to the depot is equal to $GR^r = R_C^r + (d_0 + d_A + d_B + d_C) \cdot c_{C0} = 89$ (bottom right side of Fig. 2). Route $r = (0, A, B, C, 0)$ is feasible in the RCTVRP if GR^r is at most equal to the risk threshold T.

Unlike in the traditional VRP, it is the risk constraint and not the capacity of the vehicle that prohibits visiting all customers in a single giant tour. Furthermore, the global route risk of a route r is generally not the same as the global route risk of the reversed route $\bar{r}$. For this reason, it is possible for a route to be feasible if covered in one direction and infeasible in the other one. This is an important consideration in the development of heuristic optimization approaches. Because of the cumulative risk index, the RCTVRP bears some resemblance to the travelling repairman (also called minimum latency) problem, in which the sum of visiting times at the customers is minimized. In the RCTVRP, however, the cumulative measure is treated as a constraint and not as an objective function.

3. Problem description and mathematical formulation

Formally, the RCTVRP is defined on a directed graph $G = (V, A)$. For the sake of simplicity the central depot 0 is replaced by two dummy nodes, s (start) from which all vehicle routes depart and e (end) where all routes end. Therefore the set of nodes $V = \{s, e\} \cup N$ corresponds to the set of customers $N = \{1, \ldots, n\}$, and to the depot. Each customer $i \in N$ has a non-negative demand d_i which represents the cash to be picked up by the vehicle during its visit. The demands associated to the two dummy depots are equal to zero ($d_s = d_e = 0$). The set of arcs is defined as $A = (N \times N) \cup (\{s\} \times N) \cup (N \times \{e\})$. A non-negative

distance (or travel time) c_{ij} is associated with each arc $(i, j) \in A$. All vehicles start empty ($d_s = 0$) from the depot and perform a single route, visiting a sequence of customers before returning to the depot (e) where the collected cash (demand) is deposited. At the start of their tour, each vehicle's *risk index* is equal to zero. A vehicle travelling between nodes i and j increases its risk index by a value that is equal to the product of the amount of cash it carries when it leaves node i and the distance (or travel time) c_{ij}.

To formulate this problem as a mixed integer programming (MIP) problem, two families of decision variables are defined for each node. Let D_i^r be equal to the cash carried by the vehicle when it leaves node i along route r. Let R_i^r be the value of the risk index for the vehicle when it arrives at node i along route r. Note that, for a given customer, all but one of these variables will be zero, because each customer is only visited by one vehicle.

The Boolean decision variable x_{ij}^r is equal to 1 if arc $(i, j) \in A$ is traversed by the vehicle along route r and 0 otherwise. The number of routes is determined as part of the optimization problem, and is at most equal to n in a solution in which each route contains only one node. For this reason, without loss of generality the route index r can be defined on the customer set N.

$$\min \sum_{r \in N} \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}^r \tag{6}$$

s.t.

$$\sum_{j \in N} x_{sj}^r = \sum_{i \in N} x_{ie}^r \qquad \forall r \in N \tag{7}$$

$$\sum_{j \in N} x_{sj}^1 = 1 \tag{8}$$

$$\sum_{i \in N} x_{ie}^r \geqslant \sum_{j \in N} x_{sj}^{r+1} \qquad \forall r \in N \setminus \{n\} \tag{9}$$

$$\sum_{r \in N} \sum_{j \in V \setminus \{s\}} x_{ij}^r = 1 \qquad \forall i \in N \tag{10}$$

$$\sum_{h \in V \setminus \{e\}} x_{hj}^r - \sum_{k \in V \setminus \{s\}} x_{jk}^r = 0 \qquad \forall j \in N; \forall r \in N \tag{11}$$

$$D_s^r = 0 \qquad \forall r \in N \tag{12}$$

$$D_j^r \geqslant D_i^r + d_j - (1 - x_{ij}^r) \cdot M_1 \qquad \forall (i, j) \in A; \forall r \in N \tag{13}$$

$$0 \leqslant D_i^r \leqslant M_1 \qquad \forall i \in V; \forall r \in N \tag{14}$$

$$R_s^r = 0 \qquad \forall r \in N \tag{15}$$

$$R_j^r \geqslant R_i^r + D_i^r \cdot c_{ij} - (1 - x_{ij}^r) \cdot M_2 \qquad \forall (i, j) \in A; \forall r \in N \tag{16}$$

$$0 \leqslant R_i^r \leqslant T \qquad \forall i \in V; \forall r \in N \tag{17}$$

$$x_{ij}^r \in \{0, 1\} \qquad \forall (i, j) \in A; \forall r \in N \tag{18}$$

The objective function (6) minimizes the total distance travelled along all the routes. We do not consider the cost of vehicles used in the solution. Constraints (7) impose that each route starts and ends at the depot. Constraint (8) states that the first route ($r = 1$) starts at the depot and must exist. Constraints (9) enforce that route $r + 1$ cannot exist unless route r also exists, which forces routes to be numbered consecutively. Constraints (10) ensure that each customer is visited exactly once. Constraints (11) impose that in route r, the vehicle can leave customer j only if it has previously entered it. Constraints (12)–(14) are used to define the cumulative demand from the depot to each node i along route r (M_1 is a sufficiently large number). Constraints (15)–(17) ensure that the global route risk is at most equal to the risk threshold T, with M_2 again a sufficiently large number. A sufficient choice for these large numbers is $M_1 = \sum_{i \in V} d_i$ and $M_2 = T$. Due to the cumulative demand (D_i^r) and the cumulative risk (R_i^r), which are both increasing along route r, subtours are automatically prevented. Finally, as mentioned in Section 2.1, in Constraints (16) the value of c_{ij} can be replaced by p_{ij} if detailed data about the probability of a robbery on arc (i, j) are available.

Since the RCTVRP is an extension (introducing the risk constraint) of the travelling salesman problem (TSP), which is known to be NP-hard, the RCTVRP is also NP-hard. Therefore we propose some heuristic approaches to solve medium and large problem instances in a relatively short time.

4. Metaheuristic approaches

In this section we develop some metaheuristic approaches to solve the RCTVRP. To this end we execute and report on a transparent design phase in which we develop and test several different combinations of

(a) four construction heuristics,
(b) a local search block, and
(c) two heuristic structures to escape from local optima.

The aim of reporting on this design phase is not only to ensure the quality of the final results, but also to gain additional insight into the contribution of various heuristic components on the quality of the solutions produced.

4.1. Construction phase

Four different procedures are developed to find an initial solution for the RCTVRP. The first two heuristics take the risk constraint into account during the entire construction process. They are

(1) a modified Clarke and Wright heuristic with a greedy randomized selection mechanism (CWg), and
(2) a nearest neighbour heuristic with a greedy randomized selection mechanism (NNg). The latter two heuristics temporarily relax the risk constraint, creating a giant (TSP) tour that is split into feasible tours using an optimal splitting procedure. These procedures are
(3) a TSP nearest neighbour heuristic with a greedy randomized selection mechanism and splitting (TNNg), and
(4) a TSP Lin-Kernighan heuristic with splitting (TLK).

4.1.1. Clarke and Wright heuristic with greedy randomized selection mechanism (CWg)

The first construction heuristic is a variant of the well known Clarke and Wright heuristic (Clarke & Wright, 1964), modified to take into account the risk constraint. The "standard" Clark and Wright construction heuristic starts from a star solution (in which each customer is served by its own vehicle) and iteratively merges two routes. To this end the heuristic pre-computes the saving of each pair of routes, i.e., the decrease in objective function value that would occur if two routes were merged by connecting these two customers, and sorts the list of node pairs in order of decreasing savings. At each iteration the heuristic greedily attempts to merge the pair of nodes with the largest saving. This is possible if

(1) both nodes are connected to the depot;
(2) the sum of the demand of the routes containing the nodes to merge does not exceed the vehicle capacity.

For the RCTVRP, the original Clarke and Wright heuristic is modified in three ways:

(1) in a greedy randomized selection mechanism, the next pair of nodes to merge is chosen randomly from a restricted candidate list containing the α first pairs of nodes on the sorted savings list. In particular α represents a parameter of the heuristic that allows it to generate different initial solutions;
(2) the capacity check is replaced with a risk constraint check;

Table 1
Possible merge scenarios. The reversed route of r_1 (or r_2) is denoted with $\bar{r}_1$ (or $\bar{r}_2$).

Position of node i in route r_1	Position of node j in route r_2	Possible merged routes
First	First	$\bar{r}_2 \rightarrow r_1$
		$\bar{r}_1 \rightarrow r_2$
First	Last	$r_2 \rightarrow r_1$
		$\bar{r}_1 \rightarrow \bar{r}_2$
Last	First	$\bar{r}_2 \rightarrow \bar{r}_1$
		$r_1 \rightarrow r_2$
Last	Last	$r_2 \rightarrow \bar{r}_1$
		$r_1 \rightarrow \bar{r}_2$

(3) both orientations of the resulting routes are checked, and the one with the lowest risk index is retained.

When merging routes r_1 and r_2 to obtain the saving s_{ij}, the CWg heuristic first checks which route need to be reversed, depending on the position of nodes i and j (either first customers reached right after the depot or last customers before the depot), inside r_1 and r_2 respectively. As shown in Table 1, eight different ways (two for each position of nodes i and j in r_1 and r_2 respectively) to generate a merged route starting from r_1 and r_2 are possible. Then, the CWg heuristic checks, in constant time, if those reversed routes are feasible. If at least one of the merged routes is feasible, the heuristic adds the nodes contained in the second route into the first one (or vice versa), recomputing for each of them the resulting risk indices. This operation, executed in linear time, guarantees that the risk indices associated to the nodes in the merged route are always updated.

An example of a merge operation is shown in Fig. 3, where routes r_1 and r_2 containing customers i and j are merged, yielding the saving $s_{ij} = c_{0i} + c_{0j} - c_{ij}$. Both customers i and j are the last nodes of routes r_1 and r_2 respectively. Two orientations are evaluated. A zigzag arc indicates that the initial route has been reversed. In the upper right side of Fig. 3 r_1 is merged with the reversed route $\bar{r}_2$, while in the bottom right side of Fig. 3 r_2 is merged with the reversed route $\bar{r}_1$. If both resulting routes are feasible, the one with the lower global route risk is selected to obtain the saving s_{ij}.
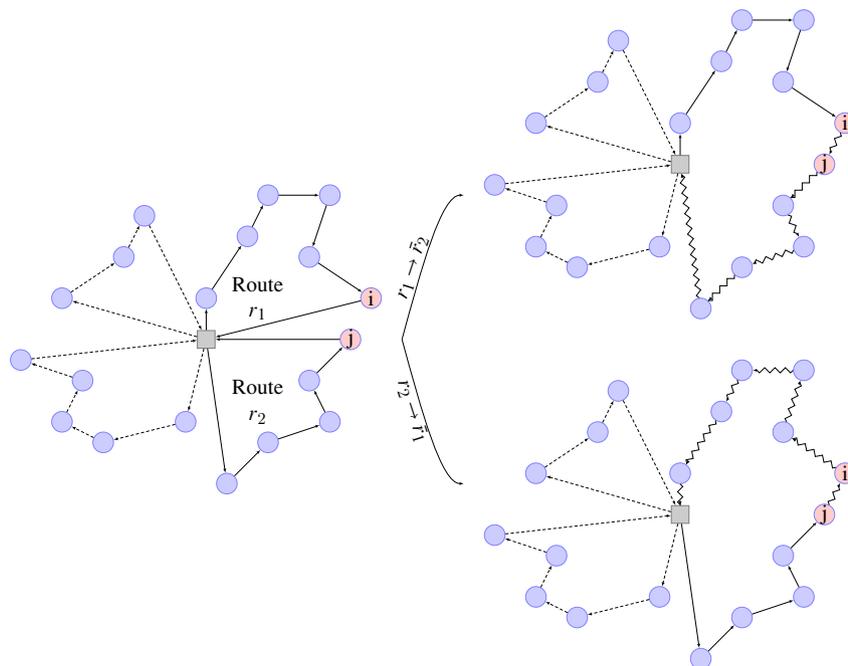
4.1.2. Nearest neighbour with greedy randomized selection mechanism (NNg)

The second constructive heuristic is a nearest neighbour heuristic combined with a greedy randomized selection mechanism. In the standard nearest neighbour heuristic, a solution is constructed by selecting the closest unvisited node at each iteration. Our version modifies the standard heuristic because a greedy randomized selection mechanism is used instead of a simple greedy procedure. This means that the next node in the solution is selected randomly from the restricted candidate list containing the first α closest unvisited nodes passing all a feasibility check with respect to the risk constraint. If it is not possible to add any node to the current route, without violating the risk constraint (i.e., the vehicle is able to reach the depot without surpassing the risk threshold), the route is closed (the vehicle drives back to the depot) and a new route is started.

4.1.3. TSP nearest neighbour with greedy randomized selection mechanism plus splitting (TNNg)

This heuristic, like the NNg, uses a nearest neighbour heuristic combined with a greedy randomized mechanism for the node selection. It does not check the risk constraint and, for this reason, creates a single (TSP) tour. This procedure is iterated a fixed number of times and the best giant TSP tour is subjected to a variant of the splitting procedure described in Prins (2004). The original procedure creates an auxiliary graph containing $n + 1$ nodes (0 to n), and adds an arc between nodes $i − 1$ and j (with $i \leqslant j$) if the route from the ith customer to the jth customer in the order they appear in the giant tour is feasible. Contrary to the original procedure, our modified version also adds an arc if the reverse route is feasible. The weight of the arc is equal to the cost of the route.

The best possible way to split the giant tour in feasible routes, is achieved by finding the shortest path from node 0 to node n in the auxiliary graph. If the shortest path contains the arc from node $i − 1$ to node j, the giant tour is split between the $(i − 1)$th and the ith nodes and between the jth and the $(j + 1)$th nodes. For a more detailed explanation of this procedure we refer to Prins (2004).

4.1.4. TSP Lin–Kernighan plus splitting (TLK)

The Lin–Kernighan heuristic described in Lin and Kernighan (1973) is a deterministic approach generally considered to be one of the

Fig. 3. Creation of a new route merging r_1 and r_2. Only two out of eight possible merge situations are shown.

most effective methods to generate optimal or near-optimal solutions for the symmetric travelling salesperson problem. The fourth constructive heuristic uses the modified Lin–Kerninghan heuristic implemented in Helsgaun (2000, 2006), solving the TSP without enforcing the risk constraint. The solution obtained by this procedure is subjected to the modified splitting heuristic described before.

4.2. Local search operators

The initial solution for the RCTVRP, obtained with one of the four construction heuristics described before, serves as input for a local search heuristic, which is composed of six of the most common local search operators for vehicle routing problems (see Bräysy & Gendreau, 2005), which have been adapted for the RCTVRP. *Two-opt* and *Or-opt* are *intra-route* operators, that attempt to improve the order in which the customers assigned to a vehicle are visited (Kindervater & Savelsbergh, 1997). On the other hand, *inter-route* operators change more than one route simultaneously. In practice, these operators improve the assignment decisions, by determining which vehicle has to serve a certain customer. Our algorithm implements *Two-opt*, *Relocate*, *Exchange* and *Cross-exchange*.

Each local search operator uses a first improvement descent strategy. As soon as a move to improve the current solution is applied, the local search heuristic is restarted from the new current solution. The heuristic stops when the solution cannot be further improved by any of the local search operators. Moreover, each of the local search operators is implemented efficiently, i.e., wherever possible, short-cut calculations are applied to efficiently determine whether a move results in an improved solution and what the new risk indices are for the different routes affected by the move. In many cases, the feasibility check can be done in O(1). For a detailed description of the local search operators used in this paper see Appendix B.

4.3. Metaheuristic structures and diversification

The construction algorithms and the local search heuristic are embedded in two different global metaheuristic structures: a multistart and a perturb-and-improve (or perturbation) structure. The multistart repeats both the construction phase and the local search phase a number of times. The perturbation structure only uses the construction heuristic once, and restarts the local search heuristic from a perturbed solution.

The perturbation operator takes the maximum percentage deterioration Δ of the objective function associated to the current solution $f(x)$ as a parameter. It then chooses a random route, removes the customer with the highest demand from this route and inserts it in a new route, thereby increasing the total distance travelled. This step is repeated until the objective function of the perturbed solution is larger than $(1 + \Delta) \cdot f(x)$. The multistart and the perturbation structures are abbreviated as m- and p-, respectively. Fig. 4 outlines the differences between the two heuristics.

Both structures are separately combined with three construction heuristics (CWg, NNg and TNNg). The TLK construction heuristic is a deterministic construction heuristic. Therefore, it does not make sense to combine it with a multistart structure, since it would

produce the same solution after every restart. As a result, seven heuristic configurations are obtained. Algorithms m-CWg, m-NNg and m-TNNg belong to the class of GRASP metaheuristics (see Feo, Bard, & Venkatraman, 1991), while the solution approaches p-CWg, p-NNg, p-TNNg and p-TLK can be classified as Iterated Local Search (ILS) metaheuristics (see Lourenço, Martin, & Stützle, 2010).

5. Experimental analysis

In this section the seven metaheuristics for the RCTVRP, presented in Section 4, are tested. Since the RCTVRP has not been studied before, no test instances are available in the literature. For this purpose, we generated two sets of instances called SET V ("VRP-lib") and SET R ("Random") which are further described in Appendix A and made available at http://antor.uantwerpen.be/RCTVRP. These two sets form a library of instances for the RCTVRP. The experimental analysis is carried out in two sequential steps described in Sections 5.2 and 5.3 respectively.

In a first stage, the parameters of each metaheuristic are calibrated in such a way that each generates the best possible results when averaged over all possible parameter combinations. This is done by running the metaheuristics on a pre-defined *tuning set* containing a subset of instances randomly selected from SET V and SET R and by analysing the results within a statistical experiment. In a second stage, once the best setting is determined for every metaheuristic separately, the seven calibrated algorithms are compared. This comparison is done in two steps: in the first stage all small instances contained in SET R are solved, while, in the second stage, the larger instances contained in SET V are tested. All experiments have been performed on an Intel core i7-2760QM 2.40GHz processor using a machine with 4 gigabyte RAM.

5.1. Evaluation of the RCTVRP model

In preliminary experiments, we tested the RCTVRP model, described in Section 3, by using IBM CPLEX (version 12.3) to solve small instances (i.e., maximum 20 nodes) contained in SET R. This set is generated in such a way that the impact of the risk threshold and the standard deviation of the customer demand (σ_d) on the computing time can be measured. In particular several risk levels have been used to test the RCTVRP model. The first risk level (RL1) is defined as the minimum risk threshold that allows a feasible solution of the instance as shown in Eq. (19). For this to be possible, the vehicle needs to be able to visit the customer that has the largest product of its distance to the depot and its demand level.

$$\text{RL1} = \max_{i \in N}\{d_i \cdot c_{i0}\} \tag{19}$$

Additional risk levels are generated by using an increasing multiplicative factor in steps of 0.5 (i.e. RL1.5 = $1.5 \cdot$ RL1, . . . , RL2 = $2 \cdot$ RL1). For further details the reader is referred to Appendix A.

The IBM CPLEX solver uses an exact approach based on a branch and bound algorithm to explore the solution space and to detect an optimal feasible solution. As expected this approach is very time consuming and very demanding with respect to memory usage, especially when the risk constraint is very tight. With regard to the solution cost,

---

Algorithm 1: Multistart
while *stopping conditions not met* **do**
generate initial solution;
improve solution using local search;
end
report best solution;

Algorithm 2: Perturbation
while *stopping conditions not met* **do**
improve solution using local search;
perturb solution;
end
report best solution;

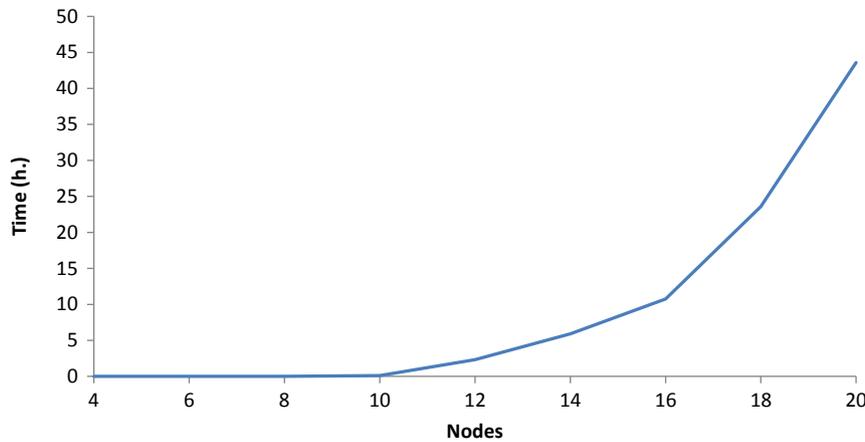**Fig. 4.** Multistart versus perturbation heuristic structure.

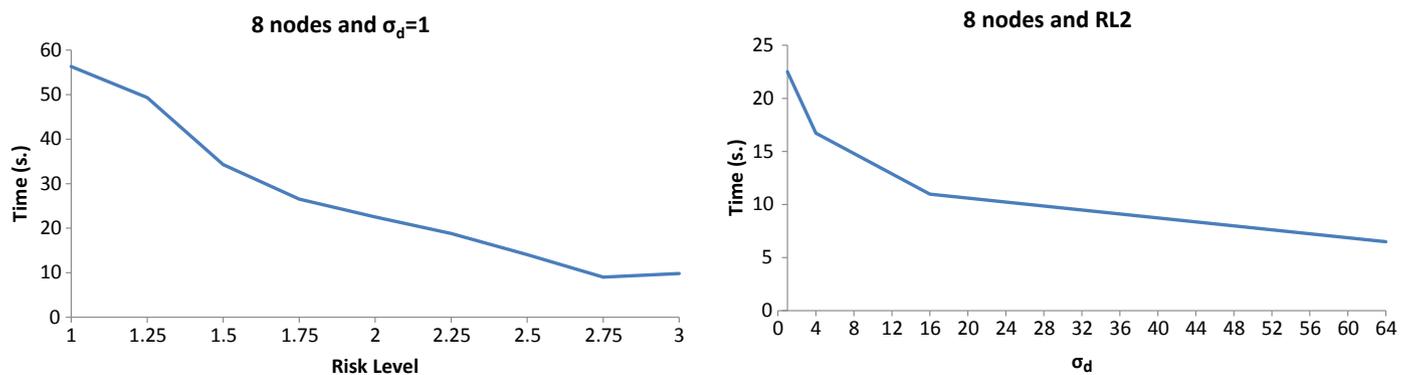**Fig. 5.** Relationship between computation time and number of nodes.

Fig. 6. IBM cplex results. (a) Relationship between CPU time and risk level. (b) Relationship between CPU time and σ_d.

given a value of σ_d, the lower the risk level the higher the solution cost due to routes with a lower number of customers. Unsurprisingly, the computation time increases exponentially with the number of nodes (see Fig. 5), which suggests that for medium and large instances (like those in SET V) the exact approach is not suitable.

Further results suggest that both the risk level and σ_d are negatively correlated with the computation time (see Fig. 6(a) and (b)). Instances with either a smaller risk level or a smaller demand variance are therefore more difficult to be solved by IBM CPLEX. In particular the lower the risk threshold, the more constrained the problem. Therefore the branch and bound algorithm, used by IBM CPLEX, might require a deeper exploration of the search tree.

Moreover, when σ_d grows, the order in which the vehicle visits the nodes along its route assumes a greater importance for the feasibility of the problem. As already mentioned in Section 2, the global route risk of a route r (e.g., (A, B, C, D)) is generally not the same as the global route risk of the reversed route $\bar{r}$ (e.g., (D, C, B, A)). Thus, when the demand associated to the nodes has a low standard deviation, it can happen that both routes $\bar{r}$ and r are feasible, resulting in higher computation times required by an exact approach.

5.2. Calibration of the metaheuristic configurations

For each metaheuristic we tested the parameters related to the instance characteristics (*instance parameters*) as well as parameters related to the metaheuristic configurations themselves (*heuristic parameters*). This allows us to study the relationship between the performance of each tested metaheuristic, for different parameter settings and the characteristics of the instance. In Table 2 an overview of these parameters is given, as well as a list of the tested values and their total number.

In order to tune each metaheuristic a tuning set was generated, separating the experimental set of instances (used to test and compare the metaheuristics in Section 5.3) from the control sample (used to tune the metaheuristics). In particular, this set was obtained by randomly selecting from the same population set (both sets SET V and SET R) a limited number of small, medium and big instances. Each instance in the tuning set (see Table 3) was solved using each metaheuristic approach in a full factorial experiment (all combinations of the heuristic parameters were tested). In order to have a fair comparison between competing algorithms, we let each metaheuristic (with a specific setting of the heuristic parameters to be tested) run on each instance for a fixed running time of 5 seconds.

Preliminary experiments revealed that all the local search operators employed in each metaheuristic have a significant effect on reducing the objective function. Although the order in which the neighbourhoods are investigated may have an impact on the quality of the obtained solutions, we did not observe such an effect in some pilot analyses. For this reason, in each metaheuristic we employed all the local search operators according to the order shown in Table 4.

To analyse the results of the experiment a full factorial ANOVA analysis was performed in SAS. The results obtained by each metaheuristic were analysed in terms of:

(1) *quality*, i.e., the capacity to find good solutions given the best possible setting of the *heuristic parameters*;
(2) *robustness*, i.e., the capacity to obtain good solutions for different levels of the *heuristic parameters*.

All metaheuristics were able to find excellent solutions for all solved instances. The largest observed difference in the lowest objective values between any pair of metaheuristics on any instance is 5.1 percent. On average, however, p-TLK and m-CWg obtained the

Table 2
Heuristic and instance parameters.

Parameter	Description	Values	#
Instance parameters			
Nodes	Number of nodes	14, 18, 22, 121, 200	5
RL	Risk level (see Table A.2)	1, 1.5, 2, 2.5, 3	5
Heuristic parameters			
Restart/Repetition	Number of restarts (multistart structure) or number of perturbations (perturbation structure)	As many as possible within 5 seconds	–
TSP-Repetition	Number of initial giant tours to generate. Only the best of these tours is subject to the splitting mechanism	2, 5, 10, 20	4
G-alpha	Size of the restricted candidate list α (greedy randomized selection mechanism)	1, 2, 3, 4, 5	5
Perturbation	Maximum deterioration Δ of the objective function	5 percent, 10 percent, 20 percent, 50 percent, 70 percent	5

Table 3
Instances contained in the tuning set.

Nodes	Instance set	Risk level
22	SET V	1
121	SET V	1.5
14	SET R	2
18	SET R	2.5
200	SET V	3

Table 4
Order of the local search operators used inside all metaheuristics.

N_λ	Local search operator
N_1	*Or-opt internal*
N_2	*Two-opt internal*
N_3	*Exchange external*
N_4	*Relocate external*
N_5	*Cross-exchange external*
N_6	*Two-opt external*

Table 6
Results obtained by solving the instances contained in SET R.

Metaheuristic	Best GAP (percent)	Avg. GAP (percent)
m-CWg	0.0	0.10
m-NNg	0.0	1.28
p-CWg	0.0	0.43
p-NNg	0.0	0.88
m-TNNg	0.0	0.58
p-TNNg	0.0	0.69
p-TLK	0.0	0.09

the number of Restart/Repetition, are relatively low, the p-CWg, p-NNg, p-TNNg and p-TLK metaheuristics can lead to different solutions. If the parameter Restart/Repetition grows larger, the differences in the initial solutions provided by the CWg, NNg, TNNg and TLK should be quickly overcome by the local search operators.

5.3. Results and metaheuristic comparison

To test and compare the calibrated metaheuristics (with all parameters set to their best possible levels) a two-step computational experiment was performed. In the first stage each metaheuristic was run 20 times on all instances of SET R, allowing a maximum running time of 5 seconds per run. The results were compared with the optimal solutions found by using IBM CPLEX. In the second stage, in order to test the behaviour of the metaheuristics on larger problems, all instances from SET V were solved.

All metaheuristics were able to find optimal solutions as shown in Table 6. For each metaheuristic we report the best percentage gap from the optimal solutions averaged over all instances in SET R and the average percentage gap from the optimal solutions over 20 runs. By comparing the metaheuristic performance and the instances parameter it can be remarked that instances are more difficult to solve when they

(1) are larger,
(2) have a low risk threshold, or
(3) have a small demand variation.

best results, both in terms of quality and robustness, while m-NNg performed worst. Statistical analysis of the relationship between instance parameters and metaheuristic performance shows that the m-CWg, p-CWg, and p-TLK metaheuristics perform better for large instances. If the risk level increases (hence the risk constraint becomes less tight) the best solution approaches are m-CWg and p-TLK.

Based on the above analysis, Table 5 summarizes the best parameter settings for each heuristic configuration. Of course, increasing the number of Restart/Repetition can only improve solution quality, but at the expense of increasing the running time. However, our experiment showed that both the GRASP metaheuristics (m-CWg, m-NNg and m-TNNg) and the ILS metaheuristics (p-CWg, p-NNg, p-TNNg and p-TLK) converge in a relatively small number of Restart/Repetition.

With reference to the group of ILS metaheuristics, the CWg, NNg, TNNg and TLK constructive procedures are applied only once, at the first step of each metaheuristic, to find an initial solution. Then, the metaheuristics are restarted from a perturbed solution. In that case, since the optimal values, for both the parameter Perturbation and

Table 5
Optimal settings for the seven metaheuristics.

Parameter	m-CWg	m-NNg	p-CWg	p-NNg	m-TNNg	p-TNNg	p-TLK
TSP-Repetition	–	–	–	–	5	10	–
G-alpha	4	3	3	2	2	3	–
Perturbation	–	–	20 percent	10 percent	–	10 percent	5 percent

Table 7
Results obtained solving the instances contained in SET V.

Risk level	m-CWg (percent)	m-NNg (percent)	p-CWg (percent)	p-NNg (percent)	m-TNNg (percent)	p-TNNg (percent)	p-TLK (percent)
(a) Best gap from best known solutions							
RL1	0.72	1.72	0.92	1.33	1.28	1.02	0.49
RL1.5	0.75	2.07	1.22	1.26	2.05	1.83	0.37
RL2	0.90	2.21	1.00	1.56	1.57	1.05	0.48
RL2.5	1.12	1.78	1.16	1.66	1.72	1.52	0.77
RL3	0.83	1.60	1.25	1.50	1.15	1.16	0.50
Average	**0.86**	**1.88**	**1.11**	**1.46**	**1.55**	**1.32**	**0.52**
(b) Average gap from best known solutions							
RL1	1.32	3.60	1.85	3.33	2.39	2.69	1.11
RL1.5	1.67	3.91	2.48	3.43	3.19	3.28	1.04
RL2	1.69	3.54	2.31	3.21	2.72	2.76v	1.11
RL2.5	2.37	3.66	2.76	3.19	2.95	2.98	1.92
RL3	1.84	3.89	2.51	3.71	2.52	2.65	1.46
Average	**1.89**	**3.75**	**2.51**	**3.38**	**2.84**	**2.92**	**1.38**

Detailed results for each instance can be found at http://antor.uantwerpen.be/RCTVRP.

In the second stage of our experiments, all seven metaheuristics were tested on the larger instances contained in SET V (see Section A.2 for more details). Each metaheuristic was run 20 times on each instance and because of the larger dimension of the instances we allowed a maximum running time of 30 seconds per run. Since the optimal solutions are not known, we compared the results offered by each metaheuristic with the best solution obtained so far for each instance.

The aggregated results for each risk level are summarized in Table 7. In particular: Table 7(a) shows for each metaheuristic the best percentage gap from the best known solutions averaged over all instances in SET V; Table 7(b) presents as a measure of the robustness of each metaheuristic, the percentage gap between the best and the average solutions found after 20 runs and averaged over all instances in SET V.

On average all metaheuristics achieved good results, finding a good balance between quality (the best gap from best known solutions averaged over all metaheuristics is equal to 1.24 percent) and robustness (the average gap from best known solutions averaged over all metaheuristics is equal to 2.67 percent).

The metaheuristics p-TLK and m-CWg outperformed the other solution approaches with respect to the quality of the obtained solutions. However, p-TLK showed a better performance than m-CWg both in terms of solution quality (0.52 percent against 0.82 percent) and robustness (1.38 percent against 1.89 percent). These results are probably due to a higher amount of `Restart/Repetition` that the p-TLK can execute within the same running time than the m-CWg metaheuristic (on average 14 times more).

The NNg heuristic both in combination with a multistart and a perturbation heuristic exhibits relatively poor performance compared to the other metaheuristics. These conclusions are in line with the results obtained when solving the instances in SET R in the first stage of our experiments.

6. Conclusions

In this paper we introduced a new risk index in order to build safe routes that have a global route risk smaller than a threshold value. We defined a new variant of the vehicle routing problem, called risk-constrained cash-in-transit vehicle routing problem or RCTVRP. We also developed a mathematical formulation for the RCTVRP and proposed a library containing two sets of instances, called respectively SET R and SET V. Using IBM CPLEX as a solver, we were able to test the accuracy of our mathematical formulation finding optimal solutions for all small instances contained in SET R.

In order to solve medium and large RCTVRP instances in a reasonable time, we developed seven different metaheuristics. We tested these solution approaches by analysing the relationship between the instance characteristics and the performance of the seven solution approaches. For each metaheuristic, a statistical experiment was set up to determine the best heuristic parameter settings.

Using the optimal parameter setting of each metaheuristic we solved all instances in SET R and SET V. All the algorithms were able to find the optimal solutions for all instances in SET R and achieved relatively small gaps from the best known solutions. In general, the best solution approaches were p-TLK and m-CWg. However, p-TLK is to be preferred since it offers a better solution quality and robustness.

Future research can be aimed at handling the risk as an additional objective to be minimized and not only as a constraint. Moreover, the problem can be extended in several ways, taking into consideration route length restrictions, time windows, precedence relations, delivery of money to customers, etc.

Appendix A. Library of RCTVRP instances

A.1. Random instances (SET R)

SET R consists of small RCTVRP instances with a number of nodes ranging from 4 to 20. The set contains 180 instances, the result of combining 9 basic instances (having 4, 6, 8, 10, 12, 14, 16, 18 and 20 nodes respectively), 5 different risk levels and 4 different values of the standard deviation of the demand vector (σ_d). For each instance the coordinates of each node are randomly selected from the interval $[-50, 50]$. The demand associated with each node is generated in such a way that the standard deviation of the demand σ_d is equal to $1, 4, 16$ and 64, depending on the instance. In addition, five different levels of the risk threshold are determined for each instance. The first one is represented by the minimum risk level that allows a feasible solution defined as $RL1 = \max_{i \in N}\{d_i \cdot c_{i0}\}$. The remaining four levels of risk threshold (named RL1.5, RL2, RL2.5 and RL3 respectively) are generated, starting from RL1 by using an increasing multiplicative factor in steps of 0.5, up to 3, as shown in Table A.2.

A.2. VRP-lib instances (SET V)

SET V is based on 14 instances from the VRP library (available at http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html) with a number of nodes ranging from 22 to 301 (see Table A.1).

These 14 basic instances are combined with 5 different levels of risk threshold as summarized in Table A.2. Therefore, the total number of instances contained in the set is equal to 70.

Table A.1
Basic instances from the VRP library.

Name	Nodes
E022-04g	22
E026-08m	26
E030-03g	30
E036-11h	36
E045-04f	45
E051-05e	51
E072-04f	72
E101-08e	101
E121-07c	121
E135-07f	135
E151-12b	151
E200-16b	200
E256-14k	256
E301-28k	301

Table A.2
Risk levels.

Risk level	Value
RL1	$\max_{i \in N}\{d_i \cdot c_{i0}\}$
RL1.5	$1.5 \cdot RL1$
RL2	$2.0 \cdot RL1$
RL2.5	$2.5 \cdot RL1$
RL3	$3.0 \cdot RL1$

Appendix B. Local search operators

In this section, the local search operators used in all metaheuristics are described. For each of them we report the complexity of the operator and the short-cut calculations which are used to efficiently determine whether a move results in an improved solution and what the new risk indices are for the different routes affected by the move. In order to make these efficient calculations possible, some additional data structures are updated throughout the local search process following the idea introduced in Kindervater and Savelsbergh (1997) on handling side constraints in arc-exchange heuristics:

(1) D_h^r, the cumulated cash carried by the vehicle when it leaves node h along route r,
(2) C_h^r, the cumulative distance travelled from the depot to node h along route r,
(3) R_h^r, the value of the risk index for the vehicle when it reaches node h along r,

(4) GR^r, the global risk of route r, and
(5) GC^r, the total length of route r.

B.1. Two-opt intra-route operator

The two-opt intra-route operator removes two edges from a given route and reconnects the route, thereby reversing part of the route. As mentioned, such a reversal can have an impact on the risk index, and therefore the feasibility, of the route. Consider a two-opt move applied to route $r = (0, \ldots, A, B, \ldots, C, D, \ldots, 0)$ (see Fig. B.1 on the left), where edges (A, B) and (C, D) are removed and the solution is reconnected through edges (A, C) and (B, D). The complexity of the move is $O(n^2)$ with the cost check performed in constant time. This move reduces the cost of the new solution only if the following condition holds:

$$c_{AB} + c_{CD} - c_{AC} - c_{BD} > 0 \tag{B.1}$$

The two-opt intra route operator generates a new route $(0, \ldots, A, C, \ldots, B, D, \ldots, 0)$, reversing the order of the nodes between B and C (Fig. B.1 on the right). This reversal, due to the nature of the risk index described in Section 2, makes it impossible to develop a short-cut calculation to determine the risk index of the new route in constant time. The complexity of the feasibility check requires thus $O(k)$ where k represents the nodes which are added, starting from node C, to create a new resulting route. The total complexity of the two-opt intra-route operator grows thus to $O(k\,n^2)$.

B.2. Or-opt intra-route operator

The Or-opt intra route operator takes a route $r = (0, \ldots, A, B, C, D, \ldots, E, F, \ldots, 0)$ (see Fig. B.2 on the left), and relocates a string of k consecutive nodes to another location in r. The relocate inter-route operator is a particular case of the Or-opt intra route operator in which $k = 1$. In our case we used $k = 2$, moving the arc (B, C) between nodes E and F, maintaining the same direction of the initial route. The complexity of the move is $O(n^2)$, while the cost evaluation can be performed in constant time. In order to reduce the cost of the current solution this move is performed only if:

$$c_{AB} + c_{CD} + c_{EF} - c_{AD} - c_{EB} - c_{CF} > 0 \tag{B.2}$$

The global route risk for the new route $(0, \ldots, A, D, \ldots, E, B, C, F, \ldots, 0)$ can be calculated in $O(1)$ as follows:

$$GR^r + D_A^r \cdot (c_{AD} - c_{AB} - c_{BC} - c_{CD}) + D_E^r \cdot (c_{EB} + c_{BC} + c_{CF} - c_{EF})$$
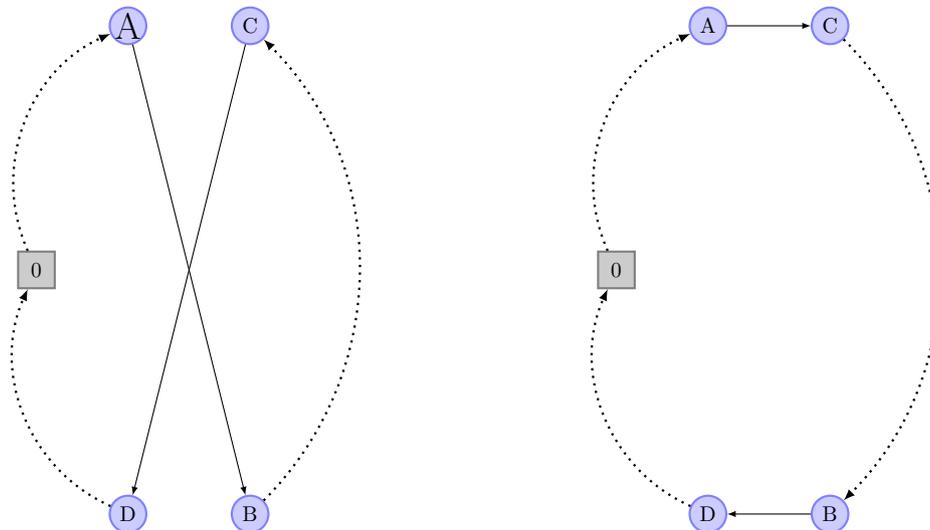$$- (d_B + d_C) \cdot (C_E^r - C_D^r + c_{BC} + c_{CD} + c_{EB}) \tag{B.3}$$

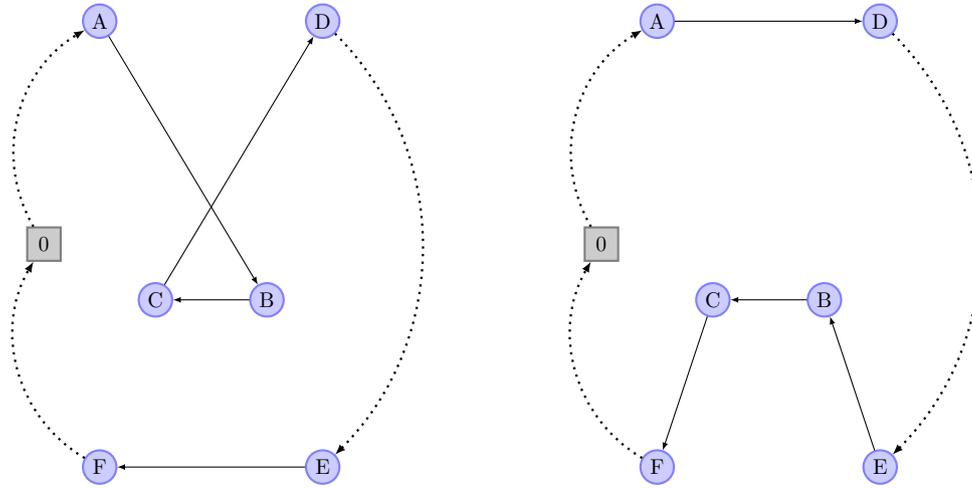**Fig. B.1.** Two-opt intra-route operator.

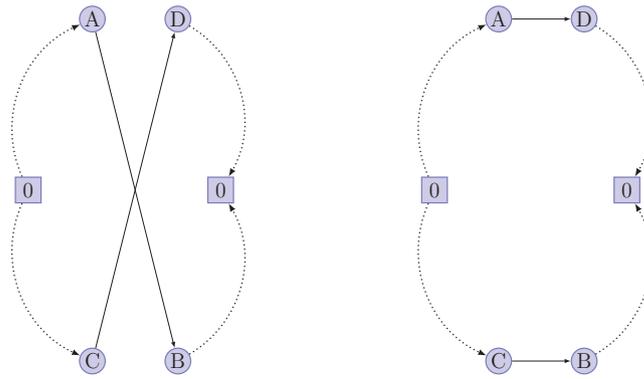**Fig. B.2.** Or-opt intra-route operator.

Fig. B.3. Two-opt inter-route operator.

Since all values in Eq. (B.3) are updated throughout the running of the algorithm, this feasibility check can be performed in O(1). The resulting route after the application of the move is depicted on the right side of Fig. B.2.

B.3. Two-opt inter-route operator

Let $r_1 = (0, \ldots, A, B, \ldots, 0)$ and $r_2 = (0, \ldots, C, D, \ldots, 0)$ (see Fig. B.3 on the left) be two routes in the current solution. A two-opt inter-route move consists in exchanging the partial routes $(B, \ldots, 0)$ and $(D, \ldots, 0)$ between r_1 and r_2, while maintaining the same direction of the initial routes. This move reduces the cost of the current solution only if the following condition is satisfied:

$$c_{AB} + c_{CD} - c_{AD} - c_{CB} > 0 \qquad (B.4)$$

The global route risk associated to the new route $(0, \ldots, A, D, \ldots, 0)$ is equal to:

$$R_A^{r_1} + D_A^{r_1} \cdot \left(c_{AD} + GC^{r_2} - C_D^{r_2}\right) + GR^{r_2} - \left(D_D^{r_2} - d_D\right) \cdot \left(GC^{r_2} - C_D^{r_2}\right) - R_D^{r_2} \qquad (B.5)$$

Similarly, the global route risk of the new route $(0, \ldots, C, B, \ldots, 0)$ is:

$$R_C^{r_2} + D_C^{r_2} \cdot \left(c_{CB} + GC^{r_1} - C_B^{r_1}\right) + GR^{r_1} - \left(D_B^{r_1} - d_B\right) \cdot \left(GC^{r_1} - C_B^{r_1}\right) - R_B^{r_1} \qquad (B.6)$$

Because the direction of the initial routes is not inverted, the feasibility check can be performed in O(1). Since the cost reduction check can also be performed in constant time, the total complexity of the

Two-opt inter-route operator is thus equal to the complexity of the neighbourhood explored $O(n^2)$. The resulting routes are shown on the right side of Fig. B.3.

B.4. Relocate inter-route operator

Starting from routes $r_1 = (0, \ldots, A, B, C, \ldots, 0)$ and $r_2 = (0, \ldots, D, E, \ldots, 0)$ contained in the current solution (see Fig. B.4 on the left), the relocation of node B from r_1 to r_2, while maintaining the same direction of the two initial routes, is desirable only if the following condition is satisfied:

$$c_{AB} + c_{BC} + c_{DE} - c_{AC} - c_{DB} - c_{BE} > 0 \qquad (B.7)$$

The complexity of the move is $O(n^2)$ with the cost reduction check and the feasibility check both performed in O(1). Taking into account the triangular inequality, it is clear that the risk constraint is always satisfied for route $(0, \ldots, A, C, \ldots, 0)$. The global route risk associated to the new route $(0, \ldots, D, B, E, \ldots, 0)$ can be calculated in O(1) time as:

$$GR^{r_2} + D_D^{r_2} \cdot \left(c_{DB} + c_{BE} - c_{DE}\right) + d_B \cdot \left(GC^{r_2} - C_E^{r_2} + c_{BE}\right) \qquad (B.8)$$

If this value is lower than the risk threshold, the relocation of node B from r_1 to r_2 can be performed obtaining the routes shown on the right side of Fig. B.4.

B.5. Exchange inter-route operator

Let $r_1 = (0, \ldots, A, B, C, \ldots, 0)$ and $r_2 = (0, \ldots, D, E, F, \ldots, 0)$ be two routes in the current solution (see Fig. B.5 on the left). This operator
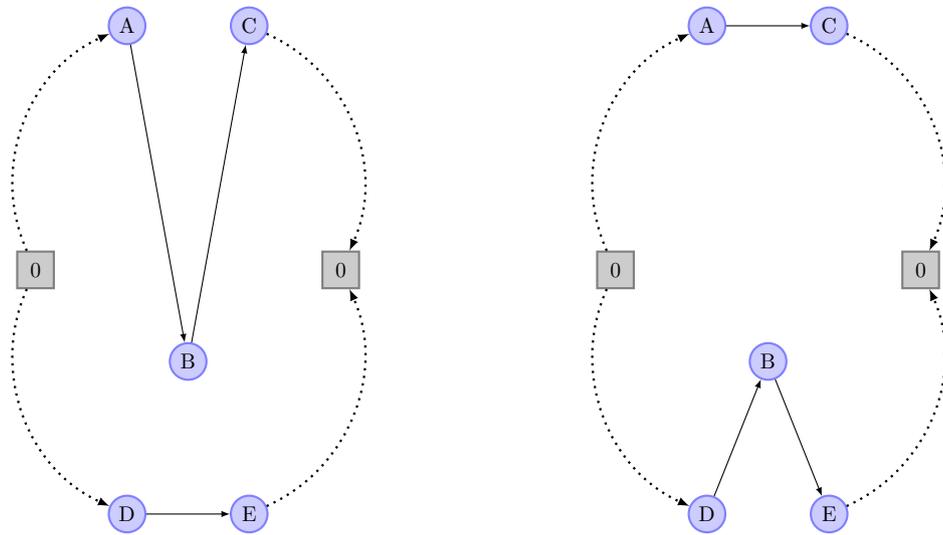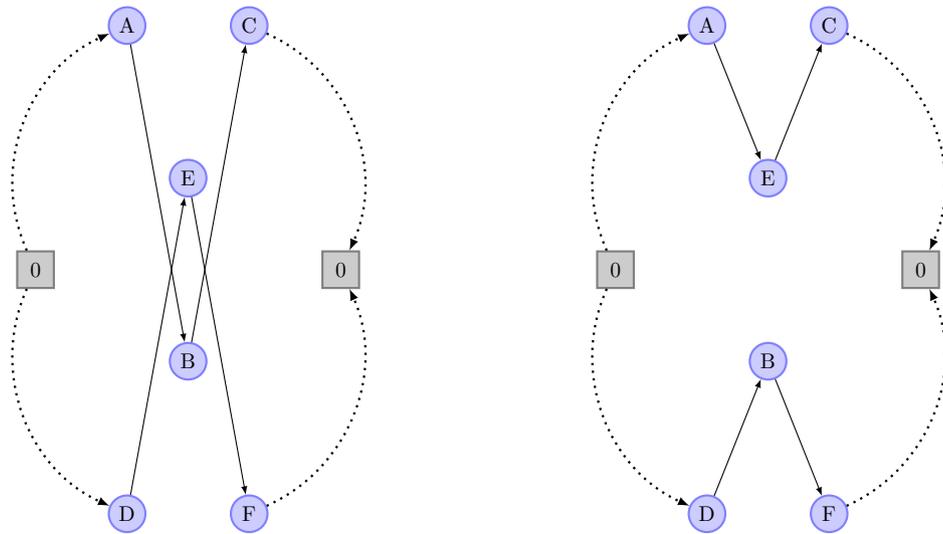
Fig. B.4. Relocate inter-route operator.

Fig. B.5. Exchange inter-route operator.

exchange nodes B and E between these routes while maintaining the same direction of the two initial routes. In order to improve the current solution, the following condition must hold:

$$c_{AB} + c_{BC} + c_{DE} + c_{EF} - c_{AE} - c_{EC} - c_{DB} - c_{BF} > 0 \quad (B.9)$$

The newly obtained routes $(0, \ldots, A, E, C, \ldots, 0)$ and $(0, \ldots, D, B, F, \ldots, 0)$ have the following global route risks respectively:

$$GR^{r_1} + D_A^{r_1} \cdot (c_{AE} - c_{AB} + c_{EC} - c_{BC}) + d_E \cdot c_{EC} - d_B \cdot c_{BC} + (d_E - d_B) \cdot (GC^{r_1} - C_C^{r_1}) \quad (B.10)$$

and

$$GR^{r_2} + D_D^{r_2} \cdot (c_{DB} - c_{DE} + c_{BF} - c_{EF}) + d_B \cdot c_{BF} - d_E \cdot c_{EF} + (d_B - d_E) \cdot (GC^{r_2} - C_F^{r_2}) \quad (B.11)$$

which both have to be smaller than the risk threshold. Both values can be calculated in O(1). In case the exchange is performed, the resulting routes are shown on the right side of Fig. B.5. Given that the feasibility check can be performed in O(1) as well as the check on the cost reduction, the overall complexity of the algorithm is thus O(n^2) (see Osman, 1993 for more details).

B.6. Cross-exchange inter-route operator

The Cross-exchange inter-route operator is similar to the Exchange inter-route operator, but a string of two consecutive nodes is exchanged between a couple of routes. Performing the exchange of arcs (B, C) and (F, G) between routes $r_1 = (0, \ldots, A, B, C, D, \ldots, 0)$ and $r_2 = (0, \ldots, E, F, G, H, \ldots, 0)$ (see Fig. B.6 on the left), while maintaining the original direction of these routes, is desirable only if:

$$c_{AB} + c_{CD} + c_{EF} + c_{GH} - c_{AF} - c_{GD} - c_{EB} - c_{CH} > 0 \quad (B.12)$$

The global route risk of the new routes $(0, \ldots, A, F, G, D, \ldots, 0)$ and $(0, \ldots, E, B, C, H, \ldots, 0)$ (see Fig. B.6 on the right) can be calculated in O(1) as follows:

$$GR^{r_1} + D_A^{r_1} \cdot (c_{AF} - c_{AB} + c_{FG} - c_{BC} + c_{GD} - c_{CD}) + d_F \cdot (c_{FG} + c_{GD}) - d_B \cdot (c_{BC} + c_{CD}) + (d_F + d_G - d_B - d_C) \cdot (GC^{r_1} - C_D^{r_1}) + d_G \cdot c_{GD} - d_C \cdot c_{CD} \quad (B.13)$$

and

$$GR^{r_2} + D_E^{r_2} \cdot (c_{EB} - c_{EF} + c_{BC} - c_{FG} + c_{CH} - c_{GH}) + d_B \cdot (c_{BC} + c_{CH}) - d_F \cdot (c_{FG} + c_{GH}) + (d_B + d_C - d_F - d_G) \cdot (GC^{r_2} - C_H^{r_2}) + d_C \cdot c_{CH} - d_G \cdot c_{GH} \quad (B.14)$$
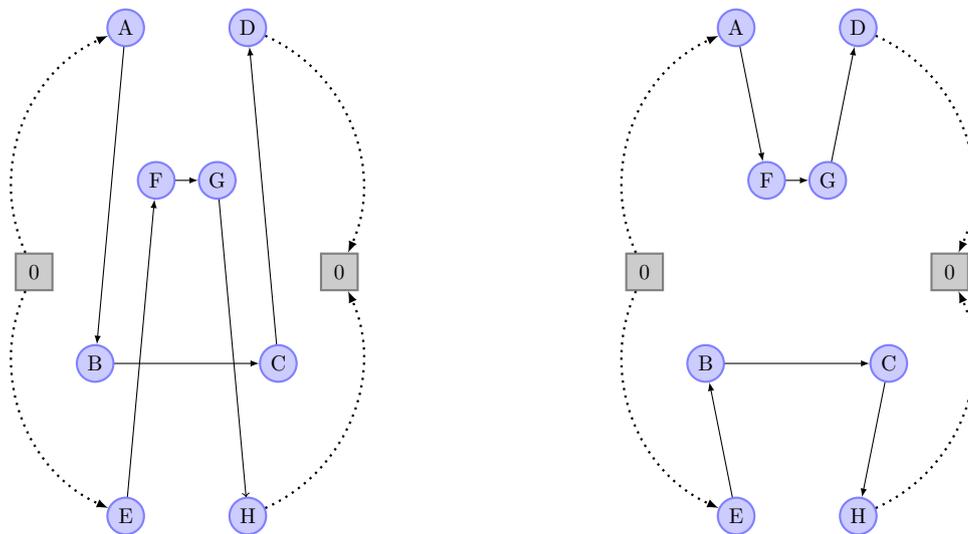
Fig. B.6. Cross-exchange inter-route operator.

The overall complexity of the move is $O(2^2 n^2)$ given that the cost check and the feasibility check can both be performed in $O(1)$. For more details about the complexity of this move the reader is referred to the λ-interchange move described in Osman (1993).

References

Akgün, V., Parekh, A., Batta, R., & Rump, C. M. (2007). Routing of a hazmat truck in the presence of weather systems. *Computers & Operations Research, 34*(5), 1351–1373.

Androutsopoulos, K. N., & Zografos, K. G. (2012). A bi-objective time-dependent vehicle routing and scheduling problem for hazardous materials distribution. *EURO Journal on Transportation and Logistics, 1*(1–2), 157–183.

Bianco, L., Caramia, M., Giordani, S., & Piccialli, V. (2013). Operations research models for global route planning in hazardous material transportation. In R. Batta, & C. Kwon (Eds.), *Handbook of OR/MS models in hazardous materials transportation*. International Series in Operations Research & Management Science (vol. 193, pp. 49–101). New York: Springer.

Bräysy, O., & Gendreau, M. (2005). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science, 39*(1), 104–118.

BSIA (British Security Industry Association). (2013). Combating cash delivery crime. http://www.bsia.co.uk/cvitcrime.

CCPS (Center for Chemical Process Safety). (2008). *Guidelines for chemical transportation safety, security, and risk management*. Hoboken, New Jersey: American Institute of Chemical Engineers.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Díaz-Ovalle, C., Vázquez-Román, R., De Lira-Flores, J., & Mannan, M. S. (2013). A model to optimize facility layouts with toxic releases and mitigation systems. *Computers & Chemical Engineering, 56*, 218–227.

Erasmus, A. (2012). The CIT industry in South Africa – Its challenges and solutions. In ESTA The Cash Management Companies Association (Ed.), *ESTA business conference, exhibition and general assembly*. Bratislava, Slovakia.

Erkut, E., & Ingolfsson, A. (2005). Transport risk models for hazardous materials: revisited. *Operations Research Letters, 33*(1), 81–89.

Feo, T. A., Bard, J. F., & Venkatraman, K. (1991). A *GRASP* for a difficult single machine scheduling problem. *Computers & Operations Research, 18*(8), 635–643.

Helsgaun, K. (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research, 126*(1), 106–130.

Helsgaun, K. (2006). An effective implementation of K-opt moves for the Lin–Kernighan TSP heuristic. *Datalogiske skrifter* (vol. 109). Roskilde University.

Kindervater, G. A. P., & Savelsbergh, M. W. P. (1997). Vehicle routing: Handling edge exchanges. *Local search in combinatorial optimization*. Chichester: Wiley.

Krarup, J. (1975). The peripatetic salesman and some related unsolved problems. In B. Roy (Ed.), *Combinatorial programming: methods and applications* (pp. 173–178). Dordrecht, The Netherlands: Reidel.

Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research, 2*(21), 498–516.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). *Iterated local search: Framework and applications*. New York: Springer.

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 & Operations Research, 41*, 196–207.

Michallet, J., Prins, C., Amodeo, L., Yalui, F., & Vitry, G. (2011). A periodic vehicle routing problem with time windows and time spread constraints on services. In *MIC 2011, the IX metaheuristics international conference*. Udine - Italy.

Milovanović, B., Jovanović, V. D., Živanović, P., & Žeželj, S. (2012). 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.

Ngueveu, S. U., Prins, C., & Wolfler Calvo, R. (2010a). A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics, 10*, 253–266.

Ngueveu, S. U., Prins, C., & Wolfler Calvo, R. (2010b). Lower and upper bounds for the m-peripatetic vehicle routing problem. *4OR, 8*(4), 387–406.

Osman, I. H. (1993). Metastrategy simulated annealing and TS algorithms for the vehicle routing problem. *Annals of OR, 41*, 421–451.

Pillay, K. (2008). The impact of stress and trauma on the occupational environment of CIT security officers – an explanatory perspective. In *International society for criminology, XV world congress*. Barcelona, Spain.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research, 31*(12), 1985–2002.

Reniers, G. L. L., De Jongh, K., Gorrens, B., Lauwers, D., Van Leest, M., & Witlox, F. (2010). Transportation Risk ANalysis tool for hazardous Substances (TRANS) – a user-friendly, semi-quantitative multi-mode hazmat transport route safety risk estimation methodology for flanders. *Transportation Research Part D: Transport and Environment, 15*(8), 489–496.

Russo, F., & Rindone, C. (2011). Planning in road evacuation: Classification of exogenous activities. In A. Pratelli, & C. A. Brebbia (Eds.), *Urban transport XVII urban transport and the environment in the 21st century*. WIT Transactions on the Built Environment (vol. 116, pp. 639–651). Southampton: WIT Press.

Smith, L., & Louis, E. (2010). Cash in transit armed robbery in Australia. *Trends and Issues in Crime and Criminal Justice, 397*, 1.

Turner, P. (2008). Insurance update. In ESTA The Cash Management Companies Association (Ed.), *ESTA business conference, exhibition and general assembly*. Nice, France.

Turner, P. (2009). Insurance update. In ESTA The Cash Management Companies Association (Ed.), *ESTA business conference, exhibition and general assembly*. Rome, Italy.

Turner, P., & Edward, D. (2005). Risk and insurance – frequently asked questions. In ESTA The Cash Management Companies Association (Ed.), *ESTA business conference, exhibition and general assembly*. Budapest, Hungary.

Van Raemdonck, K., Macharis, C., & Mairesse, O. (2013). Risk analysis system for the transport of hazardous materials. *Journal of Safety Research, 45*, 55–63.

Veroni, P. H., & Bourguet, F. (2008). Armed robberies within the bank & cash carrying sector. In ESTA The Cash Management Companies Association (Ed.), *ESTA business conference, exhibition and general assembly*. Nice, France.

Wolfer Calvo, R., & Cordone, R. (2003). A heuristic approach to the overnight security service problem. *Computers & Operation Research, 30*, 1269–1287.

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 & Industrial Engineering, 63*(2), 464–473.