INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 00 (2015) 1–18
DOI: 10.1111/itor.12214

A biobjective decision model to increase security and reduce travel costs in the cash-in-transit sector

Luca Talarico, Kenneth Sörensen and Johan Springael

University of Antwerp, Operations Research Group ANT/OR, Prinsstraat 13, 2000 Antwerp, Belgium
E-mail: luca.talarico@uantwerpen.be [Talarico]; kenneth.sorensen@uantwerpen.be [Sörensen]; johan.springael@uantwerpen.be [Springael]

Received 30 May 2015; received in revised form 30 September 2015; accepted 20 September 2015

Abstract

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 for quantifying the exposure of a vehicle to the risk of being robbed along its route. In addition, the problem is subject to a traditional capacity constraint, which limits the maximum amount of valuables that can be transported inside the vehicle. This constraint might be imposed, for example, by insurance companies. A biobjective formulation, aimed at reducing both risk and travel costs, is proposed. The objectives are conflicting since higher risk exposures would allow a reduction of the travel costs incurred by a transportation firm to visit all its customers for the collection of cash. A mathematical model of the problem is described and solved using a progressive multiobjective metaheuristic. Multiobjective optimization and multicriteria decision making are integrated into a single metaheuristic algorithm. To simplify the decision process, the decision maker's preferences are embedded into the multiobjective algorithm. Realistic instances are generated considering the geographical coordinates of several customers (e.g., stores, banks, shopping centers) located in Belgium. The proposed solution approach is tuned and tested both on these realistic instances and standard benchmark instances for the capacitated VRP.

Keywords: metaheuristic; multiobjective optimization; logistics; 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, 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, jewelery 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, though the number of episodes, risk rates, and average losses are different from country to country. For

Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

detailed statistics about the CIT sector, the reader is referred to European Security Transport Association (2014) and British Security Industry Association (2014).

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. In addition, it should be noted that the risk of being robbed, not only affects the efficiency and security at an operative level, but it has a considerable impact on CIT company's operating costs.

For these reasons, routing plans in the CIT sector should both be safe and efficient. As a result, a CIT company ought to simultaneously deal with two critical issues : the minimization of the traveled 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 that involve, in general, several hundreds of customers who should be visited each day.

Despite the large attention that researchers have devoted to vehicle routing problems (VRPs) 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 Krarup (1975) and Ngueveu et al. (2010), 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 Wolfer Calvo and Cordone (2003), 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 Michallet et al. (2014), 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. (2012). 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 for generating more flexible routing strategies in order to reduce the risk of robbery. In Talarico et al. (2015b), an index of similarity, based on the number of identical edges common between alternative solutions, is presented. Furthermore, both a mathematical formulation, which 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 VRP.

In Talarico et al. (2015a), a variant of the well-known capacitated VRP is introduced to model the problem of routing vehicles in the CIT industry. In this problem, named Risk-Constrained VRP (RCVRP), 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 time/distance covered by the vehicle transporting the cash. This risk index is used for quantifying 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 Talarico et al. (2013), an extension of the RCVRP 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 contribution of this paper is fourfold:

1. A multiobjective version of the RCVRP, considering both risk and travel costs as two conflicting objectives that should be simultaneously minimized, is described and named *multiobjective RCVRP* (MO-RCVRP). A mathematical formulation of the MO-RCVRP is also defined.
2. A progressive multiobjective metaheuristic to solve the problem is developed and named PMOO-ILS (progressive multiobjective optimization with iterative local search). The PMOO-ILS combines the features of both multiobjective metaheuristic and multicriteria decision-making approach into a single optimization approach. The outcome of the algorithm is represented by a solution, selected from a restricted archive containing high-quality alternatives, which better fits the decision maker's preferences.
3. A set of realistic test instances (SET R) is generated based on the inputs provided by a real CIT company that operates in Belgium. These instances are made publicly available.
4. Both the benchmark instances for the traditional capacitated VRP and instances contained in SET R are used for testing the PMOO-ILS algorithm.

The remainder of the paper is organized as follows. Section 2 presents the problem description. After having introduced the index used for measuring the risk of being robbed, a mathematical formulation of the problem is provided. In Section 3, a metaheuristic to solve the MO-RCVRP is presented, based on a progressive multiobjective optimization. The results of the computational experiments, together with the test instances used for solving the problem, are described in Section 4. Finally, Section 5 concludes the paper presenting also some suggestions for future research.

2. Problem definition and mathematical formulation

The MO-RCVRP problem is defined on a network $\mathcal{G} = \{V, A\}$, with $A = V \times V$ the set of available arcs and $V = N \cup D$ the set of nodes. Set N contains n customers who have to be visited, while 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., $S \equiv E \equiv D$). For each arc, a nonnegative travel cost c_{ij} is defined. In addition, for each customer $i \in N$, a strictly positive amount of cash/valuables m_i, which has to be collected by the vehicle, is known. In our model we assume to have a single vehicle, initially located at a given initial depot $o \in S$, performs all the collection routes. In real-life applications, 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 risk of being robbed and travel costs 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. The vehicle capacity represents the maximum amount of valuables, expressed in any currency, that a vehicle, depending on its characteristics, is allowed to transport.

Assuming that a vehicle only picks up cash at customer's places, a risk index R^r_j can be defined for each customer j, visited along route r, as follows (cf. Talarico et al., 2013):

$$R^r_j = R^r_i + M^r_i \cdot c_{ij}, \tag{1}$$

where M^r_i 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^r_i is defined for each customer and is obtained by summing up the amount of cash picked up by the vehicle along r from the depot $s \in S$ 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. Additional parameters, not necessarily related to the arc length, can also be used for measuring the probability of an accident on a specific roadway segment given its characteristics such as lane width, number of lanes, etc. (cf. Sattayaprasert et al., 2008; Milovanović et al., 2012).

The risk index, defined in Equation (1), is a cumulative and increasing measure of the risk faced by the vehicle while it travels along its route. Therefore, the "global route risk" R^r_e, 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^r_i be equal to the cash carried by the vehicle when it leaves customer i along route r.
2. Let R^r_i 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^r_{ij} 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. As expected, this scenario allows for the lowest risk exposure at the expense of the travel costs that reach a maximum level. For this reason, depending also on the capacity of the vehicle, the index r, which 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. A better lower bound associated to the minimum number of routes can be obtained by solving a bin packing problem given the demand associated with the customers and maximum vehicle capacity.

Hereafter, an MIP formulation for the MO-RCVRP is presented. It should be noted that the goal of the MIP formulation is to clearly state the problem using a mathematical notation.

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

$$\min \; f_2(x) = \max_{\substack{r\in N\\ e\in E}} R^r_e \tag{3}$$

s.t.

$$\sum_{j\in N} x^1_{oj} = 1 \tag{4}$$

$$\sum_{s\in S}\sum_{j\in N} x^r_{sj} = \sum_{i\in N}\sum_{e\in E} x^r_{ie} \qquad \forall r \in N \tag{5}$$

$$\sum_{i\in N} x^r_{ih} \geq \sum_{j\in N} x^{r+1}_{hj} \qquad \forall r \in N \setminus \{n\}; \forall h \in D \tag{6}$$

$$\sum_{r\in N}\sum_{j\in V} x^r_{ij} = 1 \qquad \forall i \in N \tag{7}$$

$$\sum_{h\in V} x^r_{hj} - \sum_{k\in V} x^r_{jk} = 0 \qquad \forall j \in N; \forall r \in N \tag{8}$$

$$\sum_{i\in V}\sum_{j\in N} m_j\, x^r_{ij} \leq C \qquad \forall r \in N \tag{9}$$

$$M^r_s = 0 \qquad \forall r \in N; \forall s \in S \tag{10}$$

$$x^r_{ij} = 1 \Rightarrow M^r_i + m_j = M^r_j \qquad \forall (i, j) \in A; \forall r \in N \tag{11}$$

$$R^r_s = 0 \qquad \forall r \in N; \forall s \in S \tag{12}$$

$$x^r_{ij} = 1 \Rightarrow R^r_i + M^r_i\, c_{ij} = R^r_j \qquad \forall (i, j) \in A; \forall r \in N \tag{13}$$

$$M^r_i, R^r_i \geq 0 \qquad \forall i \in V; \forall r \in N \tag{14}$$

$$x^r_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A; \forall r \in N. \tag{15}$$

The objective function (2) aims to minimize the total travel costs, while objective function (3) attempts to minimize the highest value of global route risk faced by the vehicle over all its collection routes. Constraint (4) states that the first route ($r = 1$) starts at depot $o \in D$, where the vehicle is initially located. Constraint (5) shows 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) shows that route $r + 1$ cannot exist unless route r also exists (for $r > 1$). This constraint enforces route $r + 1$ to start from the same depot where route r ends. 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) and (11) are used for defining, in a recursive way, starting from depot $s \in S$, where the vehicle begins its route r, the cumulative demand (i.e., amount of valuables inside the vehicle) associated with 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) and (13) are used for measuring, in a recursive way, starting from depot $s \in S$, where the vehicle begins its route r, the global route risk faced by the vehicle upon its return to depot $e \in E$. Constraints (11) and (13) are expressed in a compact form, similar to the notation used within the VRP formulation in Lenstra et al. (1988). However, these constraints can also be rewritten using linear inequalities and a large constant $\mathcal{P}$ as follows:

$$M^r_j \geq M^r_i + m_j - \left(1 - x^r_{ij}\right) \cdot \mathcal{P} \quad \forall (i, j) \in A; \forall r \in N \tag{16}$$

$$R^r_j \geq R^r_i + M^r_i \cdot c_{ij} - \left(1 - x^r_{ij}\right) \cdot \mathcal{P} \quad \forall (i, j) \in A; \forall r \in N. \tag{17}$$

For more details about a sufficient choice for the constant $\mathcal{P}$, the reader is referred to Talarico et al. (2015a).

Finally, subtours are automatically prevented by Constraints (10)–(13), while Constraints (14) and (15) define the domains of the decision variables.

3. A metaheuristic for the MO-RCVRP

The MO-RCVRP problem, described in Equations (2)–(15), can be considered NP-hard, since it extends (in a multiobjective fashion) the traditional VRP that is known to be NP-hard (cf. Toth and Vigo, 2001). For this reason, metaheuristics present the only viable solution approaches to deal with such kind of problems in a reasonable time.

In the context of multiobjective optimization problems, the concept of optimality needs to be abandoned in favor of the notion of domination. A solution x dominates another solution y, denoted by $x \succ y$ (or interchangeably y is dominated by x), if and only if x is no worse than y with respect to all objectives and x is strictly better than y in at least one objective. In other words, using mathematical notation and assuming that the problem requires that all η objective functions are to be minimized, $x \succ y$ if $f_j(x) \leq f_j(y), \forall j = 1, \ldots, \eta$, and $\exists j = 1, \ldots, \eta | f_j(x) < f_j(y)$.

Considering this, the goal of a multiobjective solution approach is to find a set of nondominated solutions that form as a whole the so-called Pareto frontier (or Pareto frontier approximation in case the set of solutions is really large). From this Pareto frontier containing nondominated solutions, the decision maker is left to choose one according to his preferences. In case of a high number of nondominated solutions, the selection process may be a complex task for a decision maker.

Most of the multiobjective optimization techniques proposed in the literature do not cope with the problem of selecting a solution from Pareto frontier (cf. Respício et al., 2013; Dash and Kajiji, 2014; Rath et al., 2015). As a matter of fact, these approaches implicitly assume that the decision maker is able to autonomously use a multicriteria decision-making method for this purpose. A recent study, proposed by Sörensen and Springael (2014), introduces a novel technique named PMOO, which includes both multiobjective optimization and multicriteria decision making into a single metaheuristic algorithm. This is made possible by including the decision maker's preferences directly into the multiobjective process, instead of applying an *a posteriori* method to select a solution from previously defined Pareto frontier (approximation). The main benefit of integrating the decision maker's preferences into the optimization process is due to a significant simplification of the decision process. In fact, it is preferable that the decision maker chooses his favorite solution from a restricted set of high-quality alternatives, rather than from a huge amount of nondominated solutions.

In this paper, we propose a metaheuristic approach, named, which combines the PMOO with an ILS heuristic (cf. Lourenço et al., 2010). An archive $\mathcal{A}$, containing a relatively small number of high-quality nondominated solutions, is generated 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 for generating solutions to be included in the archive $\mathcal{A}$, while a multicriteria method, the PROMETHEE II, is adopted at each iteration of the algorithm to determine whether a newly generated solution is allowed to enter the archive (see

Section 3.2). The ILS heuristic embeds and combines the following :

1. A nearest neighborhood search heuristic with a greedy randomized selection mechanism (GRASP) used for finding an initial solution.
2. A variable neighborhood descent (VND) heuristic aimed at improving 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 nondominated solutions by performing the ILS heuristic a few times. In a second phase, both PROMETHEE II method and ILS heuristic are repeated to generate solutions that fit better the decision maker's preferences, until a stopping criterion is reached (maximum number of iterations I). During these iterations, the archive $\mathcal{A}$ is updated. Finally, the "best" solution according to the decision maker's preferences is returned at the end of the algorithm. A general overview of the PMOO-ILS metaheuristic for the MO-RCVRP is shown in Algorithm 1.

Algorithm 1: PMOO-ILS metaheuristic

```
Initialize both problem and heuristic parameters;
Let A be the solution archive initially empty;
Phase 1: Populate the archive
while (| 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 A;
    end
end
Phase 2: Update the archive
Compute the net flow φ of each solution in 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 ∉ A) then
        Add x_new into A;
        Remove the solution with the lowest φ from A;
        Update φ for each solution in A by applying PROMETHEE II;
    end
end
Return the solution from A presenting the highest φ.
```

3.1. ILS and solution generation

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

The objective functions $f_1(x)$ and $f_2(x)$ (described in Equations (2) and (3), respectively) are combined 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 risk function $f_2(x)$, respectively (i.e., $w_1 \cdot f_1(x) + w_2 \cdot f_2(x)$). 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 $w_1 + w_2 = 1$. As the ILS algorithm uses a linear combination of the two objectives, it can only find supported solutions. However, this is not considered an issue for our application. A GRASP constructive heuristic (for more details, see Feo et al., 1991) is used for building feasible routes in a greedy randomized fashion. As a result, at each iteration the nodes j, to be added in the current route r after node i, are randomly selected from a restricted list containing the first α nonvisited nodes ordered by an increasing value of $w_1 \cdot c_{ij} + (1 - w_1) \cdot R^r_j$. 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 neighborhood search (VNS) metaheuristic (cf. Hansen and Mladenović, 2001). In general, VNS algorithms use a sequence of nested neighborhoods, $\mathcal{N}_1, \ldots, \mathcal{N}_{\lambda_{max}}$, with an increasing size, that is, $\mathcal{N}_\lambda \subset \mathcal{N}_{\lambda+1}$.

The VND heuristic, which is used in this paper, is composed of six of the most common local search operators for VRPs (cf. Bräysy and Gendreau, 2005). In particular "relocate and or-opt" are "intraroute" operators that attempt to improve the order in which the customers, who are assigned to a vehicle, are visited. Conversely, "two-opt," "relocate," "exchange," and "cross-exchange," are "interroute" operators that change more than one route simultaneously. In practice, the interroute 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 a single route at a time. More specifically, this operator attempts to replace the final depot of a route r with an alternative depot in set E. Additional information concerning the operators used in this paper are reported in the Appendix. Each local search operator uses a first improvement descent strategy. The quality of the neighborhood is evaluated on the basis of a linear weighted sum, which combines risk and travel costs using a w-vector, as described for the GRASP constructive heuristic. 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 used within the VND scheme is shown in Table 1.

Finally, a diversification mechanism is applied to escape local optima and the ILS reiterates the VND heuristic starting from a new perturbed solution. A perturbation is used for partially destroying a percentage number of routes from the current solution and rebuild new routes following the GRASP constructive heuristic, described earlier. 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.

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

N_λ	Local search operator	Type
N_1	Or-Opt	Intraroute
N_2	Relocate	Intraroute
N_3	Exchange	Interroute
N_4	Relocate	Interroute
N_5	Cross-exchange	Interroute
N_6	Two-opt	Interroute
N_7	Replace-one-depot	Single-route modification

Algorithm 2: ILS heuristic

Initialize a w-vector;
Let x be the current solution and $f(x) = w_1 \cdot f_1(x) + (1 - w_1) \cdot f_2(x)$ its cost;
Let x^* be the best solution found so far and $f(x^*) = w_1 \cdot f_1(x^*) + (1 - w_1) \cdot f_2(x^*)$ its cost;
Set $x, x^* \leftarrow \emptyset$ and $f(x), f(x^*) \leftarrow \infty$;
Phase 1: Generation of an initial solution
$x \leftarrow$ GRASP-heuristic();
repeat
    Phase 2: Improvement by VND
    Select a set of neighbourhood structures $\mathcal{N}_\lambda$, $\lambda = 1 \ldots 7$;
    Set $\lambda \longleftarrow 1$;
    while $(\lambda \leqslant 7)$ **do**
        $x' \longleftarrow \mathcal{N}_\lambda(x)$;
        if $(f(x') < f(x))$ **then**
            set $x \longleftarrow x'$ and $\lambda \longleftarrow 1$;
        else
            $\lambda \longleftarrow \lambda + 1$;
        end
    end
    if $(f(x) < f(x^*))$ **then**
        set $x^* \longleftarrow x$;
    end
    Phase 3: Diversification stage
    $x \leftarrow$ Perturbation(x);
until (*maximum number of iterations* τ *is reached*);
Report the best solution x^*.

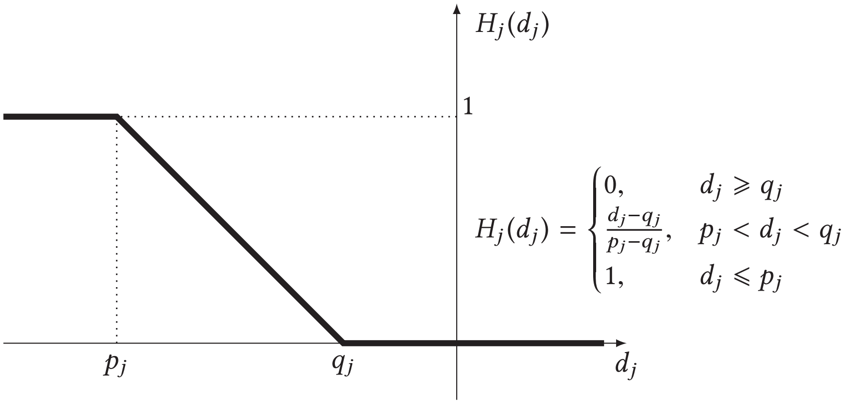


Fig. 1. Linear generalized preference function with indifference region for the minimization of f_j.

3.2. Promethee II and ILS to update the solution archive

After having populated the archive $\mathcal{A}$, a multicriteria method, based on the well-known PROMETHEE II, is used for comparing a set of solutions on the basis of the objectives $f_1(x)$ and $f_2(x)$—both described in Equation (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" (GPF).

The "GPF" is used for transforming 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 GPF with indifference region (i.e., type 5 in Brans and Mareschal, 2005) is used, as shown in Fig. 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 type of "GPF" requires two parameters that need to be set by the decision maker:

1. q_j being the minimal pairwise difference in score on a criterion for which the decision maker perceives a difference and
2. p_j, that is the strict preference threshold.

Since the MO-RCVRP 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 "GPFs" 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 "GPF" 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 the decision maker's preferences into the model . The decision maker's "intercriterion preference information" can be added in the form of criterion weights υ_i assigned to each criterion i. In our case, since we consider two objective functions, we defined a coefficient υ_1 for the travel cost criterion and a coefficient υ_2 for the risk criterion, with $\upsilon_2 = 1 - \upsilon_1$. The decision maker's preferences are used for comparing 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 x_i based on the difference between positive flow ϕ^+ (how much x_i is preferred over the other solutions) and negative flow ϕ^- (how much other solutions are preferred over x_i) as shown in Equations (18)–(20). Therefore, using the net flow, it is possible to rank all the solutions in the archive.

$$\phi^+(x_i) = \frac{1}{|\mathcal{A}| - 1} \cdot \sum_{j=1}^{2} \sum_{l=1}^{|\mathcal{A}|} \upsilon_j \cdot H_j(f_j(x_i) - f_j(x_l)) \tag{18}$$

$$\phi^-(x_i) = \frac{1}{|\mathcal{A}| - 1} \cdot \sum_{j=1}^{2} \sum_{l=1}^{|\mathcal{A}|} \upsilon_j \cdot H_j(f_j(x_l) - f_j(x_i)) \tag{19}$$

$$\phi(x_i) = \phi^+(x_i) - \phi^-(x_i) = \frac{1}{|\mathcal{A}| - 1} \cdot \sum_{j=1}^{2} \sum_{l=1}^{|\mathcal{A}|} \upsilon_j \cdot [H_j(f_j(x_i) - f_j(x_l)) - H_j(f_j(x_l) - f_j(x_i))]. \tag{20}$$

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 for determining a new search direction in the objective function space. The w_{new} vector is computed as follows:

$$w_{new} = \frac{\sum_{i=1}^{\mathcal{A}-1} \phi_i \cdot w_i}{\sum_{i=1}^{\mathcal{A}-1} \phi_i}. \tag{21}$$

In practice, in Equation (21), 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 for determining the w_{new} vector and direct the search toward 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 is not a copy of a solution already contained in the archive. The solution with the lowest net flow is removed from $\mathcal{A}$ and x_{new} is compared to the remaining solutions in the archive using again the PROMETHEE II method. All net flows associated to the solutions in the archive are updated efficiently by using the shortcut calculations described in Sörensen and Springael (2014).

For more details about PROMETHEE II methods, the reader is referred to Brans and Mareschal (2005), Figueira et al. (2005), and Sörensen and Springael (2014).

4. Computational experiments

In this section, the results of the experiments are shown. The instances, used for testing the PMOO-ILS metaheuristic, are described in Section 4.1. The test instances are grouped in the following two sets:

1. existing benchmark instances for the multidepot VRP (named SET L) and
2. realistic instances (named SET R).

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 Lis made by 33 multidepot VRP instances designed by Cordeau et al. (1997). These instances are available at http://neo.lcc.uma.es/vrp/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 R, groups five realistic instances that have been used for recreating study cases that adhere to reality as close as possible. These instances contain up to 1035 customers such as retail stores, clothes shops, supermarkets, jewelery stores—all located in Belgium. Moreover, a total number of four 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 five categories, as shown in Table 2. For each customer, the publicly available geocoordinates have been retrieved. In addition, to each customer a value of demand has been randomly assigned within the ranges (lower bounds [LBs]; upper bounds [UBs]) that are reported in Table 2 and expressed in thousands of €. Moreover, the coordinates of four different depots, which are used on a daily basis by the CIT company, are utilized. The vehicle, considered in our model, is initially located at one of these depots, which is randomly selected. In real life, the vehicles used by the CIT company transport on average 80 containers in which a maximum of €5000 can be stored. As a result, the maximum vehicle's capacity has been set to €400,000.

Table 2
Customers database for SET R

id	Category	Number	Demand [LB; UB]
1	Casinos	56	[10; 400]
2	Clothes shops	295	[5; 100]
3	Supermarkets	324	[5; 150]
4	Shopping centers	48	[20; 400]
5	Jewelry 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 instances form a set of benchmark MO-RCVRP instances, which is made publicly available at the following website: http://antor.uantwerpen.be/Downloads/MORCVRP.

4.2. Results

The PMOO-ILS metaheuristic requires two different sets of parameters:

1. decision maker's parameters and;
2. metaheuristic parameters.

Both GPF (Table 3) for each criterion and decision maker's preferences υ, associated to the objective functions, represent the "decision maker's parameters," which are used for comparing and

Table 3
Decision maker's and PMOO-ILS metaheuristic's parameters

Parameter	Description	Tuning
Decision maker's parameters		
GPF	"Generalized preference function" 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 Brans and Mareschal (2005) with $p_1 = p_2 = -100$ and $q_1 = q_2 = -1$
υ	Relative importance of risk over travel costs 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
$\lvert\mathcal{A}\rvert$	Size of the archive containing nondominated MO-RCVRP 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

rank alternative solutions inside the PROMETHEE II multicriteria 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 above, the experiments have been performed using the same "GPF," shown in Fig. 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 costs from the point of view of the decision maker is imposed, being $\upsilon = [0.5, 0.5]$.

The "metaheuristic parameters" consist of five parameters that need to be set and tuned in order to achieve the following best results:

1. The maximum number of iterations I that the PMOO-ILS metaheuristic is repeated to update the archive $\mathcal{A}$.
2. The size of the archive $\mathcal{A}$.
3. τ, which 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.
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 $I = 200$ offers a good compromise between runtime and solution quality (measured by its 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 noted that the ILS converges toward 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.

As shown in Fig. 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 "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.

As mentioned, the aim of the solution approach is not to generate Pareto frontier or a wide set of nondominated solutions, but to find a single solution with the highest net flow (see, e.g., the solutions represented by a square in Fig. 3) that better suits the decision maker's preferences. Nevertheless, Fig. 3 shows that the nondominated 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.

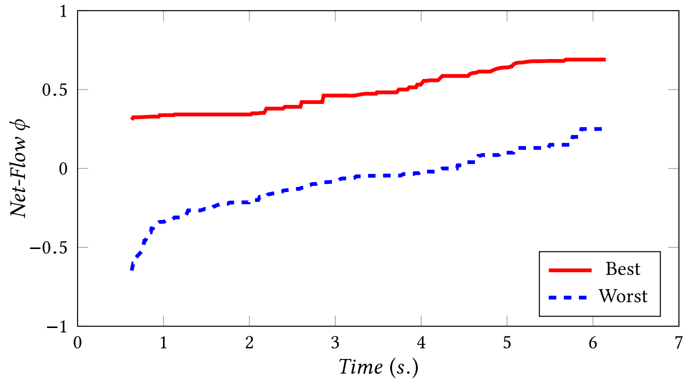


Fig. 2. Values of net flow associated to the "best" and "worst" solutions in the archive over the execution time (e.g., *pr09* in SET L).

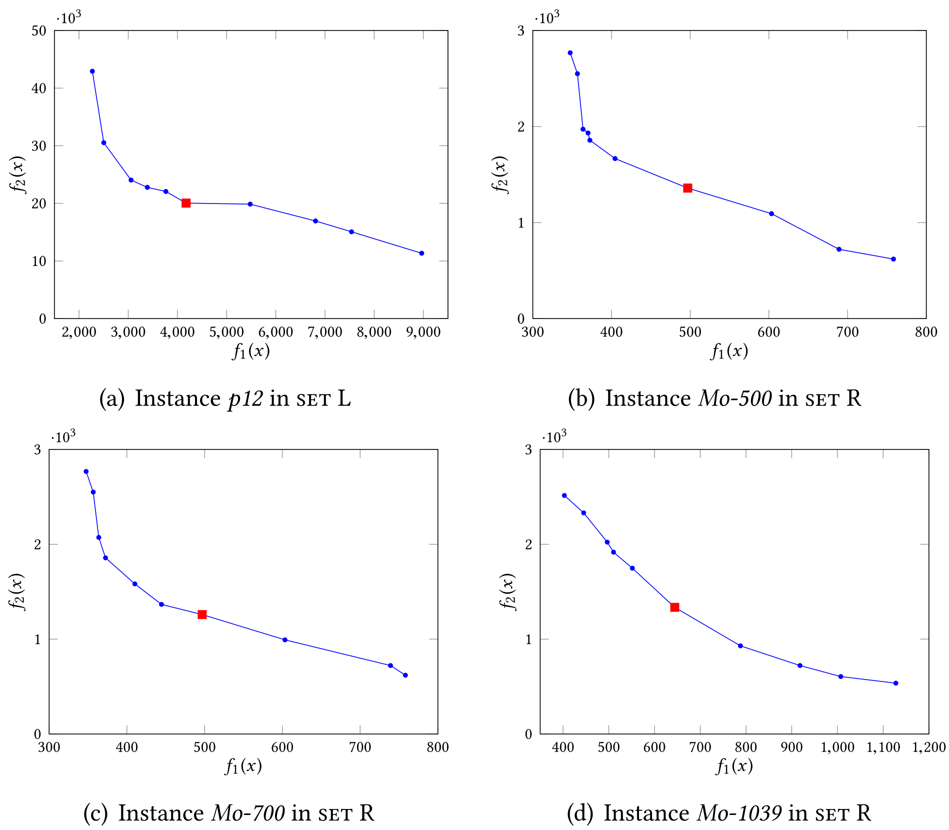


Fig. 3. Travel costs $f_1(x)$ and risk $f_2(x)$ associated to all solutions contained in $\mathcal{A}$.

5. Conclusions

In this paper, we presented a multiobjective problem, named MO-RCVRP, with practical applications in the CIT sector. The goal is to simultaneously generate relatively secure vehicle routes and minimize the total travel cost. We proposed a mathematical formulation of the problem based on the traditional capacitated VRP 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 distance covered by the vehicle carrying the cash.

A progressive multiobjective optimization that includes the decision maker's preferences is developed to solve the MO-RCVRP. 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 VRP. Some realistic instances have also been solved considering geolocations of potential customers such as retail stores, jewelry shops, supermarkets, and shopping malls that are located in Belgium. The PMOO-ILS metaheuristic is able to produce a limited set of nondominated 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 simultaneously. Moreover, the MO-RCVRP problem can be extended in several ways, taking into consideration real-life constraints such as route length restrictions, time windows, and precedence relations.

Acknowledgment

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

References

Brans, J., Mareschal, B., 2005. Promethee methods. In Figueira, J., Greco, S., Ehrgott, M. (eds) *Multiple Criteria Decision Analysis: State of the Art Surveys*, Vol. 78, *International Series in Operations Research & Management Science*. Springer, New York, pp. 163–186.

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

British Security Industry Association (BSIA), 2014. Cash and valuables in transit. Available at http://www.bsia.co.uk (accessed 29 October 2014).

Cordeau, J.-F., Gendreau, M., Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30, 2, 105–119.

Dash, G.H., Kajiji, N., 2014. On multiobjective combinatorial optimization and dynamic interim hedging of efficient portfolios. *International Transactions in Operational Research* 21, 6, 899–918.

European Security Transport Association (ESTA), 2014. ESTA guidance document. http://www.esta-cash.eu (accessed 29 October 2014).

Feo, T., Bard, J., Venkatraman, K., 1991. A *GRASP* for a difficult single machine scheduling problem. *Computer and Operations Research* 18, 8, 635–643.

Figueira, J., Greco, S., Ehrgott, M., 2005. *Multiple Criteria Decision Analysis: State of the Art Surveys*, Vol. 78. Springer Science & Business Media.

Hansen, P., Mladenović, N., 2001. Variable neighborhood search: principles and applications. *European Journal of Operational Research* 130, 3, 449–467.

Krarup, J., 1975. The peripatetic salesman and some related unsolved problems. In Roy, B. (ed.) *Combinatorial Programming: Methods and Applications*. Springer, Dordrecht, pp. 173–178.

Lenstra, J., Desroches, M., Savelbergh, M., Soumis, F., 1988. *Vehicle Routing With Time Windows: Optimization and Approximation*. Elsevier Science Publishers, North-Holland.

Lourenço, H., Martin, O., Stützle, T., 2010. Iterated local search: framework and applications. In Gendreau, M., Potvin, J. (eds) *Handbook of Metaheuristics*. Springer, New York, pp. 363–397.

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.

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., Prins, C., Wolfler Calvo, R., 2010. A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics* 10, 253–266.

Rath, S., Gendreau, M., Gutjahr, W.J., 2015. Bi-objective stochastic programming models for determining depot locations in disaster relief operations. *International Transactions in Operational Research*, doi: 10.1111/itor.12163

Respício, A., Moz, M., Vaz Pato, M., 2013. Enhanced genetic algorithms for a bi-objective bus driver rostering problem: a computational study. *International Transactions in Operational Research* 20, 4, 443–470.

Sattayaprasert, W., Taneerananon, P., Hanaoka, S., Pradhananga, R., 2008. 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.

Sörensen, K., Springael, J., 2014. Progressive multi-objective optimization. *International Journal of Information Technology & Decision Making* 13, 5, 917–936.

Talarico, L., Sörensen, K., Springael, J., 2013. The risk-constrained cash-in-transit vehicle routing problem with time window constraints. In Fink, A., Geiger, M.J. (eds) 14th Workshop of the EURO Working Group "EU/ME: the Metaheuristics Community," Hamburg, Germany, pp. 104–109.

Talarico, L., Sörensen, K., Springael, J., 2015a. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *European Journal of Operational Research* 244, 2, 457–470.

Talarico, L., Sörensen, K., Springael, J., 2015b. The *k*-dissimilar vehicle routing problem. *European Journal of Operational Research* 244, 1, 129–140.

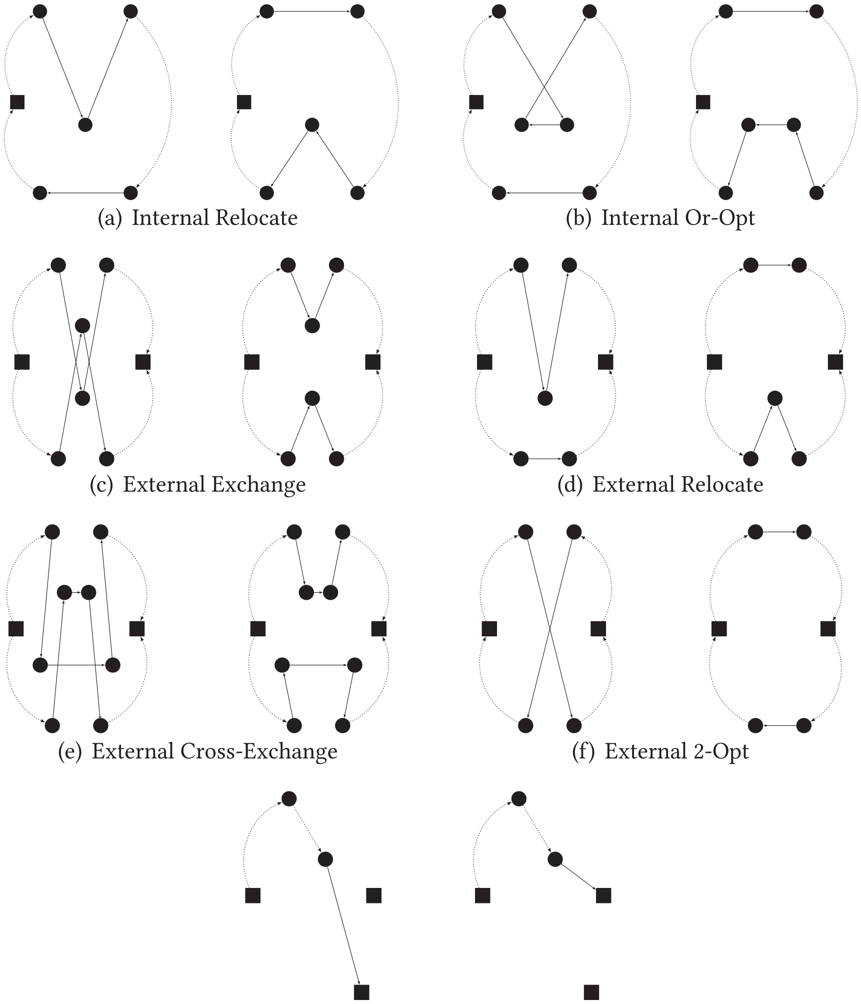
Toth, P., Vigo, D., 2001. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Wolfer Calvo, R., Cordone, R., 2003. A heuristic approach to the overnight security service problem. *Computer & Operations Research* 30, 1269–1287.

Yan, S., Wang, S., Wu, M., 2012. A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem. *Computers & Industrial Engineering* 63, 2, 464–473.

Appendix: Local search operators used inside the VND heuristic

Here, we report some illustrative examples of the intraroute and interroute local search operators that are used inside the VND heuristic. In Fig. A1, a single-route modification that is performed by the "replace-one-depot" operator is also reported. Each image shows a solution before (left side) and after (right side) the application of the corresponding local search operator. Depots are



(a) Internal Relocate (b) Internal Or-Opt

(c) External Exchange (d) External Relocate

(e) External Cross-Exchange (f) External 2-Opt

(g) Replace-One-Depot

Fig. A1. Local search operators used in the VND heuristic.

represented by black squares while black circles denote the customers. The length of the edges is proportional to their cost.