A Branch-and-Cut Algorithm for the Capacitated Vehicle Routing Problem (lecture I)

Bjørn Petersen
DIKU, University of Copenhagen

December 28, 2006

Overview

- Mathematical model, CVRP
- Valid inequalities
- Cut management
- Variable fixing
- Strong branching
- Variable pricing

Mathematical model

2-index formulation (symmetric problem)

- $x_{ij} = 1$ iff edge (i,j) is used, $i < j$

Formulation

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{1}$$

s.t.

$$\sum_{(j,i)\in E} x_{ji} + \sum_{(i,j)\in E} x_{ij} = 2 \qquad \forall i \in V\backslash\{0\} \tag{2}$$

$$\sum_{(i,j)\in\delta(0)} x_{ij} = 2K \tag{3}$$

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geq 2r(S) \qquad \forall S \subseteq V\backslash\{0\}, S \neq \emptyset \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E, i \neq 0 \tag{5}$$

$$x_{ij} \in \{0,1,2\} \qquad \forall (i,j) \in \delta(0) \tag{6}$$

- Exponential number of capacity inequalities (4)

Capacity inequalities

Identify a valid $r(S)$

- Fractional:

$$r(S) = \sum_{i \in S} d_i / C$$

  Separation is a minimum cut problem

- Rounded:

$$r(S) = \left\lceil \sum_{i \in S} d_i / C \right\rceil$$

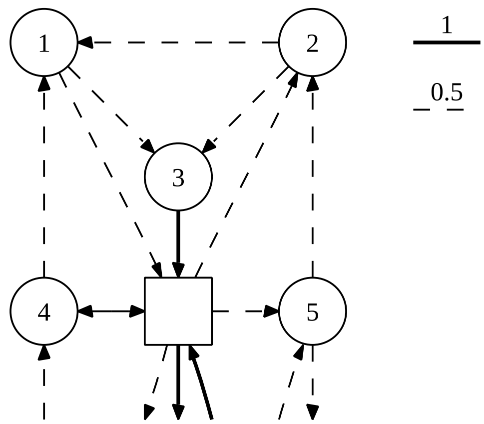
  Separation is NP-complete

- Weak: $r(S)$ equals the bin packing solution

- Capacity: $r(S)$ is based on a K-partition of all customers.

Example of violated rounded

- $C = 4$
- $d = \{2, 2, 1, 1, 1, \dots\}$
- $S = \{1, 2, 3\}$



Fractional: $r(S) = \frac{2+2+1}{2} = 1.25$

$$x(\delta(S)) = 3 \geq 2r(S) = 2.5$$

Rounded: $r(S) = \left\lceil \frac{2+2+1}{4} \right\rceil = 2$

$$x(\delta(S)) = 3 \not\geq 2r(S) = 4$$

Example of tightness

8 customers,

- $|K| = 4$
- $C = 7$
- $d = \{5, 3, 3, 3, 4, 4, 4, 2\}$

and S is the first four customers.

Fractional: $r(S) = \frac{5+3+3+3}{7} = 2$

Rounded: $r(S) = \left\lceil \frac{5+3+3+3}{7} \right\rceil = 2$

Weak: BBP solution is $\{\{5\}, \{3, 3\}, \{3\}\}$

$r(S) = 3$

Capacity: The only feasible K-partition is

$$\{\{5, 2\}, \{3, 4\}, \{3, 4\}, \{3, 4\}\}$$

$r(S) = 4$

Branch-and-Cut algorithm

- Branch on x-variables
- Branch adapted from TSP, for non-empty $S \subseteq V \backslash \{0\}$ either

$$x(\delta(S)) = 2 \text{ or } x(\delta(S)) \geq 4$$

Pseudocode of LP loop

```
LP-Loop-#1()
repeat
        Solve-LP()
        Identify-CAP-Cuts()
        if no cuts and LP solution is integral
          then UB ← Update-UB()
        LB ← Update-LB()
        if LB ≥ UB
          then return fathomed
        Add-CAP-Cuts()
    until no cuts generated
return processed
```

Valid inequalities

Improve LP bound by adding valid inequalities

- Framed capacity inequalities
- Comb inequalities
- Hypotour inequalities
- Multistar inequalities

Issues

- Separation problems are often NP-hard
- Heuristics are used

Pseudocode of LP loop

LP-Loop-#2()

```
repeat
        Solve-LP()
        Identify-CAP-Cuts()
        if no cuts and LP solution is integral
          then UB ← Update-UB()
        LB ← Update-LB()
        if LB ≥ UB
          then return fathom
        Add-CAP-Cuts()
        Generate-Cuts() ▷ NEW
   until no cuts generated
return processed
```

Framed capacity inequalities

- $S \subseteq V \backslash \{0\}$
- a partition of S, $\Omega = \{S_1, \ldots, S_n\}$
- $r(S, \Omega)$ where all capacity inequalities for each S_i holds with equality

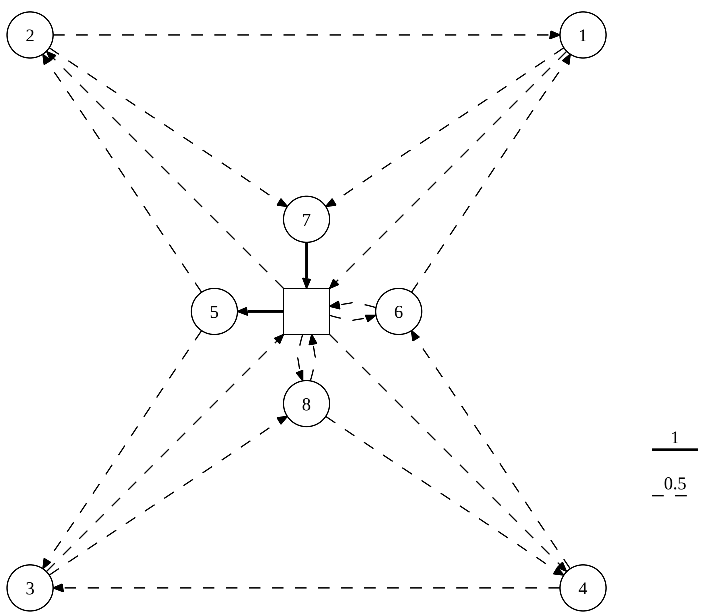
The framed capacity inequality is:

$$x(\delta(S)) + \sum_{i=1}^{n} x(\delta(S_i)) \geq 2r(S, \Omega) + \sum_{i=1}^{n} 2r(S_i)$$

Separation problem is NP-hard, involves solving bin packing problems

Example of violation

- $C = 4,\ d = \{2,2,2,2,1,1,1,1\}$
- $S = \{1,2,3,4,5,6,7,8\}$
- $\Omega = \{\{3,4,8\},\{1\},\{2\},\{5\},\{6\},\{7\}\}$
- $k(S,\Omega) = 3,\ k(S_1) = 2,\ k(S_i) = 1, i = 2,\ldots,6$



FCI:

$$x(\delta(S)) + \sum_{i=1}^{n} x(\delta(S_i)) = 6+3+2+2+2+2+2 = 19$$

$$\not\geq 2k(S,\Omega) + \sum_{i=1}^{n} 2k(S_i) = 2\cdot 3 + 2\cdot 7 = 20$$

Comb inequalities

A handle H and teeth $T_1, \ldots, T_t$

- $H, T_1, \ldots, T_t \in V$
- $T_j \backslash H \neq \emptyset$
- $T_j \cap H \neq \emptyset$
- $T_i \cap T_j = \emptyset$, $i \neq j$
- $t \geq 3$ and odd

TSP comb:

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq 3t + 1$$

VRP comb:

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq t + 1 + \sum_{i=1}^{t} 2r(T_i)$$

Hypotour inequalities

- At least on edge in F must be used

$$x(F) \geq 1$$

Extended 2-edge hypotour inequality

$$x(\delta(S)) + 2x(F) \geq 2x_{ij} + 2x_{uv} \quad i, u \in S, j, v \notin S$$

- When $x(\delta(S)) = 2$ and $x_{ij} = x_{uv} = 1$ at least one edge in F must be used
- F can be identified since a path between the depot and j and u exists

Multistar inequalities

A generalization of the fractional capacity inequalities

- Nucleus $N \in V \backslash \{0\}$
- Satellites $S \in V \backslash \{0\} \backslash N$
- Connectors $C \subseteq N$, partial when $C \subset N$
- Constants λ and ω

$$x(\delta(N)) \geq \lambda + \omega x(E(C:S))$$

- λ and ω are determined by an upper bound of $x(\delta(N))$ for a fixed $x(E(C:S))$ for different N, S, C

Cut management

- Separation:
  - Prioritize some cuts before others
  - Separate until no violated cuts
  - Depend on violation of the cut
- Storing:
  - When to remove unbinding cuts from LP
  - Keep a cut pool, local/global in branch tree

Variable fixing

Fix variables to either 0 or 1 based on some knowledge.

Reduced cost fixing

Based on reduced cost and known upper and lower bounds of the IP.

- UB and LB is the upper and lower bound of the IP
- x_{ij} is a variable
- c^{π}_{ij} is the reduced cost of variable x_{ij}

Fix to 0: Let x_{ij} be a non-basic variable with $c^{\pi}_{ij} \geq 0$.
If $LB + c^{\pi}_{ij} \geq UB$
then x_{ij} can be fixed to 0

Fix to 1: Let x_{ij} be a basic variable with $c^{\pi}_{ij} \leq 0$.
If $LB - c^{\pi}_{ij} \geq UB$
then x_{ij} can be fixed to 1

Logical fixing

Fix a variable due to logical constraints, i.e. the capacity constraints.

Let variable $x_{ij} = 1$, e.g. due to a branch/reduced cost fixing.
If $d_i + d_j + d_u > C$ then x_{ju} and x_{ui} can be fixed to 0.

Pseudocode of LP loop

```
LP-Loop-#3()
repeat
        Solve-LP()
        Identify-CAP-Cuts()
        if no cuts and LP solution is integral
           then UB ← Update-UB()
        LB ← Update-LB()
        if LB ≥ UB
           then return fathom
        Reduced-Cost-Fixing() ▷ NEW
        Logical-Fixing() ▷ NEW
        Add-CAP-Cuts()
        Generate-Cuts()
   until no cuts generated
return processed
```

Strong branching

A partly eager branch strategy

- Identify a set of branch candidates
- Presolve LP for each candidate
- (Apply cut separation during strong branching)

Pros:

- Faster convergence
- Better LP bounds

Cons:

- Each branch decision more time consuming

Variable pricing

- Handle a subset of the variables
- Similar to Branch-Cut-and-Price

Variables with negative reduced cost can price in.

The pricing problem becomes:

$$\min_{x_{ij} \in N} c^{\pi}_{ij} = c_{ij} - \sum_{r \in \mathcal{R}} \pi^{r}_{ij} a^{r}_{ij}$$

Where $\mathcal{R}$ is the set of current cuts in the LP and N is the set of non-basic variables

Pricing problem is easy, takes $O(|N||\mathcal{R}|)$

Pseudocode of LP loop

- LB only valid when no variables can price in
  - So no variable fixing when variables can price in

LP-Loop-#4()

```
repeat
        Solve-LP()
        Identify-CAP-Cuts()
        if no cuts and LP solution is integral
          then UB ← Update-UB()
        Price-In-Vars() ▷ NEW
        if no variables priced in ▷ NEW
          then LB ← Update-LB()
                 if LB ≥ UB
                   then return fathom
        Reduced-Cost-Fixing()
        Logical-Fixing()
        Add-CAP-Cuts()
        Generate-Cuts()
   until no cuts generated and no variables priced in ▷ NEW
return processed
```

Remarks

- Good for compact formulations, e.g. symmetric, few constraints
- Seldom used for VRPTW
- Important to implement speed-ups, e.g. variable fixing and strong branching
- Separation procedures can be used in Branch-Cut-and-Price

A Branch-Cut-and-Price Algorithm for the Vehicle Routing Problem with Time Windows (lecture II)

Bjørn Petersen
DIKU, University of Copenhagen

December 28, 2006

Overview

- Mathematical model, VRPTW
- Branch-Cut-and-Price
- Robust/non-robust
- 2-path cuts
- Lower bounds
- Branching

Mathematical model
3-index flow model

- x_{ijk} if edge (i,j) is used by vehicle k
- w_{ik} time when vehicle k visits customer i
- notation: split depot into node 0 and $n+1$

$$
\begin{array}{lll}
\text{minimize} & \sum_{k\in K} \sum_{(i,j)\in E} c_{ij} x_{ijk} & \\
\text{s.t.} & \sum_{k\in K} \sum_{(i,j)\in E} x_{ijk} = 1 & \forall i \in V \\
& \sum_{i\in V} d_i \sum_{(i,j)\in E} x_{ijk} \le C & \forall k \in K \\
& \sum_{(i,j)\in E} x_{ijk} = \sum_{(j,i)\in E} x_{jik} & \forall i \in V, \forall k \in K \\
& \sum_{(i,j)\in \delta(0)} x_{ijk} = 1 & \forall k \in K \\
& \sum_{(i,j)\in \delta(n+1)} x_{ijk} = 1 & \forall k \in K \\
& x_{ijk} = 1 \;\Rightarrow\; w_{ik} + t_{ij} \le w_{jk} & \forall (i,j) \in E, \forall k \in K \\
& a_i \le w_{ik} \le b_i & \forall i \in V, \forall k \in K \\
& x_{ijk} \in \{0,1\} &
\end{array}
$$

Set Partition model

- $\mathcal{R}$ is set of all routes satisfying capacity constraints and time windows
- c_r constant, cost of route r
- a_{ijr} constant, equal to use of edge (i,j) on route r
- λ_r is 1 if route $r \in \mathcal{R}$ is chosen for the solution

Yielding following model:

$$
\begin{array}{lll}
\text{minimize} & \sum_{r\in\mathcal{R}} c_r\lambda_r & \\
\text{s.t.} & \sum_{r\in\mathcal{R}} \sum_{(i,j)\in E} a_{ijr}\lambda_r \geq 1 & i \in V\backslash\{0\} \\
 & \lambda_r \in \{0,1\} & r \in \mathcal{R}
\end{array}
$$

Let π be the dual variable ($\pi_0 = 0$). The reduced cost is:

$$\widehat{c}_r = \sum_{(i,j)\in E} c_{ij}a_{ijr} - \sum_{(i,j)\in E} \pi_i a_{ijr}$$

Branch-Cut-and-Price

- Cuts on 3-index model

$$Bx \geq b_0$$

- Cuts on set partition model

$$B\lambda \geq b_0$$

Robust when complexity/structure of pricing problem does not change.

- Cuts on 3-index model that can be described on a single vehicle are robust

- Cuts on several vehicles or on the set partition model are generally non-robust

Transform set partition solution to 3-index flow model to separate inequalities.

Robust cutting

A valid inequality in the 3-index flow model

$$\sum_{k \in K} \sum_{(i,j) \in E} b_{ijk} x_{ijk} \leq b_0$$

Decomposed into the master problem we get:

$$\sum_{r \in \mathcal{R}} \sum_{(i,j) \in E} b_{ij} a_{ijr} \lambda_r \leq b_0$$

Let μ be the dual variable. The reduced cost is:

$$\begin{aligned} \widehat{c}_r &= \sum_{(i,j) \in E} c_{ij} a_{ijr} - \sum_{(i,j) \in E} \pi_i a_{ijr} - \mu \sum_{(i,j) \in E} b_{ij} a_{ijr} \\ &= \sum_{(i,j) \in E} (c_{ij} - \pi_i - \mu b_{ij}) a_{ijr} \end{aligned}$$

- Does not complicate structure in pricing problem
- Can be considered as a decomposition of the 3-index flow model after cut generation
- Approach used so far in BCP algorithms for VRP

Valid inequalities

- Valid inequalities from CVRP:
  - Capacity inequalities
  - Framed capacity inequalities
  - Comb inequalities
  - Multistar inequalities
  - Hypotour inequalities

  and VRPTW:
  - 2-path cuts
- Not all valid inequalities may be violated in column generation due to stronger lower bound from path structure

Pseudocode of LP loop

- LB only valid when no variables can price in
- Heuristic variable generation before cut generation
- Optimal pricing problems only solved when no alternative

LP-Loop()

```
repeat
        Solve-LP()
        if LP solution is integral
          then UB ← Update-UB()
        Generate-Heuristic-Vars()
        if no variables generated
          then Generate-VRP-Cuts()
               if no cuts generated
                 then Generate-Optimal-Vars()
        if no variables and cuts generated ▷ Optimal
          then LB ← Update-LB()
               if LB ≥ UB
                 then return fathom
   until no cuts and no variables generated
return processed
```

2-path cuts

- Similar to capacity inequalities
- Capacity tend not to be binding for VRPTW
- Exploits time windows to calculate $r(S)$

$$x(\delta^+(S)) \geq r(S)$$

where $r(S) = 2$

Can be generalized to k-path cuts

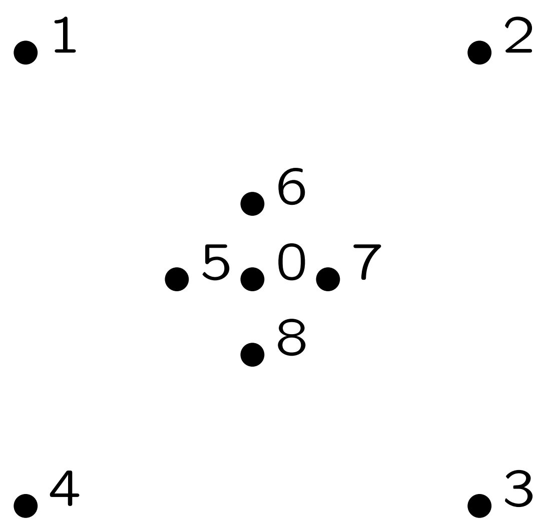
Example
$a_i, b_i = [1, 2]$ for all $i \in V \backslash \{0\}$, all travel times are 1, $S = \{1, 2, 3\}$, and $x(\delta^+(S)) = 1.5$

$r(S) = 2$, i.e. $x(\delta^+(S)) = 1.5 \not\geq r(S) = 2$

BCP Example

Vehicle capacity $C = 4$

i	a_i	b_i	d_i
0	0	30	0
1	0	12	2
2	5	25	2
3	3	27	2
4	5	14	2
5	2	20	1
6	2	12	1
7	3	25	1
8	0	15	1



The different paths:

cost	e.g. path
4	$0 \rightarrow 5 \rightarrow 0$
16	$0 \rightarrow 1 \rightarrow 0$
18	$0 \rightarrow 5 \rightarrow 1 \rightarrow 6 \rightarrow 0$
20	$0 \rightarrow 8 \rightarrow 5 \rightarrow 1 \rightarrow 0$
21	$0 \rightarrow 5 \rightarrow 1 \rightarrow 7 \rightarrow 0$
28	$0 \rightarrow 1 \rightarrow 2 \rightarrow 0$

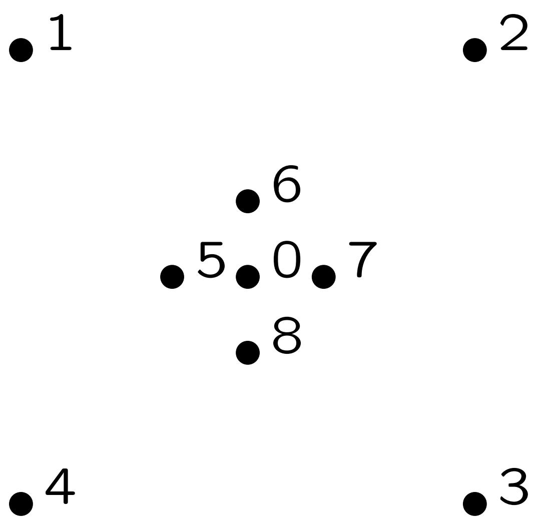
Matrix after 2 iterations:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	
16	16	16	16	4	4	4	4	18	18	18	18	28	28	28	21	
1								1				1			1	≥ 1
	1									1		1	1			≥ 1
		1									1		1	1		≥ 1
			1						1					1		≥ 1
				1				1	1						1	≥ 1
					1			1		1						≥ 1
						1				1	1				1	≥ 1
							1		1		1					≥ 1

Solution $= 64$, $\{0.5x_9, 0.5x_{10}, 0.5x_{11}, 0.5x_{12}, 0.5x_{13}, 0.5x_{15}\}$

Duals $= \{14, 14, 14, 14, 3.5, 0.5, 3.5, 0.5\}$

Induced graph:



Two rounded capacity inequalities can be found with $r(S) = 2$ due to capacity but $x(\delta(S)) = 3$.

$S = \{1, 2, 6\}$ and $S = \{3, 4, 8\}$

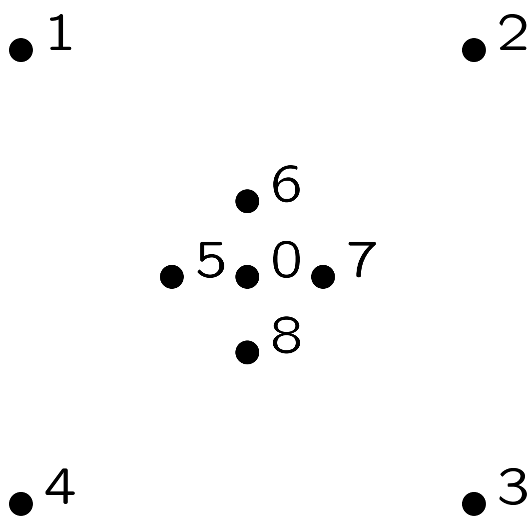
Add cuts to matrix

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	
16	16	16	16	4	4	4	4	18	18	18	18	28	28	28	21	
1								1				1			1	≥ 1
	1									1		1	1			≥ 1
		1									1		1	1		≥ 1
			1						1					1		≥ 1
				1				1	1						1	≥ 1
					1			1		1						≥ 1
						1				1	1				1	≥ 1
							1		1		1					≥ 1
2	2				2			2		2		2	2		2	≥ 4
		2	2				2	2	2							≥ 4

Solution = 66, $\{0.5x_1, 0.5x_4, 0.5x_9, 0.5x_{10}, 0.5x_{11}, 0.5x_{12}, 0.5x_{14}\}$

Duals = $\{12.5, 10.5, 10.5, 12.5, 1.5, 0.5, 3.5, 0.5, 1.75, 1.75\}$

Induced graph:



Path: $0 \to 2 \to 7 \to 8 \to 0$ Reduced cost: $\hat{c} = 20 - 10.5 - 3.5 - 0.5 - 2 \cdot 1.75 - 2 \cdot 1.75 = -1.5$

And so forth

Lower bound

- Obtain lower bound to the master problem while column generating
- Can be done when pricing problem is solved to optimality

Proposition 1. Lasdon *If* z_{LPM} *is the value of the given column generation iteration and* z_{pp} *is the value of the optimal solution of the pricing problem then* $LB_{LAS} = z_{LPM} - |K| z_{pp}$ *is a lower bound to the optimal IP solution* z_{IPM}.

Branching

- Usually done on variables x_{ijk} by removing edges in pricing problem
- Use knowledge from cutting to do TSP branch rule, i.e. for $S \subseteq V \backslash \{0\}$ either

$$x(\delta(S)) = 2 \text{ or } x(\delta(S)) \geq 4$$

Strong branching

- Perform both pricing and cut generation during strong branching
- Use LB to fathom nodes during strong branching

Remarks

- BCP is the most successful approach for VRPTW
- Becoming competitive for CVRP on harder instances
- Still unclear when to generate cuts and when to price in columns
- Recent results (Jepsen, Petersen, Spoorendonk, and Pisinger) has shown that non-robust bcp is worthwhile