Selective pricing in branch-and-price algorithms for vehicle routing

Guy Desaulniers

Polytechnique Montréal and GERAD, Canada

Joint work with D. Pecin, C. Contardo

Column Generation 2016
Búzios, Brazil, May 22, 2016

Outline

Vehicle routing

General definition

Given a set of demands (deliveries, pickups, etc.), find least-cost vehicle routes to fulfill these demands

- Large number of problem variants (CVRP, VRPTW, etc.)
- Most of the times, routes are subject to elementarity requirements (e.g., customers cannot be visited more than once)

For simplicity, we focus on variants where each customer must be visited exactly once

The ideas are, however, applicable to other variants

Vehicle routing

General definition

Given a set of demands (deliveries, pickups, etc.), find least-cost vehicle routes to fulfill these demands

- Large number of problem variants (CVRP, VRPTW, etc.)
- Most of the times, routes are subject to elementarity requirements (e.g., customers cannot be visited more than once)

For simplicity, we focus on variants where each customer must be visited exactly once

The ideas are, however, applicable to other variants

Vehicle routing

General definition

Given a set of demands (deliveries, pickups, etc.), find least-cost vehicle routes to fulfill these demands

- Large number of problem variants (CVRP, VRPTW, etc.)
- Most of the times, routes are subject to elementarity requirements (e.g., customers cannot be visited more than once)

For simplicity, we focus on variants where each customer must be visited exactly once

The ideas are, however, applicable to other variants

A simplified model for vehicle routing

E: set of feasible elementary routes

N: set of customers

c_r: cost of route r

a_{ri}: 1 if route r visits customer i, 0 otherwise

x_r: binary variable equal to 1 if route r is selected, 0 otherwise

$$\min \quad \sum_{r \in E} c_r x_r \tag{1}$$

$$\text{s.t.} \quad \sum_{r \in E} a_{ri} x_r = 1, \qquad \forall i \in N \tag{2}$$

$$x_r \in \{0, 1\}, \qquad \forall r \in E \tag{3}$$

Branch-price-and-cut (BPC)

BPC is the leading exact methodology for vehicle routing

- Branch-and-bound algorithm
- Linear relaxations are solved by column generation
  - Iterative method alternating between a restricted master problem and a pricing problem
- Cuts are added to strengthen these linear relaxations

Pricing problem

- Find feasible routes with negative reduced costs
- Corresponds to an elementary shortest path problem with resource constraints (ESPPRC)
- NP-hard in the strong sense

Path relaxations

Given the difficulty of solving the ESPPRC, a relaxation of the pricing problem is often used

- Allows non-elementary routes
  - SPPRC with k-cycle elimination (Irnich, Villeneuve, 2006)
  - ng-SPPRC (Baldacci et al., 2008)
- In model (1)–(3), set E is replaced by a larger set $R \supset E$
- Non-elementary routes can be part of linear relaxation solutions
  - Yield weaker lower bounds in the search tree
  - Removed from integer solutions through cutting and branching

Lower bounds

A lower bound $\underline{z}(R)$ is computed by solving

$$\begin{aligned} \underline{z}(R) = \min \quad & \sum_{r \in R} c_r x_r \\ \text{s.t.} \quad & \sum_{r \in R} a_{ri} x_r = 1, && \forall i \in N \\ & 0 \leq x_r \leq 1, && \forall r \in R \end{aligned}$$

Traditional stopping criterion

Reduced cost $\bar{c}_r \geq 0$ for $r \in R$

A valid lower bound is also achieved if $\bar{c}_r \geq 0$ for $r \in E$

Indeed, we can assume that ...

... this lower bound $\underline{z}(Q)$ is computed by replacing R with $Q = E \cup B$ where B is the set of routes in the basis

Remarks

- $\underline{z}(Q) \geq \underline{z}(R)$ because $E \subseteq Q \subseteq R$
- Q and thus $\underline{z}(Q)$ depend on the pricing and CG algorithms

Selective pricing

- Aims at pricing elementary paths among the relaxed paths
- Can discard a non-elementary path even if it has the least reduced cost
- Can still generate non-elementary paths

A valid lower bound is also achieved if $\bar{c}_r \geq 0$ for $r \in E$

Indeed, we can assume that ...

... this lower bound $\underline{z}(Q)$ is computed by replacing R with $Q = E \cup B$ where B is the set of routes in the basis

Remarks

- $\underline{z}(Q) \geq \underline{z}(R)$ because $E \subseteq Q \subseteq R$
- Q and thus $\underline{z}(Q)$ depend on the pricing and CG algorithms

Selective pricing

- Aims at pricing elementary paths among the relaxed paths
- Can discard a non-elementary path even if it has the least reduced cost
- Can still generate non-elementary paths

A valid lower bound is also achieved if $\bar{c}_r \geq 0$ for $r \in E$

Indeed, we can assume that ...

... this lower bound $\underline{z}(Q)$ is computed by replacing R with $Q = E \cup B$ where B is the set of routes in the basis

Remarks

- $\underline{z}(Q) \geq \underline{z}(R)$ because $E \subseteq Q \subseteq R$
- Q and thus $\underline{z}(Q)$ depend on the pricing and CG algorithms

Selective pricing

- Aims at pricing elementary paths among the relaxed paths
- Can discard a non-elementary path even if it has the least reduced cost
- Can still generate non-elementary paths

Labeling algorithm

Pricing is performed using a labeling algorithm

- Label contains all information (resource values, including reduced cost) to represent a partial path
- Labels are extended in the network using resource extension functions
- A dominance rule is applied to avoid enumerating all paths

Here we focus only on certain label components and on the dominance rule

Labeling algorithm

Pricing is performed using a labeling algorithm

- Label contains all information (resource values, including reduced cost) to represent a partial path
- Labels are extended in the network using resource extension functions
- A dominance rule is applied to avoid enumerating all paths

Here we focus only on certain label components and on the dominance rule

Additional notation

- L: a label representing a path $p(L)$
- $n(p)$: last node of path p
- $\bar{c}(p)$: reduced cost of path p
- w: a path extension
- $p \oplus w$: path resulting from the concatenation of path p and extension w (assuming that w starts in node $n(p)$)

Dominance definition without path relaxation

Let L_1 be a label such that $p(L_1) \in E$

Definition

L_1 is dominated if,
for every extension w_1 such that $p(L_1) \oplus w_1 \in E$,
there exists a label L_2 and an extension w_2 such that

1. $n(p(L_2)) = n(p(L_1))$
2. $n(p(L_2) \oplus w_2) = n(p(L_1) \oplus w_1)$
3. $p(L_2) \oplus w_2 \in E$
4. $\bar{c}(p(L_2) \oplus w_2) \leq \bar{c}(p(L_1) \oplus w_1)$

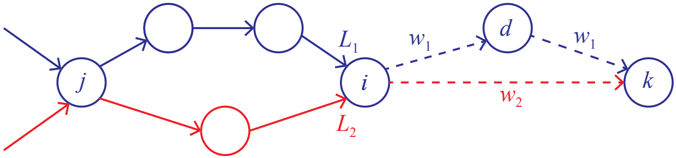
Remark

Often $w_2 = w_1$ is chosen

Example where $w_2 \neq w_1$ (Ropke and Cordeau, 2009)

Pickup and delivery problem

- $O(L)$: Requests on board at node $n(L)$ along path $p(L)$
- Triangle inequality on every resource holds at delivery nodes
- L_2 can dominate L_1 if $O(L_2) \subseteq O(L_1)$
- Here, $O(L_1) \setminus O(L_2)$ contains a single request whose delivery node is d



Dominance definitions with path relaxation

The previous definition holds when E is replaced by R

Let's see a stronger definition

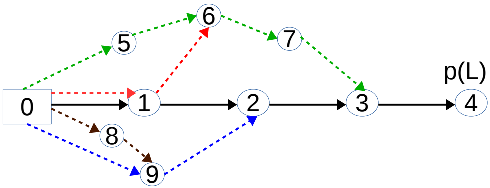
Dominance definitions with path relaxation

The previous definition holds when E is replaced by R

Let's see a stronger definition

Set $S(L)$

- $S(L)$: contains $p(L)$ and all paths identified by the algorithm as dominated by $p(L)$
- It includes extensions of paths that were dominated by a subpath of $p(L)$ and, recursively, the extensions of the paths dominated by them



S(L) = {(0,1,2,3,4),
(0,5,6,7,3,4),
(0,1,6,7,3,4),
(0,9,2,3,4),
(0,8,9,2,3,4)}

New definition (selective pricing)

L_1 is dominated if,
for every extension w_1 such that $p(L_1) \oplus w_1 \in R$, EITHER

1. there exists a label L_2 and an extension w_2 such that
    1. $n(p(L_2)) = n(p(L_1))$
    2. $n(p(L_2) \oplus w_2) = n(p(L_1) \oplus w_1)$
    3. $p(L_2) \oplus w_2 \in R$
    4. $\bar{c}(p(L_2) \oplus w_2) \leq \bar{c}(p(L_1) \oplus w_1)$

OR

2. $q \oplus w_1 \notin E, \qquad \forall q \in S(L_1)$

Remark

If the second case occurs for an extension w_1, then $p(L_1) \oplus w_1 \notin E$ and label L_1 might be discarded without identifying a path dominating $p(L_1) \oplus w_1$

ng-path

- For each customer i, let $NG(i) \subseteq N$ be the neighborhood of i, its λ closest customers
- An *ng*-route can contain a cycle $i_0 - i_1 - i_2 - \ldots - i_k = i_0$ only if there exists $j \in \{1, 2, \ldots, k-1\}$ such that $i_0 \notin NG(i_j)$

$M(L)$: set of customers (memory) to which L cannot be extended according to the NG-sets

Traditional dominance rule

A label L_1 is dominated by a label L_2 if

1. $\bar{c}(p(L_2)) \leq \bar{c}(p(L_1))$
2. $M(L_2) \subseteq M(L_1)$
3. other conditions depending on the problem

ng-path

- For each customer i, let $NG(i) \subseteq N$ be the neighborhood of i, its λ closest customers
- An *ng*-route can contain a cycle $i_0 - i_1 - i_2 - \ldots - i_k = i_0$ only if there exists $j \in \{1, 2, \ldots, k-1\}$ such that $i_0 \notin NG(i_j)$

$M(L)$: set of customers (memory) to which L cannot be extended according to the NG-sets

Traditional dominance rule

A label L_1 is dominated by a label L_2 if

1. $\bar{c}(p(L_2)) \leq \bar{c}(p(L_1))$
2. $M(L_2) \subseteq M(L_1)$
3. other conditions depending on the problem

ng-path

- For each customer i, let $NG(i) \subseteq N$ be the neighborhood of i, its λ closest customers
- An *ng*-route can contain a cycle $i_0 - i_1 - i_2 - \ldots - i_k = i_0$ only if there exists $j \in \{1, 2, \ldots, k-1\}$ such that $i_0 \notin NG(i_j)$

$M(L)$: set of customers (memory) to which L cannot be extended according to the NG-sets

Traditional dominance rule

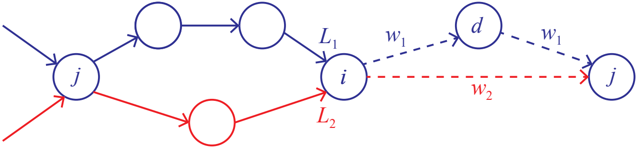
A label L_1 is dominated by a label L_2 if

1. $\bar{c}(p(L_2)) \leq \bar{c}(p(L_1))$
2. $M(L_2) \subseteq M(L_1)$
3. other conditions depending on the problem

Example from Cherkesly et al. (2015)

Pickup and delivery problem

- Same example as before except with *ng*-routes and $k = j$
- $j \in M(L_2) \subseteq M(L_1)$
- $j \notin NG(d)$
- $p(L_1) \oplus w_1 \in R$
- $p(L_2) \oplus w_2 \notin R$ but $q \oplus w_1 \notin E, \forall q \in S(L_1)$



ng-paths with selective pricing

$C(L)$: set of nodes belonging to all paths in $S(L)$

A utopian labeling algorithm

- When extending label L_i along an arc (i,j) to create a label L_j: $C(L_j) = C(L_i) \cup \{j\}$
- When label L_2 dominates label L_1: $C(L_2) = C(L_2) \cap C(L_1)$
- Dominance rule (L_2 dominates L_1):
  - $M(L_2) \subseteq M(L_1) \cup C(L_1)$
- Do not extend label L to any node in $C(L)$

Impossible because ...

$S(L)$ and thus $C(L)$ are not fully known before the end of the algorithm

ng-paths with selective pricing

$C(L)$: set of nodes belonging to all paths in $S(L)$

A utopian labeling algorithm

- When extending label L_i along an arc (i, j) to create a label L_j: $C(L_j) = C(L_i) \cup \{j\}$
- When label L_2 dominates label L_1: $C(L_2) = C(L_2) \cap C(L_1)$
- Dominance rule (L_2 dominates L_1):
  - $M(L_2) \subseteq M(L_1) \cup C(L_1)$
- Do not extend label L to any node in $C(L)$

Impossible because ...

$S(L)$ and thus $C(L)$ are not fully known before the end of the algorithm

ng-paths with selective pricing

$C(L)$: set of nodes belonging to all paths in $S(L)$

A utopian labeling algorithm

- When extending label L_i along an arc (i, j) to create a label L_j: $C(L_j) = C(L_i) \cup \{j\}$
- When label L_2 dominates label L_1: $C(L_2) = C(L_2) \cap C(L_1)$
- Dominance rule (L_2 dominates L_1):
  - $M(L_2) \subseteq M(L_1) \cup C(L_1)$
- Do not extend label L to any node in $C(L)$

Impossible because ...

$S(L)$ and thus $C(L)$ are not fully known before the end of the algorithm

ng-paths with selective pricing

An applicable labeling algorithm

- When extending label L_i along an arc (i, j) to create a label L_j: $C(L_j) = C(L_i) \cup \{j\}$
- When label L_2 dominates label L_1: $C(L_2) = C(L_2) \cap C(L_1)$
- Dominance rule (L_2 dominates L_1):
  - $M(L_2) \subseteq M(L_1) \cup C(L_1)$
  - $C(L_2) \subseteq C(L_1)$ if L_2 has been extended
- Do not extend label L to any node in $C(L)$

Heuristic pricing

Heuristic

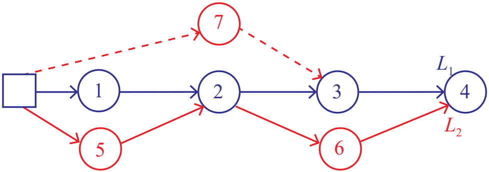
Use standard pricing except that $M(L_2) \subseteq M(L_1)$ is replaced by

$$M(L_2) \subseteq V(L_1)$$

in the dominance rule ($V(L_1)$: set of customers visited in $p(L_1)$)

Example showing why it is not exact

$M(L_1) = \{4\}$, $M(L_2) = \{2, 4\}$, L_2 dominates L_1 and cannot be extended to 2, while $0 - 7 - 3 - 4 - 2$ would be feasible



Computational experiments

Instances

- Vehicle routing problem with time windows (VRPTW)
- A few Solomon instances (100 customers) and Gehring and Homberger instances (200 customers)
- Hard-to-solve instances
  - Number of customers per route is large
  - Many possibilities to cycle

Experiments

- Comparison of
  1. Pecin et al. (2016) algorithm with standard *ng*-path pricing
  2. Pecin et al. (2016) algorithm with selective *ng*-path pricing (heuristic followed by exact algorithm)
- Root node only with rounded capacity cuts only

Results

Instance	Time (s) with SelectiveP	Time (s) with StandardP	Gain %
R208	1250	1539	18,7%
R210-10%	293	352	16,7%
R210-20%	401	416	3,6%
R211-10%	432	467	7,5%
R211-20%	498	635	21,5%
C2-2-3	8195	8487	3,4%
C2-2-4	33014	33614	1.8%
R2-2-4	31925	40012	20,2%
R2-2-8	39795	64602	38.4%
RC2-2-3	12585	14795	14,9%
RC2-2-4	57046	144783	60,6%
RC2-2-7	4210	4553	7,5%
RC2-2-8	7847	7902	0.7%
RC2-2-9	10338	11493	10.0%

Conclusion

- New paradigm for very-hard-to-solve instances
- May be applicable to other problem types
- Preliminary results show some potential

Future work

- Improve the proposed algorithm
  - Refine dominance rule
  - Filter out some non-elementary routes during column generation
- Apply principle to other problems with relaxed pricing

Thank you!
Questions?

Conclusion

- New paradigm for very-hard-to-solve instances
- May be applicable to other problem types
- Preliminary results show some potential

Future work

- Improve the proposed algorithm
  - Refine dominance rule
  - Filter out some non-elementary routes during column generation
- Apply principle to other problems with relaxed pricing

Thank you!
Questions?

Conclusion

- New paradigm for very-hard-to-solve instances
- May be applicable to other problem types
- Preliminary results show some potential

Future work

- Improve the proposed algorithm
  - Refine dominance rule
  - Filter out some non-elementary routes during column generation
- Apply principle to other problems with relaxed pricing

Thank you!
Questions?