Column Generation for the Split Delivery VRP

Alberto Ceselli, DTI – University of Milan
Giovanni Righini, DTI – University of Milan
Matteo Salani, TRANSP-OR – EPF Lausanne

CG2k8 – Aussois, June 19th 2007
The problem
The split delivery vehicle routing problem (SDVRP)

Literature review

A CG scheme
Formulations
The pricing problem

Results
Experiments
The VRP

It is given:

- a fleet of vehicles \((K)\), each having a loading capacity \((Q)\)
- a set of customers \((V)\), each requiring the delivery of goods \((d_i)\)
- a network \((G=(V,A))\)

Decide:

- a route for each vehicle

Such that:

- each customer is in a route
- the sum of demands of the customers in each route does not exceed the vehicle capacity
- the total travel distance is minimized

(BCP Fukasawa et al. ’06, Book Golden et al eds. ’08)
The Split Delivery VRP

SDVRP: each customer can belong to more than one route, and (fractionally) served by more than one vehicle:

potentially yielding a $X2$ saving.
It is a problem with several applications.
Heuristics

- Frizzel and Griffin ('95): grid network, tight multiple time windows and nonlinear loading costs, construction and local search, instances with up to 150 customers
- Bompadre Dror Orlin ('98): approximation algorithms
- Archetti Savelsbergh Speranza ('06): tabu search (up to 200 customers)
- Archetti Savelsbergh Speranza ('07): MIP based heuristic (same instances)
- Chen Golden Wasil ('07): construction and MIP heuristic (up to 200 customers)
- Jin Liu Eksioglu ('07): column generation heuristic (good for instances with large customer demands).
Exact methods

- Reduction to VRP (if data is rational in polynomial space and time)
- Dror Laporte Trudeau ('94): arc-based formulation, subtour and connectivity constraints, branching (up to 20 customers to optimality)
- Belenguer Martinez Mota ('00): polyhedral study, model for a relaxation of the problem
- Jin Lin Bowden ('06): two-stage (partitioning–routing), with 7 new classes of valid inequalities (up to 20 customers to optimality)
Column generation

- Gendreau Dejax Feillet Gueguen ('07): SDVRP with TWs
  - Set covering ILP formulation
  - Column generation and hard pricing problem
  - Relaxed model with easier pricing
  - Few instances with up to 50 customers to optimality
- Desaulniers (CG2k8): SDVRP with TWs
  - instances with up to 100 customers to optimality
Our contribution

A problem reformulation and CG scheme which:

- yields good lower bounds on the optimal value
- is ‘simple’ to compute
- allows for many VRP strategies to be applied (valid cuts, branching ...)
- ‘nicely’ fits in a branch-and-price-and-cut scheme
SDVRP flow formulation

Flow formulation (Dror Laporte Trudeau ’94):

\[ \text{min } z_{FP} = \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k \in K} x_{ijk} \]

s.t. \[ \sum_{k \in K} y_{ik} = 1 \quad \forall i \in V \]

\[ \sum_{i \in V} d_i y_{ik} \leq Q \quad \forall k \in K \] (1)

\[ \sum_{j \in V} x_{ijk} \geq y_{ik} \quad \forall i \in V, k \in K \] (2)

subtour & VRP const ... (3)

\[ x_{ijk} \in \{0, 1\}, y_{ik} \geq 0 \quad \forall i, j \in V, k \in K \] (4)
SDVRP flow formulation

Flow formulation (Dror Laporte Trudeau ’94):

\[\text{FLP}\]

\[
\min z_{FP} = \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k \in K} x_{ijk}
\]

subject to

\[
\sum_{k \in K} y_{ik} = 1 \quad \forall i \in V
\]

\[
(x_{ijk}, y_{ik}) \in \Omega_k \quad \forall k \in K
\]

LP relaxation and convexification:

\[
\Omega_k = \text{conv}\{(x_{ijk}, y_{ik}) \mid 0 \leq x_{ijk} \leq 1, y_{ik} \geq 0, (1), (2), (3)\}
\]
DW reformulation

For each $k \in K$, given an extreme point $r$: $(\bar{x}_{ij}^r, \bar{y}_i^r) \in \Omega_k$

$$c_r = \sum_{i \in V} \sum_{j \in V} c_{ij} \bar{x}_{ij}^r$$

and

$$x_{ijk} = \sum_{r \in \Omega_k} \bar{x}_{ij}^r \lambda_r \quad \forall i, j \in V$$

$$y_{ik} = \sum_{r \in \Omega_k} \bar{y}_i^r \lambda_r \quad \forall i \in V$$

s.t. \ $$\sum_{r \in \Omega_k} \lambda_r = 1$$

$$\lambda_r \geq 0 \quad \forall r \in \Omega_k$$
Extended formulation

CCLP

$$\min z_{CCLP} = \sum_{k \in K} \sum_{r \in \Omega_k} c_r \lambda_r$$

s.t.

$$\sum_{k \in K} \sum_{r \in \Omega_k} \bar{y}_i^r \lambda_r \geq 1 \quad \forall i \in V(\pi_i) \quad (1)$$

$$\sum_{r \in \Omega_k} \lambda_r \leq 1 \quad \forall k \in K$$

$$\lambda_r \geq 0 \quad \forall k \in K, r \in \Omega_k$$

(+ tightening constraints)

**Observation:** there always exists a solution in which only cols with at most 1 fract coordinate are selected (set $\bar{\Omega}$). (Jin et al '07)
Simplifying the pricing

- let be $a_i^r = \lceil \bar{y}_i^r \rceil$
- for each $k \in K$ we define $\tilde{\Omega}_k$ as the set of columns satisfying
  \[ \sum_{i \in V | a_i^r = 1} d_i - \max_{i \in V | a_i^r = 1} d_i + 1 \leq Q \]
- we observe that $\bar{\Omega}_k \subseteq \tilde{\Omega}_k$
- we substitute each covering constraint (1) as follows
  \[ \sum_{k \in K} \sum_{r \in \Omega_k} \bar{y}_i^r \lambda_r \geq 1 \quad \forall i \in V \rightarrow \]
  \[ \sum_{k \in K} \sum_{r \in \tilde{\Omega}_k} a_i^r \lambda_r \geq 1 \quad \forall i \in V \]  
  (2)
- we obtain a relaxation of the master
  (adding more vars and rounding up the lhs of $\geq$ constr.).
Final model

MP

\[
\min z_{MP} = \sum_{k \in K} \sum_{r \in \tilde{\Omega}_k} c_r \lambda_r \\
\text{s.t.} \quad \sum_{k \in K} \sum_{r \in \tilde{\Omega}_k} a_{i}^r \lambda_r \geq 1 \quad \forall i \in V (\gamma_i) \\
\sum_{r \in \tilde{\Omega}_k} \lambda_r \leq 1 \quad \forall k \in K \\
\lambda_r \geq 0 \quad \forall k \in K, r \in \tilde{\Omega}_k
\]

\[
\tilde{c}_r = \sum_{i \in V} \sum_{j \in V} c_{ij} \bar{x}_{ij}^r - \sum_{i \in V} \gamma_i a_{i}^r + \ldots
\]
Final model

MP

$$\min z_{MP} = \sum_{k \in K} \sum_{r \in \tilde{\Omega}_k} c_r \lambda_r$$

\[ \text{s.t.} \quad \sum_{r \in \tilde{\Omega}_k} a_i^r \lambda_r \geq y_{ik} \quad \forall k \in K, \forall i \in V \ (\gamma_{ik}) \]

$$\sum_{r \in \tilde{\Omega}_k} \lambda_r \leq 1 \quad \forall k \in K$$

$$\sum_{k \in K} y_{ik} = 1 \quad \forall i \in V$$

$$\lambda_r \geq 0 \quad \forall k \in K, \ r \in \tilde{\Omega}_k$$

$$y_{ik} \geq 0 \quad \forall i \in V, \ k \in K$$

$$\tilde{c}_r = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^r - \sum_{i \in V} \gamma_{ik} a_i^r + \ldots$$
Final model

MP

\[
\min z_{MP} = \sum_{k \in K} \sum_{r \in \tilde{\Omega}_k} c_r \lambda_r \\
\text{s.t.} \quad \sum_{r \in \tilde{\Omega}_k} a_i^r \lambda_r \geq y_{ik} \quad \forall k \in K, \forall i \in V (\gamma_{ik}) \\
\sum_{r \in \tilde{\Omega}_k} \lambda_r \leq 1 \quad \forall k \in K \\
\sum_{k \in K} y_{ik} = 1 \quad \forall i \in V \\
\sum_{i \in V} d_i y_{ik} \leq Q \quad \forall k \in K \\
\lambda_r \geq 0 \quad \forall k \in K, r \in \tilde{\Omega}_k \\
y_{ik} \geq 0 \quad \forall i \in V, k \in K
\]

\[\tilde{c}_r = \sum_{i \in V} \sum_{j \in V} c_{ij} \bar{x}_{ij}^r - \sum_{i \in V} \gamma_{ik} a_i^r + \ldots\]
Quality of the bound

- FLP: three-index flow based formulation
- CCLP: DW reformulation of FLP
- MP: our formulation
- GDFG: Gendreau et al formulation
- NCLP: DW reformulation of FLP leaving the capacity constraints in the master problem
The pricing problem (PP)

The PP is a resource constrained elementary shortest path problem

- labels contain both
  \(D\): the total demand of the visited customers
  \(d_{sc}\): the demand of the potential split customer
- during extension, the capacity constraint can still be respected if \(D + d_j - \max(d_j, d_{sc}) + 1 \leq Q\).
- label \(S'\) can dominate label \(S''\) only if
  \(D' \leq D''\)
  \(D' - d'_{sc} \leq D'' - d''_{sc}\)
Pricing problem - implementation

- bounded bi-directional DP
- Decremental State Space relaxation with smart core initialization (RS ’07)
- Set $U$ of unreachable customers (Feillet ’04)
- Greedy pricer
- Heuristic DP pricer (relaxed domination criteria + Fractional Knapsack Bounding)
- involved multiple pricing policy (tackle symmetries)
Computational results

We implemented the CG scheme in C using GLPK 4.16 as LP solver, subset of Solomon instances (23 r- and 4 c- instances with TWs)

<table>
<thead>
<tr>
<th></th>
<th>GDFG</th>
<th>MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg dual. gap</td>
<td>1.34%</td>
<td>1.64%</td>
</tr>
<tr>
<td>avg CPU time(s)</td>
<td>3.81</td>
<td>16.2</td>
</tr>
<tr>
<td>inst. with best bound</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>inst. with no dual. gap</td>
<td>7</td>
<td>11</td>
</tr>
</tbody>
</table>
Additional remarks

- Effect of stabilization (using GLPK interior point method for LPs):
  50% iterations reduction (but much longer LP solution times).
- Heuristics: only integrality checking.
- Branching: only naïve branching implemented,
  some instances with up to 50 customers solved to optimality.
Additional remarks

- Effect of stabilization (using GLPK interior point method for LPs):
  50% iterations reduction (but much longer LP solution times).
- Heuristics: only integrality checking.
- Branching: only naïve branching implemented, some instances with up to 50 customers solved to optimality.

Many thanks for your attention :o) Comments :?!