An iterative approach to generate $k$ dissimilar VRP solutions

Luca Talarico, Kenneth Sörensen, Johan Springael

COMEX project meeting, Brussels, Belgium
19 November 2013
Summary

Introduction

Problem Formulation

Solution approach
  Metaheuristic Components

Computational results

Future work
Introduction

**Context**  ⇨  Cash and valuables in transportation sector

**Risk**  ⇨  Crime is a real challenge

- In the U.K. alone, there is an estimated £ 500 billion transported each year (£ 1.4 billion per day);
- In 2008, there were 1,000 documented attacks against CIT couriers in the UK;
- With the downturn in the global economy, CIT crime is on the rise:
  - *February 2013 Foggia Italy: Robbery of 300 Kg of Gold;*
  - *February 2013 Brussels Belgium: Robbery of 50 Million €;*
  - *March 2013 Varese Italy: Robbery of 10 Million €.*
Introduction

- In many European countries, cash-in-transit companies must by law determine three different routes for each of their vehicles when transporting cash in order to:
  - Easily change its plans in case of unforeseen circumstances (accidents, road works, etc.);
  - Increase security by making the vehicle routes more unpredictable.
- Additionally, the same route cannot be used more than two consecutive times.

k-dissimilar Vehicle Routing Problem ($kd$-VRP) to support this optimization problem!
kd-VRP Problem

Given a similarity metric between any pair of VRP solutions, a feasible solution to the kd-VRP is a set of k feasible VRP solutions for which the difference between each pair of alternative solutions is larger than a certain threshold.

Each alternative solution must obey the traditional VRP constraints: (1) all customers are visited once; (2) all vehicles begin and end at the depot; (3) the capacity of the vehicle must not be exceeded.

The objective of the kd-VRP is to minimize the cost (total distance) of the worst alternative solution in the set.

Motivation ➞ for some real-life applications the constraint that imposes k edge-disjoint VRP solutions (no shared edges between the alternative solutions) might be too stringent!
kd-VRP Problem

- The *kd-VRP* is closely related to *m-Peripatetic Vehicle Routing Problem* (*m-PVRP*):
  - Multiple usage of an edge is a hard constraint in *m-PVRP* $\Rightarrow$ soft constraint in the *kd-VRP*;
  - *m-PVRP* minimizes the total cost over all periods $\Rightarrow$ *kd-VRP* minimizes the worst-case cost over all alternative solutions.

- **Applications:**
  - Money collection and distribution;
  - Transportation of dangerous materials to better share and mitigate the risk of accidents;
  - Design of patrol routes for security agents who must follow partially different routes over time.
Similarity indexes

The index to measure similarities between VRP solutions has been borrowed from the literature on shortest path.

\[
\text{Dis}(P_1, P_2) = 1 - \frac{1}{2} \left[ \frac{L(P_1 \cap P_2)}{L(P_1)} + \frac{L(P_1 \cap P_2)}{L(P_2)} \right]
\]  

(1)

\[
\delta(y_i, y_j) = \max_{l, m \in \mathbb{N}} \frac{1}{2} \left[ \frac{c_s(r_{y_i}^l, r_{y_j}^m)}{c(r_{y_i}^l)} + \frac{c_s(r_{y_i}^l, r_{y_j}^m)}{c(r_{y_j}^m)} \right]
\]  

(2)
A solution for the problem with $k = 3$, instance $Fn45k4$. 
kd-VRP formulation: Slave problem

\[
\begin{align*}
\min \sum_{h \in N} \sum_{(i,j) \in E} c_{ij} x_{ij}^h \\
\text{s.t.} \\
& \sum_{j \in V \setminus \{0\}} x_{0j}^h = \sum_{j \in V \setminus \{0\}} x_{j0}^h = 1 \quad \forall h \in N \\
& \sum_{h \in N} \sum_{j \in V} x_{ij}^h = \sum_{h \in N} \sum_{j \in V} x_{ji}^h = 1 \quad \forall i \in V \setminus \{0\} \\
& \sum_{i \in V \setminus \{0\}} \sum_{j \in V} d_{ij} x_{ij}^h \leq C \quad \forall h \in N \\
& \sum_{h \in N} \sum_{i \in Q} \sum_{j \notin Q} x_{ij}^h \geq 1 \quad \forall Q \subset V; Q \neq \emptyset \\
x_{ij}^h \in \{0, 1\} \quad \forall (i,j) \in E; \forall h \in N
\end{align*}
\]
kd-VRP formulation: Master problem

\[
\min \max_{\forall i \in \{1, \ldots, k\}} f(y_i) \quad (1)
\]

s.t.
\[
\delta(y_i, y_j) \leq T_s \quad \forall i, j \in \{1, \ldots, k\}; i \neq j \quad (2)
\]
\[
y_i \in \Omega \quad \forall i \in \{1, \ldots, k\} \quad (3)
\]

- **kd-VRP key controls**
  - \( k \Rightarrow \) Number of alternative solutions to be generated;
  - \( T_s \Rightarrow \) Maximum similarity threshold.
Solution approach

- The metaheuristic has been named Iterative Penalty Method for the \( kd \)-VRP (IPM\_kd).

- **Heuristic parameters**
  - \( I \) \(\rightarrow\) Number of restarts of the algorithm;
  - \( P \) \(\rightarrow\) Number of times the *Perturbation heuristic* is applied;
  - \( \alpha \) \(\rightarrow\) Number of closer neighbour vertices to be considered in the Repair heuristic;
  - \( \omega \) \(\rightarrow\) Maximum % number of routes to be destroyed;
  - \( \beta \) \(\rightarrow\) Penalty factor used in the *Penalization function*. 
Algorithm 1: IPM $k_d$ metaheuristic structure

Initialize both $k_d$-VRP and Heuristic parameters $k, T_s, I, P, \alpha, \beta$ and $\omega$;

$l \leftarrow 0$;

while ($l < I$) do

    $S \leftarrow \{\emptyset\}$;

    while ($|S| < k$) do

        $i \leftarrow |S|$;

        $p \leftarrow 0$;

        Let $y_i^*$ be the best $i$-th alternative solution found so far and $f(y_i^*)$ its cost;

        Let $y_i$ be the current $i$-th alternative solution and $f(y_i)$ its cost;

        $y_i^* \leftarrow \{\emptyset\}$, $y_i \leftarrow \{\emptyset\}$, $f(y_i^*) \leftarrow \infty$, $f(y_i) \leftarrow \infty$;

        while ($p < P$) do

            if ($p == 0$) then

                $y_i \leftarrow \text{Lin-Kernighan}(y_i) \cup \text{Splitting}(y_i)$;

            else

                $y_i \leftarrow \text{Perturbation}(y_i^*)$;

            $y_i \leftarrow \text{VND}(y_i)$;

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

                $y_i^* \leftarrow y_i$;

                $f(y_i^*) \leftarrow f(y_i)$;

                $p ++$;

            end if

        end while

        if ($i == 0$) then

            add $y_i^*$ to $S$;

            $y_i^* \leftarrow \text{PenalizationFunction}(y_i^*)$;

        else

            while ($\delta(y_i^*, y_h) > T_s \forall h \in 1, \ldots, i - 1$) do

                $y_i^* \leftarrow \text{PenalizationFunction}(y_i^*)$;

                $y_i^* \leftarrow \text{VND}(y_i^*)$;

            end while

            add $y_i^*$ to $S$;

        end if

    end while

    if ($|S| == k$) then

        $l ++$;

    end if

end while

Return the best set $S$ found so far;
IPM\_kd components

- Lin–Kernighan heuristic & Prins splitting procedure;
- Perturbation function;
- Penalization function;
- Variable Neighbourhood Descent heuristic VND.
Lin–Kernighan & Prins splitting

- Employed to find an initial alternative solution using the current cost matrix:
  - Lin-Kernighan heuristic $\rightarrow$ deterministic approach to generate optimal or nearoptimal solutions for the symmetric travelling salesman problem (TSP);
  - Prins splitting procedure $\rightarrow$ split the giant tour in feasible routes, by finding the shortest path from node 0 to node $n$ in the auxiliary graph.
Diversification mechanism to escape from local optima, while looking for the current alternative solution $y_i^*$.

- **Destroy phase**: $\omega \cdot N^+$ routes are destroyed from $y_i^*$;
- **Repair phase**: starting from the non destroyed routes of $y_i^*$ new routes are generated by applying a greedy randomized nearest neighbourhood heuristic, using $\alpha$ as a parameter.

+ $N = \text{number of routes in } y_i^*$.  

\[ \]
Penalization function

- It updates the cost matrix with a dual purpose:
  - After a new feasible alternative solution \( y_i \) is added to set \( S \):
    - It forces the search process to move to a significantly different part of the search space by increasing the cost of the edges used in \( y_i \) by a percentage \( \beta \).
  - Every time an infeasible alternative solution has been generated (similarity higher than \( T_S \)):
    - The cost of the shared edges is penalized by a percentage \( \beta \);
    - This operation forces the algorithm to discard the shared edges, guiding the VND heuristic towards a feasible solutions.
It improves a new alternative solution as soon as it is generated;
It guides the algorithm towards a feasible alternative solution.

VND → sequential Variable Neighbourhood Descent block in which 7 local search operators are used:

- *Intra Route Local Search Operators* which attempt to improve a single route;
- *Inter Route Local Search Operators* which change more than one route simultaneously.
First-improvement descent strategy.

- **Intra Route Local search:**
  - Or-Opt Operator;
  - Relocate Operator;
  - Two-Opt Operator.

- **Inter Route Local search:**
  - Exchange Operator;
  - Relocate Operator;
  - Cross-Exchange Operator;
  - Two-Opt Operator.
The IPM\_kd metaheuristic has been tested using 51 benchmark instances taken from the VRP library.

Given a similarity measure between VRP solutions, an instance of the VRP can be transformed into an instance of the kd-VRP by adding only two parameters: $k$ and $T_s$.
Computational Results

- All computational experiments were performed using a machine with an Intel core i7-2760QM 2.40GHz processor with 4GB RAM.

- The computational experiments have been carried out in three different phases:
  - In the first stage the IPM\_kd heuristic parameters were tuned by running a full factorial experiment on a subset of the benchmark instances.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
<th>#</th>
<th>best setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>Number of times the IPM_kd algorithm is restarted</td>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$P$</td>
<td>Number of times the Perturbation heuristic is applied</td>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Maximum percentage number of routes in best solution found so far to be destroyed</td>
<td>20,30,40,</td>
<td>9</td>
<td>40%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\ldots$,90,100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Number of closer neighbour vertices to be considered in the Repair heuristic</td>
<td>1,2,3,$\ldots$,9,10</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>
In the second stage, using the best configuration of the heuristic parameters, all the 51 benchmark instances are solved using different values of the $kd$-VRP parameters:

- $k \in \{1, 2, 3, 4, 5\}$
- $T_s \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$
- $\beta \in \{5, 10, 20, 40, 50, 100\}$
Figure: Relationship between the average solutions obtained and the average computational time in relation to different values of the $\beta$, while $k = 3$. 

Computational Results
Figure: Relationship between the average solutions obtained and the average computational time in relation to different values of the similarity threshold $T_s$, while $k = 3$
Figure: Average computational time, best and average percentage gap from the best known VRP solutions in relation to different values of $k$. 

<table>
<thead>
<tr>
<th>$k$</th>
<th>Avg. Sol. (%)</th>
<th>Best Sol. (%)</th>
<th>Avg. Computational Time (s.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35%</td>
<td>5%</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>30%</td>
<td>10%</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>25%</td>
<td>15%</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>20%</td>
<td>20%</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>15%</td>
<td>25%</td>
<td>6</td>
</tr>
</tbody>
</table>
In the third stage of the computational experiments, using the best parameter configuration for the IPM\(\_kd\), all the benchmark instances have been solved, fixing the problem parameters as follows:

- \(T_S = 20\%\) (and thus the \(k\) alternative solutions, contained in \(S\), must be different from each other by at least 80\%);
- \(\beta = 5\%\).
When $k$ increases the computational time grows approximately as a linear function.

The gap between the cost of the $k$-th alternative solution and the best known VRP solution is higher when solving smaller instances.

The average percentage gap from the best known VRP solutions is only 25% when $k = 5$ and $T_s = 20\%$.

Good level of robustness as the difference between the best and the average solutions’ costs (15 runs) is on average only 3.74%.

The average time needed to solve an instance in the benchmark set, when $k = 5$ and $T_s = 20\%$, is below 14 seconds.
Future work

- Introduction of risk indices to generate alternative and safe routes in the domain of the hazardous materials transportation or in the cash in transit sector.
- Variants of the $kd$-VRP problem including additional real life constraints such as:
  - Time windows;
  - Route length restrictions;
  - Precedence relations between vertices.
An iterative approach to generate $k$ dissimilar VRP solutions

Luca Talarico, Kenneth Sörensen, Johan Springael

COMEX project meeting, Brussels, Belgium
19 November 2013

luca.talarico@ua.ac.be