

DEPARTMENT OF ENGINEERING MANAGEMENT

**A bi-objective model to increase security
and reduce travel costs in the cash-in-transit sector**

Luca Talarico, Kenneth Sörensen & Johan Springael

UNIVERSITY OF ANTWERP
Faculty of Applied Economics



City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.uantwerpen.be

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

A bi-objective model to increase security and reduce travel costs in the cash-in-transit sector

Luca Talarico, Kenneth Sørensen & Johan Springael

RESEARCH PAPER 2014-026
NOVEMBER 2014

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@uantwerpen.be

**The research papers from the Faculty of Applied Economics
are also available at www.repec.org
(Research Papers in Economics - RePEc)**

D/2014/1169/026

A bi-objective model to increase security and reduce travel costs in the cash-in-transit sector

Luca Talarico*, Kenneth Sörensen, Johan Springael

University of Antwerp, Department of Engineering Management, ANT/OR - Operations Research Group

Prinsstraat 13, 2000 Antwerp, Belgium

November 2014

In this paper, we present a variant of the vehicle routing problem (VRP) to increase security in the cash-in-transit sector. A specific index is used to quantify the exposure of a vehicle to the risk of being robbed along its route. In addition, the problem is subjected to a traditional capacity constraint, according to which a maximum amount of valuables can be transported inside the vehicle. This constraint might be imposed, for example, by insurance companies.

A bi-objective formulation, aimed at reducing both the risk and the travel cost, is proposed. The objectives are conflicting since higher risk exposures allow a reduction of the travel cost needed to visit and collect valuables from all customers.

A mathematical model of the problem is proposed and solved by using a progressive multi-objective metaheuristic. Realistic instances are also generated considering the geographical coordinates of several customers (e.g., stores, banks, shopping centres) located in Belgium. The proposed solution approach is tuned and tested both on these realistic instances and on standard benchmark instances for the capacitated vehicle routing problem.

Key words: Metaheuristic, Multi-objective optimization, Combinatorial optimization, Cash-in-transit, Security.

1 Introduction and literature review

Transportation of cash and valuables by means of vehicles plays a vital role in our daily lives, supporting the economy of widespread geographical areas (e.g., cities, metropolitan areas,

* Corresponding author: Email: luca.talarico@uantwerpen.be
University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium, Tel: +32 3 265 40 17.

regions). Several cash-in-transit (CIT) companies are specialized in the physical transfer of banknotes, coins and items of value between customers (e.g., supermarkets, jewellery stores, clothes shops, shopping malls) and one or more cash deposits or banks. However, as a consequence of the nature of the transported goods, crime is a significant challenge, and CIT carriers are constantly exposed to risks such as robberies. In fact, attacks on CIT vehicles are absolutely not rare events, although the number of episodes, the risk rates and the average losses are different from country to country. For detailed statistics about the CIT sector the reader is referred to [5, 3].

In order to reduce the incidence of robberies, investments in better vehicles, equipment, infrastructure and technologies (e.g., smart strongboxes, armoured vehicles, weapons) are continuously made. Although improved security measures might work as a deterrent, a robbery cannot always be prevented. Moreover, it should be noted that the risk of being robbed, not only affects the efficiency and the security at an operative level, but it has a considerable impact on CIT company's operating costs.

For this reason, routing plans in the CIT sector should both be safe and efficient. Therefore, a CIT company ought to deal with two critical issues at the same time: the minimization of the travelled cost/time, as well as the reduction of the exposure of the transported goods to robberies. This is not an easy task due to the conflicting objectives and the inner complexity of the routing problems which involve, in general, several hundreds of customers that need to be visited each day.

Despite the large attention that researchers have devoted to vehicle routing problems to model different real-life applications, the issue of *security* during the transportation of cash or valuable goods has gained attention in academic world only very recently. However, specific contributions in this field are still few and far between.

In [9] and [13], security in the transportation of valuables is addressed in the context of the so-called *peripatetic* routing problems. More specifically, customers can be visited several times within a planning horizon, but the use of the same road segment twice is explicitly forbidden.

In [20], the risk of being attacked is reduced by generating routes that are *unpredictable* for criminals. The route *unpredictability* is achieved by defining specific 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. In [11], a similar approach is presented facing a real-case problem submitted by a software company specialized in transportation problems with security constraints. In particular, supposing that the same customer needs to be visited several times during a predefined planning horizon, regularity (in terms of time at which the visit of that customer happens) is avoided, by spreading the visit to that customer over the planning horizon within its time window.

In order to increase unpredictability, a different approach is proposed by Yan et al. [21]. In particular, a model is defined, which incorporates a new concept of similarity for routing problems, considering both time and space measures. Therefore, this approach could be used to generate more flexible routing strategies in order to reduce the risk of robbery. In [18], an

index of similarity, based on the number of identical edges that are in common between alternative solutions, is presented. Furthermore, both a mathematical formulation, that generalizes the well-known *peripatetic* routing problem, and an iterative metaheuristic are presented to generate a set of k dissimilar and not necessarily edge-disjoint solutions for the vehicle routing problem.

In [16], a variant of the well-known capacitated vehicle routing problem is introduced to model the problem of routing vehicles in the CIT industry. In this problem, named `RCTVRP`, a specific risk constraint is introduced to generate relatively safe routing plans. In particular, a risk index associated with a robbery is defined and assumed to be proportional both to the amount of cash being carried and the time/distance covered by the vehicle transporting the cash. This risk index is used to quantify the global route risk, namely the maximum exposure to risk faced by any vehicle along its route, during which a number of customers are visited and cash is collected at each customer's place. A risk constraint forces the global route risk to be not greater than a predefined risk threshold. In [17], an extension of the `RCTVRP` is presented, considering specific hard time windows for which no waiting times at the customer's location are allowed for security reasons. Two effective metaheuristics to solve this problem are also presented.

The main contributions of this paper are fourfold: (1) A multi-objective version of the `RCTVRP`, considering both risk and travel cost as two conflicting objectives that should be minimized at the same time, is described and named `MO-RCTVRP` for short. A mathematical formulation of the `MO-RCTVRP` is also defined; (2) A multi-objective metaheuristic approach to solve the problem is developed and named `PMOO-ILS`. The outcome of the algorithm is represented by a solution, selected from a restricted archive containing high quality alternatives, that better fits the decision maker's preferences; (3) A set of realistic test instances (`SET P`) is generated, based on inputs provided by a real CIT company that operates in Belgium. These instances are made publicly available at <http://antor.uantwerpen.be/Downloads/MORCTVRP>; (4) Both the benchmark instances for the traditional capacitated vehicle routing problem and the instances contained in `SET P` are used to test the `PMOO-ILS` algorithm.

The remainder of the paper is organized as follows. Section 2 is devoted to the problem description. After having introduced the index, that is used to measure the risk of being robbed, a mathematical formulation of the problem is provided. In Section 3 a metaheuristic to solve the `MO-RCTVRP` is presented, based on a progressive multi-objective optimization. The results of the computational experiments, together with the test instances used to solve the problem, are presented in Sections 4. Finally, Section 5 concludes the paper presenting some suggestions for future research.

2 Problem definition and mathematical formulation

The `MO-RCTVRP` problem is defined on a network $\mathcal{G} = \{V, A\}$, being $A = V \times V$ the set of available arcs and $V = N \cup D$ the set of nodes. In particular set N contains n customers that have to be visited and set D contains the depots from which a vehicle can start and/or end

its route. For the sake of simplicity, set D has been divided into two identical sets namely S (starting depots), from which the vehicle routes depart, and E (ending depots), where vehicle routes end (i.e. $S = E = D$). Therefore, a vehicle route is a tour that begins at a depot $s \in S$, traverses a subset of customers in a specified sequence and returns to a depot $e \in E$. For each arc, a non-negative travel cost c_{ij} is defined. In addition, for each customer i , a strictly positive amount of cash/valuables m_i , that has to be collected by the vehicle, is known. In our model we assume to have one single vehicle, initially located at a given initial depot $o \in D$, that performs all the collection routes. In real-life problems, these routes can easily be assigned to multiple vehicles if available.

The goal of the problem is to determine p vehicle routes in order to minimize both the risk of being robbed and the travel cost at once. It should be noted that each customer must be assigned to exactly one of the p routes and the vehicle capacity C must not be exceeded. Assuming that a vehicle only picks up cash at customer's places, a risk index R_j^r can be defined for each customer j , visited along route r , as follows (cf. [17]):

$$R_j^r = R_i^r + M_i^r \cdot c_{ij} \quad (1)$$

where M_i^r is the amount of money on board of the vehicle when it leaves customer i along route r and c_{ij} is the distance between two consecutive customers i and j . M_i^r is defined for each customer and it is obtained by summing the amount of cash picked up by the vehicle at each customer visited along r from the depot until customer i . It is worth observing that in our formulation we used c_{ij} expressed as the distance between two consecutive customers i and j , but it could represent any other measure, not necessarily related to the length of the roadway segment which connects i and j . Additional parameters can also be used to measure the probability of an accident on a specific roadway segment given its characteristics such as lane width, number of lanes etc. (cf. [14] and [12]).

The risk index, defined in Eq. (1), is a cumulative and increasing measure of the risk incurred by the vehicle while it travels along its route. Therefore, the *global route risk* R_e^r , associated to a route r , represents the risk incurred by the vehicle upon its return to the depot $e \in E$.

To formulate this problem as a mixed integer programming (MIP) problem, three families of decision variables are defined. (1) Let M_i^r be equal to the cash carried by the vehicle when it leaves customer i along route r . (2) Let R_i^r be the risk index associated to the vehicle when it arrives at node i along route r . Note that for a given customer i , all but one of these variables will be zero, because each customer is only visited once. (3) Let x_{ij}^r be a binary variable that is equal to 1 if arc $(i, j) \in A$ is traversed by the vehicle along route r and 0 otherwise.

It should be highlighted that the number of routes is determined as part of the optimization problem, and is at most equal to n in a particular routing plan in which each route contains only one node. Of course this scenario allows for the lowest risk values at the expense of the travel cost that reaches a maximum level. For this reason, the index r , that is used for routes, can range from 1 (in which all the customers are visited within a giant single route) up to n . Therefore, in the mathematical notation the index r is defined over the set N .

Hereafter a MIP formulation for the MO-RCTVRP is presented.

$$\min f_1(x) = \sum_{r \in N} \sum_{(i,j) \in A} c_{ij} x_{ij}^r \quad (2)$$

$$\min f_2(x) = \max_{\substack{r \in N \\ e \in E}} R_e^r \quad (3)$$

s.t.

$$\sum_{j \in N} x_{oj}^1 = 1 \quad (4)$$

$$\sum_{s \in S} \sum_{j \in N} x_{sj}^r = \sum_{i \in N} \sum_{e \in E} x_{ie}^r \quad \forall r \in N \quad (5)$$

$$\sum_{i \in N} \sum_{e \in E} x_{ie}^r \geq \sum_{s \in S} \sum_{j \in N} x_{sj}^{r+1} \quad \forall r \in N \setminus \{n\} \quad (6)$$

$$\sum_{r \in N} \sum_{j \in V} x_{ij}^r = 1 \quad \forall i \in N \quad (7)$$

$$\sum_{h \in V} x_{hj}^r - \sum_{k \in V} x_{jk}^r = 0 \quad \forall j \in N; \forall r \in N \quad (8)$$

$$\sum_{i \in V} \sum_{j \in N} m_j x_{ij}^r \leq C \quad \forall r \in N \quad (9)$$

$$M_s^r = 0 \quad \forall r \in N; \forall s \in S \quad (10)$$

$$x_{ij}^r = 1 \Rightarrow M_i^r + m_j = M_j^r \quad \forall (i,j) \in A; \forall r \in N \quad (11)$$

$$R_s^r = 0 \quad \forall r \in N; \forall s \in S \quad (12)$$

$$x_{ij}^r = 1 \Rightarrow R_i^r + M_i^r c_{ij} = R_j^r \quad \forall (i,j) \in A; \forall r \in N \quad (13)$$

$$M_i^r, R_i^r \geq 0 \quad \forall i \in V; \forall r \in N \quad (14)$$

$$x_{ij}^r \in \{0,1\} \quad \forall (i,j) \in A; \forall r \in N \quad (15)$$

The objective function (2) aims to minimize the total travel cost, while objective function (3) attempts to minimize the highest value of global route risk faced by the vehicle over p routes. Constraint (4) states that the first route ($r = 1$) starts at the initial depot $o \in D$. Constraint (5) imposes that each route starts at one depot $s \in S$ and ends at one depot $e \in E$. It may be possible that, for some routes, the initial depot s corresponds to ending depot e . Constraint (6) imposes that route $r + 1$ cannot exist unless route r also exists (for $r > 1$). This constraint forces routes to be numbered consecutively. In addition, according to Constraint (7), each customer must be visited exactly once. Constraint (8) states that the vehicle can leave node j only if it has previously entered it along the same route r . Constraint (9) imposes a restriction on the maximum amount of valuables to be transported inside the vehicle. Constraints (10)-(11) are used to define, in a recursive way, starting from the depot where the vehicle begins its route r , the cumulative demand (i.e. amount of valuables inside the vehicle) associated to each node i visited along r . As stated by Constraint (10), the amount of cash inside the vehicle at the beginning of the route is zero. Constraints (12)-(13) are used to measure, in a recursive way, starting from the depot where the vehicle begins its route r , the global route risk faced by

the vehicle along r upon its return to depot $e \in E$. Sub-tours are automatically prevented by Constraints (10)-(13). Constraints (14)-(15) define the domains of the decision variables.

3 A metaheuristic for the MO-RCTVRP

The MO-RCTVRP problem, described in Eq. (2)-(15), can be considered NP-hard, since it extends in a multi-objective fashion the traditional VRP that is known to be NP-hard (*cf.* [19]). For this reason, metaheuristics present the only viable solution approaches to deal with such kind of problem in a reasonable time.

In the context of multi-objective optimization problems, the concept of optimality needs to be abandoned in favour of the notion of domination. In particular a solution x dominates another solution y if x is at least as good as y with respect to all objectives and better with respect to at least one. In light of this, the goal of a multi-objective solution approach is to find a set of non-dominated solutions that form as a whole the so-called Pareto frontier (or a Pareto frontier approximation in case the set of solutions is really large). From this Pareto frontier containing non-dominated solutions, the decision maker is left to choose one according to his preferences. In case of a high number of non-dominated solutions, the selection process may be a complex task for a decision maker.

Most of the techniques proposed in the literature do not cope with the problem of selecting a solution from the Pareto frontier. In fact, these approaches implicitly assume that the decision maker is able to autonomously use a multi-criteria decision making method for this purpose. A recent study, proposed by Sörensen and Springael [15], introduces a novel technique named progressive multi-objective optimization (PMOO for short) that includes both multi-objective optimization and multi-criteria decision making into one single metaheuristic algorithm. This is made possible by including the user's preferences directly into the multi-objective process, instead of applying an a-posteriori method to select a solution from a previously defined Pareto frontier (approximation). The main benefit of integrating the decision maker's preferences into the optimization process relies on the fact that the decision process can be significantly simplified. In fact, the user needs to choose his favourite solution among a restricted set of high quality alternatives, rather than considering a huge amount of non-dominated solutions.

In this paper, we propose a metaheuristic approach, named PMOO-ILS, that combines the PMOO with an Iterative Local Search heuristic (ILS for short, *cf.* [10]). An archive \mathcal{A} , containing a relatively small number of high quality non-dominated solutions, is generated, maintained and updated through the metaheuristic. This archive \mathcal{A} maintains a fixed number of solutions during the whole solution process.

More specifically, the ILS (described in Section 3.1) is used to generate solutions to be included in the archive \mathcal{A} , while a multi-criteria method, the PROMETHEE II, is used at each iteration of the algorithm to determine whether a new generated solution is allowed to enter into the archive (see Section 3.2). The ILS heuristic embeds and combines: (1) a greedy nearest neighbourhood search heuristic with greedy randomized selection mechanism (GRASP), used to find

an initial solution; (2) a variable neighbourhood descent (VND) heuristic, to improve the current solution; (3) a perturbation heuristic, to escape from local optima.

The overall PMOO-ILS metaheuristic is composed of two consecutive stages. First the archive \mathcal{A} is populated with k non-dominated solutions, by performing the ILS heuristic a few times. In a second phase, both the PROMETHEE II method and the ILS heuristic are repeated, until a stopping criterion is reached (maximum number of iterations I), in order to generate solutions that fit better the decision maker's preferences, updating thus the archive \mathcal{A} . Finally, the "best" solution according to the user's preferences is returned at the end of the algorithm. A general overview of the PMOO-ILS metaheuristic for the MO-RCTVRP is shown in Algorithm 1.

Algorithm 1: PMOO-ILS metaheuristic

Initialize both problem and heuristic parameters;

Let \mathcal{A} be the solution archive initially empty;

Phase 1: Populate the archive

while ($|\mathcal{A}| < k$) **do**

 Randomly generate a w vector;

 Generate a solution x by applying the ILS given w ;

if (x not dominated) **then**

 Add x into \mathcal{A} ;

end

end

Phase 2: Update the archive

Compute the net-flow ϕ of each solution in \mathcal{A} by applying PROMETHEE II;

while (*max number of iterations I not reached*) **do**

 Compute a new vector w_{new} ;

 Find a new solution x_{new} by using the ILS given w_{new} ;

if ($x_{new} \notin \mathcal{A}$) **then**

 Add x_{new} into \mathcal{A} ;

 Remove the solution with the lowest ϕ from \mathcal{A} ;

 Update ϕ for each solution in \mathcal{A} by applying PROMETHEE II;

end

end

Return the solution from \mathcal{A} presenting the highest ϕ

3.1 ILS and solution generation

The first step of the PMOO-ILS algorithm consists in populating the archive \mathcal{A} by using the ILS heuristic. Therefore, the ILS is repeated until k non-dominated solutions are added to \mathcal{A} .

The objective functions $f_1(x)$ and $f_2(x)$ (described in Eq. (2) and (3) respectively) are combined together by means of a linear weighed sum function, with w_1 and w_2 being the weights for the cost function $f_1(x)$ and the risk function $f_2(x)$ respectively. These weights are stored in a

w_i vector that is defined for each solution i . Moreover, each w_i vector is randomly generated such that $w_1 + w_2 = 1$. A GRASP constructive heuristic (see [6] for more details) is used to build feasible routes in a greedy randomized fashion. In fact, at each iteration the nodes to be added to the current route are randomly selected from a restricted list containing the first α non visited nodes ordered by a decreasing value of $w_1 \cdot f_1(x) + (1 - w_1) \cdot f_2(x)$. After the GRASP procedure, a VND heuristic is performed to improve the current solution. The VND is a deterministic variant of the well-known variable neighbourhood search (VNS) metaheuristic (cf. [8]). In general, VNS algorithms use a sequence of nested neighbourhoods, $\mathcal{N}_1, \dots, \mathcal{N}_{\lambda_{max}}$ with an increasing size, i.e., $\mathcal{N}_\lambda \subset \mathcal{N}_{\lambda+1}$.

The VND heuristic, that is used in this paper, is composed of six of the most common local search operators for vehicle routing problems (cf. [2]). In particular *Relocate* and *Or-opt* are *Intra-route* operators, that attempt to improve the order in which the customers, that are assigned to a vehicle, are visited. Conversely, *Two-opt*, *Relocate*, *Exchange* and *Cross-Exchange*, are *Inter-route* operators that change more than one route simultaneously. In practice, the inter-route operators improve the assignment decisions by determining on which vehicle route a customer needs to be included. An additional local search operator, named *Replace-One-Depot*, has been defined in order to modify one single route at a time. In practice, this operator attempts to replace the final depot of a route r with an alternative depot in set E . 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 VND heuristic is restarted from the new current solution. The local search stops when the solution cannot be further improved by any of the local search operators. The order in which the local search operators are employed within the VND schema is shown in Table 1.

Table 1: Order of the local search operators inside the VND heuristic

N_λ	Local Search Operator	Type
N_1	<i>Or-Opt</i>	<i>Intra-route</i>
N_2	<i>Relocate</i>	<i>Intra-route</i>
N_3	<i>Exchange</i>	<i>Inter-route</i>
N_4	<i>Relocate</i>	<i>Inter-route</i>
N_5	<i>Cross-Exchange</i>	<i>Inter-route</i>
N_6	<i>Two-opt</i>	<i>Inter-route</i>
N_7	<i>Replace-One-Depot</i>	<i>Single-route modification</i>

Finally, a diversification mechanism is applied to escape from local optima and the ILS reiterates the VND heuristic starting from a new perturbed solution. A perturbation is used to partially destroy a percentage number of current routes from the current solution and to rebuild new routes following the GRASP constructive heuristic, described before. The whole ILS heuristic is repeated a number of times, until a stopping condition (maximum number of iterations denoted by parameter τ) is met and the best solution found so far is reported. The structure of the ILS heuristic is shown in Algorithm 2.

Algorithm 2: ILS heuristic

Initialize a w -vector;

Let x be the current solution and $f(x) = f_1(x) + (1 - w_1) \cdot f_2(x)$ its cost;

Let x^* be the best solution found so far and $f(x^*) = f_1(x^*) + (1 - w_1) \cdot f_2(x^*)$ its cost;

Set $x, x^* \leftarrow \emptyset$ and $f(x), f(x^*) \leftarrow \infty$;

Phase 1: Generation of an initial solution

$x \leftarrow \text{GRASP-Heuristic}()$;

repeat

Phase 2: Improvement by VND

 Select a set of neighbourhood structures $\mathcal{N}_\lambda, \lambda = 1 \dots 7$;

 Set $\lambda \leftarrow 1$;

while ($\lambda \leq 7$) **do**

$x' \leftarrow \mathcal{N}_\lambda(x)$;

if ($f(x') < f(x)$) **then**

 set $x \leftarrow x'$ and $\lambda \leftarrow 1$;

else

$\lambda \leftarrow \lambda + 1$;

end

end

if ($f(x) < f(x^*)$) **then**

 set $x^* \leftarrow x$;

end

Phase 3: Diversification stage

$x \leftarrow \text{Perturbation}(x)$;

until (*maximum number of iterations τ not reached*);

Report the best solution x^*

3.2 ILS PROMETHEE II to update the solution archive

After having populated the archive \mathcal{A} , a multi-criteria method, based on the well-known PROMETHEE II, is used to compare a set of solutions on the basis of objectives $f_1(x)$ and $f_2(x)$ described in Eq. (2). These objective functions represent the criteria that the PROMETHEE II uses to execute pairwise comparisons among the solutions in \mathcal{A} . The starting point of the method is to evaluate all pairwise differences between any couple of solutions on each criterion. In other words, the values of the objective functions associated to each solution are used as the basis on which this comparison is made. Using these pairwise differences, a degree of preference of an alternative over the other on a specific criterion is computed by adopting a *generalized preference function*.

The *generalized preference function* is used to transform the pairwise difference which can be expressed in any unit, into a normalized number between 0 and 1. This function represents the way in which a decision maker might perceive differences in scores of different solutions with respect to a single criterion. In this paper, a linear generalized preference function with

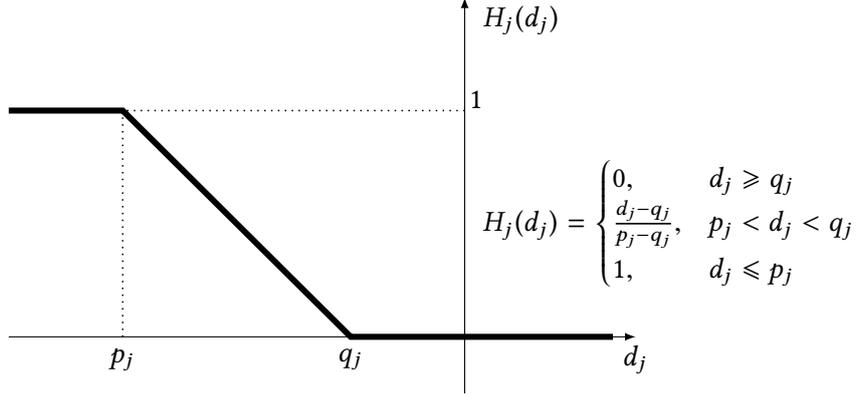


Figure 1: Linear generalized preference function with indifference region (for minimization of f_j)

indifference region (i.e., type 5 in [1]) is used, as shown in Figure 1 where the axes represent the pairwise difference (d_j) between a couple of alternative solutions on criterion j and the perceived difference in score ($H_j(d_j)$) of different alternative solutions with respect to a single criterion j . This function requires two parameters that need to be set by the decision maker: (i) q_j , being the minimal pairwise difference in score on a criterion for which the decision maker perceives a difference and (ii) p_j , that is the strict preference threshold. Since the MO-RCTVRP concerns the minimization of $f_1(x)$ and $f_2(x)$, both q_j and p_j are negative values. If the difference in scores of two alternative solutions on a criterion is greater than q_j , the decision maker is indifferent and the degree of preference is 0. In case the difference is lower than p_j , the degree of preference is 1. Different *generalized preference functions* can be selected by the decision maker and associated to each criterion (objective of the problem). However, for the sake of simplicity, in the remainder of this paper we used the same *generalized preference function* for both objectives $f_1(x)$ and $f_2(x)$ with $p_1 = p_2 = -100$ and $q_1 = q_2 = -1$ (see Section 4.2).

Starting from the perceived differences in score for each couple of alternative solutions on each criterion, the PROMETHEE II method evaluates and ranks these solutions, by including into the model the decision maker's preferences. The decision maker's *inter-criterion preference information* can be added in the form of criterion weights v_i assigned to each criterion i . In our case, since we consider two objective functions, we defined a coefficient v_1 for the travel cost criterion and a coefficient v_2 for the risk criterion, with $v_2 = 1 - v_1$. The decision maker's preferences are used to compare the solutions contained in \mathcal{A} , assigning a relative importance of cost over risk. In practice, the PROMETHEE II method assigns a total score (named net-flow ϕ) to each solution based on the difference between positive flow (how much the solution is preferred over the others) and negative flow (how much other solutions are preferred over this one). Therefore, by using the net-flow, it is possible to rank all the solutions in the archive.

After having associated a net-flow to each solution in \mathcal{A} , a new weight vector w_{new} is defined. This new vector is used to determine a new search direction in the objective function space. The w_{new} vector is computed as follows:

$$w_{new} = \frac{\sum_{i=1}^{\mathcal{A}-1} \phi_i \cdot w_i}{\sum_{i=1}^{\mathcal{A}-1} \phi_i} \quad (16)$$

In practice, inside Eq. (16), the net-flow ϕ_i associated to each solution i is used as a weight for the w_i vectors. In addition, the solution with the lowest net-flow in \mathcal{A} (in which we suppose that the solutions are ordered by decreasing values of ϕ) is excluded from the computation. In other words, the output of the PROMETHEE II method is used to determine the w_{new} vector and direct the search towards a promising area.

Thereafter, the ILS heuristic is reapplied to find a new solution x_{new} , by linearly combining the objective functions $f_1(x_{new})$ and $f_2(x_{new})$ using the weights contained in w_{new} . This new solution is added into \mathcal{A} if it is not a copy of a solution already contained in the archive. The solution with the lowest net-flow in the archive is removed and x_{new} is compared with the remaining solutions in the archive by using again the PROMETHEE II method. All net-flows associated to the solutions in the archive are updated efficiently by using the short-cut calculations described in [15].

For more details about PROMETHEE II methods, the reader is referred to [1, 7] and [15].

4 Computational experiments

In this section the experiments are described. The instances, used to test the PMOO-ILS meta-heuristic, are described in Section 4.1. The test instances are grouped in two sets: (i) existing benchmark instances for the multi-depot VRP problem and; (ii) realistic instances.

The solution approach has been coded in Java and the experiments have been performed on an Intel core i7-2760QM 2.40 GHz processor with 8 GB RAM. All the instances contained in both benchmark sets are solved during the experiments and the results are summarized in Section 4.2.

4.1 Test instances

We tested our solution approach on two different sets of instances. The first set, named SET L, contains existing benchmark instances that are well-known in the literature on VRP. More specifically, SET L is made by 33 multi-depot VRP instances designed by Cordeau et al. [4] and it is available at <http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances>. This set contains small, medium and large instances with a number of nodes between 48 and 360 and a number of depots varying from 2 up to 9.

An additional set, named SET P, groups 5 plausible instances that have been used to recreate study cases that adhere reality as close as possible. These instances contain up to 1035 customers such as retail stores, clothes shops, supermarkets, jewellery stores, all located in Belgium. Moreover, a total number of 4 depots, where the collected valuables can be deposited,

are considered. These realistic instances have been generated following the indications of a real CIT company operating in Belgium.

First, a database of customers has been generated containing 1035 different potential CIT customers, grouped in 5 categories, as shown in Table 2. For each customer, the publicly available geo-coordinates have been retrieved. In addition, to each customer a value of demand has been randomly assigned within the ranges ([lower-bounds (LB); upper-bounds (UB)]) reported in Table 2 and expressed in thousands of €. Moreover, the coordinates of four different depots, that are used on a daily basis by the CIT company, are considered. The vehicle, considered in our model, is initially located at one of these depots, that is randomly selected. In real-life, the vehicles used by the CIT company transport on average 80 containers in which a maximum of 5000 € is stored. For this reason, the maximum vehicle’s capacity has been set to 400 thousand €.

Table 2: Customers database for SET P

id	Category	Number	Demand [LB; UB]
1	Casinos	56	[10; 400]
2	Clothes shops	295	[5; 100]
3	Supermarkets	324	[5; 150]
4	Shopping-centres	48	[20; 400]
5	Jewellery stores	312	[5; 200]
Total		1035	

Five instances, presenting an increasing size of 100, 200, 500, 700 and 1039 nodes respectively, have been generated by including all the depots and a certain number of customers, randomly selected from the database. These medium and large size instances form a set of benchmark MO-RCTVRP instances that is made publicly available at the following website <http://antor.uantwerpen.be/Downloads/MORCTVRP>.

4.2 Results

The PMOO-ILS metaheuristic requires two different sets of parameters: (1) *decision maker’s parameters* and; (2) *metaheuristic parameters*.

Both the *generalized preference function* (GPF for short in Table 3) for each criterion and the decision maker’s preferences v , associated to the objective functions, represent the *decision maker’s parameters*, which are used to compare and rank alternative solutions inside the PROMETHEE II multi-criteria method. Since these parameters need to be defined by the decision maker, before the optimization process, the PMOO-ILS metaheuristic handles them as input parameters in the same way as the problem data. As mentioned before, the experiments have been performed by using the same *generalized preference function*, shown in Figure 1, for

both criteria, $f_1(x)$ and $f_2(x)$, with $p_1 = p_2 = -100$ and $q_1 = q_2 = -1$. Moreover, an equal importance of risk over travel cost from the point of view of the decision maker is imposed being $v = [0.5, 0.5]$.

The *metaheuristic parameters* consist of five parameters that need to be set and tuned in order to achieve the best results: (1) the maximum number of iterations I that the PMOO-ILS metaheuristic is repeated to compute a new solution and update the archive \mathcal{A} ; (2) the size of the archive \mathcal{A} ; (3) τ , that is the maximum number of times that the ILS heuristic is repeated to generate a solution; (4) the percentage number (denoted by letter γ) of routes contained in the current solution to be destroyed during the perturbation stage of the ILS heuristic; and (5) the parameter α used inside the GRASP constructive heuristic. In order to speed up the solution approach, a relative small archive has been used ($|\mathcal{A}| = 10$). As expected, higher values of I make the PMOO-ILS metaheuristic find better solutions at the expense of a longer computation time. In preliminary experiments, we observed that the PMOO-ILS metaheuristic converges quickly and that $I = 200$ offers a good compromise between runtime and solution quality (measured by the net-flow ϕ). Similarly, the higher the value of parameter τ , the better the solutions (presenting higher values of $f(x) = w_1 \cdot f_1(x) + (1 - w_1) \cdot f_2(x)$) that the ILS can generate. During the preliminary tests, we noticed that the ILS converges towards stable solutions for $\tau = 50$. Moreover, we set $\alpha = 3$ and $\gamma = 50\%$ inside the ILS heuristic, as this delivered good results in the pretests. In Table 3, the algorithm’s parameter settings, which were determined in a limited pilot study, are presented together with a short description of these parameters.

Table 3: Decision maker’s and PMOO-ILS metaheuristic’s parameters

Parameter	Description	Tuning
Decision maker’s parameters		
GPF	<i>Generalized preference function</i> associated to each criterion, used inside PROMETHEE II, to transform the pairwise difference between couples of alternative solutions, into a normalized number between 0 and 1	type 5 in [1] with $p_1 = p_2 = -100$ and $q_1 = q_2 = -1$
v	Relative importance of risk over travel cost from the point of view of the decision maker	[0.5,0.5]
PMOO-ILS metaheuristic’s parameters		
I	Maximum number of iterations of the PMOO-ILS metaheuristic	200
$ \mathcal{A} $	Size of the archive containing non-dominated MO-RCTVRP solutions	10
τ	Maximum number of times that the ILS heuristic is repeated	50
γ	Maximum percentage of routes to be removed from the current solution during the perturbation phase inside the ILS heuristic	50%
α	Size of the restricted candidate list inside the GRASP heuristic	3

As shown in Figure 2, the quality (measured by the net-flow ϕ) of the “best” solution in the archive improves during the execution of the algorithm as well as the net-flow of the “worst” solution in \mathcal{A} . As expected, the distance measured by the relative net-flow difference between

the “worst” and the “best” solutions in \mathcal{A} decreases over the running time. This means that the overall quality of the solutions contained in the archive improves during the execution of the PMOO-ILS metaheuristic.

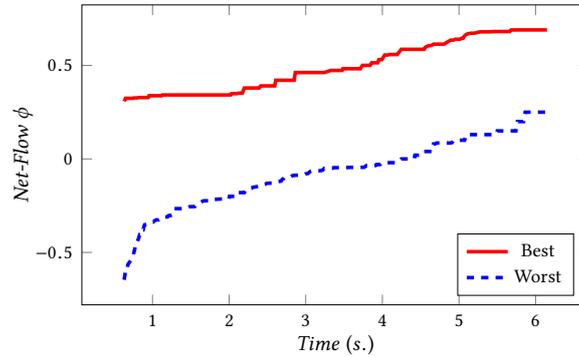


Figure 2: Values of net-flow associated to the “best” and “worst” solutions in the archive over the execution time (instance *pr09* in SET L)

As mentioned, the aim of the solution approach is not to generate a Pareto frontier or a wide set of non-dominated solutions, but to find a single solution with the highest net-flow (see e.g., the solutions highlighted with a square in Figure 3) that better suits the decision maker’s preferences. Nevertheless, Figure 3 shows that the solutions (represented by circles) contained in \mathcal{A} , at the end of the PMOO-ILS metaheuristic, represent a satisfactory and well spread Pareto frontier approximation that can be used by the decision maker during the decision making process. For example, the decision maker can use the solutions in the archive for further investigation/analysis or for possible comparisons with other alternative solutions.

5 Conclusion

In this paper, we presented a multi-objective problem, named MO-RCTVRP, with practical applications in the CIT sector to generate relatively secure vehicle routes and minimize the travel cost at the same time. We proposed a mathematical formulation of the problem based on the traditional capacitated vehicle routing problem where, beside the minimization of travel costs, a second objective associated to the minimization of the risk exposure to robberies is considered. The risk associated with a robbery is assumed to be proportional both to the amount of cash being carried and the distance covered by the vehicle carrying the cash.

A progressive multi-objective optimization that includes the decision maker’s preferences is developed to solve the MO-RCTVRP. This method can be used as a decision support tool by CIT companies in order to select the most appropriate vehicle route plans depending on the relative importance of travel costs over the risk exposure to robberies, expressed by the decision maker. The solution approach has been tested on a set of small, medium and large instances for the capacitated vehicle routing problem. Some realistic instances have also been solved considering

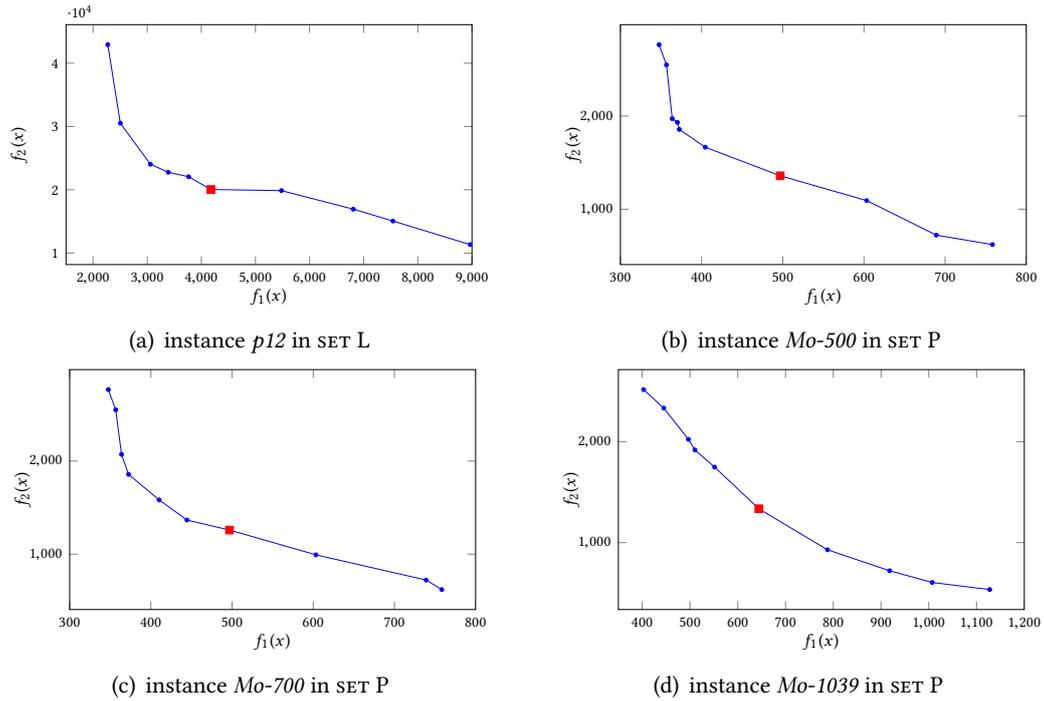


Figure 3: Travel cost $f_1(x)$ and risk $f_2(x)$ of the solutions contained inside archive \mathcal{A}

geo-locations of potential customers such as retail stores, jewellery shops, supermarkets and shopping malls that are located in Belgium. The PMOO-ILS metaheuristic is able to produce a limited set of non-dominated solutions presenting good quality in a short running time.

Future research can be aimed at adapting the model to other domains (e.g., chemical sector, transportation of dangerous goods) where risk and cost also represent two conflicting objectives to be minimized at the same time. Moreover, the MO-RCTVRP problem can be extended in several ways, taking into consideration real-life constraints such as route length restrictions, time windows and precedence relations.

Acknowledgement

This research is supported by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office (COMEX Project).

References

- [1] J. Brans and B. Mareschal. Promethee methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 163–186. Springer New York, 2005.

- [2] 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.
- [3] BSIA: British Security Industry Association. Cash and Valuables in Transit. 2014. <http://www.bsia.co.uk>, visited on 2014-10-29.
- [4] J.-F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997.
- [5] ESTA: European Security Transport Association. ESTA Guidance document. 2014. <http://www.esta-cash.eu>, visited on 2014-10-29.
- [6] T. Feo, J. Bard, and K. Venkatraman. A GRASP for a difficult single machine scheduling problem. *Computer and Operations Research*, 18(8):635–643, 1991.
- [7] J. Figueira, S. Greco, and M. Ehrgott. *Multiple criteria decision analysis: state of the art surveys*, volume 78. Springer, 2005.
- [8] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.
- [9] J. Krarup. The peripatetic salesman and some related unsolved problems. In *Combinatorial programming: methods and applications*, pages 173–178. Springer, 1975.
- [10] H. Lourenço, O. Martin, and T. Stützle. *Iterated Local Search: Framework and Applications*. Springer New York, 2010.
- [11] J. Michallet, C. Prins, L. Amodeo, F. Yalaoui, and G. Vitry. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations Research*, 41(0):196–207, 2014.
- [12] 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.
- [13] S. Ngueveu, C. Prins, and R. Wolfler Calvo. A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics*, 10:253–266, 2010.
- [14] W. Sattayaprasert, P. Taneerananon, S. Hanaoka, 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.
- [15] K. Sörensen and J. Springael. Progressive multi-objective optimization. *International Journal of Information Technology & Decision Making*, pages 1–20, 2013.
- [16] L. Talarico, K. Sörensen, and J. Springael. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. In *Verolog 2012: The first meeting of the EURO working Group on Vehicle Routing and Logistics Optimization*, Bologna - Italy, 2012.
- [17] L. Talarico, K. Sörensen, and J. Springael. The risk-constrained cash-in-transit vehicle routing problem with time window constraints. In A. Fink and M. J. Geiger, editors, *14th Workshop of the EURO Working Group “EU/ME: the metaheuristics community”*, pages 104–109, Hamburg - Germany, 2013.
- [18] L. Talarico, K. Sörensen, and J. Springael. The k-dissimilar vehicle routing problem. Working Papers 2013029, University of Antwerp, Faculty of Applied Economics, Dec. 2013.

- [19] P. Toth and D. Vigo. *The vehicle routing problem*. Siam, 2001.
- [20] R. Wolfer Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computer & Operation Research*, 30:1269–1287, 2003.
- [21] S. Yan, S. Wang, and M. Wu. A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem. *Computers & Industrial Engineering*, 63(2):464 – 473, 2012.