

DEPARTMENT OF ENGINEERING MANAGEMENT

**A Large Neighbourhood Metaheuristic for the Risk-constrained  
Cash-in-Transit Vehicle Routing Problem**

**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](http://www.uantwerpen.be)

# **FACULTY OF APPLIED ECONOMICS**

DEPARTMENT OF ENGINEERING MANAGEMENT

## **A Large Neighbourhood Metaheuristic for the Risk-constrained Cash-in-Transit Vehicle Routing Problem**

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

RESEARCH PAPER 2014-024  
OCTOBER 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](mailto:joeri.nys@uantwerpen.be)

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

**D/2014/1169/024**

# A Large Neighbourhood Metaheuristic for the Risk-constrained Cash-in-Transit Vehicle Routing Problem

L. Talarico, K. Sörensen, J. Springael

ANT/OR – University of Antwerp Operations Research group,

University of Antwerp, Faculty of Applied Economics,

Prinsstraat 13, 2000 Antwerp, Belgium

October 2014

In this paper, we propose a new metaheuristic to solve the Risk constrained Cash-in-Transit Vehicle Routing Problem (RCTVRP). The RCTVRP is a variant of the well-known capacitated vehicle routing problem and models the problem of routing vehicles in the cash-in-transit sector. In the RCTVRP, the risk associated with a robbery represents a critical aspect that is treated as a limiting factor instead of the vehicle capacity which is typical of capacitated vehicle routing problems. The risk of being robbed is assumed to be proportional both to the amount of cash being transported and the time/distance covered by the vehicle carrying the cash. The maximum vehicle exposure to risk limited by a certain risk threshold.

A new metaheuristic, called aLNS (Ant colony heuristic with Large Neighbourhood Search), is described. The aLNS metaheuristic combines the ant colony heuristic for the travelling salesman problem and a large neighbourhood search heuristic within an iterated local search heuristic framework.

A new library of RCTVRP instances with known optimal solutions is proposed, and split in two sets named SET O and SET S respectively. The aLNS algorithm is extensively tested on small, medium and large benchmark instances and compared with all existing solution approaches for the RCTVRP problem.

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

## 1 Introduction

Vehicle routing problems (VRP) have been widely studied by a large number of researchers during the last years. Different variants of the problem have been developed to cope with many real life applications (e.g. waste collection, freight distribution,

dangerous materials transportation). An interesting field of research, which has not received much attention so far, concerns the issue of *security* during the transportation of cash/valuables between banks, large retailers, ATMs, jewellers, casinos, etc. The cash-in-transit (CIT for short) sector refers to the physical transfer of banknotes, coins and items of value from one location to another by specialized transportation companies. In general, cash and valuables are moved from customer locations (e.g. shopping centres, retail stores and other premises holding large amounts of cash) to one or more depots/banks by the means of armoured vehicles. Due to the nature of the transported goods, CIT companies are constantly exposed to real risks such as robbery and armed assault. The attacks carried on by crime organizations may also cause serious injuries to retail staff, customers and CIT personnel and so forth.

During the last decades, several efforts have been made to drive the attacks to CIT vehicles down by improving the vulnerability of the vehicles i.e. reducing the probability that a robbery succeeds, given that it occurs. Substantial investments have been made by CIT companies in vehicles, equipment, infrastructure and technology. Although the implementation of such measures (e.g. armoured vehicle, weapons on board, on-board drop safes and interlocking doors, active vehicle tracking) can serve as a deterrent, the risk reduction can be also achieved by implementing other preventative measures aimed at reducing the occurrence of a robbery and/or at mitigating the consequences of a CIT attack (Erasmus, 2012; Smith and Louis, 2010). In this paper, we attempt to reduce the risk and its components, in particular occurrence and consequences, by generating safe routes directly in the routes planning phase.

An approach suggested in the literature is to reduce the risk of being attacked by building routes that are “unpredictable” for criminals. In so-called “peripatetic” routing problems (Wolfer Calvo and Cordone, 2003; Krarup, 1975; Ngueveu et al., 2010), customers are visited several times, but the use of the same road segment twice is explicitly forbidden.

In a previous work of Talarico et al. (2013), the Risk-constrained Cash-in-Transit Vehicle Routing Problem has been introduced and named for short RCTVRP. The RCTVRP is a variant of the VRP problem, where the total risk, that any vehicle may incur during its operations, is limited by a pre-specified *risk threshold*. Using a particular kind of risk index, which is proportional both to the amount of cash being carried (consequence) and the time/distance covered by the vehicle carrying the cash (occurrence), we can measure the global risk faced by each vehicle along its route. The aim of the RCTVRP is to determine  $K$  routes, where a route is a tour that begins at the depot, traverses a subset of the customers in a specified sequence and returns to the depot. Each customer must be assigned to exactly one of the  $K$  routes. The total risk for each route must not exceed the risk threshold  $T$ . The routes should be chosen to minimize total travel cost. In Talarico et al. (2013), seven different metaheuristics to solve the problem have also been proposed. Computational results show that the metaheuristics named p-TLK and m-CWg obtain the best performances (in term of solution quality and robustness) while the heuristic named m-NNg is the worst.

The p-TLK metaheuristic initially relaxes the risk constraint, effectively creating a giant tour. By using a variant of the well known Lin-Kerningham heuristic, implemented in Helsgaun (2000), and subjecting this tour to a variant of the splitting procedure, described in Prins (2004), an initial feasible solution for the RCTVRP is obtained. The m-CWg metaheuristic determines an initial solution by applying a variant of the well known Clarke and Wright heuristic while the m-NNg uses a constructive heuristic based on a nearest neighbour heuristic combined with a greedy randomized selection mechanism of unvisited nodes. In all these metaheuristics the initial solutions are then subjected to an improvement stage to find local optima.

The idea followed in this paper is to generate initial solutions for the RCTVRP that balance both the solution quality (that is characteristic of p-TLK and m-CWg) and the solution variability (that is characteristic of m-NNg), making the diversification and intensification stages more effective. For this purpose, a new metaheuristic (aLNS for short) is developed. The aLNS algorithm uses a natural inspired heuristic, the ant colony optimization described in Dorigo and Gambardella (1996), to generate an initial solution for the RCTVRP. The ant colony optimization outperforms other nature-inspired algorithms such as simulated annealing and evolutionary algorithms (Dorigo and Gambardella, 1997) and is able to generate initial solutions for the RCTVRP with a better quality than the m-NNg (but worse than the m-CWg and p-TLK), and with a higher variability than the m-CWg and p-TLK heuristics (but lower than the m-NNg heuristic). The initial solution for the RCTVRP is then improved by an effective large neighbourhood search (LNS) heuristic. Both the ant colony and the LNS heuristics are embedded into an iterated local search framework (Lourenço et al., 2010) that uses two diversification mechanisms to escape from local optima and explore different areas of the search space (see Section 3 for more details).

The aLNS metaheuristic has been tested and compared with the existing metaheuristics for the RCTVRP by using the benchmark instances (contained in SET R and SET V) described in Talarico et al. (2013). Moreover, an additional library of medium and large instances, with known optimal solutions, is proposed in this paper and used in the computational experiments. The aLNS metaheuristic, developed in this paper, outperforms the existing algorithms used to solve the RCTVRP problem, considering all the benchmark instances.

The main contributions of this paper are twofold. (1) An effective metaheuristic for the RCTVRP has been developed; (2) A new set of benchmark instances, with known optimal solutions, is presented and used to test the solution approaches for the RCTVRP.

The remainder of the paper is organized as follows. In Section 2, the literature on vehicle routing in risk-prone situations is surveyed, and the concept of *risk constraint* is introduced. In Section 3, the different components of the solution approach are described. In Section 4, the algorithm is tested and computational results are reported. Section 5 concludes the paper and includes some suggestions for future research.

## 2 Risk constraint

The concept of risk has received limited attention in the context of vehicle routing problems. An early contribution to define the risk on a path from an origin to a destination node, is due to Pijawka et al. (1985). In their work, the concept of “vulnerability assessment” is introduced and used to evaluate transportation risks and capabilities to respond and mitigate the consequence of an unwanted event. According to the Center for Chemical Process Safety (1995), the risk can be defined as a cardinal measure of potential economic loss, human injury or environmental damages in terms of both incident probability and magnitude of the loss, injury, or damage.

In Russo and Rindone (2011), risk is defined in terms of three main components: (1) *occurrence* of an event in term of probability/frequency of a specific unwanted event happening; (2) *vulnerability*, that is the resistance that objects to be protected exhibit, when the event occurs; (3) *exposure* (or possible consequences) that quantifies an equivalent homogeneous weighted value of people, goods and infrastructure affected during and after the event. Depending on the specific application domains, risk has to be assessed in all its components (occurrence, vulnerability and exposure) and a numerical value needs to be determined to express a cardinal measure of safety and security levels. The higher the risk, the lower the safety and security levels.

The general concept of risk can be adapted to the CIT sector to support specific optimization applications aimed at generating safe vehicle routes. Some preliminary remarks need to be done. First of all, the concept of *vulnerability* depends on several factors such as: (1) the modus operandi of the criminals, that is not under the direct control of the CIT company; (2) the type of vehicle (e.g. heavy armoured, light armoured or not armoured), the weapons on board, the preparedness of the crew, that are all under the control of the CIT companies, even though they can be modified only in the medium-long term. Since all these factors are either not under the control of the CIT companies or can be modified only in the medium-long term, for short-term operational decisions concerning the route planning, the vulnerability is considered as a fixed parameter and thus neglected in the formula of the risk described later on.

Moreover, the concept of *exposure* can be distinguished between: (1) *foreseeable consequences* of a robbery that are related to the amount of cash/valuables being transported. These consequences can be mitigated by CIT companies by selecting appropriate routes; (2) *unforeseeable consequences* of a robbery related to the criminal activity itself, involving significant additional costs (e.g. damaged vehicles and/or equipment, costs of policing, ambulance and hospital treatments, the opportunity cost as a result of inactive CIT personnel). Since the *unforeseeable consequences* are not under the direct control of the CIT companies and because they cannot be quantified a priori, only the *foreseeable consequences* of a robbery are considered in the formula of the risk for CIT applications. For these reasons, the concept of risk, proposed in Russo and Rindone (2011), can be simplified as follows:

$$R = P \cdot M \quad (1)$$

where  $P$  is the probability that a robbery occurs and  $M$  is the magnitude defined as a cardinal measure associated to the consequences of a robbery. As mentioned before, the probability of being robbed is directly related to the time/distance that the vehicle spends/travels along its route, being a potential target for criminals. Therefore, the time/ distance covered by the vehicle can be used as an approximation for  $P$ , while  $M$  is approximated by the amount of cash carried inside the vehicle that will be lost in case of robbery. Considering Eq.(1), two types of measures for risk reduction may be adopted by designing safe routes: (i) *prevention*, that consists in reducing the level of  $P$ ; (ii) *protection*, that consists in reducing the level of  $M$ . In this paper, both measures are addressed.

Let  $m_i$  be the amount of cash that is collected by the vehicle at customer  $i$ . It is assumed that a vehicle only picks up cash along its route and only deposits these valuables at the end of its route, at a depot. Under this assumption, a risk index  $R_i^r$  can be defined for each node  $i$  in route  $r$  as follows. Let  $M_i^r$  be the total amount of money on board of the vehicle when it leaves node  $i$  along  $r$ . Suppose also that route  $r$  contains arc  $(i, j)$  whose length is  $d_{ij}$ . The risk index for node  $j$  can then be computed as follows (see Talarico et al. (2013) for more details).

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

Since the risk index in Eq. (2) is a cumulative measure of the risk incurred by the vehicle while it travels along its route, the *global route risk* of route  $r$ , denoted by  $GR^r$ , is the risk incurred by the vehicle upon its return to the depot. In the RCTVRP, the global route risk of every route is limited to a certain maximum value, named the *risk threshold*  $T$ . This value represents the maximum risk level that CIT companies accept for the risk faced by any vehicle along its route during which valuables are collected.

The example in Figure 1 may clarify the calculation of the global route risk. A vehicle is supposed to travel along route  $r$ , where nodes  $A$ ,  $B$ , and  $C$  are visited before returning to depot 0. The risk index at node  $A$  is zero because the vehicle is empty along the arc  $(0, A)$  ( $R_A^r = 0 \cdot 7 = 0$ ). At node  $A$ , the vehicle collects one unit of money and continues to customer  $B$ . The risk index at node  $B$  is therefore equal to  $R_B^r = R_A^r + M_A \cdot d_{AB} = 1 \cdot 8 = 8$ . Similarly, the risk index at node  $C$  is equal to  $R_C^r = R_B^r + M_B \cdot d_{BC} = 8 + (1 + 3) \cdot 9 = 44$ . The global route risk is equal to the risk incurred by the vehicle upon its return to the depot:  $GR^r = R_C^r + M_C \cdot d_{C0} = 89$ . Route  $r = (0, A, B, C, 0)$  is feasible in the RCTVRP if  $GR^r \leq T$ . The global route risk of a route  $r$  depends on the order in which nodes are visited and generally it is not the same as the global route risk of the reversed route  $\bar{r}$ . For this reason, it is possible for a route to be feasible if covered in one direction and infeasible in the other one.

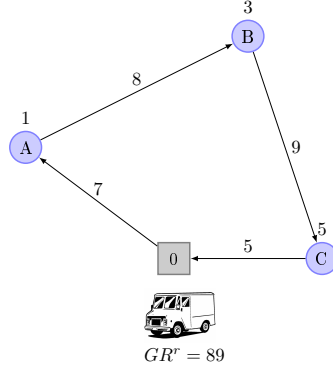


Figure 1: Illustrative example. The numbers on the arcs represent the travel times between nodes, while the values on the nodes denote the cash to be picked up by the vehicle at each customer place

### 3 aLNS metaheuristic

The aLNS metaheuristic, described in this section, uses the ant colony heuristic for the travelling salesman problem introduced by Dorigo and Gambardella (1996) to generate an initial solution for the RCTVRP (see Section 3.1). The risk constraint is initially relaxed and, since no capacity constraint is imposed on the vehicle (or any other constraint that restricts the route length), a giant TSP (Travelling Salesman Problem) tour is obtained. Then, the risk constraint is considered and feasible routes, presenting a global route risk not greater than the risk threshold  $T$ , are generated by subjecting the giant TSP tour to a variant of the splitting procedure, described in Prins (2004).

The solution obtained after the splitting heuristic serves as input for the large neighbourhood search heuristic (see Section 3.2), which is composed of seven of the most common local search operators for vehicle routing problems, modified for the RCTVRP. *Two-opt*, *Or-opt*, and *Relocate* are *intra-route* operators, that attempt to improve a single route. *Inter-route* operators change more than one route simultaneously. Our algorithm implements *Two-opt*, *Relocate*, *Exchange*, and *Cross-Exchange*.

The intensification stage stops when the current solution cannot be further improved by any of the local search operators and a local optimum is reached. In order to examine unexplored areas of the solution space and escape from local optima, two diversification mechanisms are employed (see Section 3.3). The first one perturbs the current solution by partially destroying and rebuilding a restricted number of routes contained in it. If a maximum number of iterations without having obtained any improvement of the current solution is reached, the second diversification mechanism is used. It consists in generating a new initial solution for the RCTVRP by using the ant colony optimization followed by the splitting procedure.

The overall schema of the aLNS metaheuristic is shown in Algorithm 1.



---

**Algorithm 1:** aLNS metaheuristic pseudo-code

---

```
Let  $x$  be the current solution and  $f(x)$  its cost;
Let  $x^*$  be the best solution found so far and  $f(x^*)$  its cost;
Set  $x, x^* \leftarrow \emptyset$  and  $f(x), f(x^*) \leftarrow \infty$ ;
Phase 1: Generation of an initial solution
Generate a giant TSP tour using the ant colony heuristic;
 $x \leftarrow$  split the giant TSP tour in feasible routes;
repeat
  Phase 2: Intensification stage
  Select a set of neighbourhood structures  $\mathcal{N}_l, l = 1 \dots l_{max}$ ;
  Set  $l \leftarrow 1$ ;
  repeat
     $x' \leftarrow \mathcal{N}_l(x)$ ;
    if ( $f(x') < f(x)$ ) then
      | set  $x \leftarrow x'$  and  $l \leftarrow 1$ 
    else
      |  $l \leftarrow l + 1$ ;
  until ( $l = l_{max}$ );
  if ( $f(x) < f(x^*)$ ) then
    | set  $x^* \leftarrow x$ ;
  update number of iterations without improvement;
  Phase 3: Diversification stage
  if (max number of iterations without improvement not reached) then
    |  $x \leftarrow$  Perturbation( $x$ );
  else
    | Generate a giant TSP tour using the ant colony heuristic;
    |  $x \leftarrow$  split the giant TSP tour in feasible routes;
until (stopping criterion not met);
Report the best solution  $x^*$ 
```

---

### 3.1 Generation of an initial solution for the RCTVRP

The generation of an initial solution for the RCTVRP follows a two steps procedure. In the first step (described in Section 3.1.1), the risk constraint is temporarily relaxed and the ant colony heuristic is used to create a giant TSP tour. In the second step (described in Section 3.1.2), this giant tour is split into feasible routes for the RCTVRP using an optimal splitting procedure.

In order to test the initial solutions provided by the aLNS metaheuristic and compare them with the solutions provided by other known metaheuristics for the RCTVRP, a preliminary experiment has been conducted. Ten instances have been randomly selected from the test sets SET V, SET R, SET S and SET O (all described in Section 4.1) and used for this test. Each test instance has been run 20 times by using all the existing metaheuristics for the RCTVRP. The best solutions and the average solutions over these 20 runs have been used to compute the best and the average percentage gaps from the best known solutions that are reported in Table 1. It can be observed that the average quality of the initial RCTVRP solutions (measured by the best percentage gap from best known solutions) provided by the aLNS metaheuristic is better than the m-NNg (or p-NNg, both using the same constructive heuristic to generate an initial RCTVRP solution) but

worse than the m-CWg (or p-CWg, both using the same constructive heuristic to generate an initial RCTVRP solution) and the p-TLK. In this way, the large neighbourhood search heuristic may have more room for improvement during the intensification stage than m-CWg and p-TLK, without being far from promising search areas. With regard to the robustness of the initial RCTVRP solutions (measured by the average percentage gap from best known solutions), the aLNS metaheuristic produces a wider variety of initial solutions than the m-CWg (or p-CWg) and the p-TLK (that uses a deterministic constructive heuristic), helping the large neighbourhood search heuristic to efficiently escape from local optima.

Table 1: Quality and robustness of initial solutions provided by the existing metaheuristics for the RCTVRP

Metaheuristic	Best % gap	Average % gap
p-TLK	2.39%	2.39%
m-CWg (p-CWg)	3.16%	4.57%
<b>aLNS</b>	<b>5.26%</b>	<b>7.73%</b>
m-TNng (p-TNng)	8.72%	12.26%
m-NNg (p-NNg)	9.52%	13.68%

### 3.1.1 Giant TSP creation

In the first step of the constructive heuristic, the risk constraint is relaxed and a giant TSP tour is obtained by using the ant colony heuristic, as described in Dorigo and Gambardella (1996), to which the reader is referred for more details. In the ant colony heuristic, a set of agents (ants) cooperate to find good solutions for the TSP. Ants cooperate using an indirect form of communication mediated by the pheromone that they deposit on the arcs of the graph while building solutions.

A predefined number of ants (parameter  $nAnts$ ) is initially selected and then each ant is positioned on a randomly chosen network node. Each ant builds a TSP tour by repeatedly applying a stochastic greedy rule, named the state transition rule. While constructing its tour, an ant also modifies the amount of pheromone on the visited arcs by applying a local pheromone updating rule, that also considers the quality of the obtained tour. The number of times each ant applies both the state transition rule, to incrementally build a solution, and the local pheromone updating rule is a parameter of the heuristic named *step number* in the remainder of the paper.

Once all ants have terminated their tour, the amount of pheromone on arcs is modified again by applying the global updating rule, during which a fraction of the pheromone evaporates on all arcs. Ants are guided, in building their tours, by both heuristic information (e.g. preference of short edges), and by pheromone information: an edge with a high amount of pheromone is a very desirable choice. The pheromone updating rules are designed so that they tend to give more pheromone to edges which should be visited by ants.

The process is then iterated a number of time (parameter *antRepetition*) and the best found TSP tour represents the input for the splitting procedure (described in Section 3.1.2) used to generate a feasible solution for the RCTVRP.

### 3.1.2 Splitting Procedure

In the second step of the constructive heuristic, implemented in aLNS, the best found giant TSP tour is subjected to a variant of the splitting procedure described in Prins (2004). The original procedure creates an auxiliary graph containing  $n + 1$  nodes (0 to  $n$ ), and adds an arc between nodes  $i - 1$  and  $j$  (with  $i \leq j$ ) if the route visiting the  $i$ -th customer to the  $j$ -th customer in the order they appear in the giant tour is feasible. Contrary to the original procedure, the modified procedure also adds an arc if the reverse route is feasible. The weight of the arc is equal to the cost of the route.

The best possible way to split the giant tour in feasible routes is achieved by finding the shortest path from node 0 to node  $n$  in the auxiliary graph. If the shortest path contains the arc from  $i - 1$  to  $j$ , the giant tour is split between the  $i - 1$ -th node and the  $i$ -th node and between the  $j$ -th and the  $j + 1$ -th node. The auxiliary graph is useful to understand how to split a big TSP tour in feasible routes. However, in practice, the generation of an auxiliary graph is not needed. In order to find the optimal splitting, we adapted a labelling algorithm that is described in Chang and Chen (2007).

## 3.2 Intensification stage

After having generated an initial RCTVRP solution, during the intensification stage, a large neighbourhood search (LNS) heuristic is employed to improve the current solution. Different from the local search heuristics used in Talarico et al. (2013), the LNS, employed in the aLNS metaheuristic, contains: (1) a *Reverse* operator, that is executed at the beginning of the LNS heuristic and every time the current solution gets improved; (2) an additional local search operator, the *Internal Relocate Operator*, for which specific short-cut calculations are developed to check if the movement is both desirable (from the point of view of the solution cost) and feasible (from the point of view of the risk constraint) in  $O(1)$  (see Section 3.2.1).

The *Reverse* operator takes advantage of the fact that the global risk associated to a generic route  $r$  (e.g.  $(0, A, B, C, D, 0)$ ) is different from the global risk of the reversed route  $\bar{r}$ , in which the order of the nodes is inverted (i.e.  $(0, D, C, B, A, 0)$ ). The *Reverse* operator is executed only if the reversed route of  $r$  (named  $\bar{r}$ ) presents a global risk  $GR^{\bar{r}} < GR^r$ . In case the *Reverse* operator is applied,  $\bar{r}$  will present the same cost of  $r$  (whether the RCTVRP is symmetric), but also a lower value of the global route risk. Therefore,  $\bar{r}$  will offer more room for additional nodes to be added in  $\bar{r}$  resulting in higher chances to reduce the solution cost.

The LNS heuristic, employed in this paper, uses 7 local well-known local search operators for vehicle routing problems (see Bräysy and Gendreau (2005) for more details) and the *Reverse* operator, described before, which are explored sequentially in the order shown in Algorithm 2. Each local search operator uses a first improvement strategy and every improvement encountered is accepted. Moreover, short-cut calculations (similar to the ones described in Talarico et al. (2013)) are implemented by each local search operator to efficiently determine whether a move results in an improved solution and what the new risk indices are for the different routes, affected by the move. In many cases, the feasibility checks can be performed in  $O(1)$ . If none of the operators can lead to a better solution, the current solution is saved as a local optimum.

---

**Algorithm 2:** LNS structure

---

```

Let  $x$  be the current solution;
Set  $stop = false$ ;
repeat
     $\forall$  routes in  $x$  apply the Reverse operator;
    Apply Internal Or-Opt to  $x$ ;
    if no improvement then
        Apply Internal Relocate to  $x$ ;
        if no improvement then
            Apply Internal Two-Opt to  $x$ ;
            if no improvement then
                Apply External Exchange to  $x$ ;
                if no improvement then
                    Apply External Relocate to  $x$ ;
                    if no improvement then
                        Apply External Cross-Exchange to  $x$ ;
                        if no improvement then
                            Apply External Two-Opt to  $x$ ;
                            if no improvement then
                                set  $stop = true$ ;
until ( $stop == true$ );

```

---

### 3.2.1 Internal Relocate Operator

In this section, the short-cut calculations, used inside the internal relocate operator to test the desirability and the feasibility of a move, are described.

Let  $r = (0, \dots, A, B, C \dots, D, E, \dots, 0)$  (see Figure 2 on the left) be a route in the current solution, where node 0 represents the depot. A relocate intra-route move consists in the relocation of node  $B$  within  $r$ , e.g. between  $D$  and  $E$ , maintaining the same direction of the initial route. The movement is desirable only if the following condition is satisfied:

$$d_{AB} + d_{BC} + d_{DE} - d_{AC} - d_{DB} - d_{BE} > 0. \quad (3)$$

In order to evaluate if the new route  $r_{new} = (0, \dots, A, C \dots, D, B, E, \dots, 0)$  (see Figure 2 on the right) is feasible, from the risk constraint point of view, a short-cut calculation

can be applied to evaluate the global route risk for  $r_{new}$ , without computing all the risk labels for each node in  $r_{new}$ . To make this calculation possible, the following data structures are used: (1)  $D_h^r$ , which represents the cumulative distance travelled from the depot 0 to node  $h$  along route  $r$ ; (2)  $GR^r$ , which is the global route risk of route  $r$ ; (3)  $M_A^r$ , which denotes the amount of money on board of the vehicle when it leaves node  $A$  along route  $r$ . The global route risk for  $r_{new}$  can be calculated as follows:

$$GR^{r_{new}} = GR^r - (M_A^r + m_A + m_B) \cdot d_{BC} - m_B \cdot (D_D^r - D_C^r) + M_E^r \cdot (d_{BE} - d_{DE}) + (M_A^r + m_A) \cdot (d_{AC} - d_{AB}) + (M_D^r + m_B + m_D) \cdot d_{DB} \quad (4)$$

All data structures in Eq. (4) are updated throughout the running of the algorithm. Therefore, this feasibility check can be performed in  $O(1)$ . If the global route risk for  $r_{new}$  is lower than the risk threshold  $T$ , the move can be executed. The resulting route, after the application of the move, is depicted on the right side of Figure 2.

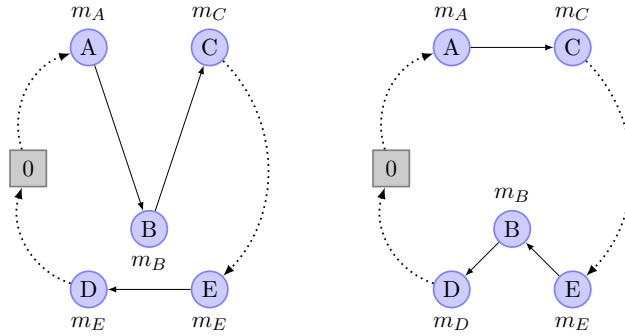


Figure 2: Relocate inter-route operator

### 3.3 Diversification strategies

As mentioned before, the whole metaheuristic is repeated until a stopping condition is met. In particular, the aLNS metaheuristic stops its execution when a predefined maximum number of iterations (parameter  $\rho$ ) is reached. Alternately, a maximum allowed running time, as done in Section 4, can be employed as a stopping criterion. At each iteration, a new solution for the RCTVRP is obtained. At the end of the  $\rho$  repetitions, the best solution found so far is reported.

During each repetition, two different structures are used to escape from local optima. The first one, named *multi-start*, is adopted in case a maximum number of iterations without having obtained any improvement in the current solution (parameter *noImpr*) is reached. In this case, a new initial solution is built from scratch by applying the ant

colony heuristic followed by the splitting procedure. This new solution is then improved by reapplying the LNS heuristic, described before. In case the maximum number of iterations without improvements is not passed, a second diversification mechanism, named *perturbation* is employed.

The *perturbation* attempts to explore the solution space moving from the current solution ( $x$ ) to another one ( $x'$ ) obtained after having partially destroyed and then repaired the current solution. The perturbation mechanism takes two parameters as input: (1)  $\xi$ , which represents the percentage number of routes to be destroyed from  $x$  and (2)  $\alpha$ , that is used to control the balance between greediness and randomness during the repair phase[1].

During the destruction phase, a random route  $r$  is selected from the current solution  $x$ . All nodes are removed from  $r$  and inserted in a list of unvisited nodes ( $U$ ). This step is repeated  $\xi \cdot K$  times, where  $K$  represents the total number of routes in  $x$ . During the repair phase, a new current solution  $x'$  is built, starting from the non destroyed routes of  $x$  and adding new routes which contain the unvisited nodes in  $U$ . These new routes are generated by applying a nearest neighbourhood approach combined with a greedy randomized selection of the unvisited nodes in  $U$ . This means that the nodes are selected randomly from a restricted candidate list containing the first  $\alpha$  closest unvisited nodes passing all a feasibility check with respect to the risk constraint. If it is not possible to add any node to the current route without violating the risk constraint (i.e. the vehicle can reach the depot without surpassing the risk threshold), the route is closed (the vehicle drives back to the depot) and a new route is started until  $U$  is empty. Once a new current solution  $x'$  has been generated, the LNS heuristic is used to improve  $x'$ .

## 4 Computational experiment

In this section, the experiments that have been performed to test the aLNS metaheuristic are described. The solution approach has been coded using Java language and the experiments were executed on an Intel core i7-2760QM 2.40GHz processor using a machine with 4GB RAM.

First, two additional sets of medium-large benchmark RCTVRP instances, named SET O and SET S respectively, have been generated. These instances, with known optimal solutions, together with the existing benchmark instances available for the RCTVRP (SET V and SET R) have been used to tune and test the aLNS metaheuristic (see Section 4.1).

The experimental analysis is carried out in two sequential steps. In a first stage, the parameters for the aLNS metaheuristic are calibrated in order to achieve the best results (see Section 4.2). In the second stage, the aLNS metaheuristic with its “best” parameter setting is used to solve all test instances contained in SET V, SET R, SET O and SET S. In Section 4.3, the obtained results are compared with the solutions provided by the existing metaheuristics for the RCTVRP.

## 4.1 Test instances

The test instances that are used in this paper are based both on existing benchmark instances for the RCTVRP, namely SET R and SET V, and two additional set of benchmark instances with known optimal solutions, namely SET O and SET S.

SET R includes 180 small RCTVRP instances with a number of nodes ranging from 4 to 20. For these instances, optimal solutions are known. SET V contains 70 medium and large instances with a number of nodes ranging from 22 to 301. Both these instance sets have been introduced in Talarico et al. (2013) and downloadable at <http://antor.uantwerpen.be/Download>.

In order to have benchmark instances, having optimal known solutions to which compare the quality of the obtained results, that are larger than the instances contained in SET R, two additional test sets are proposed in this paper and named SET O and SET S respectively. These 2 sets form a library of instances for the RCTVRP that is made available at <http://antor.uantwerpen.be/Download>.

These sets include in total 64 different instances. Both SET O and SET S contain 32 instances ranging from 10 to 337 nodes. These instances are generated in such a way that the optimal solutions are known in advance. As can be seen from Figure 3 SET O and SET S differ from each other with respect to the topology of the graph. The special structure of the instances contained in SET O makes them really difficult to be solved, since the distances from neighbouring nodes (e.g. the ones on the left/right or above/below a given node) along each branch of the obelisk shapes is identical. The risk threshold is defined so as to obtain a predefined number of customers in each route. The optimal solutions of the instances in SET O and SET S contain as many routes as there are either obelisk shapes or the spiral arms (3, 4, 6, 8 and 16) visible in Figure 3. The demand vector is composed of integer values specifically defined for each node in order to have an increasing demand that is proportional to the order of visit of the nodes in each obelisk/spiral shape that composes the optimal solution. In other words, we assigned a demand equal to 1 to the first node visited in each route contained in optimal solution (the spiral/obelisk shown in Figure 3), the value 2 to the second node, 3 to the third node and so on.

## 4.2 Parameter configuration

The aLNS metaheuristic uses some internal parameters that need to be tuned in order to have a good compromise between solution quality and computation time.

Using 8 instances randomly selected from all the instances sets (SET R, SET V, SET O and SET S), a full factorial statistical experiment has been performed. For each parameter, different values have been tested as shown in Table 2. All the parameters resulted having a significant effect on both the solution cost and the running time. The

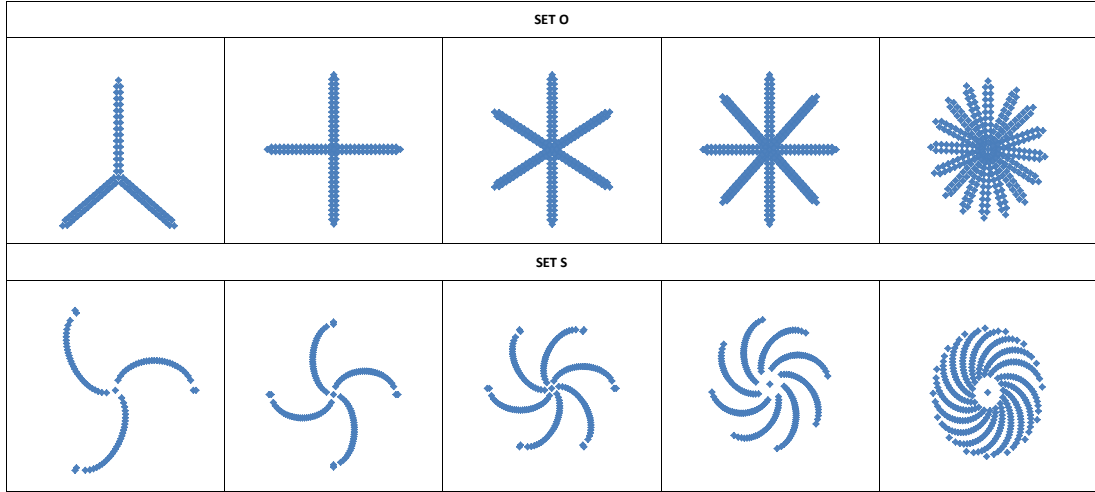


Figure 3: Topology of instances contained in SET O and SET S

“best” setting, that has been selected for the aLNS metaheuristic, is the one reported in last column of Table 2.

Parameter	Description	Tested Values	“Best”
$\rho$	Number of repetitions of the aLNS metaheuristic	50	<b>50</b>
$\xi$	Maximum percentage number of routes in the current solution to be destroyed	10%, 20%, ..., 90%	<b>60%</b>
$\alpha$	Size of the restricted candidate list (repair heuristic)	1, 2, ..., 5	<b>3</b>
noImpr	Maximum number of iterations without improvements	$\rho/2$ , $\rho/3$ , $\rho/5$ , $\rho/10$	$\rho/5$
antRepetition	Number of initial giant tours to generate	2, 5, 10, 20	<b>5</b>
nAnts	Number of ants	4, 8, 16, 20	<b>16</b>
Step number	Number of times each ant applies the state transition rule and the local pheromone updating rule to incrementally build a giant tour	3, 4, 5, 6	<b>6</b>

Table 2: Metaheuristic parameters

It is worth observing that the higher the parameters  $\rho$ , the better the solution quality at the expense of the computational time. In Figure 4, the evolution of the solution cost obtained with different values of  $\rho$  is reported. It can be noticed that the aLNS metaheuristic converges towards stable solutions when  $\rho$  assumes high values. It should be noted that the marginal improvement of the solutions significantly decreases when  $\rho$  assumes values greater than 1000.



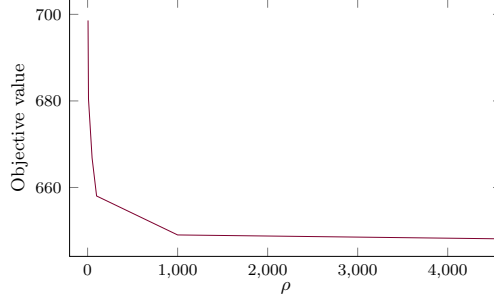


Figure 4: Solution cost over the number of repetitions ( $\rho$ ) of the aLNS metaheuristic

### 4.3 Results

The aLNS metaheuristic in his “best” parameter setting, described before, has been tested and compared with the existing metaheuristics for the RCTVRP, namely m-NNg, p-NNg, m-TNng, p-TNng, m-CWg, p-CWg and p-TLK (see Talarico et al. (2013) for more details). Moreover, since each metaheuristic involves various elements of randomness, finding better solutions may also be achieved by repeating the solution process a number of times. For this reason, each solution approach is run 20 times on each instance. The average solutions and the best solutions over these 20 runs are used to evaluate the robustness and the solution quality associated to each metaheuristic.

The comparison between the different metaheuristics is done by first solving the existing benchmark instances for the RCTVRP namely SET V and SET R. In order to make a fair comparison between competing metaheuristics, a fixed computation time is allowed for all the solution approaches to solve each instance.

Since instances contained in SET R are small instances, with a maximum of 20 nodes, a limited computation time of 5 seconds per run on each instance is allowed. In Table 3 both the best percentage gap from optimal solutions averaged over all instances in SET R (column Best GAP) and the average percentage gap from the optimal solutions over 20 runs (column Avg. GAP) are reported. As it can be observed, the aLNS metaheuristic produces results that are comparable with the best existing solution approaches p-TLK and m-CWg. The aLNS metaheuristic is able to find all the optimal solutions for the instances in SET R, as well as the other solution approaches, and the robustness of the aLNS metaheuristic (looking at the average gap from optimal solutions over 20 runs) is equal to the m-CWg metaheuristic and only 0.01% worse than p-TLK.

Successively, the aLNS metaheuristic is compared with the existing metaheuristics by solving the medium and large instances contained in SET V. Due to the larger dimension of the instances, a maximum running time of 30 seconds per run is allowed. Since the optimal solutions for the instances contained in SET V are not known, the results obtained by each metaheuristic are compared with the best solution obtained for each instance. Also in this case, both the best percentage gap from best known solutions

Table 3: Results obtained by solving the instances contained in SET R

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

averaged over all instances in SET V (column Best GAP) and the average percentage gap from the best known solutions over 20 runs (column Avg. GAP) are reported in Table 4. With regard to the instances contained in SET V, the aLNS metaheuristic is able to produce better solutions than the existing metaheuristics. Only the robustness is slightly worse than the p-TLK due to a higher variety of solutions that the aLNS metaheuristic can generate.

Table 4: Results obtained by solving the instances contained in SET V

Metaheuristic	Best GAP	Avg. GAP
m-CWg	0.86%	1.89%
m-NNg	1.88%	3.75%
p-CWg	1.11%	2.51%
p-NNg	1.46%	3.38%
m-TNNg	1.55%	2.84%
p-TNNg	1.32%	2.92%
p-TLK	0.52%	1.38%
aLNS	0.50%	1.44%

Finally, all the metaheuristics are tested and compared using the newer sets of benchmark instances, that have been introduced in this paper, namely SET O and SET S. Since the size of the instances contained in these sets is comparable with the instances in SET V, a maximum running time of 30 seconds per run on each instance is also allowed. For each metaheuristic, the following results have been reported in Table 5: (1) the best percentage gap from optimal solutions averaged over all instances in the sets (column Best GAP); (2) the average percentage gap from the best known solutions over 20 runs (column Avg. GAP); (3) the number of optimal solutions that have been found (column # Optimal).

As shown in Table 5, the aLNS metaheuristic finds a higher number of optimal solutions for SET O, while the m-CWg metaheuristic finds the highest number of optimal solutions for set SET S (4 more than aLNS and 2 more than p-TLK). On average, for both benchmark sets, the solution quality obtained by aLNS is better than the other existing approaches. In terms of robustness, the aLNS metaheuristic obtained lower average

Table 5: Results obtained solving the instances contained in SET O and SET S

(a) Results for SET O			
Metaheuristic	Best GAP	Avg. GAP	# Optimal
m-CWg	4.91%	5.29%	8/32
m-NNg	6.35%	7.29%	3/32
p-CWg	5.21%	5.63%	6/32
p-NNg	5.88%	6.65%	2/32
m-TNNg	5.83%	6.49%	1/32
p-TNNg	5.57%	6.22%	4/32
p-TLK	4.25%	4.57%	15/32
aLNS	3.92%	4.54%	17/32
(b) Results for SET S			
Metaheuristic	Best GAP	Avg. GAP	# Optimal
m-CWg	3.48%	4.26%	13/32
m-NNg	4.90%	5.93%	7/32
p-CWg	3.95%	4.62%	8/32
p-NNg	4.78%	5.81%	6/32
m-TNNg	4.64%	5.57%	7/32
p-TNNg	3.97%	4.80%	8/32
p-TLK	3.20%	3.86%	11/32
aLNS	2.91%	3.75%	9/32

percentage gaps from the best known solutions over 20 runs. Also in this case, the m-NNg resulted being the metaheuristic with the poorest performances.

The overall results obtained by solving all the benchmark instances, using all solution approaches, are summarized in Table 6. It is possible to observe that the aLNS metaheuristic, on average, produces better solutions than the existing solution approaches for the RCTVRP over all the benchmark sets. The best gap, averaged over all benchmark sets, is equal to 1.83%, while the average gap is only 2.46%. The second best metaheuristic resulted being the p-TLK, presenting a best gap, averaged over all the benchmark sets, of 2.08%. The third place is occupied by the p-CWg metaheuristic, with a best gap of 2.23%. The m-NNg performed the worst also if the solution quality is, on average, not so far from the best known solutions (with a best gap of only 3.28%).

## 5 Conclusions

In this paper, an iterated local search metaheuristic for the RCTVRP is proposed. This solution approach has been named aLNS and uses a bio-inspired heuristic, the ant colony heuristic, to find a solution for the underlying TSP problem. Then, the best found giant TSP tour is split in feasible routes and then a large neighbourhood search heuristic is

Table 6: Results obtained solving the instances contained in all the benchmark sets

(a) Best GAP

Test Set	m-CWg	m-NNg	p-CWg	p-NNg	m-TNNg	p-TNNg	p-TLK	aLNS
SET R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
SET V	0.52%	1.88%	1.11%	1.46%	1.55%	1.32%	0.86%	0.50%
SET O	4.91%	6.35%	5.21%	5.88%	5.83%	5.57%	4.25%	3.92%
SET S	3.48%	4.90%	3.95%	4.78%	4.64%	3.97%	3.20%	2.91%
<b>Average</b>	<b>2.23%</b>	<b>3.28%</b>	<b>2.57%</b>	<b>3.03%</b>	<b>3.01%</b>	<b>2.71%</b>	<b>2.08%</b>	<b>1.83%</b>

(b) Avg. GAP

Test Set	m-CWg	m-NNg	p-CWg	p-NNg	m-TNNg	p-TNNg	p-TLK	aLNS
SET R	0.10%	1.28%	0.43%	0.88%	0.58%	0.69%	0.09%	0.10%
SET V	1.38%	3.75%	2.51%	3.38%	2.84%	2.92%	1.89%	1.44%
SET O	5.29%	7.29%	5.63%	6.65%	6.49%	6.22%	4.57%	4.54%
SET S	4.26%	5.93%	4.62%	5.81%	5.57%	4.80%	3.86%	3.75%
<b>Average</b>	<b>2.76%</b>	<b>4.57%</b>	<b>3.30%</b>	<b>4.18%</b>	<b>3.87%</b>	<b>3.66%</b>	<b>2.60%</b>	<b>2.46%</b>

applied. Finally, two different diversification mechanism are used to explore the solution space and to escape from local optima.

Two additional sets, named SET O and SET S respectively, containing medium and large RCTVRP instances, have been introduced. The aLNS metaheuristic has been tested and compared with the existing solution approaches for the RCTVRP by using four different benchmark instance having a total number of instances equal to 314.

The aLNS metaheuristic outperformed the existing solving approaches for the RCTVRP for what concern the solution quality and the robustness of the solution approach. The aLNS was also able to find a total number of optimal solutions for the instances contained in both SET O and SET S in line with the best known solution approach for the RCTVRP (26 optimal solutions found in total as the p-TLK). These results can be explained by the fact that the ant colony heuristic produces initial solutions presenting on average a better quality than the m-NNg (that results the worst solution approach for the RCTVRP) but worse than the m-CWg and p-TLK (that result among the best metaheuristics for the RCTVRP problem). Similar reasoning holds for the solution variability. In fact, the variability of solutions that the aLNS can produce is higher than the m-CWg and p-TLK heuristics, but lower than the m-NNg heuristic. These two properties, combined together, makes both the diversification and intensification stages used inside the aLNS heuristic more effective.

Further development can be oriented at extending the RCTVRP by modelling some real life constraints (e.g. route length restrictions, precedence relations, delivery of money to customers). Another interesting research path concerns the definition of a multi objective version of the RCTVRP where e.g. both the risk and the travel cost/distance need to be minimized.

## References

- O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.
- Center for Chemical Process Safety. *Guidelines for Chemical Transportation Risk Analysis*. American Institute of Chemical Engineers, New York, 1995.
- Y. Chang and L. Chen. Solve the vehicle routing problem with time windows via a genetic algorithm. *Discrete and continuous dynamical systems*, pages 240 – 249, 2007.
- M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the travel salesman problem. *TR/IRIDIA/1996-5*, 1996.
- M. Dorigo and L.M. Gambardella. Ant colonies for the travelling salesman problem. *Biosystems*, 43(2):73 – 81, 1997.
- A. Erasmus. The CIT industry in South Africa - its challenges and solutions. In ESTA The Cash Management Companies Association, editor, *ESTA Business Conference, Exhibition and General Assembly*, Bratislava - Slovakia, 2012.
- K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.
- J. Krarup. The peripatetic salesman and some related unsolved problems. In B. Roy (Ed.), editor, *Combinatorial programming: methods and applications*, page 173–178, Reidel: Dordrecht, The Netherlands, 1975.
- H.R. Lourenço, O.C. Martin, and T. Stützle. *Iterated Local Search: Framework and Applications*. Springer New York, 2010.
- S.U. Nguveu, C. Prins, and R. Wolfer Calvo. A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics*, 10:253–266, 2010.
- K. D. Pijawka, S. Foote, and A. Soesila. Risk assessment of transporting hazardous materials: Route analysis and hazard management. In Transportation Research Board, editor, *Transportation Research Record 1020*, National Research Council, Washington, D.C., 1985.
- C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computer and Operations Research*, (31):1985 –2002, 2004.
- F. Russo and C. Rindone. Planning in road evacuation: classification of exogenous activities. In Pratelli and Brebbia C.A., editors, *WIT Transactions on the Built Environment Volume 116, Urban Transport XVII, Urban Transport and the Environment in 21st Century*, pages 639–651, WIT Press Southampton, 2011.
- L. Smith and E. Louis. Cash in transit armed robbery in Australia. *Trends and issues in crime and criminal justice*, 397:1, 2010.

- L. Talarico, K. Sörensen, and J. Springael. Metaheuristics for the Risk-constrained Cash-in-Transit Vehicle Routing Problem. Working Papers 2013005, University of Antwerp, Faculty of Applied Economics, March 2013.
- R. Wolfer Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computer and Operation Research*, 30:1269–1287, 2003.