Column Generation for Extended Formulations

R. Sadykov F. Vanderbeck

team RealOpt

INRIA

UNIVERSITÉ BORDEAUX 1 Sciences Technologies

Reformulation involving **extra variables**

$\Downarrow$

tighter relations between variables

- Variable Splitting (binary or unary expansion)
- Network Flow (Multi-Commodity)
- Dynamic Programming Solver **[Martin et al, OR90]**
- Union of Polyhedra **[Balas]**
- Polyhedral Branching Systems **[Kaibel & Loos, LNCS10]**
- ...

often rely on problem **decomposition**

Extended formulations: **why & how**

Reformulation involving **extra variables**

$\Downarrow$

tighter relations between variables

- Variable Splitting (binary or unary expansion)
- Network Flow (Multi-Commodity)
- Dynamic Programming Solver **[Martin et al, OR90]**
- Union of Polyhedra **[Balas]**
- Polyhedral Branching Systems **[Kaibel & Loos, LNCS10]**
- ...

often rely on problem **decomposition**

Extended formulations: **why & how**

Reformulation involving **extra variables**

$\Downarrow$

tighter relations between variables

- Variable Splitting (binary or unary expansion)
- Network Flow (Multi-Commodity)
- Dynamic Programming Solver **[Martin et al, OR90]**
- Union of Polyhedra **[Balas]**
- Polyhedral Branching Systems **[Kaibel & Loos, LNCS10]**
- ...

often rely on problem **decomposition**

Extended formulation in practice

1. Use a **direct MIP-solver** approach: **size is an issue**.

2. Use **projection tools**: Benders' cuts.
   $\longrightarrow$ **dynamic outer approximation** of the intended form.

3. Use **column generation** (and row management)
   $\longrightarrow$ **dynamic inner approximation** of the intended form.

Extended formulation in practice

1. Use a **direct MIP-solver** approach: **size is an issue**.
2. Use **projection tools**: Benders' cuts.
   $\longrightarrow$ **dynamic outer approximation** of the intended form.
3. Use **column generation** (and row management)
   $\longrightarrow$ **dynamic inner approximation** of the intended form.

Extended formulation in practice

1. Use a **direct MIP-solver** approach: **size is an issue**.
2. Use **projection tools**: Benders' cuts.
   $\longrightarrow$ **dynamic outer approximation** of the intended form.
3. Use **column generation** (and row management)
   $\longrightarrow$ **dynamic inner approximation** of the intended form.

Column generation for extended formulations

A **generalization** of the **standard column generation**
(where the extended formulation = Dantzig-Wolfe reformulation)

- **The literature on "column-and-row generation"**
  - Bin Packing **[Carvalho, AOR99]**,
  - Multi-Commodity Flow **[Mamer & Mcbride, MS00]**,
  - Split Delivery Vehicle Routing **[Feillet et al, WP06]**,
  - Network Design **[Feillet et al, ORL10]**, **[Frangioni & Gendron, DAM09, WP10]**.
- **This paper**
  - Formalized algorithm for column-and-row generation
  - Termination criteria that extends to approximate reform.
  - Analysis of the interest of column-and-row generation
  - Comparative computational results across applications

Column generation for extended formulations

A **generalization** of the **standard column generation**
(where the extended formulation = Dantzig-Wolfe reformulation)

- **The literature on "column-and-row generation"**
  - Bin Packing **[Carvalho, AOR99]**,
  - Multi-Commodity Flow **[Mamer & Mcbride, MS00]**,
  - Split Delivery Vehicle Routing **[Feillet et al, WP06]**,
  - Network Design **[Feillet et al, ORL10]**, **[Frangioni & Gendron, DAM09, WP10]**.
- **This paper**
  - Formalized algorithm for column-and-row generation
  - Termination criteria that extends to approximate reform.
  - Analysis of the interest of column-and-row generation
  - Comparative computational results across applications

Column generation for extended formulations

A **generalization** of the **standard column generation**
(where the extended formulation = Dantzig-Wolfe reformulation)

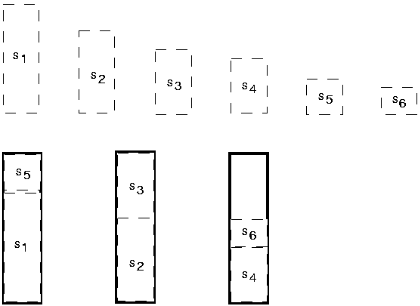
- **The literature on "column-and-row generation"**
  - Bin Packing **[Carvalho, AOR99]**,
  - Multi-Commodity Flow **[Mamer & Mcbride, MS00]**,
  - Split Delivery Vehicle Routing **[Feillet et al, WP06]**,
  - Network Design **[Feillet et al, ORL10]**, **[Frangioni & Gendron, DAM09, WP10]**.
- **This paper**
  - Formalized algorithm for column-and-row generation
  - Termination criteria that extends to approximate reform.
  - Analysis of the interest of column-and-row generation
  - Comparative computational results across applications

Outline

Outline

Bin Packing

$$
\begin{aligned}
\textbf{[F]} \equiv \min\{\sum_k \delta_k \quad &: \\
\sum_k x_{ik} &= 1 \quad \forall i \\
\sum_i s_i\, x_{ik} &\leq C\,\delta_k \quad \forall k \\
x_{ik} &\in \{0,1\} \quad \forall i,k \\
\delta_k &\in \{0,1\} \quad \forall k\}
\end{aligned}
$$



Bin Packing: standard column generation

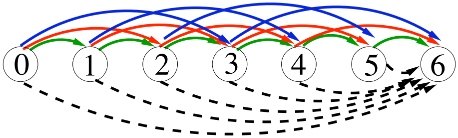
$$
\begin{aligned}
[\mathrm{F}] \equiv \min\{\sum_k \delta_k \quad : & \\
\sum_k x_{ik} &= 1 \quad \forall i \\
\sum_i s_i\, x_{ik} &\leq C\,\delta_k \quad \forall k \\
x_{ik} &\in \{0,1\} \quad \forall i,k \\
\delta_k &\in \{0,1\} \quad \forall k\}
\end{aligned}
$$

$$
[\mathrm{SP}] \equiv \min\{\delta - \sum_i \pi_i x_i : \sum_i s_i x_i \leq C\,\delta, (x,\delta) \in \{0,1\}^{n+1}\}
$$

$$
\begin{aligned}
[\mathrm{M}] \equiv \min\{\sum_g \lambda_g : & \\
\sum_g x_i^g\,\lambda_g &= 1 \quad \forall i \in I \\
\lambda_g &\in \{0,1\} \quad \forall g \in G.\}
\end{aligned}
$$

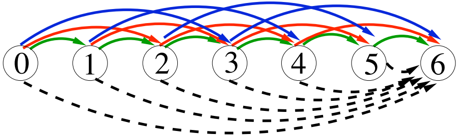
Bin Packing: Network Flow Reformulation

$$
\begin{aligned}
\textbf{[SP]} \equiv \min\{\delta - \sum_{iuv} \pi_i f^i_{uv} : \quad & (f,\delta) \in \{0,1\}^{n*m+1} \\
\sum_{i,v} f^i_{0v} + f_{0C} &= \delta \\
\sum_{i,u} f^i_{uv} &= \sum_{i,u} f^i_{vu} + f_{v,C} \quad v = 1,\dots,C-1 \\
\sum_{i,u} f^i_{uC} + \sum_{v} f_{vC} &= \delta \\
0 \le f^i_{uv} &\le 1 \quad \forall i,u,v = u + s_i \}
\end{aligned}
$$



A relaxation to an unbounded knapsack Problem

Let $F^i_{uv} = \sum_k f^{ik}_{uv}$, $F_{vC} = \sum_k f^k_{vC}$, and $\Delta = \sum_k \delta^k$.



$$\textbf{[R]} \equiv \min\{\Delta : $$

$$\sum_{(u,v)} F^i_{uv} = 1 \quad \forall i$$

$$\sum_{i,v} F^i_{0v} + F_{0C} = \Delta$$

$$\sum_{i,u} F^i_{uv} = \sum_{i,u} F^i_{vu} + F_{vC} \quad v = 1, \ldots, C-1$$

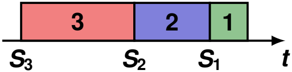
$$\sum_{i,u} F^i_{uC} + \sum_{v} F_{vC} = \Delta$$

$$F^i_{uv} \in \{0,1\} \quad \forall i, (u,v) : v = u + s_i \}.$$

[Valerio de Carvalho, AOR99]

Single Machine Scheduling: reformulations

$$[F] \equiv \min\{\sum_j c(S_j) : \; S_j + p_j \leq S_i \text{ or } S_i + p_i \leq S_j \; \forall (i,j) \in J \times J\}$$



$$[R] \equiv \min\{\sum_{jt} c_{jt} \, z_{jt}$$

$$\sum_{t=1}^{T-p_j+1} z_{jt} = 1 \quad \forall j$$

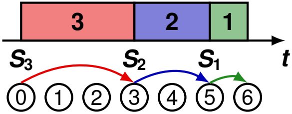
$$\sum_j z_{j0} = 1, \; \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0 \quad = \quad 0, \; t > 0$$

$$z_{jt} \in \{0,1\} \quad \forall j, t\}$$

$$[M] \equiv \min\{\sum_g c^g \lambda_g : \sum_g \sum_{t=1}^{T-p_j+1} z_{jt}^g \lambda_g = 1 \; \forall j, \; \sum_{g \in G} \lambda_g = 1, \; \lambda_g \in \{0,1\} \; \forall g\}$$

Single Machine Scheduling: reformulations

$$[\mathbf{F}] \equiv \min\{\sum_j c(S_j) : \; S_j + p_j \leq S_i \;\text{ or }\; S_i + p_i \leq S_j \; \forall (i,j) \in J \times J\}$$



$$[\mathbf{R}] \equiv \min\{\sum_{jt} c_{jt} \, z_{jt}$$

$$\sum_{t=1}^{T-p_j+1} z_{jt} \quad = \quad 1 \qquad \forall j$$

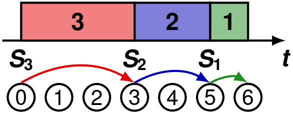
$$\sum_j z_{j0} = 1, \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0 \quad = \quad 0, \; t > 0$$

$$z_{jt} \quad \in \quad \{0,1\} \quad \forall j, t\}$$

$$[\mathbf{M}] \equiv \min\{\sum_g c^g \lambda_g : \; \sum_g \sum_{t=1}^{T-p_j+1} z^g_{jt} \lambda_g = 1 \; \forall j, \sum_{g \in G} \lambda_g = 1, \lambda_g \in \{0,1\} \; \forall g\}$$

Single Machine Scheduling: reformulations

$$\textbf{[F]} \equiv \min\{\sum_j c(S_j) : \; S_j + p_j \le S_i \text{ or } S_i + p_i \le S_j \; \forall (i,j) \in J \times J\}$$



$$\textbf{[R]} \equiv \min\{\sum_{jt} c_{jt} \, z_{jt}$$

$$\sum_{t=1}^{T-p_j+1} z_{jt} \; = \; 1 \qquad \forall j$$

$$\sum_j z_{j0} = 1, \; \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0 \; = \; 0, \; t > 0$$

$$z_{jt} \; \in \; \{0,1\} \quad \forall j,t\}$$

$$\textbf{[M]} \equiv \min\{\sum_g c^g \lambda_g : \sum_g \sum_{t=1}^{T-p_j+1} z^g_{jt} \lambda_g = 1 \; \forall j, \sum_{g \in G} \lambda_g = 1, \lambda_g \in \{0,1\} \; \forall g\}$$

Single Machine Scheduling: Hybrid Approach

1. Generate a column for $[\overline{M}]$ (obtain a pseudo schedule):

$$[SP] \equiv \min\{\sum_{jt}(c_{jt} - \pi_j) z_{jt} : \sum_j z_{j0} = 1, \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0 \ \forall t\},$$

2. Disaggregate the SP solution in arc **variables** z for $[\overline{R}]$

3. Add the associated flow conservation **constraints** to $[\overline{R}]$

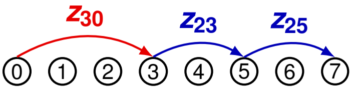
4. Solve the restricted $[\overline{R}]$ to obtain **dual prices** π

Single Machine Scheduling: Hybrid Approach

1. Generate a column for [$\overline{M}$] (obtain a pseudo schedule):

$$[\mathrm{SP}] \equiv \min\Big\{\sum_{jt}(c_{jt} - \pi_j)z_{jt} \,:\, \sum_j z_{j0} = 1,\ \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0\ \forall t\Big\},$$

2. Disaggregate the SP solution in arc **variables** z for [$\overline{R}$]



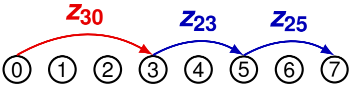
3. Add the associated flow conservation **constraints** to [$\overline{R}$]

4. Solve the restricted [$\overline{R}$] to obtain **dual prices** π

1. Generate a column for $[\overline{M}]$ (obtain a pseudo schedule):

$$[\mathrm{SP}] \equiv \min\{\sum_{jt}(c_{jt}-\pi_j)z_{jt} : \sum_j z_{j0} = 1, \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0 \ \forall t\},$$

2. Disaggregate the SP solution in arc **variables** z for $[\overline{R}]$



3. Add the associated flow conservation **constraints** to $[\overline{R}]$

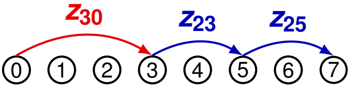
4. Solve the restricted $[\overline{R}]$ to obtain **dual prices** π

Single Machine Scheduling: Hybrid Approach

1. Generate a column for $[\overline{M}]$ (obtain a pseudo schedule):

$$[SP] \equiv \min\{\sum_{jt}(c_{jt} - \pi_j) z_{jt} : \sum_j z_{j0} = 1, \sum_j z_{jt} - \sum_j z_{j,t-p_j} = 0 \ \forall t\},$$

2. Disaggregate the SP solution in arc **variables z** for $[\overline{R}]$



3. Add the associated flow conservation **constraints** to $[\overline{R}]$

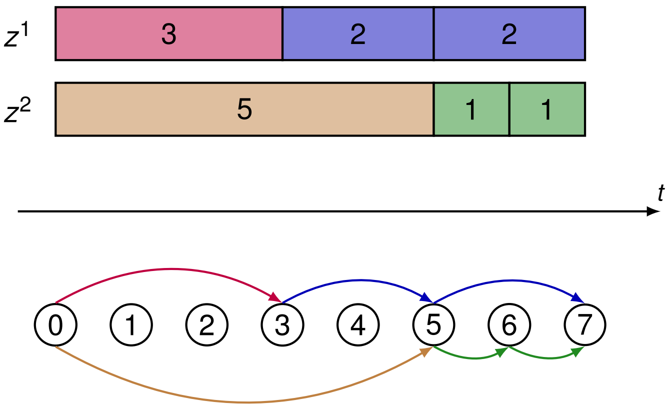
4. Solve the restricted $[\overline{R}]$ to obtain **dual prices** π

Standard Column Gen. versus Hybrid Approach

	Column generation for **[M]**	Column-and-row generation for **[R]**
Initial solution		
Iteration	Subproblem solution	Subproblem solution
1		
2		
3		
...	...	
10		
11		
Final solution		

Interest of the Hybrid Approach: Flow recombination

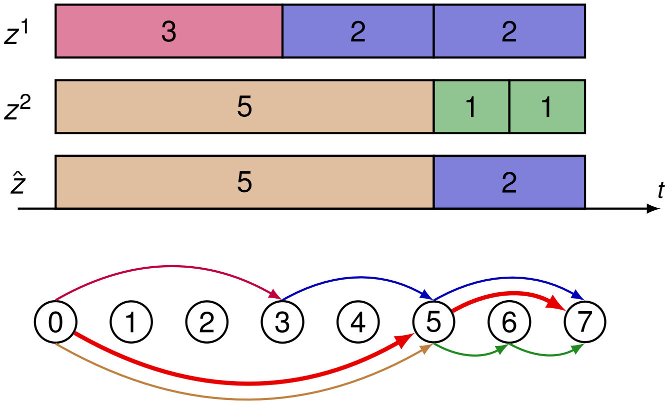
$$Z(\overline{S}) = \{z^1, z^2\}, \qquad \hat{z} \in \overline{R}_{LP}$$



Such recombinations are not feasible in $\overline{M}_{LP}$.

Interest of the Hybrid Approach: Flow recombination

$$Z(\overline{S}) = \{z^1, z^2\}, \qquad \hat{z} \in \overline{R}_{LP}$$



Such recombinations are not feasible in $\overline{\boldsymbol{M}}_{\boldsymbol{LP}}$.

Outline

Assumption 1: ∃ Extended Formulation for a SP

$$[F] \equiv \min\{c\,x \\ A\,x \geq a \\ B\,x \geq b \\ x \in N^n\}$$

Subproblem : $P = \{x \in \mathbb{R}^n_+ : Bx \geq b\}$ and $X = P \cap \mathbb{Z}^n$

Extended Formulation for a Subproblem

∃ a polyhedron $Q = \{z \in \mathbb{R}^e_+ : Hz \geq h, z \in \mathbb{R}^e_+\}$ and transformation T s.t.: Q defines an **extended formulation** for $conv(X)$:

$$conv(X) = proj_x Q = \{x = Tz : Hz \geq h, z \in \mathbb{R}^e_+\}$$

Special Case: **Dantzig-Wolfe reformulation**, let $X = \{x^g\}_{g \in G}$

$$conv(X) = \{x = \sum_g x^g \lambda_g : \sum_g \lambda_g = 1, \ \lambda_g \geq 0\}$$

Assumption 1: $\exists$ Extended Formulation for a SP

$$
\begin{aligned}
[F] \equiv \min\{c\,x \\
A\,x &\geq a \\
B\,x &\geq b \\
x &\in N^n\}
\end{aligned}
$$

Subproblem : $P = \{x \in \mathbb{R}^n_+ : Bx \geq b\}$ and $X = P \cap \mathbb{Z}^n$

Extended Formulation for a Subproblem

$\exists$ a polyhedron $Q = \{z \in \mathbb{R}^e_+ : Hz \geq h, z \in \mathbb{R}^e_+\}$ and transformation T s.t.: Q defines an **extended formulation** for $conv(X)$:

$$conv(X) = proj_x Q = \{x = Tz : Hz \geq h, z \in \mathbb{R}^e_+\}$$

Special Case: Dantzig-Wolfe reformulation, let $X = \{x^g\}_{g \in G}$

$$conv(X) = \{x = \sum_g x^g \lambda_g : \sum_g \lambda_g = 1,\ \lambda_g \geq 0\}$$

Assumption 1: $\exists$ Extended Formulation for a SP

$$[F] \equiv \min\{c\,x \quad A\,x \geq a \quad B\,x \geq b \quad x \in N^n\}$$

$$\textbf{Subproblem}: P = \{x \in \mathbb{R}^n_+ : Bx \geq b\} \quad \text{and} \quad X = P \cap \mathbb{Z}^n$$

Extended Formulation for a Subproblem

$\exists$ a polyhedron $Q = \{z \in \mathbb{R}^e_+ : Hz \geq h, z \in \mathbb{R}^e_+\}$ and transformation T s.t.: Q defines an **extended formulation** for $conv(X)$:

$$conv(X) = proj_x Q = \{x = Tz : Hz \geq h, z \in \mathbb{R}^e_+\}$$

Special Case: **Dantzig-Wolfe reformulation**, let $X = \{x^g\}_{g \in G}$

$$conv(X) = \{x = \sum_g x^g \lambda_g : \sum_g \lambda_g = 1, \; \lambda_g \geq 0\}$$

Extended or Dantzig-Wolfe reformulation

$$
[F] \equiv \min\{c\,x \quad A\,x \geq a, \quad B\,x \geq b, \quad x \in N^n\}
$$

$$
x = T\,z \qquad H\,z \geq h
$$

$$
[R] \equiv \min\{c\,T\,z \quad A\,T\,z \geq a, \quad H\,z \geq h, \quad z \in \mathbb{Z}_+^e\}
$$

$$
x = \sum_g x^g\,\lambda_g \qquad \sum_g \lambda_g = 1
$$

$$
[M] \equiv \min\{\textstyle\sum_{g\in G} c\,x^g\,\lambda_g \quad \sum_{g\in G} A\,x^g\,\lambda_g \geq a, \quad \sum_{g\in G}\lambda_g = 1, \quad \lambda \in \{0,1\}^{|G|}\}
$$

Extended or Dantzig-Wolfe reformulation

$$[F] \equiv \min\{c\,x \quad A\,x \geq a \quad B\,x \geq b \quad x \in N^n\}$$

$$x = T\,z$$

$$H\,z \geq h$$

$$[R] \equiv \min\{c\,T\,z \quad A\,T\,z \geq a \quad H\,z \geq h \quad z \in \mathbb{Z}_+^e\}$$

$$x = \sum_g x^g \lambda_g$$

$$\sum_g \lambda_g = 1$$

$$[M] \equiv \min\{\textstyle\sum_{g \in G} c\,x^g \lambda_g \quad \sum_{g \in G} A\,x^g \lambda_g \geq a \quad \sum_{g \in G} \lambda_g = 1 \quad \lambda \in \{0,1\}^{|G|}\}$$

Extended or Dantzig-Wolfe reformulation

$$
\begin{aligned}
[F] \equiv \min\{c\, x \\
A\, x &\geq a \\
B\, x &\geq b \\
x &\in N^n\}
\end{aligned}
$$

$$
\begin{aligned}
x &= T\, z \\
H\, z &\geq h
\end{aligned}
$$

$$
\begin{aligned}
[R] \equiv \min\{c\, T\, z \\
A\, T\, z &\geq a \\
H\, z &\geq h \\
z &\in \mathbb{Z}_+^e\}
\end{aligned}
$$

$$
x = \sum_g x^g \lambda_g
$$

$$
\sum_g \lambda_g = 1
$$

$$
\begin{aligned}
[M] \equiv \min\{\textstyle\sum_{g \in G} c\, x^g \lambda_g \\
\textstyle\sum_{g \in G} A\, x^g \lambda_g &\geq a \\
\textstyle\sum_{g \in G} \lambda_g &= 1 \\
\lambda &\in \{0, 1\}^{|G|}\}
\end{aligned}
$$

Restricted reformulations

$$[\mathbf{F}] \equiv \min\{c\,x \quad A\,x \geq a \quad B\,x \geq b \quad x \in N^n\}$$

$$[\overline{R}] \equiv \min\{c\,\overline{T}\,\overline{z} \quad A\,\overline{T}\,\overline{z} \geq a \quad \overline{H}\,\overline{z} \geq \overline{h} \quad z \in \mathbb{Z}_+^e\}$$

$$[\overline{M}] \equiv \min\{\sum_{g\in\overline{G}} c\,x^g\,\lambda_g \quad \sum_{g\in\overline{G}} A\,x^g\,\lambda_g \geq a \quad \sum_{g\in\overline{G}} \lambda_g = 1 \quad \lambda \in \{0,1\}^{|\overline{G}|}\}$$

Hybrid Approach: two points of view

1. An alternative to a direct extended formulation approach
   - Dynamic **generation of the variables** of **[$\overline{R}$]**:
     generated **in bunch** by optimizing over a Sub-Problem
   - **Adding rows** that are active for the current SP solution.

$\longrightarrow$ **a specific pricing strategy + delayed constraint generation**

2. An alternative to standard column generation
   - Perform Column Generation for **[$\overline{M}$]**
   - "Project up" **(lift)** the Master Program in space **[$\overline{R}$]**

Assumption 2: $\exists$ Tight Reformulation for a SP

$$
[\mathrm{F}] \equiv \min\{c\,x \quad
\begin{array}{rcl}
A\,x & \geq & a \\
B\,x & \geq & b \\
x & \in & N^n\}
\end{array}
$$

$$\textbf{Subproblem} : P = \{x \in \mathbb{R}^n_+ : Bx \geq b\} \quad \text{and} \quad X = P \cap \mathbb{Z}^n$$

Reformulation for a Subproblem

$\exists$ a polyhedron $Q = \{z \in \mathbb{R}^e_+ : Hz \geq h, z \in \mathbb{R}^e_+\}$ and transformation T s.t.: Q defines an **tighter formulation** for X:

$$conv(X) \subset proj_x Q = \{x = Tz : Hz \geq h, z \in \mathbb{R}^e_+\} \subset P$$

Outline

Pros and Cons of the Hybrid Approach

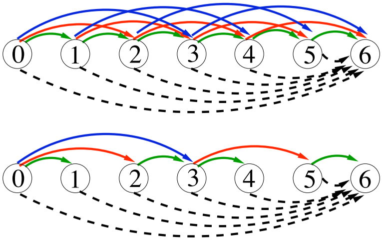
"+"

- Recombinations of SP solutions
- Extra variables for branching **[Valerio de Carvalho, AOR99]**
- Extra variables for defining cuts **[Uchoa et al]**

$$\alpha\, z \geq \alpha_0 \text{ for } \textbf{[R]} \quad \Leftrightarrow \quad \sum_g \alpha\, z^g\, \lambda^g \geq \alpha_0 \text{ for } \textbf{[M]}$$

"-"

- Specific pricing oracle in the extended space Z
- Larger formulation and dynamic row generation
- Symmetries



Outline

Restricted reformulations

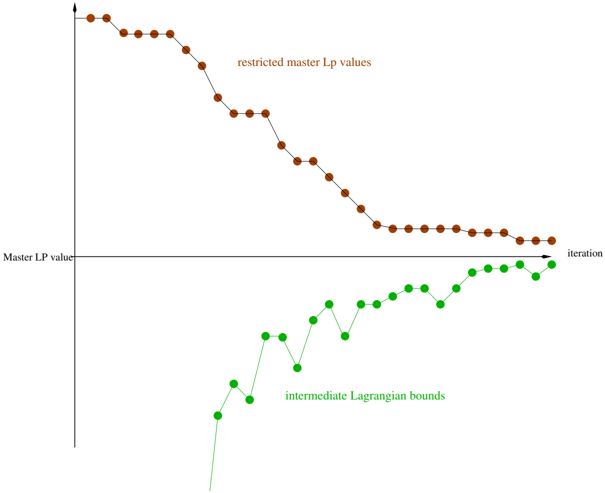
$$\overline{X} = \{x^g\}_{g \in \overline{G}} = \{T\, z^g\}_{g \in \overline{G}}$$

$$\begin{array}{rcll}
[\overline{R}] \equiv \min\{c\, \overline{T}\, \overline{z} & & & \\
A\, \overline{T}\, \overline{z} & \geq & a & \pi \\
\overline{H}\, \overline{z} & \geq & \overline{h} & \sigma \\
z & \in & \mathbb{Z}_+^e\} &
\end{array}$$

$$\begin{array}{rcll}
[\overline{M}] \equiv \min\{\sum_{g \in \overline{G}} c\, x^g\, \lambda_g & & & \\
\sum_{g \in \overline{G}} A\, x^g\, \lambda_g & \geq & a & \pi \\
\sum_{g \in \overline{G}} \lambda_g & = & 1 & \\
\lambda & \in & \{0,1\}^{|\overline{G}|}\} &
\end{array}$$

Hybrid column generation: convergence

$$v_{LP}^{\overline{R}} \geq L(\pi) = \pi\, a + \min_{x \in X}\{(c - \pi\, A)\, x\}$$



Hybrid column generation procedure

Step 0: **Initialize** the dual bound, $\beta := -\infty$, and the subproblem solution set $\overline{G}$ so that the linear relaxation of $[\overline{R}]$ is feasible.

Step 1: **Solve** $[\overline{R}_{LP}]$ and collect the dual solution π.

Step 2: Solve the **pricing problem**: $z^* \leftarrow \min\{(c - \pi A)\, T\, z :\ z \in Z\} = \min\{(c - \pi A)\, x :\ x \in X\}$.

Step 3: Compute the Lagrangian **dual bound**:
$L(\pi) = \pi\, a + (c - \pi A)\, T\, z^*$, and update the dual bound $\beta := \max\{\beta, L(\pi)\}$ (Lagrangian dual value estimator).
If $v_{LP}^{\overline{R}} \leq \beta$, **STOP**.

Step 4: **Update** the current bundle $\overline{G}$ by adding solution $z^s := z^*$ and update the resulting restricted reformulation $[\overline{R}]$. Goto Step 1.

Proposition (Termination)

- *Either $v_{LP}^{\overline{R}} \leq \beta$, or $[(c - \pi A)\, T - \sigma\, H]z^* < 0$ and some of the components of z^* have negative reduced cost in $[\overline{R}_{LP}]$.*
- *On termination, $v_{LP}^{\overline{R}} \beta v_{LP}^{M} = \min\{cx : Ax \geq a, x \in conv(X)\}$ (under Assumption).*

Hybrid column generation procedure

Step 0: **Initialize** the dual bound, $\beta := -\infty$, and the subproblem solution set $\overline{G}$ so that the linear relaxation of $[\overline{R}]$ is feasible.

Step 1: **Solve** $[\overline{R}_{LP}]$ and collect the dual solution π.

Step 2: Solve the **pricing problem**: $z^* \leftarrow$
$\min\{(c - \pi A)\, T\, z : \; z \in Z\} = \min\{(c - \pi A)\, x : \; x \in X\}$.

Step 3: Compute the Lagrangian **dual bound**:
$L(\pi) = \pi\, a + (c - \pi A)\; T\, z^*$, and update the dual bound
$\beta := \max\{\beta, L(\pi)\}$ (Lagrangian dual value estimator).
If $v^{\overline{R}}_{LP} \leq \beta$, **STOP**.

Step 4: **Update** the current bundle $\overline{G}$ by adding solution $z^s := z^*$ and update the resulting restricted reformulation $[\overline{R}]$. Goto Step 1.

Proposition (Termination)

- *Either $v^{\overline{R}}_{LP} \leq \beta$, or $[(c - \pi A)\; T - \sigma\; H] z^* < 0$ and some of the components of z^* have negative reduced cost in $[\overline{R}_{LP}]$.*
- *On termination, $v^{\overline{R}}_{LP} = \beta = v^{M}_{LP} = \min\{cx : Ax \geq a, x \in conv(X)\}$ (under Assumption 1).*

Hybrid column generation procedure

Step 0: **Initialize** the dual bound, $\beta := -\infty$, and the subproblem solution set $\overline{G}$ so that the linear relaxation of $[\overline{R}]$ is feasible.

Step 1: **Solve** $[\overline{R}_{LP}]$ and collect the dual solution π.

Step 2: Solve the **pricing problem**: $z^* \leftarrow$
$\min\{(c - \pi A)\, T\, z : \; z \in Z\} = \min\{(c - \pi A)\, x : \; x \in X\}$.

Step 3: Compute the Lagrangian **dual bound**:
$L(\pi) = \pi\, a + (c - \pi A)\, T\, z^*$, and update the dual bound
$\beta := \max\{\beta, L(\pi)\}$ (Lagrangian dual value estimator).
If $v_{LP}^{\overline{R}} \leq \beta$, **STOP**.

Step 4: **Update** the current bundle $\overline{G}$ by adding solution $z^s := z^*$ and update the resulting restricted reformulation $[\overline{R}]$. Goto Step 1.

Proposition (Termination)

- *Either $v_{LP}^{\overline{R}} \leq \beta$, or $[(c - \pi A)\, T - \sigma\, H] z^* < 0$ and some of the components of z^* have negative reduced cost in $[\overline{R}_{LP}]$.*
- *On termination, $v_{LP}^{\overline{R}} \leq \beta \leq v_{LP}^{M} = \min\{cx : Ax \geq a, x \in conv(X)\}$ (under Assumption 2, one has inequalities).*

Lifting Procedure

$$\boldsymbol{x} \in \boldsymbol{X} \longrightarrow \boldsymbol{z} \in \boldsymbol{p}^{-1}(\boldsymbol{x})$$

where

$$\boldsymbol{p}^{-1}(\boldsymbol{x}) := \{\boldsymbol{z} \in \mathbb{Z}_+^{\boldsymbol{e}} : \boldsymbol{T}\,\boldsymbol{z} = \boldsymbol{x};\ \boldsymbol{H}\,\boldsymbol{z} \geq \boldsymbol{h}\}\ .$$

NB: A generic procedure is to solve this feasibility problem.

Desirable property (in an application specific context):

Property ("Lifting")

A lifting procedure is available to transform any $\boldsymbol{x} \in \boldsymbol{X}$ into a solution to the extended system $\boldsymbol{z} \in \boldsymbol{Z}$ such that $\boldsymbol{x} = \boldsymbol{T}\,\boldsymbol{z}$.

- Then one can use an oracle for $\mathbf{min}\{(\boldsymbol{c} - \pi \boldsymbol{A})\,\boldsymbol{x} :\ \boldsymbol{x} \in \boldsymbol{X}\}$
- It can break symmetries (at least partially)

Outline

Why Col Gen for [R] converges faster than for [M]

Property (recombination)

Given $\overline{G} \subset G$, $\exists \hat{z} \in R_{LP}(\overline{G})$, such that $\hat{z} \notin conv(Z(\overline{G}))$.

$$\overline{R}_{LP} \supset \overline{M}_{LP} \quad \Rightarrow \quad v^{\overline{R}}_{LP} \leq v^{\overline{M}}_{LP}$$

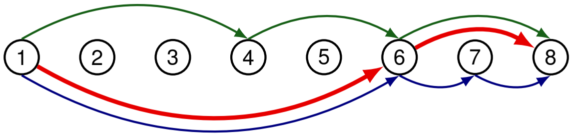
True for Network Flow based reformulations: $w = z^1 - z^2$ is a cycle flow; w decomposes into elementary cycle flow w^A, $w^B, \ldots$; $\hat{z} = z^1 + \alpha w^A \in \overline{Q}$; but, $\hat{z} \notin conv(z^1, z^2)$

Why Col Gen for [R] converges faster than for [M]

Property (recombination)

Given $\overline{G} \subset G$, $\exists \hat{z} \in R_{LP}(\overline{G})$, such that $\hat{z} \notin conv(Z(\overline{G}))$.

$$\overline{R}_{LP} \supset \overline{M}_{LP} \quad \Rightarrow \quad v_{LP}^{\overline{R}} \leq v_{LP}^{\overline{M}}$$

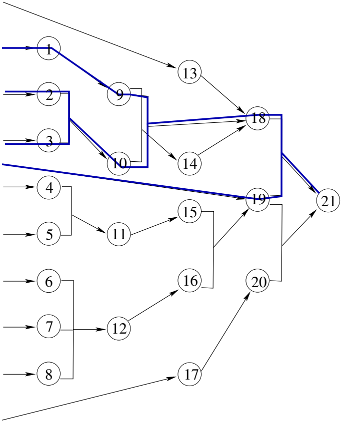


True for Network Flow based reformulations: $w = z^1 - z^2$ is a cycle flow; w decomposes into elementary cycle flow w^A, $w^B, \ldots$; $\hat{z} = z^1 + \alpha w^A \in \overline{Q}$; but, $\hat{z} \notin conv(z^1, z^2)$

[Martin et al OR90]
When the subproblem can be solved by dynamic programming,

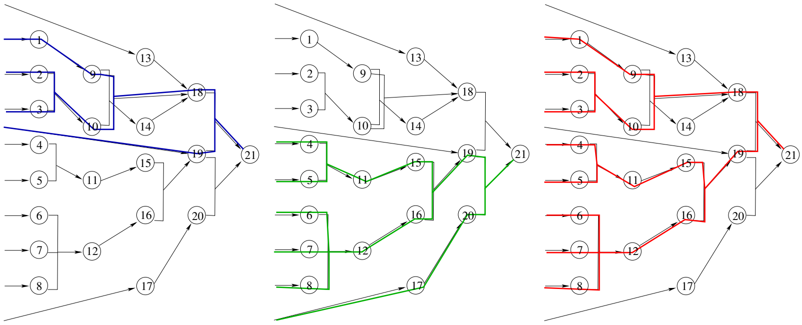
$$\gamma(I) = \min_{(J,I)\in\mathcal{A}} \{\sum_{j\in J} \gamma(j) + c(J,I)\},$$

an extended formulation is to model a decision tree in an hyper-graph:



Recombinations for DP based reformulations

The recombination of DP sol 1 and DP sol 2 into DP sol 3



[Vanderbeck & Savelsbergh, ORL06]

- **Disaggregation helps** as when exploiting block-diag.

$$\begin{array}{ccccc} B^1 x^1 & & & & \geq b^1 \\ & B^2 x^2 & & & \geq b^2 \\ & & \ddots & & \geq \vdots \\ & & & B^K x^K & \geq b^K \end{array}$$

- **Exchange vectors** (also known as dual cuts)

$$v = x^{g_1} - x^{g_2} \quad \text{for } x^{g_1}, x^{g_2} \in X$$

$$x = \sum_g x^g \lambda_g + \sum_r v^r \mu_r$$

- **Base-generators** as in the network design example

$$x^g \quad \text{for } (x, y) \in X \text{ with } x = x^g$$

- **State space relaxation**, under Assumption 2

$$x^g \in X^R$$

Outline

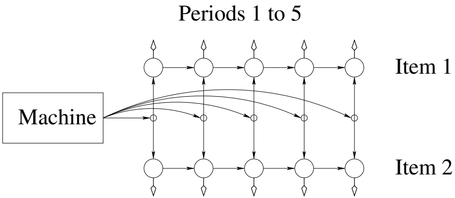
Machine Scheduling

n	p_{max}	Cplex 12.1 for [R_{LP}] cpu	Col gen for [M_{LP}] it	cpu	Col-and-row gen for [R_{LP}] it	%var	cpu
1 machine							
25	50	3	352	2	54	8.9%	1
50	50	35	1559	42	82	6.7%	6
100	50	382	9723	2531	112	6.1%	47
25	100	11	378	2	75	5.9%	2
50	100	155	1418	44	114	4.6%	18
100	100	2040	10375	3436	155	4.5%	182
2 machines							
25	100	7	208	1	62	5.4%	1
50	100	199	641	10	93	4.5%	3
100	100	4038	2697	198	115	4.5%	30
4 machines							
50	100	35	441	3	90	4.4%	2
100	100	726	1353	47	113	4.3%	10
200	100	22442	4306	685	151	3.1%	80

class $C = 4000$	n	Cplex 12.1 for $[F_{LP}]$ *gap*	*cpu*	Col gen for $[M_{LP}]$ (*gap*=0) *it*	*cpu*	Col-and-row gen for $[R_{LP}]$ (*gap*=0) *it*	*%z*	*cpu*
"a2"	200	5.6	0.1	944	0.6	736	0.4	1.1
$s_i \in$	400	8.6	0.8	2863	3.3	1726	0.4	4.8
$[1000, 3000]$	800	6.6	10.4	10358	31.5	4918	0.4	34.6
"a3"	200	4.0	0.1	148	0.1	116	0.1	0.2
$s_i \in$	400	8.6	0.6	283	1.0	186	0.1	1.0
$[1000, 1500]$	800	17.4	7.7	578	9.3	294	0.1	5.9
"a4"	200	0.8	0.1	501	1.0	279	0.2	1.0
$s_i \in$	400	1.8	0.6	1045	6.8	431	0.2	3.8
$[800, 1300]$	800	2.8	5.8	2070	54.3	656	0.1	17.4

Generalized Assignment

		Cplex 12.1 for [F_{LP}]		Col gen for [M_{LP}]			Col-and-row gen for [R_{LP}]			
m	*n*	*%gap*	*cpu*	*it*	*%gap*	*cpu*	*it*	*%gap*	*%z*	*cpu*
20	100	1.17	0.05	78	0.09	0.8	32	0.49	2.26	1.4
10	100	0.55	0.03	195	0.10	1.3	35	0.38	1.95	1.1
5	100	0.26	0.01	712	0.05	6.2	34	0.23	1.63	1.1
20	200	0.28	0.10	231	0.02	10.8	40	0.18	1.25	9.3
10	200	0.17	0.05	959	0.04	68.9	41	0.15	1.06	8.7
5	200	0.07	0.02	6389	0.02	2401.9	39	0.07	0.88	8.7
40	400	0.15	0.51	291	0.03	120.3	42	0.12		88.5
20	400	0.09	0.23	1078	0.03	1077.9	46	0.08		79.2
10	400	0.04	0.11				46	0.04		71.4



		Cplex 12.1 for [R_{LP}]	Col gen for [M_{LP}]		Col-and-row gen for [R_{LP}]		
n	*p*	*cpu*	*sp*	*cpu*	*sp*	*%var*	*cpu*
20	100	6.3	2170	2.5	764	2.9%	2.3
40	200	161.9	6216	21.0	2192	1.1%	32.0
10	100	2.2	2444	4.5	348	3.6%	1.1
20	200	60.5	7024	43.5	1006	1.6%	15.4
40	400	1847.9	18668	544.2	2672	0.6%	232.5

$\frac{p}{n}$ is an indication of the # of setups per item
(= number of arcs in a path defining a production planning)

Multi-Echelon Multi-Item Lot-Sizing – Small Bucket

		Cplex 12.1 for [R_{LP}]	Col. gen. for [M_{LP}]		Col-and-row gen for [R_{LP}]		
n	p	*cpu*	*it*	*cpu*	*it*	*%z*	*cpu*
2 echelons							
10	50	$>$1h	164	2.3	29	0.59%	1.6
20	50	$>$1h	70	1.5	25	0.46%	2.8
10	100	$>$1h	1039	254.1	37	0.15%	7.1
20	100	$>$1h	404	65.1	34	0.15%	19.9
3 echelons							
10	50	$>$1h	306	20.8	39	0.18%	5.2
20	50	$>$1h	129	12.1	31	0.13%	9.1
10	100	$>$1h	1874	3600.3	47	0.03%	35.6
20	100	$>$1h	952	1415.0	47	0.02%	98.7
5 echelons							
10	50	$>$1h	1045	774.0	49	0.12%	19.1
20	50	$>$1h	682	281.2	38	0.08%	30.2
10	100	$>$1h	$>$3000	$>$9h	68	0.02%	152.0
20	100	$>$1h	2998	32667.4	56	0.01%	428.9

Conclusion

1. Column generation for an extended formulation is **to be considered when**:
   - The extended formulation $\leftarrow$ **decomposition principle**,
   - SP solutions can **be recombined** into alternative ones.

2. Computational results (ours and in the literature) show that this can be **competitive approach**.

   $$\text{Gain factors: } \boldsymbol{\Delta} it \in [\mathbf{2}, \mathbf{100}], \quad \boldsymbol{\Delta} cup \in [\mathbf{1}, \mathbf{50}]$$

3. It can be interpreted a **stabilization method** for column generation.

4. It can be **applied to an approximation** of an extended formulation trade-off *"speed and recombinations"* versus *"dual bound quality"*

Multi-Commodity Capacitated Network Design

$$
\begin{aligned}
[\mathbf{F}] \equiv \min\{\sum_{ijk} c^k_{ij}\, x^k_{ij} + \sum_{ij} f_{ij}\, y_{ij} \\
\sum_j x^k_{ji} - \sum_j x^k_{ij} &= d^k_i \quad \forall i, k \\
\sum_k x^k_{ij} &\leq u_{ij}\, y_{ij} \quad \forall i, j \\
x^k_{ij} &\geq 0 \quad \forall i, j, k \\
y_{ij} &\in N \quad \forall i, j\}
\end{aligned}
$$

$$
\begin{aligned}
[SP^{ij}] \equiv \min\{\sum_k c^k\, x^k + f\, y : \\
\sum_k x^k &\leq u\, y \\
x^k &\leq \min\{d^k, u\, y\} \forall k\}
\end{aligned}
$$

Network Design: Extended form. for the SPs

Let $y_{ij}^s = 1$ and $x_{ij}^{ks} = x_{ij}^k$ if $y_{ij} = s$.

$$[SP^{ij}] \equiv \min\{\sum_{ks} c_{ij}^k \, x_{ij}^{ks} + \sum_{s} f_{ij} \, s \, y_{ij}^s :$$

$$\begin{aligned} \sum_s y_{ij}^s &\leq 1 \\ (s-1)\, u_{ij}\, y_{ij}^s \leq \sum_k x_{ij}^{ks} &\leq s \, u_{ij} \, y_{ij}^s \quad \forall s \\ x_{ij}^{ks} &\leq \min\{d^k, s\, u_{ij}\} \, y_{ij}^s \quad \forall k, s\} \end{aligned}$$

Extended formulation for the arc design subproblem (Union of Polyhedra)
[Croxton, Gendron and Magnanti OR07]

Network Design: extended formulation

$$
\begin{aligned}
\textbf{[R]} \equiv \min\{\sum_{ijks} c_{ij}^{k}\, x_{ij}^{ks} + \sum_{ijs} f_{ij}\, s\, y_{ij}^{s} \\
\sum_{js} x_{ji}^{ks} - \sum_{js} x_{ij}^{ks} &= d_i^k \quad \forall i, k \\
(s-1)\, u_{ij}\, y_{ij}^{s} \le \sum_{k} x_{ij}^{ks} &\le s\, u_{ij}\, y_{ij}^{s} \quad \forall i, j, s \\
0 \le x_{ij}^{ks} &\le d^k y_{ij}^{s} \quad \forall i, j, k, s \\
\sum_{s} y_{ij}^{s} &= 1 \quad \forall i, j \\
y_{ij}^{s} &\in \{0, 1\} \quad \forall i, j, s\}
\end{aligned}
$$

[Frangioni & Gendron, DAM09]

- solved by col-and-row generation
- better performance than by adding Benders' cut to **[F]**

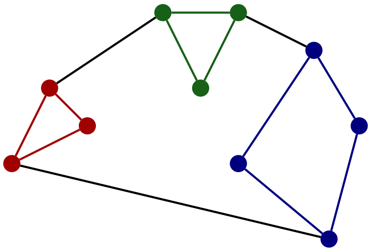
Network Design: standard col. gen. formulation

$$
\begin{aligned}
\textbf{[M]} \equiv \min\{ \sum_{i,j,s,g \in G^{ij}} (c_{ij}^k \, x_{ks}^g + f_{ij} \, s \, y_s^g) \, \lambda_g^{ij} & & \\
\sum_{js} \sum_{g \in G^{ij}} x_{ks}^g \, \lambda_g^{ij} - \sum_{js} \sum_{g \in G^{ij}} x_{ks}^g \, \lambda_g^{ij} & = d_i^k & \forall i, k \\
\sum_{g \in G^{ij}} \lambda_g^{ij} & \leq 1 & \forall i, j \\
\lambda_g^{ij} & \in \{0, 1\} & \forall i, j, g \in G^{ij} \}
\end{aligned}
$$

[Frangioni & Gendron WP10]

- col-and-row generation for **[R]** outperforms standard col gen for **[M]**

Network Design: Union of Polyhedra with [R]



Network Design: Union of Polyhedra with [M]