The Vehicle Routing Problem with Conflicts

Khaoula Hamdi-Dhaoui∗, Nacima Labadie∗, Alice Yalaoui∗.

∗Laboratory of Industrial Systems Optimization, University of Technology of Troyes, Charles Delaunay Institute, UMR STMR, CNRS, 12 rue Marie Curie, BP 2060 10 010 TROYES Cedex, France.
(e-mail: (khaoula.dhaoui, nacima.labadie, alice.yalaoui)@utt.fr).

Abstract: The Vehicle Routing Problem (VRP) is a classical problem for which several exact and approximation methods were proposed. In real life applications, such as in Hazardous Material transportation, transported items may be incompatible. This paper summarizes the first methods developed for this extension called the Vehicle Routing Problem with Conflicts (VRPC). This study presents a mathematical model, heuristics, metaheuristics (an Iterated Local Search (ILS) and a Greedy Randomized Adaptive Search Procedure - Evolutionary Local Search (GRASP-ELS)). A lower bound is also introduced in this article.

Keywords: Vehicle Routing, conflicts, heuristics, ILS, GRASP-ELS, branch and cut.

1. INTRODUCTION

The Vehicle Routing Problem consists in designing the optimal set of routes for a fleet of vehicles in order to serve a given set of customers. There are many variants of the problem (Toth and Vigo 2002); Capacitated VRP (CVRP) with limited capacity constraint for vehicles, VRP with Time Windows (VRPTW) where each customer has to be served in a time interval, VRP with Pickup and Delivery (VRPPD) where a number of goods need to be moved from given pickup locations to other delivery locations, etc. In the classical VRP, the aim is to satisfy customers demands transporting their items while minimizing the travel cost. The items nature is not precise and does not interfere with the problem’s objective. The problem studied in this paper takes into account the physical properties of the transported items, especially when the materials nature is crucial and has heavy consequences on the transport conditions. This is the case in Hazardous Materials Transportation, for example.

A Hazardous Material (HazMat) is any chemical, radiological or biological material that poses a wide range of health hazards (such as irritation, sensitization, and carcinogenicity) and physical hazards (such as flammability, corrosion, and reactivity). Specific guidelines exist on transport and shipping hazardous materials. HazMat is classified into different categories according to their chemical and physical properties. Some of these materials are incompatible so they can not be stored together in the same storage building or room nor be transported in the same vehicle. In the literature, there are very few papers concerning HazMat routing. Most of the existing works on HazMat transportation focus on constructing a path between an origin and a destination which optimizes a risk criterion (Zografos and Androutsopoulos 2004, 2008) and (Zhang et al. 2005).

We propose in this paper a study based on the existence of conflicts between different kinds of HazMat to transport. The concept of conflicts between items was introduced in another classical combinatorial problem: the Bin Packing Problem (BPP). The Bin Packing Problem with Conflicts (BPPC) is a natural generalization of the BPP that aims to pack items into the minimum number of bins subject to incompatibility restrictions (Gendreau et al. 2004) (Khanafer et al. 2008).

The association of conflicts with Vehicle routing leads to a new problem. This kind of constraints has been noticed in some industrial cases studies (Ceselli et al. 2009) (Giuseppe et al. 2008). We present in this communication the first real theoretical study. We first describe the problem and present the mathematical model. Then, based on basic heuristics (Hamdi et al. 2009), we present an ILS and an hybrid GRASP-ELS (Hamdi et al. 2010a, 2010b). Finally, a lower bound based on a branch and cut algorithm is developed (Hamdi et al. 2010c).

2. THE VEHICLE ROUTING PROBLEM WITH CONFLICTS

This extension of the VRP can be formally defined on an undirected graph \( G = (V, E) \) with \( V = \{0, 1, \ldots, n\} \) the set of vertices and \( E \) the set of edges. The vertex 0 represents the depot, where a fleet of \( M \) identical vehicles with capacity \( W \) is provided, whereas the other vertices represent the customers. Let \( F \) be the set of vehicles. A demand \( d_i \) is associated to each customer \( i \). The vertices are characterized with their Euclidean coordinates on the plane. The cost \( c_{ij} \) associated to each edge \([i,j]\) corresponds to the traveling cost from \( i \) to \( j \). Here, we consider symmetric costs, in other words, \( c_{ij} = c_{ji}, \forall [i, j] \in E \).

Without loss of generality, we consider that \( c_{ij} = d_{ij} \) is the distance separating \( i \) and \( j \). Let \( A \) be the matrix of
conflicts whose elements equal 0 or 1. An element $a_{ij}$ of
the matrix takes the value 1 if and only if the demands
of customers $i$ and $j$ are conflicting. We assume that
the request of a customer corresponds to a single type of
HazMat. Given the graph $G$, the vehicle routing problem
with conflicts consists in finding the routes satisfying the
following constraints: (1) each route starts and ends at
the depot (2) each customer is visited only once by a
single vehicle (3) the total demand of customers assigned
to the same route can not exceed the vehicle capacity $W$
(4) customers with conflicting items must be assigned
different routes. The aim is to minimize the total traveling
cost.

2.1 Model

The variable $x_{ij}^k$ equals to 1 if, and only if, the vehicle $k$
uses the edge $[i, j]$ going from $i$ to $j$. The second variable
$y_i^k$ takes the value 1 if, and only if, the vertex $i$ is assigned
to the vehicle $k$.

The objective function (1) corresponds to the minimization
of the total traveled distance. The respect of the vehicle’s capacity is ensured by the constraints (2). The
constraints (3) guarantee that every customer is serviced
by a single vehicle. The constraints (4) make sure that
exactly $M$ routes are built. If a vehicle services a customer,
it must arrive to the customer (5) and leave it (6). The
elimination of sub-tours is ensured by the constraints (7).
The constraints (8) guarantee that no conflicting items are
assigned to the same vehicle. Finally, the constraints (9)
and (10) fix the nature of the decision variables.

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=1}^{M} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^k \\
\text{subject to} & \quad \sum_{i=1}^{n} d_i y_i^k \leq W, \quad \forall k \in F \\
& \quad \sum_{k=1}^{M} y_i^k = 1, \quad \forall i \in V \setminus \{0\} \\
& \quad \sum_{k=1}^{M} y_i^k = M \\
& \quad \sum_{j=1}^{n} x_{ij}^k = y_i^k, \quad \forall j \in V \setminus \{0\}, \forall k \in F \\
& \quad \sum_{j=1}^{n} x_{ij}^k = y_i^k, \quad \forall i \in V \setminus \{0\}, \forall k \in F \\
& \quad \sum_{i,j \in S} x_{ij}^k \leq |S| - 1, \quad S \subseteq V \setminus \{0\}; 2 \leq |S| \leq n - 1; \forall k \in F \\
& \quad \sum_{j=1}^{n} a_{ij} y_j^k \leq n(1 - y_i^k), \quad \forall i \in V; \forall k \in F \\
& \quad x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in V^2; i \neq j; \forall k \in F \\
& \quad y_i^k \in \{0, 1\}, \quad \forall i \in V; \forall k \in F
\end{align*}
\]

3. CONSTRUCTIVE HEURISTICS

We started by adapting 3 well-known heuristics for the
classical vehicle routing problem (Hamdi et al. 2009). A
fourth heuristic based on the partition of the items into
sets, corresponding to HazMat classes, was developed.

3.1 Modified Clarke and Wright heuristic

The Clarke and Wright algorithm (Clarke and Wright
1964) starts by assigning each customer to a different
route. Then the routes are sorted out in increasing order
of costs. An evaluation of the savings, resulting from pos-
sible mergers of couples of routes, is then performed. The
one that allows the best cost reduction is chosen. In our
version, at each iteration the algorithm verifies that there
are no conflict between the customers of both routes to be
merged, using compatibility test. The method stops when
no more merger allowing total cost reduction is possible.

3.2 Best insertion heuristic

At each iteration, the additional cost produced by the
insertion of each non-inserted customer, in all the possible
positions with regard to the already inserted customers, is
calculated. The insertion which gives the minimum cost
increase, without generating conflicts, is chosen. If the
creation of a new route has a lower cost, the route is
created. The method stops when all the customers are
inserted into routes.

3.3 Modified Sweep Heuristic

This heuristic is inspired from (Gillett and Miller 1974)
and starts by grouping the customers into sets. Then,
an algorithm for the TSP is applied to form a route
going through all the customers of the same set. Sets of
customers with compatible items are formed by adding
at each step the customer that increases the less a lower
bound of the TSP, defined by the customers already in
the set plus the considered one. The lower bound used is
the minimal cost 1-Spanning Tree. A spanning tree of a
subgraph $G_k$ (set of customers assigned to the vehicle $k$)
is a selection of edges of $G_k$ that form a tree spanning every
vertex. That is, every vertex lies in the tree, but no cycles
(or loops) are formed. A 1-spanning tree is a spanning tree
to which one free edge incident to the root is added. The
lower bound increases by a value that equals the cost of
the cheapest edge linking the current customer to a node
already assigned to the cluster. Once the sets are formed,
a route is built for each set using the convex hull. The
convex hull is the minimal convex set containing all the
points of the sets (customers are represented with points
corresponding to their coordinates in the plane). Then,
the remaining inside customers (in the convex hull but
not on the borders) are inserted in the route in positions
with minimum deformation of the envelope. The tests
of compatibility are performed at the first phase of the
algorithm. Before the insertion of a customer in the tree,
we verify that it is not conflicting with the other customers
already in the tree.
3.4 A Class Based heuristic

Another heuristic grouping customers, in the same or compatible classes, is developed. The main idea of this heuristic and its three tested versions is based on the use of a priority rule. For this purpose we define the class degree as the number of classes conflicting with it. Then, the classes are sorted out in decreasing order of their degrees of conflict. The assignment of customers to vehicles is done by Best Insertion technique in the three algorithms. The main difference between them lies on the list of customers candidate to insertion. In the Variant 1: The priority is given to the customers of the first class in the sorted list, in Variant 2: The classes which are compatible with all other classes are removed from the list. At each stage, the first class in the sorted list is considered, at the same time with the customers of $C_0$. Finally, in Variant 3, set containing the class of highest degree and the classes which are compatible with it (and compatible between them) is formed. When all the customers of this set are assigned, the same procedure is repeated with the remaining classes of the sorted list, until all the customers are inserted.

4. METAHEURISTIC ILS

An Iterated Local Search was developed for the problem (Algorithm 1). $Soln$ denotes the solution and $Cost(Soln)$ its cost. The best found solution is $BestSoln$. $H$ stands for the used heuristic and $LS$ the local search. The number of iterations is $n_i$ and $Perturb$ a perturbation procedure.

Algorithm 1: Iterated Local Search

Initialization

$H(BestSoln)$
$LS(BestSoln)$
For $j := 1$ to ($n_i$) do
  $Soln := BestSoln$
  $Perturb(Soln)$
  $LS(Soln)$
  If $Cost(Soln) < Cost(BestSoln)$ then
    $BestSoln := Soln$
  End if
End for

The local search explores the neighborhood of the current solution with 3 kinds of moves, sorted out in an increasing order of complexity (Customer relocation, Position exchange, 2-opt) (Hamdi et al. 2010a). The neighborhood exploration starts by the first customer of the first route. The first move that improves the solution cost is performed. After each update of the solution, the neighborhood exploration is restarted. The local search stops when all the routes are scanned and there is no more possible improvement.

The perturbation consists in performing one or several random moves on the current solution, based on customer relocation (Customers to relocate and insertion positions are picked randomly). Two perturbation procedures are used. In the first one, the cost variation generated by the move is evaluated at each iteration. If the move improves the solution, the perturbation is as soon stopped and the neighbourhood is immediately explored with the local search. When there is no improvement, the perturbation can perform up to $MaxP$ relocations. The second procedure, performs at least $MaxP$ moves. When a move improves the solution cost, the counter is handed to $MinP$ and $MaxP$ new moves are performed.

5. METAHEURISTIC GRASP-ELS

A Hybrid GRASP-ELS (Prins 2009) has also been developed to solve the VRPC (algorithm 2) (Hamdi et al. 2010a). In a GRASP, random solutions are generated using randomized heuristics and improved by local search. In the ELS, given an initial solution, $n_i$, $n_c$ local searches and perturbations are applied to improve a solution.

Algorithm 2: GRASP ELS

Initialization

Cost($BestSoln$) := Infinity
For $i := 1$ to $n_i$ do
  If $i = 1$ then $H(Soln)$
  Else $HR(Soln)$
  End if
  $LS(Soln)$
  $p := p_{min}$
  For $j := 1$ to $n_i$ do
    $BestChildCost := Infinity$
    For $k := 1$ to $n_c$ do
      $ChildSoln := Soln$
      $Perturb(ChildSoln, p)$
      $LS(ChildSoln)$
      If $Cost(ChildSoln) < Cost(BestChildSoln)$ then
        $BestChildSoln := ChildSoln$
      End if
    End for
    If $Cost(Soln) < Cost(BestSoln)$ then
      $Soln := BestChildSoln$
      $p := p_{min}$
    Else $p := min(p_{nax}, p+1)$
    End if
  End for
End for

$Soln$ denotes the current solution, $Cost(Soln)$ its cost, the best solution is $BestSoln$. $H$ stands for the used heuristic and $HR$ for the randomized version of the same heuristic. The number of iterations is $n_i$ and the number of child solutions constructed in parallel at each iteration is $n_c$. $LS$ is the local search and $Perturb$ the perturbation.

The GRASP part of the metaheuristic consists in creating different independent initial solutions by calling $H$ and $HR$. $n_i$ initial solutions are created and local search is applied to find better solutions in the neighborhood. In the classical GRASP, the best solution found so far is updated after the local search is called. In this approach, an ELS metaheuristic is applied on the initial solutions instead of the simple local search. The method starts by creating $n_c$ copies ($ChildSoln$) of every initial solution. Each one of the $n_c$ Child Solutions is perturbed. The resulting solution undergoes the local search and the incumbent solution
replaces the best child solution \( \text{BestChildSoln} \) if its cost is better. This procedure is repeated \( n_i \) times for each initial solution.

To generate initial solutions, we used two randomized versions of the modified Clarke and Wright heuristic and the Class Based heuristic. The Local search procedure is identical to the one described for ILS. Two perturbation procedures are called. The first one consists in performing one or more random moves, that can be those of the local search, on the obtained solution. The second one selects the least loaded route and aims to empty it by inserting its customers in the other routes.

6. LOWER BOUND

A method of branching-and-cutting for the VRPC is proposed in this section, in order to evaluate the quality of the solutions obtained with the metaheuristics. This method improves significantly the lower bound obtained with the linear relaxation. The algorithms are implemented with the softwares Visual Studio and Gurobi Optimizer and brought us to modify the Lysgaard’s CVRPSEP functions library for detection of violated cuts and to obtain a new library for the VRPC (Lysgaard et al. 2004).

This algorithm corresponds to a procedure of separation and evaluation where the lower bound for each subproblem is strengthened by means of cutting planes. To generate them, separation algorithms are needed. The cutting planes applied to the initial relaxation have to find an integer solution and to add capacity constraints progressively such that the final solution becomes feasible for the initial problem, well before about \( 2^n \) capacity constraints are considered. As the VRPC is a generalization of the VRP, families of inequalities whose efficiency for the VRP were proved are used. Different types of constraints are iteratively added in order to improve the bound’s quality. The most important are capacity inequalities, Strengthened comb inequalities and Multistar inequalities (Hamdi et al. 2010c).

6.1 Model

Let \( V_c = V \setminus \{0\} \) denote the set of customers. Given a set of customers \( S \subseteq V_c \) let \( \delta(S) \) denote the set of edges in \( G \) with exactly one end-vertex in \( S \) and \( r(S) \) denote the minimum number of vehicles required to serve the customers in \( S \). Given two sets of customers \( S \) and \( T \), let \( (S : T) \) denote the set of edges with one end vertex in \( S \) and the other in \( T \). Let \( x_{ij} \) be the number of times a vehicle uses the link \( e = [i,j] \). We note by \( x(\delta(S)) \) the \( \sum_{e \in \delta(S)} x_e \). The initial integer programming formulation \( LP \) is then:

\[
\begin{align*}
\text{minimize} & \quad \sum_{e \in E} c_e x_e \\
\text{subject to} & \quad x(\delta(\{i\})) = 2 \quad \forall i \in V \setminus \{0\} \\
& \quad x(\delta(S)) \geq 2 r(S) \quad S \subseteq V \setminus \{0\}, |S| \geq 2 \\
& \quad x_{ij} \in \{0,1\} \quad 1 \leq i \leq j \leq n \\
& \quad x_{ij} \in \{0,1,2\} \quad i = 0, j \in V \setminus \{0\} \\
\end{align*}
\]

The degree equations (12) ensure that customers are visited exactly once. The capacity inequalities (13) impose the vehicle capacity restrictions and ensure also that the routes are connected. Finally, the constraints (14) and (15) are the integrality conditions.

6.2 Lower bound on vehicles number

The exact number of vehicles necessary to carry the demand of a set \( S \) of customers is the optimal solution of the Bin Packing Problem (BPP) defined by the demands of these customers. Since the BPP is NP-hard, (Lysgaard et al. 2004) used instead an obvious lower bound given by \( r(S) = \lceil q(S)/W \rceil \) where \( q(S) \) is the sum of demands of customers in \( S \).

In order to strengthen the capacity inequalities, the compatibilities between items are partially taken into account by two manners. First, a new constraint is added to the model. Let us recall that \( a_{ij} \) takes the value 1 if and only if \( i \) and \( j \) are conflicting, and 0 otherwise. When \( i \) and \( j \) are conflicting, \( x_{ij} \) is forced to be nil thanks to the following inequalities:

\[
\sum_{j \in V \setminus \{0\}} x_{ij} \leq 1 - a_{ij} \quad (16)
\]

Second, a lower bound on the bin packing problem with conflicts is used to improve \( r(S) \). The considered lower bound is the one developed by Gendreau et al. (2004) and denoted as Constrained Packing Lower Bound. This algorithm starts by computing a maximal clique on the extended conflict graph. In this graph, two vertices are connected by an edge if they are incompatible. The set of edges contains the given input conflicts and the conflicts imposed for those items whose weight sum is greater than the bin capacity. A bin is initialized for every item in the clique, then a transportation algorithm that takes into consideration both the weights and the given conflicts is solved to assign the remaining customer (or fractions of customers) to the bins.

7. EXPERIMENTAL RESULTS

7.1 Benchmark generation

Since the VRPC is a new problem, we have generated our own benchmark in order to evaluate the quality of our heuristics. It was elaborated to take into account real incompatibilities of HazMat as stated in the MSDS (Materials Safety Data Sheet) data base (Environment Canada 2003). This method is based on the fact that HazMat are classified into different categories, according to their chemical and physical properties. Demands are categorized using the Transportation of Dangerous Goods Act (TDGA) - Hazard Classification System (Environment Canada 2003). These materials are sorted in one of nine categories.

For our tests, we use the instances (denoted by CMT) of (Christofides et al., 1979). The number of customers ranges from 50 to 199. We associate randomly a class index to each customer. Each class has a percentage of customers between 4% and 26% of the total number of customers.
7.2 Heuristics results

In the following tables CW stands for Clarke and Wright heuristic, BI for Best Insertion Heuristic, SH for Sweep heuristic. The three new algorithms are denoted respectively by CB1, CB2 and CB3. For each instance, \( N_v \) is the number of used vehicles and \( Cost \) is the solution’s cost.

### Table 1: Heuristics results

<table>
<thead>
<tr>
<th></th>
<th>CW</th>
<th>BI</th>
<th>SH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n )</td>
<td>( N_v )</td>
<td>Cost</td>
</tr>
<tr>
<td>CMT1</td>
<td>50</td>
<td>7</td>
<td>784</td>
</tr>
<tr>
<td>CMT2</td>
<td>75</td>
<td>12</td>
<td>1160</td>
</tr>
<tr>
<td>CMT3</td>
<td>100</td>
<td>9</td>
<td>1187</td>
</tr>
<tr>
<td>CMT4</td>
<td>150</td>
<td>14</td>
<td>1539</td>
</tr>
<tr>
<td>CMT5</td>
<td>199</td>
<td>19</td>
<td>1780</td>
</tr>
<tr>
<td>CMT11</td>
<td>100</td>
<td>9</td>
<td>1700</td>
</tr>
<tr>
<td>CMT12</td>
<td>120</td>
<td>12</td>
<td>1413</td>
</tr>
</tbody>
</table>

The computing time for all the instances was inferior to \( 10^{-5} \) seconds. Generally, SH and BI have very close results for all the instances, while the Clarke and Wright heuristic is widely better. The adapted heuristics also give more interesting results than the new ones concerning the cost. But the new variants are more efficient for the minimization of the number of used vehicles. In fact, their main idea is based on the management of conflicts, and the cost minimization comes in the second place. The results of the three new algorithms are very close concerning the solutions costs and it is difficult to conclude that one is better than the others.

7.3 Metaheuristics results

Several tests for ILS have been done with different values of \( MaxP \) and \( n_t \). We notice that the solution quality increases with the number of iterations. In general, the best solutions are obtained with \( MaxP = 5 \). In fact, the procedure \( Perturb \) have to modify enough the solution in order to escape from the attraction pool of the current local optimum, and in the same time, the perturbation should not be so strong that the solution changes completely by jumping randomly in the solutions space.

Two versions of GRASP-ELS are tested. Each one uses a different heuristic to obtain the initial solutions. The first version calls the modified Clarke and Wright heuristic and the second one the Class Based heuristic. In each version, the classical heuristic is called in the first run of the GRASP, and the randomized heuristic is called in the following \( n_t - 1 \) runs. The first perturbation procedure is used in the two versions. In our tests, the parameter \( n_t \) takes the value 10 (10 initial solutions are created in each version). After several tests, we decided that \( n_t \) and \( n_c \) will take different values in the set 50, 100, 200. These values were chosen in order to improve the ILS results in a reasonable time. In fact, we noticed that for \( n_t \) or \( n_c \) smaller than 50 the cost improvement remains very small. With \( n_t \) or \( n_c \) larger than 200, the obtained cost improvement is insufficient seen the required computing time. Different combinations were tested to determine the best parameters regulation.

In order to reduce the metaheuristic computational time, two counters are used. The first one counts the number of iterations without improvement of \( Soln \) and the second one counts the number of child copies explored at each iteration without improvement of \( BestChildSoln \). The two limit values for the counters are fixed in function of \( n_t \) and \( n_c \). Different values are tested and the chosen ones are: \( Cmax1 = n_t / 4 \) and \( Cmax2 = n_c / 4 \).

In general, the version with the modified Clarke and Wright (CW) heuristic provides more interesting results in both cost and vehicles number minimization. In the following tests, the version with CB1 initial solutions will be omitted. The best statistics are provided by setting the parameters \((n_c, n_t) = (50, 200)\) with 4.5% of cost improvement. This parameters setting will be conserved for the coming tests. The algorithm with these parameters and with initial solutions issued of CW heuristic is denoted \( G-E 1 \). In addition, at each iteration, the perturbation to apply is picked randomly between the two possible procedures.

Although \( G-E 1 \) is more complicated and have much more components than the ILS, its computing time is much less important. This time economy is due to the integration of the counters \( Cmax1 \) and \( Cmax2 \). The interruption of a solution exploration after \( Cmax1 \) or \( Cmax2 \) fruitless local searches allows a significant reduction of the computing time.

### Table 2: Metaheuristics results.

<table>
<thead>
<tr>
<th></th>
<th>ILS</th>
<th>GRASP-ELS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( N_v )</td>
<td>Cost</td>
</tr>
<tr>
<td>CMT1</td>
<td>6</td>
<td>742</td>
</tr>
<tr>
<td>CMT2</td>
<td>12</td>
<td>1129</td>
</tr>
<tr>
<td>CMT3</td>
<td>9</td>
<td>1119</td>
</tr>
<tr>
<td>CMT4</td>
<td>14</td>
<td>1517</td>
</tr>
<tr>
<td>CMT5</td>
<td>18</td>
<td>1712</td>
</tr>
<tr>
<td>CMT11</td>
<td>8</td>
<td>1576</td>
</tr>
<tr>
<td>CMT12</td>
<td>11</td>
<td>1386</td>
</tr>
</tbody>
</table>

In Table 2, the best solution found by the two metaheuristics are presented. The column \( Imp. \) represents the improvement in percentage obtained by GRASP-ELS metaheuristic with regards to ILS results. The ILS improves the results of CW heuristic in all instances (average of 4.3% of improvement). In average, the improvement of \( N_v \) is about 6.25%. The results of the GRASP-ELS metaheuristic show a cost improvement of the solution for all the instances with an average of 4.5% with regard to the ILS and an improvement on the number of vehicles that reaches 7%.

7.4 Lower bound results

In Table 3, the results of two branch-and-cut algorithms are compared with the lower bound obtained with the linear relaxation of the linear program model of the VRPC (11)-(15). \( A1 \) stands for the complete branch-and-cut algorithms that integrates Gendreau’s bound on vehicles number and \( A2 \) is the same algorithm without the bound
on the vehicles number. Both A1 and A2 improve strongly the linear relaxation (average of 112.37% for A1). The impact of the integration of the bound on number of vehicles is only about 3% (columns 5 and 6 give the improvements obtained by A1 with regards to A2 and the linear relaxation).

The gap between the lower bound and the metaheuristic remains important (Table 4, average of 27%). In fact, the branch-and-cut algorithm takes into account the existing conflicts between items only by adding equation (16) and using Gendreau’s lower bound for the BPC to improve the trivial value of \( r(s) \). In the compact formulation (11)-(15), there is no index for the vehicles so it seems difficult to develop valid inequalities that avoid transporting two conflicting items in the same vehicle.

Table 3: Lower bound results.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>Relax.</th>
<th>Improvement by A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imp.</td>
<td>A2</td>
<td>Imp. Relax.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMT1</td>
<td>652.14</td>
<td>601.68</td>
<td>380.74</td>
<td>8.35%</td>
</tr>
<tr>
<td>CMT2</td>
<td>919.39</td>
<td>884.83</td>
<td>486.82</td>
<td>3.91%</td>
</tr>
<tr>
<td>CMT3</td>
<td>932.83</td>
<td>932.31</td>
<td>580.78</td>
<td>0.06%</td>
</tr>
<tr>
<td>CMT4</td>
<td>1136.11</td>
<td>1093.31</td>
<td>576.97</td>
<td>3.91%</td>
</tr>
<tr>
<td>CMT5</td>
<td>1288.76</td>
<td>1271.94</td>
<td>640.23</td>
<td>1.32%</td>
</tr>
<tr>
<td>CMT11</td>
<td>953.67</td>
<td>949.91</td>
<td>358.32</td>
<td>0.4%</td>
</tr>
<tr>
<td>CMT12</td>
<td>918.41</td>
<td>888.43</td>
<td>309.42</td>
<td>3.36%</td>
</tr>
</tbody>
</table>

Table 4: Distance between LB and GRASP-ELS.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>E</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMT1</td>
<td>652</td>
<td>738</td>
<td>17.17%</td>
</tr>
<tr>
<td>CMT2</td>
<td>919</td>
<td>1084</td>
<td>17.90%</td>
</tr>
<tr>
<td>CMT3</td>
<td>933</td>
<td>1090</td>
<td>16.85%</td>
</tr>
<tr>
<td>CMT4</td>
<td>1136</td>
<td>1439</td>
<td>26.66%</td>
</tr>
<tr>
<td>CMT5</td>
<td>1289</td>
<td>1669</td>
<td>29.50%</td>
</tr>
<tr>
<td>CMT11</td>
<td>954</td>
<td>1505</td>
<td>57.81%</td>
</tr>
<tr>
<td>CMT12</td>
<td>918.13</td>
<td>1303</td>
<td>41.88%</td>
</tr>
</tbody>
</table>

8. CONCLUSION

This paper introduces a new problem which is The Vehicle Routing Problem with Conflicts. We started by adapting some classical VRP heuristics and proposed a new heuristic more suitable to HazMat transportation. Then two metaheuristics are proposed to improve the heuristic results. The results of the GRASP-ELS metaheuristic show a cost improvement of the solution for all the instances with an average of 4.5% with regard to the ILS.

The first lower bound for the VRPC improves strongly the bound obtained by the linear relaxation. Its performance can be improved by the creation of new cuts based on the concept of conflicts detection.

REFERENCES


