Une heuristique de génération de colonnes pour le problème de tournées de véhicules avec faisabilité boîte noire

JFPC 2012, Toulouse

Florence Massen$^1$ $^a$ Yves Deville$^1$ Pascal Van Hentenryck$^2$

$^1$ICTEAM, Université catholique de Louvain, Belgique

$^2$Optimization Research Group, NICTA, University of Melbourne, Australie

$^a$Supported by the National Research Fund, Luxembourg
Vehicle Routing is still an open problem

- After 50 years Vehicle Routing still a very active research domain
- Past decades concentrated on different variants of the problem
- Research now focuses on rich problems
- Often combination of a number of different VRP variants
- But also combination of VRP with combinatorial side-problems
Pheromone-based Column Generation for the VRPBB

1 Problem Description

2 VRPBB as a Set-Partitioning Problem

3 Column Generation Heuristic for the VRPBB

4 Collector Ants & Pheromones

5 Experimental Results
1 Problem Description
   - Capacitated Vehicle Routing Problem
   - VRPs with combinatorial side-problems
   - VRP with Black Box Feasibility

2 VRPBB as a Set-Partitioning Problem

3 Column Generation Heuristic for the VRPBB

4 Collector Ants & Pheromones

5 Experimental Results
CVRP – Problem Formulation

Given:
- a complete, undirected, weighted graph $G = (V, E)$
- $V = \{v_0, v_1, \ldots, v_n\}$, $v_0 =$ the depot vertex, $v_1, \ldots, v_n =$ the customers
- weight of edge $e_{ij} =$ cost of traveling $(t_{ij})$ from vertex $i$ to vertex $j$ or vice-versa
- for each customer $i$ a demand $d_i =$ total weight of the items to be delivered at $i$
- a set of $K$ vehicles of capacity $Q$
CVRP – Problem Formulation

Find a set of routes such that:

- number of routes \( \leq K \)
- each customer is visited by exactly one vehicle
- the sum of the demands on a route does not exceed the vehicle’s capacity
- the total travel cost is minimized

Total travel cost = 9
VRPs with combinatorial side-problems covered in literature

**VRP with Loading Constraints, side-problem: Loading**

[Iori et al., 2007, Gendreau et al., 2006]

- 2D: given item sequence, existence of feasible loading verified in $O(m^3)$
- 3D: given item sequence, existence of feasible loading verified in $O(m^4)$

where $n \leq m$, $m =$ # number of items in route, $n =$ # customers visited in route

**VRPTW with Driver Rules, side-problem: Scheduling**

[Archetti and Savelsbergh, 2009, Prescott-Gagnon et al., 2010]

- existence of feasible schedule verified in $O(n^3)$ (US regulation)
- existence of feasible schedule verified in $O(n^k)$ (European regulation)

where $n =$ # customers visited in route, $k =$ possible (short, long) break/rest options
Typically dedicated approaches are developed

- These problems have been tackled with dedicated approaches
- Demanding the effort to find for the combinatorial side-problem
  - lower bounds
  - functions evaluating the quality of a solution
  - functions evaluating the violation degree of an infeasible solution
  - ...
- Often not possible to find good bounds or evaluations for complicated side-problems

⇒ Need for a generic approach
VRP, or variant, with set of unknown constraints $BB$

A route can be feasible only if it satisfies all constraints in $BB$

Black box function $\text{feas}(r)$ verifying feasibility of route $r$ wrt $BB$ is provided

$\text{feas}(r)$ is of non-linear time complexity in the length of $r$
Advantages of VRP with Black Box Feasibility

- Can accommodate combinations of routing and any combinatorial problems
- Switching from one problem to another by plugging different black box function
- Optimization approach for the VRPBB generic to all possible combinations
A route is:

- VRP-feasible if feasible wrt underlying VRP
  - e.g. respects capacity constraints
  - e.g. no customer appears twice
- $BB$-feasible if feasible wrt $BB$
- feasible if $(VRP, BB)$-feasible

This easily extends to solutions
Pheromone-based Column Generation for the VRPBB

1. Problem Description
2. VRPBB as a Set-Partitioning Problem
3. Column Generation Heuristic for the VRPBB
4. Collector Ants & Pheromones
5. Experimental Results
Metaheuristic approach difficult

We're not looking for the optimal solution, just a good one.

⇒ Metaheuristics are the preferred solution methodology for VRPs [Laporte, 2007], but:

- At each step of the metaheuristic we would need to find a set of routes
  - that are $BB$-feasible (and VRP-feasible)
  - that can be combined into a feasible solution
Reformulate VRPBB as a Set-partitioning problem (SPP)

Among all feasible solutions select the one that
- minimizes the total travel cost is minimized
Reformulate VRPBB as a Set-partitioning problem (SPP)

Among all feasible solutions select the one that
- minimizes the total travel cost is minimized

Shift focus on routes ↓
Reformulate VRPBB as a Set-partitioning problem (SPP)

Among all feasible solutions select the one that
- minimizes the total travel cost is minimized

Shift focus on routes ↓

Among all feasible routes select the subset such that
- each customer is visited once
- subset contains \( \leq K \) routes
- the total travel cost is minimized
Reformulate VRPBB as a Set-partitioning problem (SPP)

\[
\begin{align*}
\text{Min} & \quad \sum_{r \in \mathcal{R}} c_r x_r \\
\text{s.t.} & \quad \sum_{r \in \mathcal{R}} v_{ir} x_r = 1 \quad \forall i \in V \setminus 0 \\
& \quad \sum_{r \in \mathcal{R}} x_r \leq K \\
& \quad x_r \in \{0, 1\} \quad \forall r \in \mathcal{R}
\end{align*}
\]

\[v_{ir} = \begin{cases} 
1 & \text{if customer } i \text{ visited in route } r \\
0 & \text{else}
\end{cases}\]

Where:
- \(\mathcal{R}\) the set of all feasible routes,
- \(x_r\) a variable indicating if route \(r\) is selected in the solution,
- \(c_r\) the cost of route \(r\)
Reformulate VRPBB as a Set-partitioning problem (SPP)

Min \sum_{r \in \mathcal{R}} c_r x_r

s.t. \sum_{r \in \mathcal{R}} v_{ir} x_r = 1 \quad \forall i \in \mathcal{V} \setminus 0

\sum_{r \in \mathcal{R}} x_r \leq K

x_r \in \{0, 1\} \quad \forall r \in \mathcal{R}

v_{ir} = \begin{cases} 
1 & \text{if customer } i \text{ visited in route } r \\
0 & \text{else}
\end{cases}

Where:

- \mathcal{R} the set of all feasible routes,
- \(x_r\) a variable indicating if route \(r\) is selected in the solution,
- \(c_r\) the cost of route \(r\)
Reformulate VRPBB as a Set-partitioning problem (SPP)

\[\begin{align*}
\text{Min} & \quad \sum_{r \in \mathcal{R}} c_r x_r \\
\text{s.t.} & \quad \sum_{r \in \mathcal{R}} v_{ir} x_r = 1 \quad \forall i \in \mathcal{V}\setminus 0 \\
& \quad \sum_{r \in \mathcal{R}} x_r \leq K \\
& \quad x_r \in \{0, 1\} \quad \forall r \in \mathcal{R}
\end{align*}\]

\[v_{ir} = \begin{cases} 
1 & \text{if customer } i \text{ visited in route } r \\
0 & \text{else}
\end{cases}\]

Where:

- \(\mathcal{R}\) the set of all feasible routes,
- \(x_r\) a variable indicating if route \(r\) is selected in the solution,
- \(c_r\) the cost of route \(r\)
In practice work on restricted set $\mathcal{R}^*$

- Generating $\mathcal{R}$, the set of all feasible routes is intractable
In practice work on restricted set \( \mathcal{R}^* \)

- Generating \( \mathcal{R} \), the set of all feasible routes is intractable

\[ \Rightarrow \] Use column generation-based approach

- Work on restricted set of feasible routes \( \mathcal{R}^* \)
- Iteratively enrich \( \mathcal{R}^* \) with new feasible routes
- New route in \( \mathcal{R}^* \) corresponds to new \( x_r \) variable in SPP
- New route in \( \mathcal{R}^* \) corresponds to new column in SPP constraint matrix
Pheromone-based Column Generation for the VRPBB

1. Problem Description

2. VRPBB as a Set-Partitioning Problem

3. Column Generation Heuristic for the VRPBB

4. Collector Ants & Pheromones

5. Experimental Results
Column Generation - basic reminder

- Method to optimize a linear problem (Master Problem, MP) with too many variables
- Works on a restricted version of the problem (Restricted Master Problem, RMP)
- Each iteration consists in
  1. solving the current RMP to optimality, which provides dual costs
  2. using the dual costs to generate new interesting columns (new variables)
  3. adding the generated columns to the current RMP
- In our case generating columns = generating feasible routes
Column Generation Heuristic for the VRPBB - solving the RMP

High-Level Algorithm

1: initialize $R^*$, $\Pi$
2: while ¬ stopping criterion do
3: $R^* \leftarrow R^* \cup \text{generateRoutes}(\Pi)$
4: $\Pi \leftarrow \text{solveRMP}(R^*)$
5: end while
6: return $\text{solveSPP}(R^*)$

- Restricted Master Problem (RMP) = relaxed SPP on $R^*$
- $\Pi$ denotes the dual costs obtained by solving the RMP
- generateRoutes heuristically generates feasible routes
- solveRMP solves the current RMP
- solveSPP solves the integer SPP on the set of collected feasible routes $R^*$
Column Generation Heuristic for the VRPBB - generating & adding columns

High-Level Algorithm

1: initialize $\mathcal{R}^*$, $\Pi$

2: while ¬ stopping criterion do

3: $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \text{generateRoutes}(\Pi)$

4: $\Pi \leftarrow \text{solveRMP}(\mathcal{R}^*)$

5: end while

6: return $\text{solveSPP}(\mathcal{R}^*)$

- Restricted Master Problem (RMP) = relaxed SPP on $\mathcal{R}^*$
- $\Pi$ denotes the dual costs obtained by solving the RMP
- $\text{generateRoutes}$ heuristically generates feasible routes
- $\text{solveRMP}$ solves the current RMP
- $\text{solveSPP}$ solves the integer SPP on the set of collected feasible routes $\mathcal{R}^*$
High-Level Algorithm

1: initialize $R^*$, $\Pi$
2: \textbf{while} \neg \text{ stopping criterion} \textbf{do}
3: \qquad $R^* \leftarrow R^* \cup \text{generateRoutes}(\Pi)$
4: \qquad $\Pi \leftarrow \text{solveRMP}(R^*)$
5: \textbf{end while}
6: \textbf{return} \text{solveSPP}(R^*)

- Restricted Master Problem (RMP) = relaxed SPP on $R^*$
- $\Pi$ denotes the dual costs obtained by solving the RMP
- \text{generateRoutes} heuristically generates feasible routes
- \text{solveRMP} solves the current RMP
- \text{solveSPP} solves the integer SPP on the set of collected feasible routes $R^*$
Generating **feasible** routes

`generateRoutes` should return only \((\text{VRP}, \text{BB})\)-feasible routes

- Generating \textit{VRP}-feasible routes is trivial
- But we want the routes to be \textit{BB}-feasible too!
- Black box function \textit{only} provides \textit{feasibility test}
Generating feasible routes

generateRoutes should return only \((\text{VRP}, \text{BB})\)-feasible routes

- Generating \text{VRP}-feasible routes is trivial
- But we want the routes to be \text{BB}-feasible too!
- Black box function only provides feasibility test

- Routes in \(\mathcal{R}^*\) are \((\text{VRP}, \text{BB})\)-feasible routes
- Routes in current RMP solution are optimal given current \(\mathcal{R}^*\) and are feasible
- Similar routes could have similar qualities
Generating **feasible** routes

generateRoutes should return only \((VRP, BB)\)-feasible routes

- Generating \(VRP\)-feasible routes is trivial
- But we want the routes to be \(BB\)-feasible too!
- Black box function **only** provides feasibility test

- Routes in \(\mathcal{R}^*\) are \((VRP, BB)\)-feasible routes
- Routes in current RMP solution are **optimal** given current \(\mathcal{R}^*\) and are **feasible**
- Similar routes could have similar qualities

\[ \Rightarrow \text{generate routes similar to those in RMP solutions} \]
Generating routes similar to those in RMP solutions

- We want to generate routes similar to those in the RMP solutions
- heuristic in generateRoutes must be guided to do this
- One way to implement this is using:
  - repeated heuristic executions called ants
  - pheromones influencing the decisions of the ants
Column Generation Approach for the VRPBB - Pheromones

High-Level Algorithm

1: initialize $\mathcal{R}^*$, $\Pi$, $\tau$, $Sol$
2: while ¬ stopping criterion do
3: $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup$ generateRoutes($\Pi$, $\tau$)
4: $\Pi$, $Sol \leftarrow$ solveRMP($\mathcal{R}^*$)
5: $\tau \leftarrow$ updatePheromones($Sol$)
6: end while
7: return solveSPP($\mathcal{R}^*$)

- $\tau$ contains the pheromone quantity for each edge in problem graph
High-Level Algorithm

1: initialize $\mathcal{R}^*$, $\Pi$, $\tau$, $Sol$, $\Psi$

2: while ¬ stopping criterion do

3: $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup$ generateRoutes($\Pi$, $\tau$, $\Psi$)

4: $\Pi$, $Sol \leftarrow$ solveRMP($\mathcal{R}^*$)

5: $\tau \leftarrow$ updatePheromones($Sol$)

6: end while

7: return solveSPP($\mathcal{R}^*$, $\Psi$)

- $\Psi$ contains the feasibility information of checked routes
Pheromone-based Column Generation for the VRPBB

1. Problem Description
2. VRPBB as a Set-Partitioning Problem
3. Column Generation Heuristic for the VRPBB
4. Collector Ants & Pheromones
5. Experimental Results
Column Generation Approach for the VRPBB

High-Level Algorithm

1. initialize $\mathcal{R}^*$, $\Pi$, $\tau$, $Sol$, $\Psi$

2. while ¬ stopping criterion do

3. $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \text{generateRoutes}(\Pi, \tau, \Psi)$ // via collector ants

4. $\Pi, Sol \leftarrow \text{solveRMP}(\mathcal{R}^*)$

5. $\tau \leftarrow \text{updatePheromones}(Sol)$

6. end while

7. return $\text{solveSPP}(\mathcal{R}^*, \Psi)$
Generating Columns via Collector Ants

- Collector ants are a variant of savings-based ants [M. Reimann et al., 2002]
- Generate and collect feasible routes to enrich $\mathcal{R}^*$
- Each ant executes a randomized version of the savings heuristic
Collector ants execute savings heuristic

- ant starts from an initial state where each customer is in its own route
Collector ants execute savings heuristic

- at each step, the ant merges two routes in current state
- merge consists in adding a new edge (and removing two) to the current state
Collector ants execute savings heuristic

- ant continues merging routes until no further merges is possible
Collector ants execute savings heuristic

- ant continues merging routes until no further merges is possible
An ant’s life

- At each step ant computes list $\Omega$ of most attractive merges

- Attractiveness depends on:
  - distance saved through merge
  - pheromone deposit on introduced edge

- One of the merges in $\Omega$ is randomly (probability based on attractiveness) selected and executed

- Dual costs from the current RMP are used in the construction of $\Omega$

- To compute list $\Omega$ ant must evaluate $BB$-feasibility of several routes

- Ant collects all routes that are positively evaluated for $BB$-feasibility
Column Generation Approach for the VRPBB

High-Level Algorithm

1. initialize $R^*$, $\Pi$, $\tau$, $Sol$, $\Psi$

2. while ¬ stopping criterion do

3. $R^* \leftarrow R^* \cup \text{generateRoutes}(\Pi, \tau, \Psi)$ // via collector ants

4. $\Pi, Sol \leftarrow \text{solveRMP}(R^*)$

5. $\tau \leftarrow \text{updatePheromones}(Sol)$

6. end while

7. return $\text{solveSPP}(R^*, \Psi)$
Pheromones guide ants

- Each ant disposes of the pheromone matrix $\tau$
- Pheromones influence probability of a merge to be chosen
- Pheromones should lead ants to build routes similar to those in RMP solutions
- Put pheromones on edges appearing in routes selected in current RMP solution
Pheromone Update

For every edge $ij$ ($i = 0..n$, $j = 0..n$, $i \neq j$) we do:

$$
\tau_{ij} = \rho \tau_{ij} + \sigma_{ij} \varepsilon
$$

where

- $0 \leq \rho \leq 1 = \text{trail persistence}$
- $\varepsilon = \text{a small constant}$
- $\sigma_{ij} = \text{number of times } ij \text{ appears in current RMP solution}$
1 Problem Description

2 VRPBB as a Set-Partitioning Problem

3 Column Generation Heuristic for the VRPBB

4 Collector Ants & Pheromones

5 Experimental Results
   ▪ Application to 3L-CVRP
   ▪ Application to MP-VRP
3L-CVRP : CVRP + 3-dimensional loading

Q = 25
3L-CVRP: CVRP + 3-dimensional loading

\[ d_1 = 12 \]
\[ d_2 = 6 \]
\[ d_3 = 4 \]
\[ d_4 = 5 \]
\[ d_5 = 10 \]
\[ d_6 = 8 \]
\[ d_7 = 2 \]

\[ Q = 25 \]
3L-CVRP : CVRP + 3-dimensional loading
Existing Work on the 3L-CVRP

- [Gendreau et al., 2006] : Tabu Search + Tabu Search for Loading
- [Tarantilis et al., 2009] : Guided Tabu Search. + Heuristics for Loading
- [Bortfeldt, 2010] : Hybrid Tabu Search + Bounded Tree Search for Loading
Experimental Results – Setup

- Described approach implemented in C++, using CPLEX as LP solver
- Loading black box implemented (in C++) as described in [Bortfeldt, 2010]
- Tested on benchmark instances from [Gendreau et al., 2006]
- Time limits imposed on Column Generation phase correspond to total time limits used in [Gendreau et al., 2006]
- Averages over 10 Runs
Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$z_{min}$</th>
<th>$z_{avg}$</th>
<th>$sec_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTS ([Bortfeldt, 2010])</td>
<td>939.66</td>
<td>959.05</td>
<td>219.5</td>
</tr>
<tr>
<td>ACO ([Fuellerer et al., 2010])</td>
<td>960.87</td>
<td>966.67</td>
<td>1793.1</td>
</tr>
<tr>
<td>GTS ([Tarantilis et al., 2009])</td>
<td>997.18</td>
<td>–</td>
<td>2415.9</td>
</tr>
<tr>
<td>TS ([Gendreau et al., 2006])</td>
<td>1042.26</td>
<td>1042.26</td>
<td>4200</td>
</tr>
<tr>
<td>(this work)</td>
<td><strong>935.68</strong></td>
<td><strong>939.66</strong></td>
<td><strong>2840.3</strong></td>
</tr>
</tbody>
</table>

Average values over all instances

- $z$ is the solution cost (total distance)
- Best solution value: proposed generic approach improves best known
- Average solution value: proposed generic approach improves best known
- Total execution time: expected slowdown due to genericity
MP-VRP : VRP + multi-pile loading
MP-VRP: VRP + multi-pile loading
MP-VRP : VRP + multi-pile loading
Existing Work on the MP-VRP

- [Doerner et al., 2007] : Ant Colony Optimization and Tabu Search
  + Heuristics for Loading

- [Tricoire et al., 2009] : Variable Neighborhood Search
  + Heuristics and Exact Method for Loading
Experimental Results – Setup

- Described approach implemented in C++, using CPLEX as LP solver
- Loading black box (in C++) kindly provided by Fabien Tricoire
- Tested on benchmark instances from [Doerner et al., 2007]
- Time Limit imposed on Column Generation phase as in [Tricoire et al., 2009]
- Averages over 10 Runs
## Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$Z_{min}$</th>
<th>$Z_{avg}$</th>
<th>$sec_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNS ([Tricoire et al., 2009])</td>
<td>1369.12</td>
<td>1376.16</td>
<td>&lt;1800</td>
</tr>
<tr>
<td>ACO ([Doerner et al., 2007])</td>
<td>1385.25</td>
<td>1392.13</td>
<td>1899.8</td>
</tr>
<tr>
<td>TS ([Doerner et al., 2007])</td>
<td>1402.53</td>
<td>1402.53</td>
<td>4954.5</td>
</tr>
<tr>
<td>(this work)</td>
<td>1385.72</td>
<td>1399.83</td>
<td>1765</td>
</tr>
</tbody>
</table>

Average values over all instances

- $z$ is the solution cost (total distance)
- Best solution value: proposed generic approach comparable to existing approaches
- Average solution value: proposed generic approach comparable to existing app.
- Total execution time : comparable to existing approaches
Conclusion & Future Work

Conclusion

- New generic VRP variant, VRP with Black Box Feasibility
- Allows to accommodate combination of VRP and other Combinatorial problems
- New algorithm designed especially for this generic problem
- Experiments on 3L-CVRP and MP-VRP show validity of approach

Future Work

- Reduction of number of costly calls to black box function
- Extend with Branch & Generate scheme
- Test on other applications e.g. Routing combined with Scheduling
Une heuristique de génération de colonnes pour le problème de tournées de véhicules avec faisabilité boîte noire

JFPC 2012, Toulouse

Florence Massen\textsuperscript{1} \textsuperscript{a} Yves Deville\textsuperscript{1} Pascal Van Hentenryck\textsuperscript{2}

\textsuperscript{1}ICTEA, Université catholique de Louvain, Belgique
\textsuperscript{2}Optimization Research Group, NICTA, University of Melbourne, Australie

\textsuperscript{a}Supported by the National Research Fund, Luxembourg


F. Massen, Y. Deville, P. Van Hentenryck (UCL) Heuristique de génération de colonnes pour VRPBB

53