COLUMN GENERATION 2016

May 22-25, 2016, Búzios, Brazil

p-step formulations for vehicle routing problems: a generalized class of models based on column generation

Pedro Munari

Federal University of São Carlos (UFSCar), Brazil

munari@dep.ufscar.br

Outline

- VRP formulations: main models and their features;

Outline

- VRP formulations: main models and their features;
- p-step formulations for VRP;

Outline

- VRP formulations: main models and their features;
- p-step formulations for VRP;
- Column generation for p-step formulations;

Outline

- VRP formulations: main models and their features;
- p-step formulations for VRP;
- Column generation for p-step formulations;
- Conclusions and future work.

VRP formulations

- Most used formulations:
  - Vehicle flow formulations;
  - Set partitioning formulations.

VRP formulations

- Most used formulations:
  - Vehicle flow formulations;
  - Set partitioning formulations.

- Vehicle flow formulations: variables correspond to arcs;

$$x_{ij} = \begin{cases} 1, & \text{if a route goes from node } i \text{ directly to node } j; \\ 0, & \text{o.w.} \end{cases}$$

VRP formulations

- Most used formulations:
  - Vehicle flow formulations;
  - Set partitioning formulations.
- Vehicle flow formulations: variables correspond to arcs;

$$x_{ij} = \begin{cases} 1, & \text{if a route goes from node } i \text{ directly to node } j; \\ 0, & \text{o.w.} \end{cases}$$

- Set partitioning formulations: variables correspond to routes;

$$\lambda_r = \begin{cases} 1, & \text{if route } r \text{ is used in the solution;} \\ 0, & \text{o.w.} \end{cases}$$

Two-index vehicle flow formulation (CVRP)

$$\min \quad \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ij} = 1, \qquad i = 1, \ldots, n,$$

$$\sum_{\substack{i=0 \\ i \neq h}}^{n} x_{ih} - \sum_{\substack{j=1 \\ j \neq h}}^{n+1} x_{hj} = 0, \qquad h = 1, \ldots, n,$$

$$\sum_{j=1}^{n} x_{0j} \leq K,$$

$$y_j \geq y_i + q_j x_{ij} - Q(1 - x_{ij}), \qquad i, j = 0, \ldots, n+1,$$

$$d_i \leq y_i \leq Q, \qquad i = 0, \ldots, n+1,$$

$$x_{ij} \in \{0, 1\}, \qquad i, j = 0, \ldots, n+1.$$

Set partitioning formulation (CVRP)

$$
\begin{aligned}
\min \quad & \sum_{r \in \mathcal{R}} c_r \lambda_r \\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} a_{ri} \lambda_r = 1, \quad i = 1, \ldots, n, \\
& \sum_{r \in \mathcal{R}} \lambda_r \leq K, \\
& \lambda_r \in \{0, 1\}, \qquad r \in \mathcal{R}.
\end{aligned}
$$

- R: set of feasible (complete) routes;
- Routes generated by solving Resource Constrained Elementary Shortest Path Problems (or relaxations).

Main features

VF formulation	SP formulation

Main features

VF formulation

- Weak linear relaxation;

SP formulation

- Strong linear relaxation;

Main features

VF formulation

- Weak linear relaxation;
- Few variables;

SP formulation

- Strong linear relaxation;
- Many variables (may be huge);

Main features

VF formulation

- Weak linear relaxation;
- Few variables;
- Many constraints (may be huge);

SP formulation

- Strong linear relaxation;
- Many variables (may be huge);
- Few constraints;

Main features

VF formulation

- Weak linear relaxation;
- Few variables;
- Many constraints (may be huge);
- Compact formulations or branch-and-cut;

SP formulation

- Strong linear relaxation;
- Many variables (may be huge);
- Few constraints;
- Sophisticate, specialized branch-price-and-cut;

Main features

VF formulation

- Weak linear relaxation;
- Few variables;
- Many constraints (may be huge);
- Compact formulations or branch-and-cut;

SP formulation

- Strong linear relaxation;
- Many variables (may be huge);
- Few constraints;
- Sophisticate, specialized branch-price-and-cut;

- Hence, we can say that they are completely different types of formulations, right?

Main features

VF formulation

- Weak linear relaxation;
- Few variables;
- Many constraints (may be huge);
- Compact formulations or branch-and-cut;

SP formulation

- Strong linear relaxation;
- Many variables (may be huge);
- Few constraints;
- Sophisticate, specialized branch-price-and-cut;

- Hence, we can say that they are completely different types of formulations, right?
- Actually, they are not!

Main features

VF formulation	SP formulation
▶ Weak linear relaxation;	▶ Strong linear relaxation;
▶ Few variables;	▶ Many variables (may be huge);
▶ Many constraints (may be huge);	▶ Few constraints;
▶ Compact formulations or branch-and-cut;	▶ Sophisticate, specialized branch-price-and-cut;

- Hence, we can say that they are completely different types of formulations, right?
- Actually, they are not! :o

Main features

VF formulation

- Weak linear relaxation;
- Few variables;
- Many constraints (may be huge);
- Compact formulations or branch-and-cut;

SP formulation

- Strong linear relaxation;
- Many variables (may be huge);
- Few constraints;
- Sophisticate, specialized branch-price-and-cut;

- Hence, we can say that they are completely different types of formulations, right?
- Actually, they are not! :o
- They are particular cases of a generalized formulation.

p-step formulations

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5)$,

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5), (3, 9)$,

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0,5),(3,9),(7,2),$

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5), (3, 9), (7, 2), (5, 101), \ldots$

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5), (3, 9), (7, 2), (5, 101), \ldots$
  - 3-step paths: $(0, 5, 3, 9),$

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5), (3, 9), (7, 2), (5, 101), \ldots$
  - 3-step paths: $(0, 5, 3, 9), (3, 9, 7, 2),$

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0,5),(3,9),(7,2),(5,101),\ldots$
  - 3-step paths: $(0,5,3,9),(3,9,7,2),(10,8,5,101),\ldots$

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5), (3, 9), (7, 2), (5, 101), \ldots$
  - 3-step paths: $(0, 5, 3, 9), (3, 9, 7, 2), (10, 8, 5, 101), \ldots$
- Let $\mathcal{S}$ be the set of:

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0,5),(3,9),(7,2),(5,101),\dots$
  - 3-step paths: $(0,5,3,9),(3,9,7,2),(10,8,5,101),\dots$
- Let $\mathcal{S}$ be the set of:
  - all p-step paths (starting and ending at any node);

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0, 5), (3, 9), (7, 2), (5, 101), \ldots$
  - 3-step paths: $(0, 5, 3, 9), (3, 9, 7, 2), (10, 8, 5, 101), \ldots$
- Let $\mathcal{S}$ be the set of:
  - all p-step paths (starting and ending at any node);
  - all k-step paths ($1 \leq k < p$) that starts on 0 (depot).

p-step formulations

- Based on variables associated to partial paths: up to p (sequential) *steps* in the network;
- A *step* corresponds to traversing a given arc in the network;
- p-step path: traverses p arcs sequentially;
  - 1-step paths: $(0,5),(3,9),(7,2),(5,101),\dots$
  - 3-step paths: $(0,5,3,9),(3,9,7,2),(10,8,5,101),\dots$
- Let $\mathcal{S}$ be the set of:
  - all p-step paths (starting and ending at any node);
  - all k-step paths ($1 \leq k < p$) that starts on 0 (depot).
- For each $s \in \mathcal{S}$ we have the decision variable:

$$\lambda_s = \begin{cases} 1, & \text{if partial path } s \text{ is used in the solution;} \\ 0, & \text{o.w.} \end{cases}$$

p-step formulation (CVRP)

$$
\begin{aligned}
\text{min} \quad & \sum_{s \in \mathcal{S}} c_s \lambda_s \\
\text{s.t.} \quad & \sum_{s \in \mathcal{S}} e_i^s \lambda_s = 1, && i = 1, \dots, n, \\
& \sum_{s \in \mathcal{S}} a_i^s \lambda_s = 0, && i = 1, \dots, n, \\
& \sum_{s \in \mathcal{S}} a_0^s \lambda_s \le K, && \\
& \varphi_j \ge \varphi_i + q_j \sum_{s \in \mathcal{S}_{ij}} \lambda_s - Q(1 - \sum_{s \in \mathcal{S}_{ij}} \lambda_s), && i = 0, \dots, n, \; j = 1, \dots, n+1, \\
& q_i \le \varphi_i \le Q, && i = 1, \dots, n, \\
& \lambda_s \in \{0, 1\}, && s \in \mathcal{S}.
\end{aligned}
$$

p-step formulation (CVRP)

- c_s is the total cost of traversing arcs in path s;
- e^s and a^s are $n+1$-vectors defined as

$$e_i^s = \begin{cases} +1, & \text{if } i \text{ is visited by path } s\text{, but it is not the last node of } s, \\ 0, & \text{otherwise,} \end{cases}$$

$$a_i^s = \begin{cases} +1, & \text{if } i \text{ is the first node visited by path } s, \\ -1, & \text{if } i \text{ is the last node visited by path } s, \\ 0, & \text{otherwise,} \end{cases}$$

  for all $i = 0, 1, \ldots, n$ and $s \in \mathcal{S}$.
- $\mathcal{S}_{ij} \subset \mathcal{S}$ contains only the paths that traverse arc (i,j) for a pair $i,j \in \mathcal{N}$.

p-step formulation (CVRP)

$$
\begin{aligned}
\text{min} \quad & \sum_{s \in \mathcal{S}} c_s \lambda_s \\
\text{s.t.} \quad & \sum_{s \in \mathcal{S}} e_i^s \lambda_s = 1, && i = 1, \ldots, n, \\
& \sum_{s \in \mathcal{S}} a_i^s \lambda_s = 0, && i = 1, \ldots, n, \\
& \sum_{s \in \mathcal{S}} a_0^s \lambda_s \leq K, \\
& \varphi_j \geq \varphi_i + q_j \sum_{s \in \mathcal{S}_{ij}} \lambda_s - Q(1 - \sum_{s \in \mathcal{S}_{ij}} \lambda_s), && i = 0, \ldots, n, \; j = 1, \ldots, n+1, \\
& q_i \leq \varphi_i \leq Q, && i = 1, \ldots, n, \\
& \lambda_s \in \{0, 1\}, && s \in \mathcal{S}.
\end{aligned}
$$

p-step formulation (VRPTW)

$$
\begin{array}{lll}
\text{min} & \sum_{s \in \mathcal{S}} c_s \lambda_s & \\
\text{s.t.} & \sum_{s \in \mathcal{S}} e_i^s \lambda_s = 1, & i = 1, \ldots, n, \\
& \sum_{s \in \mathcal{S}} a_i^s \lambda_s = 0, & i = 1, \ldots, n, \\
& \sum_{s \in \mathcal{S}} a_0^s \lambda_s \leq K, & \\
& \varphi_j \geq \varphi_i + q_j \sum_{s \in \mathcal{S}_{ij}} \lambda_s - Q(1 - \sum_{s \in \mathcal{S}_{ij}} \lambda_s), & i = 0, \ldots, n, \; j = 1, \ldots, n+1, \\
& \omega_j \geq \omega_i + (s_i + t_{ij}) \sum_{s \in \mathcal{S}_{ij}} \lambda_s - M(1 - \sum_{s \in \mathcal{S}_{ij}} \lambda_s), & i = 0, \ldots, n, \; j = 1, \ldots, n+1, \\
& q_i \leq \varphi_i \leq Q, & i = 1, \ldots, n, \\
& w_i^a \leq \omega_i \leq w_i^b, & i = 0, \ldots, n+1, \\
& \lambda_s \in \{0, 1\}, & s \in \mathcal{S}.
\end{array}
$$

Special cases of p-step formulations

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Proof. In the 1-step formulation, the set $\mathcal{S}$ is given by all the single feasible arcs (i, j), with $i, j = 0, 1, \ldots, n+1$.

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Proof. In the 1-step formulation, the set $\mathcal{S}$ is given by all the single feasible arcs (i, j), with $i, j = 0, 1, \ldots, n+1$. Hence, all variables in this formulation can be rewritten as $\lambda_s = x_{i_s j_s}$, where (i_s, j_s) is the arc traversed by the 1-step path $s \in \mathcal{S}$.

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Proof. In the 1-step formulation, the set $\mathcal{S}$ is given by all the single feasible arcs (i, j), with $i, j = 0, 1, \ldots, n+1$. Hence, all variables in this formulation can be rewritten as $\lambda_s = x_{i_s j_s}$, where (i_s, j_s) is the arc traversed by the 1-step path $s \in \mathcal{S}$. By replacing this in the p-step formulation and noticing that s can be expressed uniquely by its corresponding pair of nodes i_s and j_s, we obtain the vehicle flow formulation.

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Proof. (cont.) On the other hand, in the $(n + 1)$-step formulation, set $\mathcal{S}$ is given by all the k-paths that start at node 0, $1 < k < n + 1$, as no $(n + 1)$-step path exists.

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Proof. (cont.) On the other hand, in the $(n + 1)$-step formulation, set $\mathcal{S}$ is given by all the k-paths that start at node 0, $1 < k < n + 1$, as no $(n + 1)$-step path exists. Hence, $\mathcal{S}$ can be reduced to all feasible complete routes, as in the usual set partitioning formulation.

Special cases of p-step formulations

- We have a family of formulations, as for each $p = 1, \ldots, n$ we have a different p-step formulation;
- Two particular choices are well-known: $p = 1$ and $p = n + 1$;

Proposition 1. The vehicle flow formulation and the set partitioning formulation are particular cases of the p-step formulation, with $p = 1$ and $p = n + 1$, respectively.

Proof. (cont.) On the other hand, in the $(n+1)$-step formulation, set $\mathcal{S}$ is given by all the k-paths that start at node 0, $1 < k < n+1$, as no $(n+1)$-step path exists. Hence, $\mathcal{S}$ can be reduced to all feasible complete routes, as in the usual set partitioning formulation. In addition to that, since any path in $s \in \mathcal{S}$ is a proper feasible route, resource constraints in the p-step formulation are redundant and can be discarded, which leads to the set partitioning formulation. $\square$

Special cases of p-step formulations

- The basic difference among the p-step formulations for different values of p lies in the *level of arc coupling* in the partial paths;

Special cases of p-step formulations

- The basic difference among the p-step formulations for different values of p lies in the *level of arc coupling* in the partial paths;
- In the 1-step formulation the arcs are totally detached, so the model has the freedom to decide what is the most convenient way to connect them (and in the linear relaxation this can be very convenient);

Special cases of p-step formulations

- The basic difference among the p-step formulations for different values of p lies in the *level of arc coupling* in the partial paths;
- In the 1-step formulation the arcs are totally detached, so the model has the freedom to decide what is the most convenient way to connect them (and in the linear relaxation this can be very convenient);
- The good point is that the number of partial paths is small (quadratic), so path generation is easy and quick;

Special cases of p-step formulations

- The basic difference among the p-step formulations for different values of p lies in the *level of arc coupling* in the partial paths;
- In the 1-step formulation the arcs are totally detached, so the model has the freedom to decide what is the most convenient way to connect them (and in the linear relaxation this can be very convenient);
- The good point is that the number of partial paths is small (quadratic), so path generation is easy and quick;
- In the $(n+1)$-step formulation all arcs are already attached, so the model has less freedom when assigning value to them (and the feasible set of the linear relaxation becomes more restricted);

Special cases of p-step formulations

- The basic difference among the p-step formulations for different values of p lies in the *level of arc coupling* in the partial paths;
- In the 1-step formulation the arcs are totally detached, so the model has the freedom to decide what is the most convenient way to connect them (and in the linear relaxation this can be very convenient);
- The good point is that the number of partial paths is small (quadratic), so path generation is easy and quick;
- In the $(n + 1)$-step formulation all arcs are already attached, so the model has less freedom when assigning value to them (and the feasible set of the linear relaxation becomes more restricted);
- The number of paths is huge (exponential) and requires column generation, label-setting and branch-and-price.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. By contradiction, let us assume that $\tilde{z}_{p+1} < \tilde{z}_p$.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. By contradiction, let us assume that $\tilde{z}_{p+1} < \tilde{z}_p$. The p-step formulation has the following partial paths:

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. By contradiction, let us assume that $\tilde{z}_{p+1} < \tilde{z}_p$. The p-step formulation has the following partial paths:

$$(0, v_1),$$
$$(0, v_1, \ldots, v_k), \text{ for } 1 < k \leq p,$$
$$(v_0, v_1, \ldots, v_p), \text{ with } v_0 > 0.$$

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. By contradiction, let us assume that $\tilde{z}_{p+1} < \tilde{z}_p$. The p-step formulation has the following partial paths:

$$(0, v_1),$$
$$(0, v_1, \ldots, v_k), \text{ for } 1 < k \leq p,$$
$$(v_0, v_1, \ldots, v_p), \text{ with } v_0 > 0.$$

In the $(p+1)$-step formulation, we have the partial paths:

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. By contradiction, let us assume that $\tilde{z}_{p+1} < \tilde{z}_p$. The p-step formulation has the following partial paths:

$$(0, v_1),$$
$$(0, v_1, \ldots, v_k), \text{ for } 1 < k \leq p,$$
$$(v_0, v_1, \ldots, v_p), \text{ with } v_0 > 0.$$

In the $(p+1)$-step formulation, we have the partial paths:

$$(0, v_1),$$
$$(0, v_1, \ldots, v_k), \text{ for } 1 < k \leq p,$$
$$(0, v_1, \ldots, v_p, v_{p+1}),$$
$$(v_0, v_1, \ldots, v_p, v_{p+1}), \text{ with } v_0 > 0.$$

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation. Since it is optimal, no partial path with negative reduced cost can be generated by solving the subproblem using u^*.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation. Since it is optimal, no partial path with negative reduced cost can be generated by solving the subproblem using u^*. On the other hand, since $\tilde{z}_{p+1} < \tilde{z}_p$, there must exist a path $(v_0, v_1, \ldots, v_{p+1})$ with a negative reduced cost for u^*.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation. Since it is optimal, no partial path with negative reduced cost can be generated by solving the subproblem using u^*. On the other hand, since $\tilde{z}_{p+1} < \tilde{z}_p$, there must exist a path $(v_0, v_1, \ldots, v_{p+1})$ with a negative reduced cost for u^*. This is a *contradiction*,

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation. Since it is optimal, no partial path with negative reduced cost can be generated by solving the subproblem using u^*. On the other hand, since $\tilde{z}_{p+1} < \tilde{z}_p$, there must exist a path $(v_0, v_1, \ldots, v_{p+1})$ with a negative reduced cost for u^*. This is a *contradiction*, as any p-step path visiting only vertices in $\{v_0, v_1, \ldots, v_{p+1}\}$ has a reduced cost greater than or equal to zero.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation. Since it is optimal, no partial path with negative reduced cost can be generated by solving the subproblem using u^*. On the other hand, since $\tilde{z}_{p+1} < \tilde{z}_p$, there must exist a path $(v_0, v_1, \ldots, v_{p+1})$ with a negative reduced cost for u^*. This is a *contradiction*, as any p-step path visiting only vertices in $\{v_0, v_1, \ldots, v_{p+1}\}$ has a reduced cost greater than or equal to zero. Indeed, the sum of $p+1$ terms cannot be negative if the sum of each combination having p terms of them is non-negative.

Special cases of p-step formulations

Proposition 2. Let $\tilde{z}_p$ be the optimal value of the linear relaxation of the p-step formulation, for $p = 1, \ldots, n+1$. For any $p \in \{1, \ldots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. (cont.) Hence, both formulations have all k-step paths starting on 0, for $k = 1, \ldots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let u^* be a dual optimal solution of the p-step formulation. Since it is optimal, no partial path with negative reduced cost can be generated by solving the subproblem using u^*. On the other hand, since $\tilde{z}_{p+1} < \tilde{z}_p$, there must exist a path $(v_0, v_1, \ldots, v_{p+1})$ with a negative reduced cost for u^*. This is a *contradiction*, as any p-step path visiting only vertices in $\{v_0, v_1, \ldots, v_{p+1}\}$ has a reduced cost greater than or equal to zero. Indeed, the sum of $p+1$ terms cannot be negative if the sum of each combination having p terms of them is non-negative. Therefore, $\tilde{z}_{p+1} \geq \tilde{z}_p$. □

Special cases of p-step formulations

- We show some empirical evidence of the difference in the optimal values for p-step formulations for different values of p;

Special cases of p-step formulations

- We show some empirical evidence of the difference in the optimal values for p-step formulations for different values of p;
- The experiments use the Solomon's instances of the VRPTW, considering only the first 25 customers;

Special cases of p-step formulations

- We show some empirical evidence of the difference in the optimal values for p-step formulations for different values of p;
- The experiments use the Solomon's instances of the VRPTW, considering only the first 25 customers;
- We solve p-step formulations with $p = 1, \ldots, 10$;

Special cases of p-step formulations

- We show some empirical evidence of the difference in the optimal values for p-step formulations for different values of p;
- The experiments use the Solomon's instances of the VRPTW, considering only the first 25 customers;
- We solve p-step formulations with $p = 1, \ldots, 10$;
- All partial paths were generated by full enumeration (feasible only);

Special cases of p-step formulations

- We show some empirical evidence of the difference in the optimal values for p-step formulations for different values of p;
- The experiments use the Solomon's instances of the VRPTW, considering only the first 25 customers;
- We solve p-step formulations with $p = 1, \ldots, 10$;
- All partial paths were generated by full enumeration (feasible only);
- The formulation with all generated paths was solved by a MIP solver (IBM CPLEX v.12.4);

Special cases of p-step formulations

- We show some empirical evidence of the difference in the optimal values for p-step formulations for different values of p;
- The experiments use the Solomon's instances of the VRPTW, considering only the first 25 customers;
- We solve p-step formulations with $p = 1, \ldots, 10$;
- All partial paths were generated by full enumeration (feasible only);
- The formulation with all generated paths was solved by a MIP solver (IBM CPLEX v.12.4);
- Linux OS on a Dell Precision T7600 with Intel E5-6280 2.70 GHz CPU and 192 GB RAM.

Special cases of p-step formulations

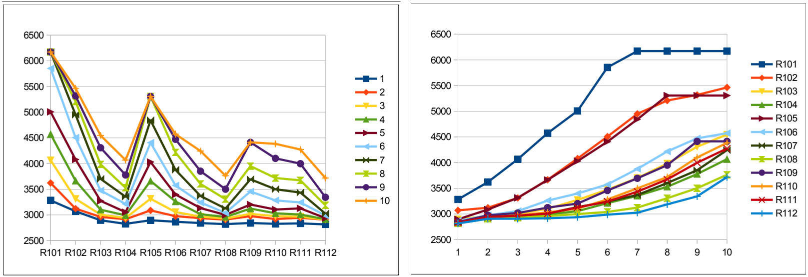


Figure 1. Objective values of the linear relaxations for class R1.

Special cases of p-step formulations

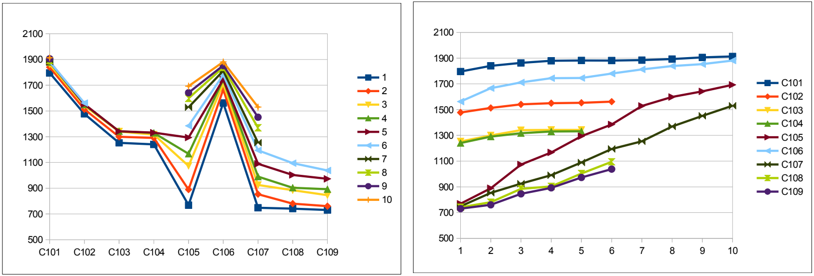


Figure 2. Objective values of the linear relaxations for class C1.

Special cases of p-step formulations

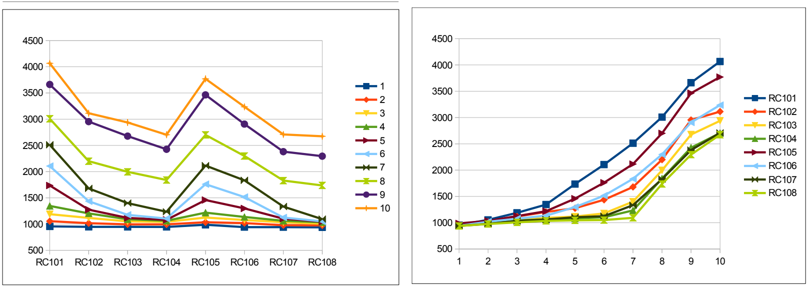
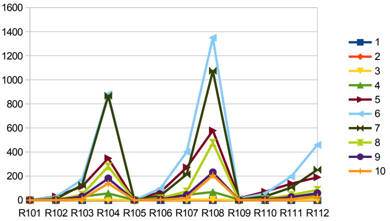


Figure 3. Objective values of the linear relaxations for class RC1.

Special cases of p-step formulations



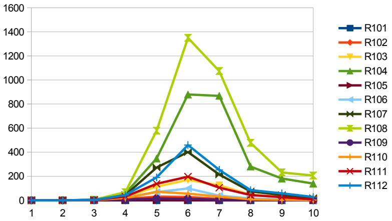


Figure 4. Computational times to solve the linear relaxations for class R1.

Special cases of p-step formulations

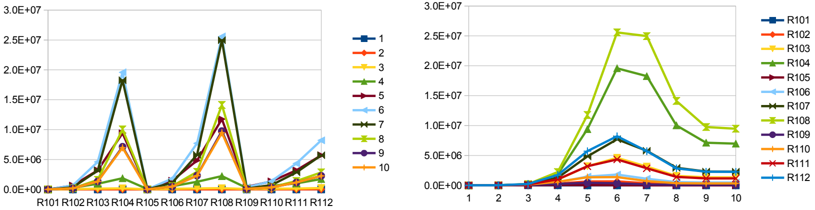


Figure 5. Number of paths generated for class R1.

Column generation for p-step formulations

Restricted master problem, with $\overline{\mathcal{S}} \subset \mathcal{S}$:

$$
\begin{aligned}
\text{min} \quad & \sum_{s \in \overline{\mathcal{S}}} c_s \lambda_s && \\
\text{s.t.} \quad & \sum_{s \in \overline{\mathcal{S}}} e_i^s \lambda_s = 1, && i = 1, \ldots, n, \\
& \sum_{s \in \overline{\mathcal{S}}} a_i^s \lambda_s = 0, && i = 1, \ldots, n, \\
& \sum_{s \in \overline{\mathcal{S}}} a_0^s \lambda_s \leq K, && \\
& \varphi_i - \varphi_j + \sum_{s \in \overline{\mathcal{S}}_{ij}} (q_j + Q) \lambda_s \leq Q, && i = 0, \ldots, n, \ j = 1, \ldots, n+1, \\
& q_i \leq \varphi_i \leq Q, && i = 1, \ldots, n, \\
& \lambda_s \geq 0, && s \in \overline{\mathcal{S}}.
\end{aligned}
$$

Column generation for p-step formulations

Restricted master problem, with $\overline{\mathcal{S}} \subset \mathcal{S}$:

$$
\begin{aligned}
\min \quad & \sum_{s \in \overline{\mathcal{S}}} c_s \lambda_s & \\
\text{s.t.} \quad & \sum_{s \in \overline{\mathcal{S}}} e_i^s \lambda_s = 1, & i = 1, \ldots, n, \\
& \sum_{s \in \overline{\mathcal{S}}} a_i^s \lambda_s = 0, & i = 1, \ldots, n, \\
& \sum_{s \in \overline{\mathcal{S}}} a_0^s \lambda_s \leq K, & \\
& \varphi_i - \varphi_j + \sum_{s \in \overline{\mathcal{S}}_{ij}} (q_j + Q) \lambda_s \leq Q, & i = 0, \ldots, n, \; j = 1, \ldots, n+1, \\
& q_i \leq \varphi_i \leq Q, & i = 1, \ldots, n, \\
& \lambda_s \geq 0, & s \in \overline{\mathcal{S}}.
\end{aligned}
$$

Column generation for p-step formulations

Restricted master problem, with $\overline{\mathcal{S}} \subset \mathcal{S}$:

$$
\begin{aligned}
\text{min} \quad & \sum_{s \in \overline{\mathcal{S}}} c_s \lambda_s & & \\
\text{s.t.} \quad & \sum_{s \in \overline{\mathcal{S}}} e_i^s \lambda_s = 1, & (u^1) \quad & i = 1, \dots, n, \\
& \sum_{s \in \overline{\mathcal{S}}} a_i^s \lambda_s = 0, & (u^2) \quad & i = 1, \dots, n, \\
& \sum_{s \in \overline{\mathcal{S}}} a_0^s \lambda_s \leq K, & (u^3) \quad & \\
& \varphi_i - \varphi_j + \sum_{s \in \overline{\mathcal{S}}_{ij}} (q_j + Q) \lambda_s \leq Q, & (u^4) \quad & i = 0, \dots, n, \ j = 1, \dots, n+1, \\
& q_i \leq \varphi_i \leq Q, & & i = 1, \dots, n, \\
& \lambda_s \geq 0, & & s \in \overline{\mathcal{S}}.
\end{aligned}
$$

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;
  - additional dual values in the reduced cost computation.

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;
  - additional dual values in the reduced cost computation.
- Reduced cost associated to partial path $(v_0, v_1, \ldots, v_k)$:

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;
  - additional dual values in the reduced cost computation.
- Reduced cost associated to partial path $(v_0, v_1, \ldots, v_k)$:

$$\bar{z}_{SP} \quad = \quad \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \right)$$

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;
  - additional dual values in the reduced cost computation.
- Reduced cost associated to partial path $(v_0, v_1, \ldots, v_k)$:

$$\begin{aligned}\bar{z}_{SP} &= \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \right) \\ &\quad -(+1)\overline{u}^3 \quad \text{(if } v_0 = 0\text{)}\end{aligned}$$

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;
  - additional dual values in the reduced cost computation.
- Reduced cost associated to partial path $(v_0, v_1, \dots, v_k)$:

$$
\begin{aligned}
\bar{z}_{SP} \quad = \quad & \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \right) \\
& -(+1)\overline{u}^3 \quad (\text{if } v_0 = 0) \\
& -(+1)\overline{u}^2_{v_0} \quad (\text{if } v_0 \neq 0)
\end{aligned}
$$

Column generation for p-step formulations

- Partial paths are generated by solving a Resource Constrained Elementary Shortest Partial Path Problem (RCESPPP);
- Main differences to usual RCESPP:
  - paths can start at any node and finish at any node;
  - path size becomes a resource;
  - additional dual values in the reduced cost computation.
- Reduced cost associated to partial path $(v_0, v_1, \dots, v_k)$:

$$\begin{aligned}\bar{z}_{SP} \quad = \quad & \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \right) \\ & -(+1)\overline{u}^3 \quad \text{(if } v_0 = 0\text{)} \\ & -(+1)\overline{u}^2_{v_0} \quad \text{(if } v_0 \neq 0\text{)} \\ & -(-1)\overline{u}^2_{v_k} \quad \text{(if } v_k \neq n+1\text{)}\end{aligned}$$

Column generation for p-step formulations

- The reduced cost expression shows that the generation of a p-step path uses more dual information than the generation of a k-step path, for $1 < k < p$;

Column generation for p-step formulations

- The reduced cost expression shows that the generation of a p-step path uses more dual information than the generation of a k-step path, for $1 < k < p$;
- "*Larger routes use more of the dual information*":

Column generation for p-step formulations

- The reduced cost expression shows that the generation of a p-step path uses more dual information than the generation of a k-step path, for $1 < k < p$;
- "*Larger routes use more of the dual information*":
  - 1-step path (v_0, v_1):

Column generation for p-step formulations

- The reduced cost expression shows that the generation of a p-step path uses more dual information than the generation of a k-step path, for $1 < k < p$;
- "*Larger routes use more of the dual information*":
  - 1-step path (v_0, v_1):

$$\begin{aligned} \bar{z}_{SP} \quad = \quad & \left(c_{v_0 v_1} - (+1)\overline{u}^1_{v_0} - (q_{v_1} + Q)\overline{u}^4_{v_0 v_1}\right) \\ & -(+1)\overline{u}^3 \quad \text{(if } v_0 = 0\text{)} \\ & -(+1)\overline{u}^2_{v_0} \quad \text{(if } v_0 \neq 0\text{)} \\ & -(-1)\overline{u}^2_{v_1} \quad \text{(if } v_1 \neq n+1\text{)} \end{aligned}$$

Column generation for p-step formulations

- The reduced cost expression shows that the generation of a p-step path uses more dual information than the generation of a k-step path, for $1 < k < p$;
- "*Larger routes use more of the dual information*":
  - k-step path $(v_0, v_1, \ldots, v_k)$:

$$\begin{aligned} \bar{z}_{SP} \quad = \quad & \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \right) \\ & -(+1)\overline{u}^3 \quad \text{(if } v_0 = 0) \\ & -(+1)\overline{u}^2_{v_0} \ \ \text{(if } v_0 \neq 0) \\ & -(-1)\overline{u}^2_{v_k} \ \ \text{(if } v_k \neq n+1) \end{aligned}$$

Column generation for p-step formulations

- The reduced cost expression shows that the generation of a p-step path uses more dual information than the generation of a k-step path, for $1 < k < p$;
- "*Larger routes use more of the dual information*":
  - p-step path $(v_0, v_1, \ldots, v_k, \ldots, v_p)$:

$$\begin{aligned}
\bar{z}_{SP} \quad = \quad & \sum_{j=0}^{k-1} \Big(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \Big) \\
& + \sum_{j=k}^{p-1} \Big(c_{v_j v_{j+1}} - (+1)\overline{u}^1_{v_j} - (q_{v_{j+1}} + Q)\overline{u}^4_{v_j v_{j+1}} \Big) \\
& -(+1)\overline{u}^3 \quad (\text{if } v_0 = 0) \\
& -(+1)\overline{u}^2_{v_0} \quad (\text{if } v_0 \neq 0) \\
& -(-1)\overline{u}^2_{v_p} \quad (\text{if } v_p \neq n+1)
\end{aligned}$$

Advantages of p-step formulations

Advantages of p-step formulations

- Reduces the label extension, as paths have limited size now;

Advantages of p-step formulations

- Reduces the label extension, as paths have limited size now;
- We can solve $n+1$ subproblems in parallel, one for each starting node;

Advantages of p-step formulations

- Reduces the label extension, as paths have limited size now;
- We can solve $n + 1$ subproblems in parallel, one for each starting node;
- Resources are in the master problem as well, so we send dual information regarding them to subproblem;

Advantages of p-step formulations

- Reduces the label extension, as paths have limited size now;
- We can solve $n+1$ subproblems in parallel, one for each starting node;
- Resources are in the master problem as well, so we send dual information regarding them to subproblem;
- Allow different types of capacity constraints (and valid inequalities, which can be based on p-steps instead of single arcs);

Advantages of p-step formulations

- Reduces the label extension, as paths have limited size now;
- We can solve $n+1$ subproblems in parallel, one for each starting node;
- Resources are in the master problem as well, so we send dual information regarding them to subproblem;
- Allow different types of capacity constraints (and valid inequalities, which can be based on p-steps instead of single arcs);
- Can be easily extended to rich variants of VRP problems (IRP, location routing, etc.);

Advantages of p-step formulations

- Reduces the label extension, as paths have limited size now;
- We can solve $n + 1$ subproblems in parallel, one for each starting node;
- Resources are in the master problem as well, so we send dual information regarding them to subproblem;
- Allow different types of capacity constraints (and valid inequalities, which can be based on p-steps instead of single arcs);
- Can be easily extended to rich variants of VRP problems (IRP, location routing, etc.);
- Turning point: any p-step formulation can be converted to a $p + k$, $k > 1$, at any moment to improve bounds;

Disadvantages of p-step formulations

Disadvantages of p-step formulations

- Linear relaxation bound depends on p;

Disadvantages of p-step formulations

- Linear relaxation bound depends on p;
- Master problem becomes larger, so it takes more time to solve the restricted master problems;

Disadvantages of p-step formulations

- Linear relaxation bound depends on p;
- Master problem becomes larger, so it takes more time to solve the restricted master problems;
  - Interior point methods can help!

Disadvantages of p-step formulations

- Linear relaxation bound depends on p;
- Master problem becomes larger, so it takes more time to solve the restricted master problems;
  - Interior point methods can help!
  - Large-scale master problems:
    Gondzio, J.; González-Brevis, P.; Munari, P. *Large-Scale Optimization with the Primal-Dual Column Generation Method*. Mathematical Programming Computation 8, 2016.

Disadvantages of p-step formulations

- Linear relaxation bound depends on p;
- Master problem becomes larger, so it takes more time to solve the restricted master problems;
  - Interior point methods can help!
  - Large-scale master problems:
    Gondzio, J.; González-Brevis, P.; Munari, P. *Large-Scale Optimization with the Primal-Dual Column Generation Method*. Mathematical Programming Computation 8, 2016.
  - Interior-point BPC:
    Munari, P.; Gondzio, J. *Using the primal-dual interior point algorithm within the branch-price-and-cut method*. Computers & Operations Research 40, 2013.

Generalization of p-step formulations

- In a p-step formulation, the size of the partial paths are limited by the limit on the number of traversed arcs;

Generalization of p-step formulations

- In a p-step formulation, the size of the partial paths are limited by the limit on the number of traversed arcs;
- Since this can be seen as a resource, other types of resource may be used to limit a path: capacity, timing, etc.;

Generalization of p-step formulations

- In a p-step formulation, the size of the partial paths are limited by the limit on the number of traversed arcs;
- Since this can be seen as a resource, other types of resource may be used to limit a path: capacity, timing, etc.;
- For example, we could generate partial paths in which the maximum load is up to a percentage of the capacity, or the total travel time is less than a percentage of the final time instant;

Generalization of p-step formulations

- In a p-step formulation, the size of the partial paths are limited by the limit on the number of traversed arcs;
- Since this can be seen as a resource, other types of resource may be used to limit a path: capacity, timing, etc.;
- For example, we could generate partial paths in which the maximum load is up to a percentage of the capacity, or the total travel time is less than a percentage of the final time instant;
- This is somehow a generalization of what we do in the bidirectional label-extension!

Generalization of p-step formulations

- In a p-step formulation, the size of the partial paths are limited by the limit on the number of traversed arcs;
- Since this can be seen as a resource, other types of resource may be used to limit a path: capacity, timing, etc.;
- For example, we could generate partial paths in which the maximum load is up to a percentage of the capacity, or the total travel time is less than a percentage of the final time instant;
- This is somehow a generalization of what we do in the bidirectional label-extension!
- In this case the joining of paths is done on the master problem;

Generalization of p-step formulations

- In a p-step formulation, the size of the partial paths are limited by the limit on the number of traversed arcs;
- Since this can be seen as a resource, other types of resource may be used to limit a path: capacity, timing, etc.;
- For example, we could generate partial paths in which the maximum load is up to a percentage of the capacity, or the total travel time is less than a percentage of the final time instant;
- This is somehow a generalization of what we do in the bidirectional label-extension!
- In this case the joining of paths is done on the master problem;
- This allows for any type of partitioning in the label-extension, instead of two partitions only (e.g. several partitions of time).

Conclusions and future developments

- New generalized formulation for VRP: p-step formulations;

Conclusions and future developments

- New generalized formulation for VRP: p-step formulations;
- We have shown that the well-known types of VRP formulations are special cases of them;

Conclusions and future developments

- New generalized formulation for VRP: p-step formulations;
- We have shown that the well-known types of VRP formulations are special cases of them;
- Theoretical results and computational evidence relating the bounds provided by p-step formulations;

Conclusions and future developments

- New generalized formulation for VRP: p-step formulations;
- We have shown that the well-known types of VRP formulations are special cases of them;
- Theoretical results and computational evidence relating the bounds provided by p-step formulations;
- Column generation for p-step formulations;

Conclusions and future developments

- New generalized formulation for VRP: p-step formulations;
- We have shown that the well-known types of VRP formulations are special cases of them;
- Theoretical results and computational evidence relating the bounds provided by p-step formulations;
- Column generation for p-step formulations;
- Future developments: branch-price-and-cut for p-step formulations; extending the idea of p-step formulations to other types of optimization problems.

Thank you!

Acknowledgments

FAPESP CNPq ufscar

My webpage

`http://www.dep.ufscar.br/docentes/munari`

PDCGM webpage

`http://www.maths.ed.ac.uk/~gondzio/software/pdcgm.html`