Extended Formulations, Column Generation, and stabilization: synergies in the benefit of large scale applications

François Vanderbeck

University of Bordeaux INRIA Bordeaux-Sud-Ouest

Co-authors: **A. Pessoa, R. Sadykov, E. Uchoa, L.A. Wolsey**

EURO/INFORMS, Rome, July 2013

References

- F. Vanderbeck and L. A. Wolsey, Reformulation and Decomposition of Integer Programs, *50 Years of Integer Programming*, editors Junger et al., (2010).
- R. Sadykov and F. Vanderbeck, Column Generation for Extended Formulations, *EURO Journal on Computational Optimization* (2013).
- A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, In-Out Separation and Column Generation Stabilization by Dual Price Smoothing, Symp. on Experimental Algor. (SEA), *Lect. Notes in Comp. Sc.* (2013).
- R. Sadykov and al., Solving a Freight Railcar Flow Problem Variant Arising in Russia, *Working Paper* (2013).

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by combining ideas of
  - Restriction / Relaxation,
  - Benders projection, and
  - Dantzig-Wolfe dynamic generation.
- With dynamic generation, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for efficiency enhancement features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Outline

1. Extented Formulations
   - Definitions
   - Interests
   - Coping with its large size

2. Dynamic Row-and-Column Generation
   - Methodology
   - Practical issues

3. Large-scale application
   - Freight transport by rail in Russia

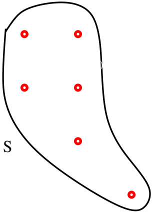
Outline

1. Extented Formulations
   - Definitions
   - Interests
   - Coping with its large size
2. Dynamic Row-and-Column Generation
   - Methodology
   - Practical issues
3. Large-scale application
   - Freight transport by rail in Russia

Combinatorial Optimization Problem

$$(CO) \equiv \min\{c(s): \; s \in \mathbf{S}\}$$

where $\mathbf{S}$ is the "discrete" set of feasible solutions.



Formulation

Combinatorial Optimization Problem

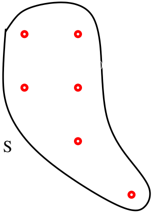
$$(CO) \equiv \min\{c(s) : \; s \in \mathbf{S}\}$$

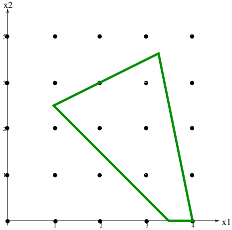
where $\mathbf{S}$ is the “discrete” set of feasible solutions.

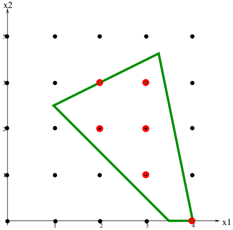
Formulation

A **polyhedron** $\mathbf{P} = \{x \in \mathbb{R}^n : Ax \geq a\}$ is a formulation for (CO) iff

$$\min\{c(s) : \; s \in \mathbf{S}\} \quad \equiv \quad \min\{cx : x \in \mathbf{P_I} = \mathbf{P} \cap \mathbb{N}^{\mathbf{n}}\}.$$





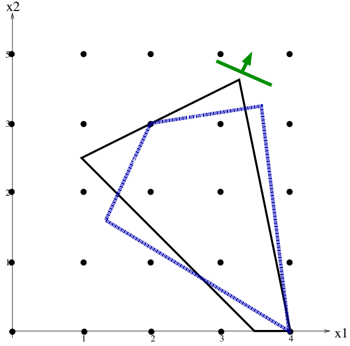


Alternative formulations

A formulation is typically not unique

P and P' can be **alternative formulations** for (CO) if

$$(CO) \quad \equiv \quad \min\{cx : x \in P \cap \mathbb{N}^n\} \quad \equiv \quad \min\{c'x' : x' \in P' \cap \mathbb{N}^{n'}\}$$

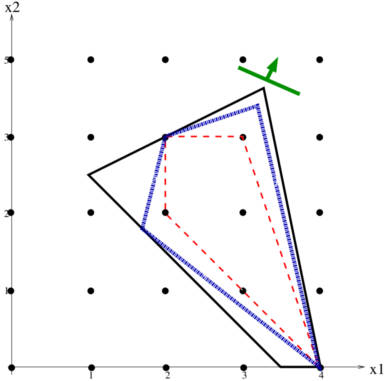


warning: can expressed in different variable-spaces.

Stronger formulation (in the same space)

Formulation $P' \subseteq \mathbb{R}^n$ is a **stronger** than $P \subseteq \mathbb{R}^n$ if $P' \subset P$. Then,

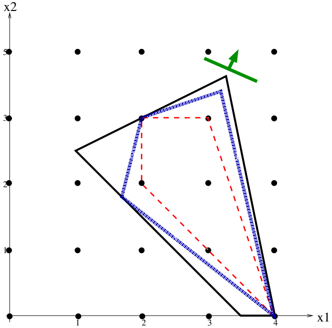
$$\min\{cx' : x' \in P'\} \geq \min\{cx : x \in P\}$$



Ideal Formulation

The Convex hull of an IP set, P_I

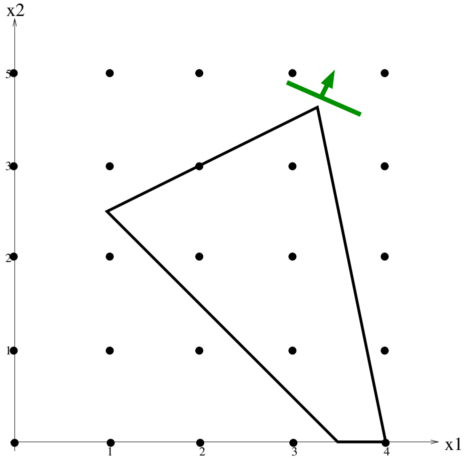
$\mathrm{conv}(P_I)$ is the smallest closed convex set containing P_I.



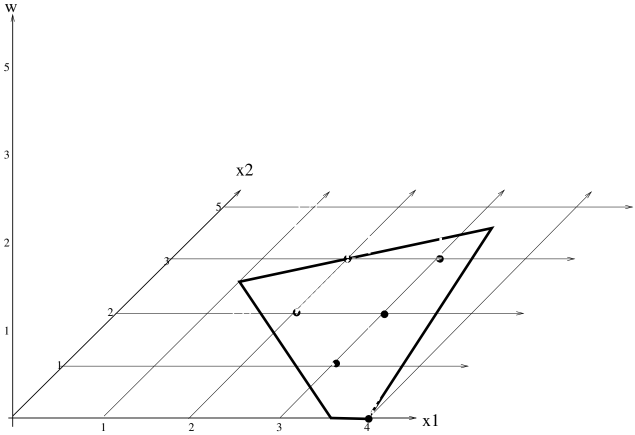
$\mathbf{conv}(P_I)$ is an ideal polyhedron / formulation

If P_I is defined by rational data, $\mathrm{conv}(P_I)$ is a polyhedron.

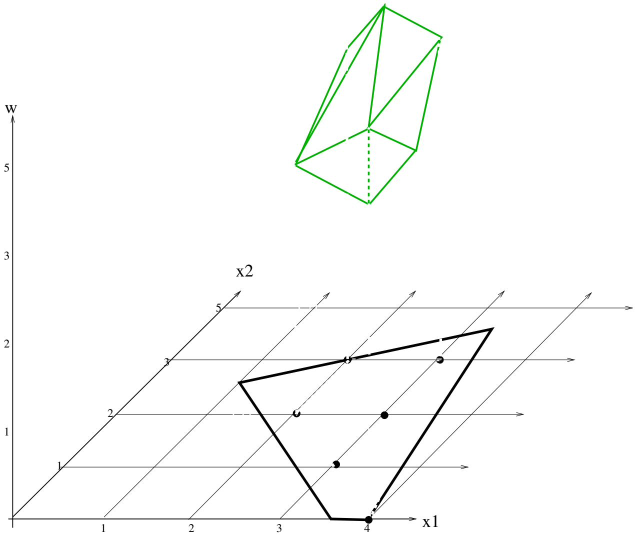
Given an initial **compact formulation**:



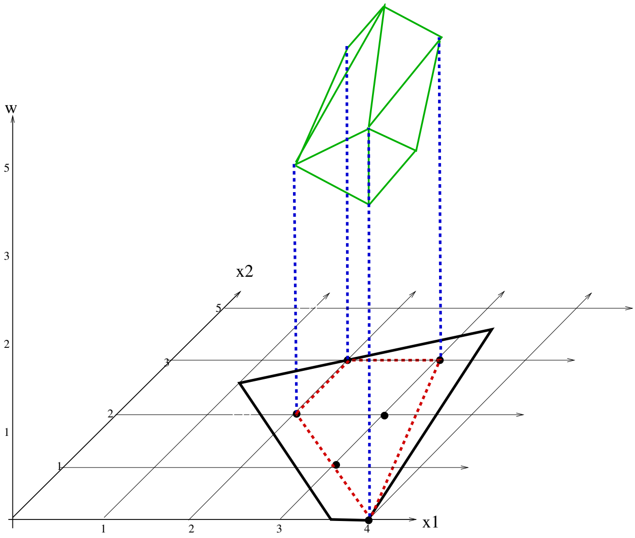
Extended Formulation



Extended Formulation



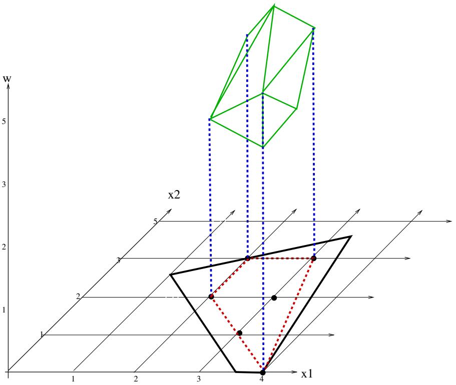
Extended Formulation



The Projection

of $Q = \{(x, w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ on the x-space is:

$$\text{proj}_x(Q) := \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^e \text{ such that } (x, w) \in Q\}.$$



Projection

The Projection

of $Q = \{(x, w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ on the x-space is:

$$\text{proj}_x(Q) := \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^e \text{ such that } (x, w) \in Q\}.$$

Farka's Lemma

Given $\tilde{x}$,

$$\{w \in \mathbb{R}^n_+ : Hw \geq (d - G\,\tilde{x}\,)\} \neq \emptyset$$

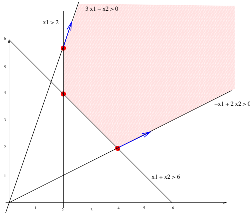
if and only if

$$\forall v \in \mathbb{R}^m_+ : vH \leq 0, \quad v(d - G\,\tilde{x}\,)\ \leq 0.$$

Hence, a polyhedral description of the projection in the x-space is:

$$\text{proj}_x(Q) = \{x \in \mathbb{R}^n : v^j(d - Gx)\ \leq 0 \quad j \in J\}$$

$\{v^j\}_{j \in J}$, exteme rays. of $\{v \in \mathbb{R}^m_+ : vH \leq 0\}$.

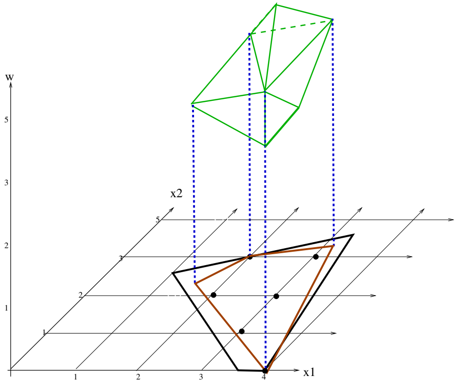


Extended Formulations

An extended formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is a polyhedron $Q = \{(x, w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ such that

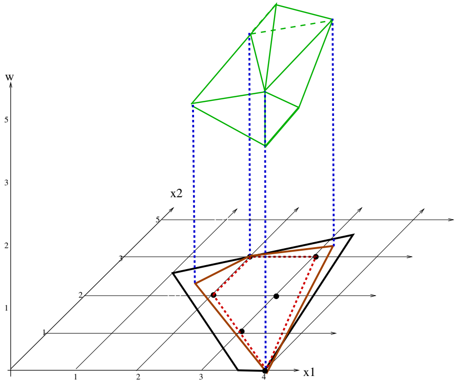
$$P_I = \mathrm{proj}_x(Q) \cap \mathbb{N}^n.$$



An extended formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is a polyhedron $Q = \{(x,w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ such that

$$P_I = \text{proj}_x(Q) \cap \mathbb{N}^n.$$

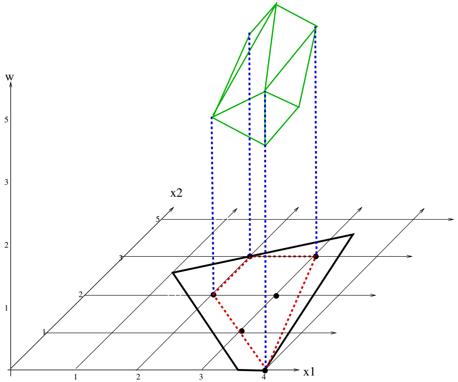


Tight Extended Formulations

A tight extended formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is a polyhedron $Q = \{(x,w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ such that

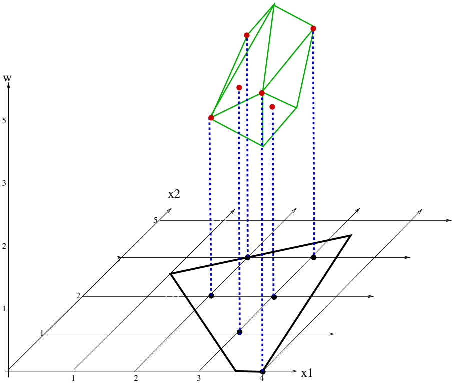
$$\mathbf{conv}(P_I) = \mathrm{proj}_x(Q).$$



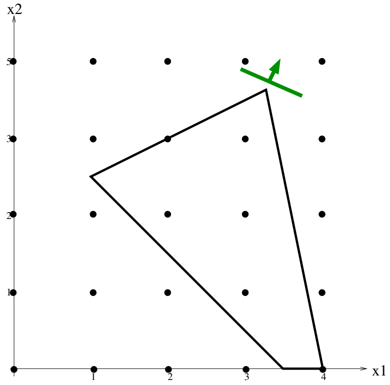
An extended IP-formulation for an IP set $P_I \subseteq \mathbb{N}^n$

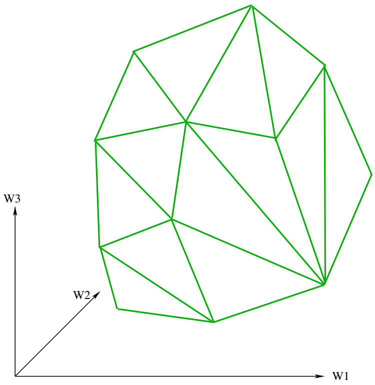
is an IP-set $Q_I = \{(x,w) \in \mathbb{R}^n \times \mathbb{N}^e : Gx + Hw \geq b\}$ s.t.

$$P_I = \text{proj}_x Q_I.$$



Change of variables: $x=T\,w$





Reformulation: a special case of extended formulation

An extended formulation based on a **change of variables**: $\mathbf{x} = \mathbf{T}\mathbf{w}$.

$$Q = \{(x, w) \in \mathbb{R}^{n+e} : \begin{array}{rcl} Tw & = & x \\ Hw & \geq & h \end{array}\}.$$

Then,

$$\text{proj}_x(Q) = T(W) := \{x = Tw \in \mathbb{R}^n : \underbrace{Hw \geq h, w \in \mathbb{R}^e}_{w \in W}\}.$$

A reformulation for an IP-set $P_I \subseteq \mathbb{N}^n$

is a polyhedron W along a linear transformation, $\mathbf{x} = \mathbf{T}\mathbf{w}$, s.t.

$$\boldsymbol{P_I = T(W) \cap \mathbb{N}^n}$$

A IP-reformulation for an IP-set $P_I \subseteq \mathbb{N}^n$

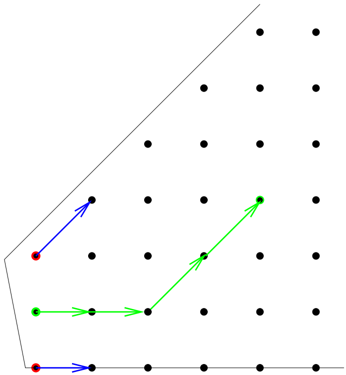
is an IP-set $W_I = W \cap \mathbb{N}^e$ along a linear transformation, $\mathbf{x} = \mathbf{T}\mathbf{w}$, s.t.,

$$\boldsymbol{P_I = T(W_I)}$$

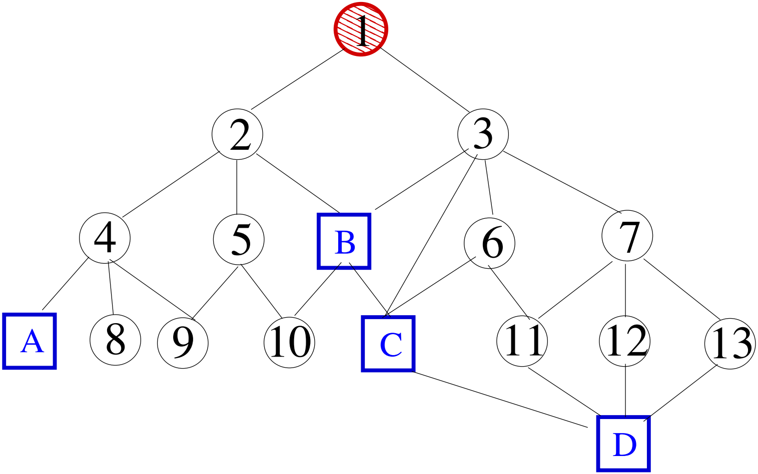
Minkowski's representation: a special case of reformulation

Polyhedron $\mathrm{conv}(P_I)$ can be defined by its extreme points and rays:

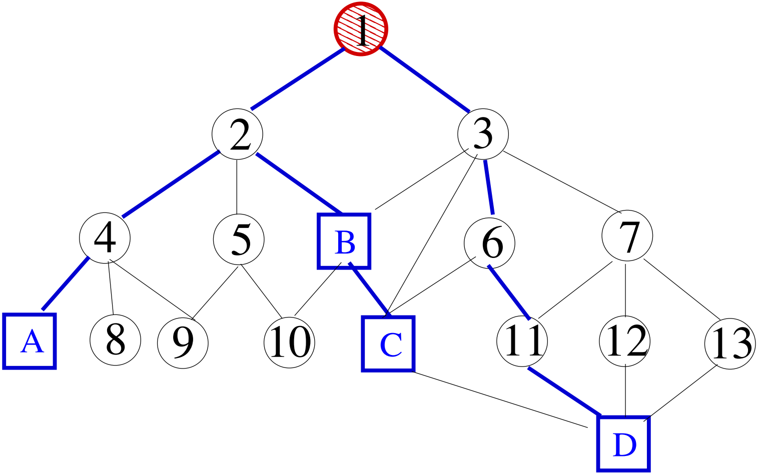
$$Q = \{(x,\lambda,\mu) \in \mathbb{R}^n \times \mathbb{R}_+^{|G|} \times \mathbb{R}_+^{|R|} : x = \sum_{g\in G} x^g \lambda_g + \sum_{r\in R} v^r \mu_r, \ \sum_{g\in G} \lambda_g = 1\}$$



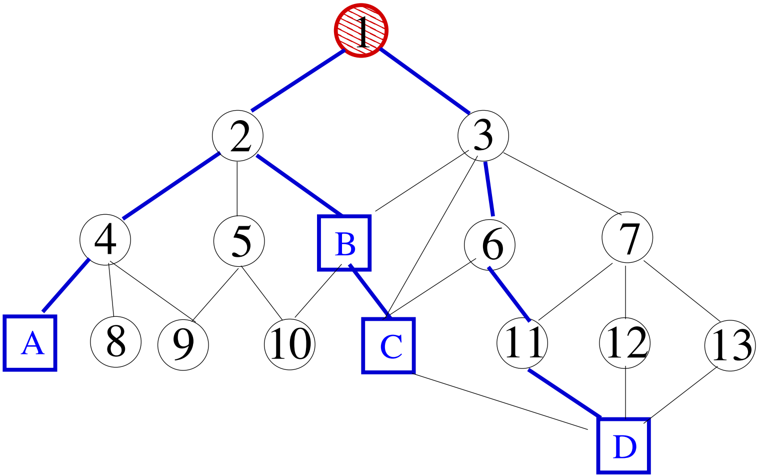
change of variables: $\mathbf{x} = \mathbf{X}\,\lambda + \mathbf{V}\,\mu$.


1
2
3
4
5
B
6
7
A
8
9
10
C
11
12
13
D

Example: Steiner Tree



Example: Steiner Tree



Special cases:

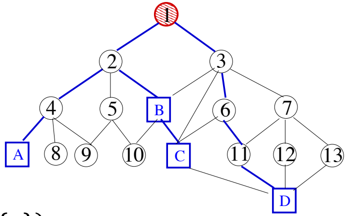
- $T = \{i\}$: shortest path from r to i
- $T = V \setminus \{r\}$: minimum cost spanning tree

Steiner Tree: Arc flow formulation

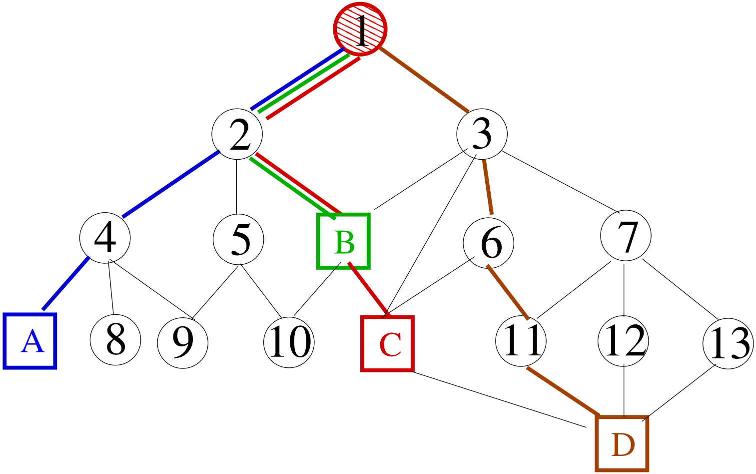
Variables

- $x_{ij} \in \{0,1\}$ — arc (i,j) is used or not
- $y_{ij} \in \mathbb{N}$ — number of connections going through (i,j)

$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij} x_{ij} & & \\
\sum_{j\in V^+(r)} y_{rj} &= |T| & \\
\sum_{j\in V^-(i)} y_{ji} - \sum_{j\in V^+(i)} y_{ij} &= 1 & i \in T \\
\sum_{j\in V^-(i)} y_{ji} - \sum_{j\in V^+(i)} y_{ij} &= 0 & i \in V \setminus (T \cup \{r\}) \\
y_{ij} &\leq |T|\, x_{ij} & (i,j) \in A \\
y &\in \mathbb{R}_+^{|A|}, & \\
x &\in \{0,1\}^{|A|} &
\end{aligned}
$$



Example: Steiner Tree

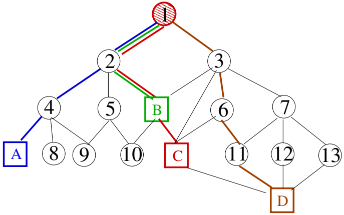


Steiner Tree: Multi commodity flow formulation

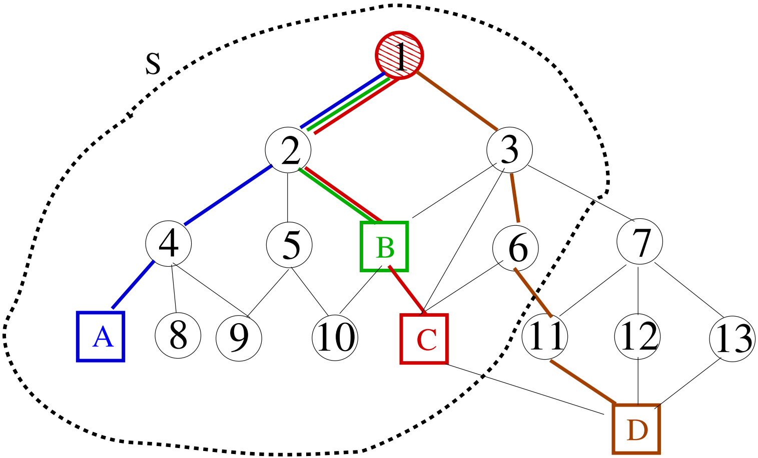
Variable splitting

- $w^t_{ij} \in \{0,1\}$ — arc (i,j) is used to connect terminal t
- $y_{ij} = \sum_k w^t_{ij}$ — defines a linear transformation

$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij} x_{ij} & & \\
\sum_{j\in V^+(r)} w^t_{rj} &= 1 \quad t \in T \\
\sum_{j\in V^-(i)} w^t_{ji} - \sum_{j\in V^+(i)} w^t_{ij} &= 1 \quad i = t \in T \\
\sum_{j\in V^-(i)} w^t_{ji} - \sum_{j\in V^+(i)} w^t_{ij} &= 0 \quad i \in V \setminus \{r,k\},\, t \in T \\
w^t_{ij} &\leq x_{ij} \quad (i,j) \in A,\, t \in T \\
w &\in \mathbb{R}_+^{|K|\times|A|}, \\
x &\in \{0,1\}^{|A|}
\end{aligned}
$$



Example: Steiner Tree



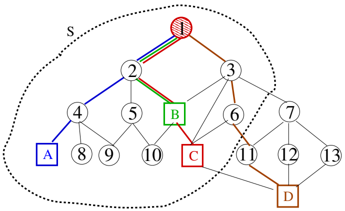
Steiner Tree: Network design formulation

projection in the x-space

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1 \quad S \ni r, T \setminus S \neq \emptyset$$

$$x \in \{0,1\}^{|A|},$$



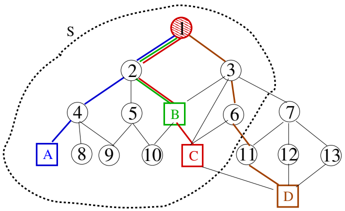
Steiner Tree: Network design formulation

projection in the x-space

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1 \quad S \ni r, T \setminus S \neq \emptyset$$

$$x \in \{0,1\}^{|A|},$$



Note: This projection onto the x space

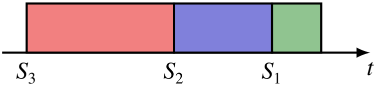
- has the **same LP value** than the multi-commodity flow formulation
- is **better than** the initial **compact** aggregate flow formulation.

Ways to obtain extended formulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion: $x = \sum_{q=0}^{u} q\, w_q,\ \sum_{q=0}^{u} w_q = 1, w \in \{0,1\}^{u+1}$
  - Binary expansion: $x = \sum_{p=0}^{\log\lfloor u \rfloor} w_p,\ , w \in \{0,1\}^{\log u}$
- Dynamic Programming Solver → Network Flow LP [Martin et al]
- Separation is easy → Separation LP [Martin et al]
- Reduced coefficent / basis reformulations [Aardal et al]
- Union of Polyhedra [Balas]
- ...

Unary expansion: Time-Indexed Formulation

Single machine scheduling problem (with integer data):

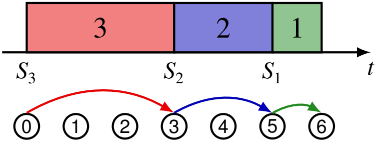


$$S_j \geq S_i + p_i \text{ or } S_i \geq S_j + p_j \ \forall\ i,j$$

requires big M formulation: $S_j \geq S_i + p_i - M(1 - x_{ij})$.

Unary expansion: Time-Indexed Formulation

Single machine scheduling problem (with integer data):



$$S_j \geq S_i + p_i \text{ or } S_i \geq S_j + p_j \ \forall\ i,j$$

Change of variables: $S_j = \sum_t t\, w_{jt}$

with $w_{jt} = 1$ iff job j starts at the beginning of $[t, t+1]$.

$$\begin{array}{rcl} \sum_{j \in J} w_{j0} & = & 1 \\ \sum_{j \in J} w_{jt} - \sum_{j \in J} w_{j,t-p_j} & = & 0 \quad \forall t \geq 1 \end{array}$$

Ways to obtain extended formulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion: $x = \sum_{q=0}^{u} q\, w_q,\ \sum_{q=0}^{u} w_q = 1, w \in \{0,1\}^{u+1}$
  - Binary expansion: $x = \sum_{p=0}^{\log\lfloor u \rfloor} w_p,\ , w \in \{0,1\}^{\log u}$
- Dynamic Programming Solver → Network Flow LP [Martin et al]
- Separation is easy → Separation LP [Martin et al]
- Reduced coefficent / basis reformulations [Aardal et al]
- Union of Polyhedra [Balas]
- ...

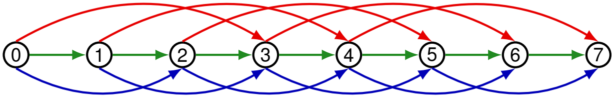
DP based reformulation: the knapsack example

$$\max\{\sum_i p_i x_i : \sum_i a_i x_i \le b, x_i \in \mathbb{N}\}$$

- **DP Recursion:** $V(c) = \max_{i=1,\ldots,n: c \ge a_i}\{V(c-a_i)+p_i\}$
- **in LP form:**

$$\begin{aligned} &\min V(b) \\ V(c) - V(c-a_i) &\ge p_i \qquad i = 1,\ldots,n,\ c = a_i,\cdots,b \\ V(0) &= 0 \end{aligned}$$

- **its Dual:** "longest path problem"



DP based reformulation: the knapsack example

$$\max\{\sum_i p_i x_i : \sum_i a_i x_i \leq b, x_i \in \mathbb{N}\}$$

- **DP Recursion:** $V(c) = \max_{i=1,\dots,n: c \geq a_i}\{V(c-a_i) + p_i\}$
- **in LP form:**

$$\begin{array}{rcll} \min V(b) & & & \\ V(c) - V(c-a_i) & \geq & p_i & i = 1,\dots,n,\ c = a_i,\cdots,b \\ V(0) & = & 0 & \end{array}$$

- **its Dual:** “longest path problem”

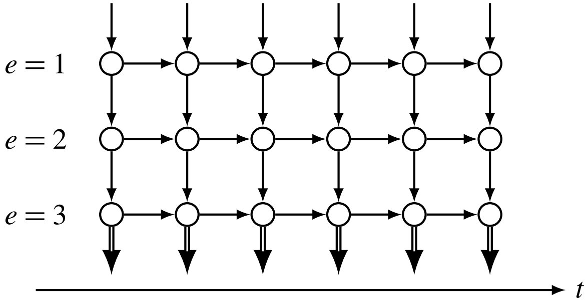
$$\begin{array}{rcll} \max \sum_{j=1}^{n} \sum_{r=0}^{b-a_i} c_i w_{ic} & & & \\ \sum_i w_{ic} & = & 1 & c = 0 \\ \sum_i w_{ic} - \sum_i w_{i,c-a_i} & = & 0 & c = 1,\cdots,b-1 \\ \sum_i w_{i,c-a_i} & = & 1 & c = b \\ w_{ic} & \geq & 0 & i = 1,\cdots,n;\ c = 0,\cdots,b-a_i \end{array}$$

DP based reformulation: Multi-Echelon Lot-Sizing

Variables

- $x_{e,t}$ — production of intermediate product of echelon e in period t
- $s_{e,t}$ — stock of echelon e product at the end of period t

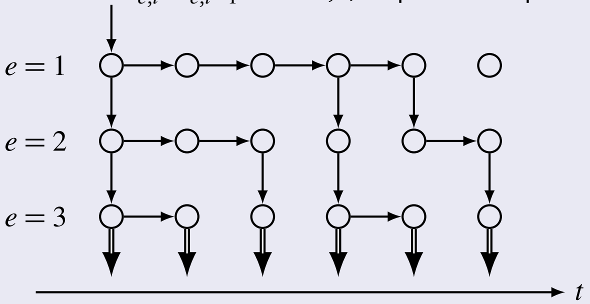
$$
\begin{array}{rcll}
x_{e,t} + s_{e,t-1} & = & x_{e+1,t} + s_{e,t} & \text{for } e = 1, \ldots, E-1 \\
x_{e,t} + s_{e,t-1} & = & d_t + s_{e,t} & \text{for } e = E
\end{array}
$$



DP based reformulation: Multi-Echelon Lot-Sizing

Dominance property

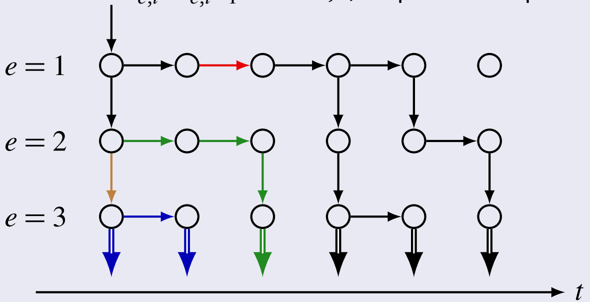
$\exists$ opt solution where $x_{e,t} \cdot s_{e,t-1} = 0 \; \forall e, t, \Rightarrow$ production plan is a tree:



DP based reformulation: Multi-Echelon Lot-Sizing

Dominance property

$\exists$ opt solution where $x_{e,t} \cdot s_{e,t-1} = 0 \;\forall e,t, \Rightarrow$ production plan is a tree:



Dynamic programming

State (e,t,a,b) corresponds to accumulating at echelon e in period t a production covering exactly the demand of periods $a,\ldots,b$.

$$
\begin{aligned}
V(e,t,a,b) \quad = \quad & \min\{V(e,t+1,a,b), \\
& \min_{l=a,\ldots,b}\{V(e+1,t,a,l) + c^k_{et}\, D^k_{al} + f^k_{et} + V(e,t+1,l+1,b)\}\}
\end{aligned}
$$

DP based reformulation: Multi-Echelon Lot-Sizing

- **DP Recursion:**

$$
\begin{aligned}
V(e,t,a,b) &= \min\{V(e,t+1,a,b), \\
&\quad \min_{l=a,\dots,b}\{V(e+1,t,a,l) + c^k_{et} D^k_{al} + f^k_{et} + V(e,t+1,l+1,b)\}\}
\end{aligned}
$$

- **in LP form:**

$$
\begin{aligned}
\max V(1,1,1,T) & \\
V(e,t,a,b) &\leq V(e,t+1,a,b) \ \forall e,t,a,b \\
V(e,t,a,b) &\leq V(e+1,t,a,l) + c^k_{et} D^k_{al} + f^k_{et} + V(e,t+1,l+1,b) \ \forall e,t,a,b,l \\
V(E+1,t,a,b) &= 0 \ \forall t,a,b
\end{aligned}
$$

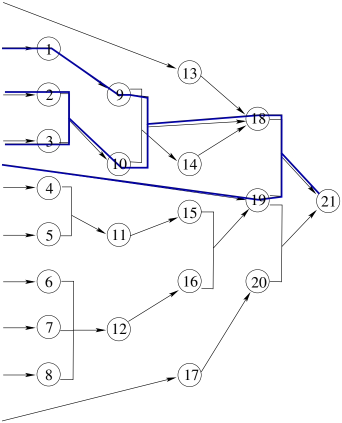
- **its Dual:** flow on hyper-arcs
  $w_{e,t,a,l,b} = 1$ if at echelon e in period t production covers demands from period a to period l, while the rest of demand up to b, shall be covered in the future.

DP based reformulations

[Martin et al OR90] When a problem can be solved by dynamic programming,

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

an extended formulation consist in modeling a decision tree in an hyper-graph



Ways to obtain extended formulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion: $x = \sum_{q=0}^{u} q\, w_q,\ \sum_{q=0}^{u} w_q = 1, w \in \{0,1\}^{u+1}$
  - Binary expansion: $x = \sum_{p=0}^{\log\lfloor u \rfloor} w_p,\ , w \in \{0,1\}^{\log u}$
- Dynamic Programming Solver → Network Flow LP [Martin et al]
- Separation is easy → Separation LP [Martin et al]
- Reduced coefficent / basis reformulations [Aardal et al]
- Union of Polyhedra [Balas]
- ...

Outline

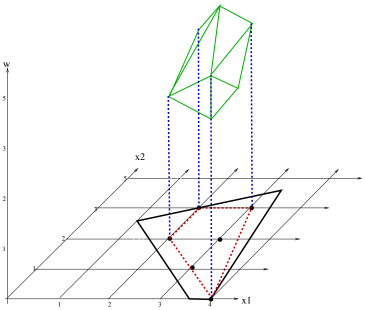
1. Extented Formulations
   - Definitions
   - Interests
   - Coping with its large size
2. Dynamic Row-and-Column Generation
   - Methodology
   - Practical issues
3. Large-scale application
   - Freight transport by rail in Russia

1. **Improved formulation** (better LP bound & rounding heuristic)

extra variables

↓

tighter relations, linearisation



Extended formulation: Interests

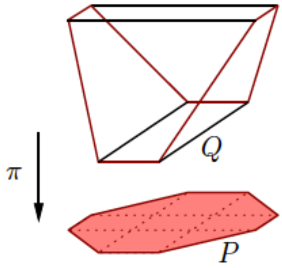
1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)

extra variables

↓

fewer constaints

structure built into var. definitions



Extended formulation: Interests

1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)
3. **Direct use of a MIP-Solver** (solved by standard tools)

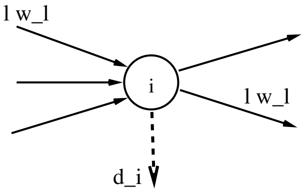
Extended formulation: Interests

1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)
3. **Direct use of a MIP-Solver** (solved by standard tools)
4. **Rich variable space** (to express cuts or branching)

Vehicle routing: $x_a = \sum_{l=0,\dots,C} w_l^a$
$w_q^a = 1$ if vehicle on arc a with load l,

$$\sum_l \sum_{a \in \delta^-(i)} l w_l^a - \sum_l \sum_{a \in \delta^+(i)} l w_l^a \quad = \quad d_i$$

$\longrightarrow$ knapsack cover cuts.



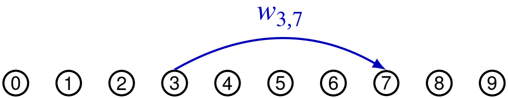
[Uchoa]

Outline

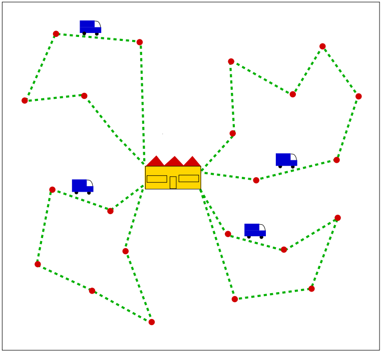
[Macedo, Alves, Valerio de Carvalho, Clautiaux, Hanafi. EJOR2011]

Variables

- w^r_{st} — nb of vehicles using route r that starts in s and ends in t



$$
\begin{aligned}
\min \sum_{rst} c^r_{st}\, w^r_{st} & & & \\
\sum_{r \ni i, s, t} w^r_{st} &= 1 & & \forall \text{order } i \\
\sum_{rt} w^r_{0t} &= V & & \\
\sum_{r,t} w^r_{\tau t} - \sum_{r,s} w^r_{s\tau} &= 0 & & \forall \tau > 1 \\
w^r_{st} &\in \{0,1\} & & r, s, t
\end{aligned}
$$

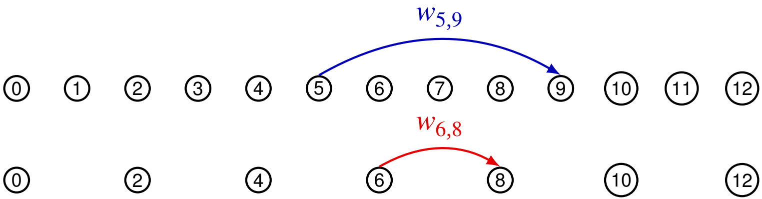


Coping with size: Related work on Multi-Route-VRP

[Macedo, Alves, Valerio de Carvalho, Clautiaux, Hanafi. EJOR2011]

Relaxation

- round-up start time: $S = \{s : \lceil s \rceil = S\}$
- round-down termination time: $T = \{t : \lfloor t \rfloor = T\}$
- define relaxed route arcs : $w^r_{S,T} = \sum_{s \in S, t \in T} w^r_{s,r}$.



Automatic Desaggragation Algorithm:

1. Solve problem over **aggregate time** periods.
2. Try to build a **desaggregate feasible solution.**
3. If it fails, **desaggregate the time period** of conflict.

Mastering the size extended formulations

1. Use of a relaxation [Van Vyve & Wolsey MP06]
    - Drop some of the constraints
    - Aggregate commodities/nodes (down-rounding of durations)
    - Partial reformulation

   $\longrightarrow$ static outer approximation of the extended formulation

Mastering the size extended formulations

1. Use of a relaxation [Van Vyve & Wolsey MP06]
   - Drop some of the constraints
   - Aggregate commodities/nodes (down-rounding of durations)
   - Partial reformulation

   → static outer approximation of the extended formulation

2. Use of a restriction
   - define only some transitions in a dynamic program
   - up-rounding of durations

   → static inner approximation of the extended formulation

Mastering the size extended formulations

1. Use of a relaxation [Van Vyve & Wolsey MP06]
   - Drop some of the constraints
   - Aggregate commodities/nodes (down-rounding of durations)
   - Partial reformulation

   → static outer approximation of the extended formulation

2. Use of a restriction
   - define only some transitions in a dynamic program
   - up-rounding of durations

   → static inner approximation of the extended formulation

3. Projection: Benders' cuts (applying Farkas Lemma)
   → dynamic outer approximation of the extended formulation

Mastering the size extended formulations

1. Use of a relaxation [Van Vyve & Wolsey MP06]
   - Drop some of the constraints
   - Aggregate commodities/nodes (down-rounding of durations)
   - Partial reformulation

   ⟶ static outer approximation of the extended formulation

2. Use of a restriction
   - define only some transitions in a dynamic program
   - up-rounding of durations

   ⟶ static inner approximation of the extended formulation

3. Projection: Benders' cuts (applying Farkas Lemma)
   ⟶ dynamic outer approximation of the extended formulation

4. Dynamic generation: delayed column-and-row generation.
   ⟶ dynamic inner approximation of the extended formulation

Outline

Extended formulation based on a subproblem

Original formulation

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

Subproblem

$$P \equiv \left\{ \begin{array}{rcl} B\,x & \geq & b \\ x & \in & {\mathbb{R}_+}^n \end{array} \right\}$$

$$P_I = P \cap \mathbb{N}^n$$

Decomposition + SP Reformulation

Extended formulation based on a subproblem

Original formulation

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

Subproblem

$$\mathsf{P} \equiv \left\{ \begin{array}{rcl} B\,x & \geq & b \\ x & \in & \mathbb{R}_+{}^n \end{array} \right\}$$

$$P_I = \mathsf{P} \cap \mathbb{N}^n$$

Assumption

Subproblem P_I admits an **IP-reformulation** W_I: $\exists$ polyhedron

$$W = \left\{ Hw \geq h, w \in \mathbb{R}_+{}^e \right\}$$

and **a linear transformation** T, such that

$$P_I = \mathrm{proj}_x(W_I) = T(W_I) = \left\{ x = Tw : Hw \geq h, w \in \mathbb{N}^e \right\}$$

Extended formulation based on a subproblem

Original formulation

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

Extended reformulation

$$
[R] \equiv \min \left\{ c\,T\,w \quad \begin{array}{rcl} A\,T\,w & \geq & a \\ H\,w & \geq & h \\ w & \in & \mathbb{N}^e \end{array} \right\}
$$

Assumption

Subproblem P_I admits an **IP-reformulation**, W_I: $\exists$ polyhedron

$$
W = \{Hw \geq h, w \in \mathbb{R}_+{}^e\}
$$

and **a linear transformation,** T, s.t.

$$
P_I = \text{proj}_x(W_I) = T(W_I) = \left\{ x = Tw : Hw \geq h, w \in \mathbb{N}^e \right\}
$$

Extended formulation based on a subproblem

Original formulation

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

Extended reformulation

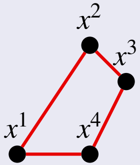
$$[R] \equiv \min \left\{ c\,T\,w \quad \begin{array}{rcl} A\,T\,w & \geq & a \\ H\,w & \geq & h \\ w & \in & \mathbb{N}^e \end{array} \right\}$$

Special case: Dantzig-Wolfe Reformulation

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

Applying Minkowski

$x = \sum_{g \in G} x^g\,\lambda_g$



Extended formulation based on a subproblem

Original formulation

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

Extended reformulation

$$[R] \equiv \min\left\{ c\,T\,w \quad \begin{array}{rcl} A\,T\,w & \geq & a \\ H\,w & \geq & h \\ w & \in & \mathbb{N}^e \end{array} \right\}$$

Column-and-row generation

- Dynamic generation of the variables of [R] by bunch, solving the column generation subproblem of [M] over W_I.
- Adding rows that become active.

$\overline{S} = \{w^s\}_{s \in \overline{S}}$: a subset of integer solutions to W_I.
$\overline{w}$ = restriction of w to the non-zero components in $\overline{S}$.
$\overline{G} = \{g \in G : x^g = T\, w^s, s \in \overline{S}\}$

$$[\overline{R}_{LP}] \equiv \min \left\{ c\, \overline{T}\, \overline{w} \right. \quad \begin{array}{rcl} A\, \overline{T}\, \overline{w} & \geq & a \\ \overline{H}\, \overline{w} & \geq & \overline{h} \\ \overline{w} & \in & \left. \mathbb{R}_+^{\overline{e}} \right\} \end{array}$$

$$[\overline{M}_{LP}] \equiv \min \left\{ \sum_{g \in \overline{G}} c\, x^g\, \lambda_g \right. \quad \begin{array}{rcl} \sum_{g \in \overline{G}} A\, x^g\, \lambda_g & \geq & a \\ \sum_{g \in \overline{G}} \lambda_g & = & 1 \\ \lambda & \in & \left. \mathbb{R}_+^{|\overline{G}|} \right\} \end{array}$$

Restricted reformulations

$\overline{S} = \{w^s\}_{s \in \overline{S}}$: a subset of integer solutions to W_I.
$\overline{w}$ = restriction of w to the non-zero components in $\overline{S}$.
$\overline{G} = \{g \in G : x^g = T\, w^s, s \in \overline{S}\}$

$$[\overline{R}_{LP}] \equiv \min \left\{ c\,\overline{T}\,\overline{w} \quad \begin{array}{rcl} A\,\overline{T}\,\overline{w} & \geq & a \\ \overline{H}\,\overline{w} & \geq & \overline{h} \\ \overline{w} & \in & \mathbb{R}^{\overline{e}}_{+} \end{array} \right\}$$

$$[\overline{M}_{LP}] \equiv \min \left\{ \sum_{g \in \overline{G}} c\, x^g\, \lambda_g \quad \begin{array}{rcl} \sum_{g \in \overline{G}} A\, x^g\, \lambda_g & \geq & a \\ \sum_{g \in \overline{G}} \lambda_g & = & 1 \\ \lambda & \in & \mathbb{R}^{|\overline{G}|}_{+} \end{array} \right\}$$

Proposition 1

$$v^{[M_{LP}]} =_* v^{[R_{LP}]} \leq v^{[\overline{R}_{LP}]} \leq v^{[\overline{M}_{LP}]}$$ $(*)$ if tight reformulation

Column-and-row generation procedure

Step 1: Solve $[\overline{R}_{LP}]$ and collect the **dual solution** $\overline{\pi}$ associated to constraints $A\,\overline{T}\,\overline{z} \geq a$, only.

Step 2: Obtain a solution w^* of the pricing problem:

$$\min\{(c - \overline{\pi}A)\,T\,w : \; w \in W_I\}$$

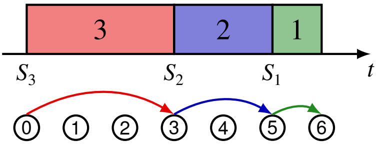
Step 3: Compute the Lagrangian dual bound:
$L(\overline{\pi}) \leftarrow \overline{\pi}\,a + (c - \overline{\pi}A)\,T\,w^*$, $\beta \leftarrow \max\{\beta, L(\overline{\pi})\}$.
If $v^{[\overline{R}_{LP}]} \leq \beta$, STOP.

Step 4: Update $\overline{S}$ by adding solution w^* and iterate

Proposition 2

Either $v^{\overline{R}}_{LP} \leq \beta$ (stopping condition),
or some of the components of w^* have negative reduced cost in $[\overline{R}_{LP}]$.

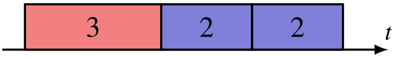
Example: parallel machine scheduling



$$
\begin{aligned}
[R] \equiv \min \Big\{ \sum_{jt} c_{jt}\, w_{jt} & & & \\
\sum_{t=0}^{T-p_j} w_{jt} &= 1 & & \forall j \in J \\
\sum_{j \in J} w_{j0} &= m & & \\
\sum_{j \in J} w_{jt} - \sum_{j \in J} w_{j,t-p_j} &= 0 & & \forall t \geq 1 \\
w_{jt} &\in \{0,1\} & & \forall j, t \Big\}
\end{aligned}
$$

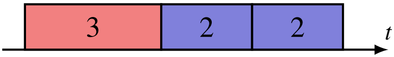
$$
\begin{aligned}
[M] \equiv \min \Big\{ \sum_{g \in G} c^g \, \lambda_g & & \\
\sum_{g \in G} \sum_{t=0}^{T-p_j} w_{jt}^g \, \lambda_g &= 1 & \forall j \in J \\
\sum_{g \in G} \lambda_g &= m & \\
\lambda_g &\in \{0,1\} \ \forall g \in G \Big\} &
\end{aligned}
$$

1. Solve the pricing subproblem (obtain a pseudo schedule)

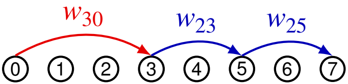


Machine scheduling: column-and-row generation

1. Solve the pricing subproblem (obtain a pseudo schedule)

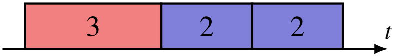


2. Disaggregate the subproblem solution in arc variables w.

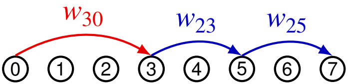


Machine scheduling: column-and-row generation

1. Solve the pricing subproblem (obtain a pseudo schedule)



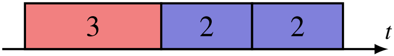
2. Disaggregate the subproblem solution in arc variables w.



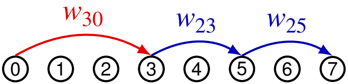
3. Add them to $[\overline{R}]$ along with the associated flow conservation constraints.

Machine scheduling: column-and-row generation

1. Solve the pricing subproblem (obtain a pseudo schedule)



2. Disaggregate the subproblem solution in arc variables w.



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

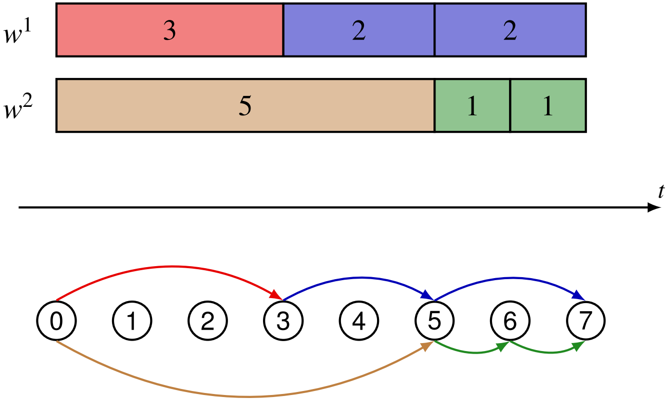
4. Solve the restricted extended formulation $[\overline{R}]$ and update dual prices.

Machine scheduling: example of convergence

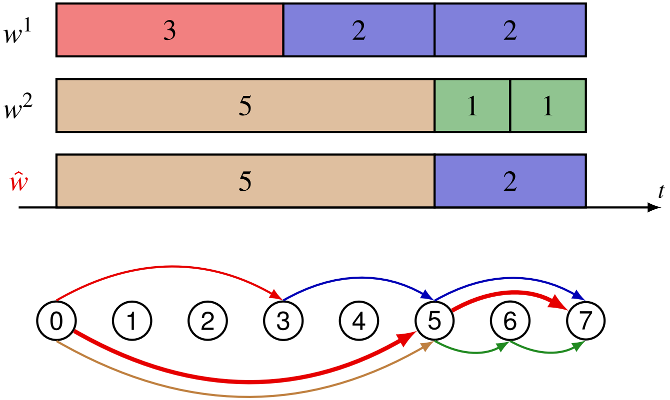
	Column generation for [M]	Column-and-row generation for [R]
Initial solution		
Iteration	Subproblem solution	Subproblem solution
1		
2		
3		
. . .	. . .	
10		
11		
Final solution		

Machine scheduling: recombination property

$$\overline{S} = \{w^1, w^2\}$$



$$\overline{S} = \{w^1, w^2\}, \qquad \hat{w} \in \overline{W} \setminus \mathrm{conv}(w^1, w^2)$$



Machine Scheduling: numerical results

- Averages on 25 instances (OR-library) with $p_j \in [1, \ldots, 100]$.

		Cplex 12.1 for [R]	Colomn gen. for [M]		Column-and-row generation for [R]		
m	n	*cpu*	*#it*	*cpu*	*#it*	*vars*	*cpu*
1	25	7.1	337	0.9	124	3.8%	0.8
1	50	132.6	1274	24.2	246	2.7%	8.6
1	100	2332.0	8907	1764.4	455	1.9%	61.3
2	25	4.1	207	0.3	97	3.9%	0.2
2	50	109.2	645	5.7	173	2.8%	1.9
2	100	**3564.4**	2678	**115.5**	319	2.1%	**14.9**
4	50	18.7	433	1.5	167	3.0%	0.7
4	100	485.7	1347	27.9	295	2.2%	5.2
4	200	$>$**2h**	4315	**409.4**	561	1.5%	**39.4**

#it number of column generation iterations
vars percentage of w variables that are generated
cpu solution time, in seconds

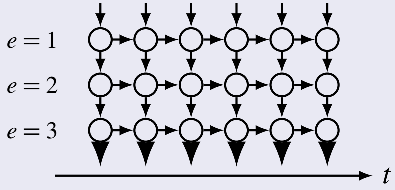
Machine Scheduling: results with stabilization

		Colomn gen. for [M]		Column-and-row generation for [R]		
m	n	*#it*	*cpu*	*#it*	*vars*	*cpu*
1	25	150	0.2	96	2.6%	0.4
1	50	354	3.8	172	1.7%	4.0
1	100	781	39.5	299	1.3%	31.1
2	25	142	0.2	87	3.3%	0.2
2	50	323	1.7	158	2.2%	1.6
2	100	715	**17.3**	275	1.6%	**11.3**
4	50	287	0.6	154	2.6%	0.6
4	100	638	8.7	264	1.8%	4.6
4	200	1553	**87.7**	481	1.2%	**33.4**

- x^i_{et}, s^i_{et} — production/stock for item i at echelon e in period t
- $y^i_{et} \in \{0,1\}$ — setup for item i at echelon e in period t

coupling constraints: $\sum_i y^i_{et} \leq 1 \quad \forall e,t$

Subproblems

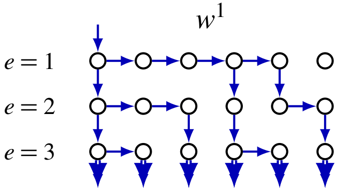


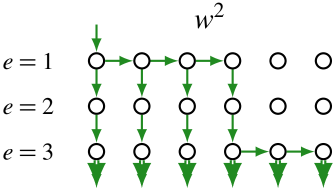
DP based **extended formulation** as a **flow in a hypergraph**:

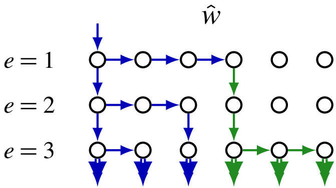
- $w^i_{e,t,a,l,b} = 1$ if at echelon e in period t production covers demands for item i from period a to period l, while the rest of demand up to b, shall be covered in the future.

Multi-echelon lot-sizing: recombination property

$$\overline{S} = \{w^1, w^2\}, \qquad \hat{w} \in \overline{W} \setminus conv(w^1, w^2)$$







Multi-echelon lot sizing: results with stabilization

Averages for 10 instances are given

			Colomn gen. for [M]		Column-and-row generation for [R]		
E	K	T	*#it*	*cpu*	*#it*	*vars*	*cpu*
2	10	50	126	1.7	29	0.57%	1.6
2	20	50	79	1.8	27	0.44%	3.1
2	10	100	332	38.0	43	0.15%	8.1
2	20	100	232	31.5	38	0.14%	20.0
3	10	50	187	11.8	38	0.16%	5.5
3	20	50	112	12.0	33	0.12%	9.8
3	10	100	509	454.5	49	0.02%	36.4
3	20	100	362	**520.4**	48	0.02%	**103.1**
5	10	50	296	62.6	48	0.10%	16.3
5	20	50	223	66.8	42	0.07%	34.3
5	10	100	882	4855.9	61	0.01%	134.0
5	20	100	362	**4657.8**	56	0.01%	**386.1**

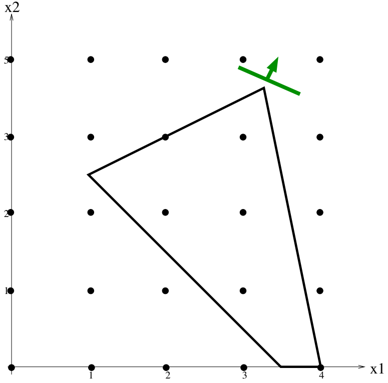
Outline

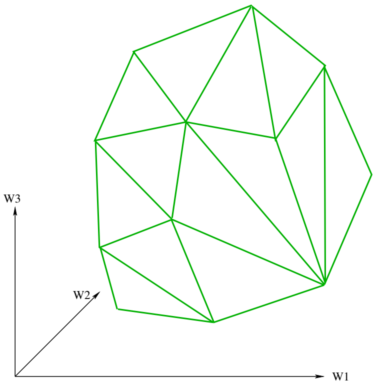
Coping with the size of the Subproblem

1. Solve the **compact formulation**

   Step 2: Obtain a solution x^* of the pricing problem:

$$\min\{(c - \overline{\pi}A)\,x : \; x \in P_I\}.$$



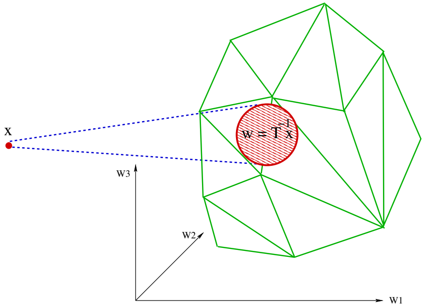


Coping with the size of the Subproblem

1. Solve the **compact formulation**

   Step 2: Obtain a solution x^* of the pricing problem:

$$\min\{(c - \overline{\pi}A)\,x : \; x \in P_I\}.$$



1. Solve the **compact formulation**

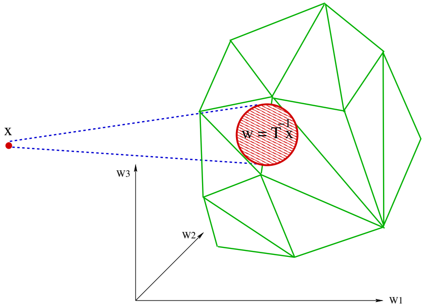
   Step 2: Obtain a solution x^* of the pricing problem:

$$\min\{(c - \overline{\pi}A)\,x : \; x \in P_I\}.$$

Lifting set:

$$T^{-1}(x) := \{w \in \mathbb{N}^e : T\,w = x;\; H\,w \geq h\}$$

Solving a “preprocessed” feasibility MIP



1. Solve the **compact formulation**

   Step 2: Obtain a solution x^* of the pricing problem:

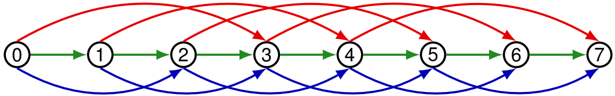
$$\min\{(c - \overline{\pi}A)\,x : \; x \in P_I\}.$$

Lifting set:

$$T^{-1}(x) := \{w \in \mathbb{N}^e : T\,w = x;\; H\,w \geq h\}$$

Solving a "preprocessed" feasibility MIP

Example of the Knapsack Problem:



1. Solve the **compact formulation**

   Step 2: Obtain a solution x^* of the pricing problem:

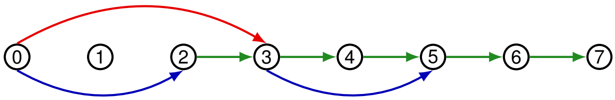
$$\min\{(c - \overline{\pi}A)\,x : \; x \in P_I\}.$$

Lifting set:

$$T^{-1}(x) := \{w \in \mathbb{N}^e : T\,w = x;\; H\,w \geq h\}$$

Solving a "preprocessed" feasibility MIP

Example of the Knapsack Problem:



1. Solve the **compact formulation** (while no master constr. on w)

   Step 2: Obtain a solution x^* of the pricing problem:

$$\min\{(c - \overline{\pi} A)\, x : \; x \in P_I\}.$$

Lifting set:

$$T^{-1}(x) := \{w \in \mathbb{N}^e : T\, w = x; \; H\, w \geq h\}$$

Solving a "preprocessed" feasibility MIP

Lifting operator:

$$\boxed{x^* \rightarrow w^* \in T^{-1}(x^*)}$$

Breaking symmetries

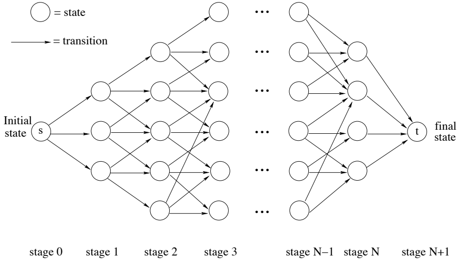
Coping with the size of the Subproblem

1. Solve the **compact formulation** (while no master constr. on w)

$$\min\{(c - \overline{\pi}A)\,x : \; x \in X\}.$$

$$\boxed{x^* \longrightarrow w^* \in T^{-1}(x^*)}$$

2. Use a **forward labelling** Dynamic Program



Handling the underlying graph implicitly

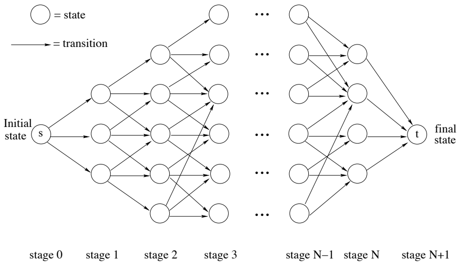
Coping with the size of the Subproblem

1. Solve the **compact formulation** (while no master constr. on w)

$$\min\{(c - \overline{\pi}A)\,x : \; x \in X\}.$$

$$\boxed{x^* \longrightarrow w^* \in T^{-1}(x^*)}$$

2. Use a **forward labelling** Dynamic Program

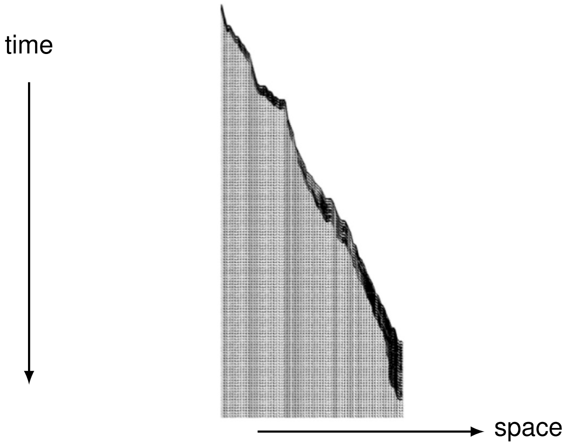


Handling the underlying graph implicitly

3. Use **successive approximations**: restrictions or relaxations

Coping with the Subproblem: Related work

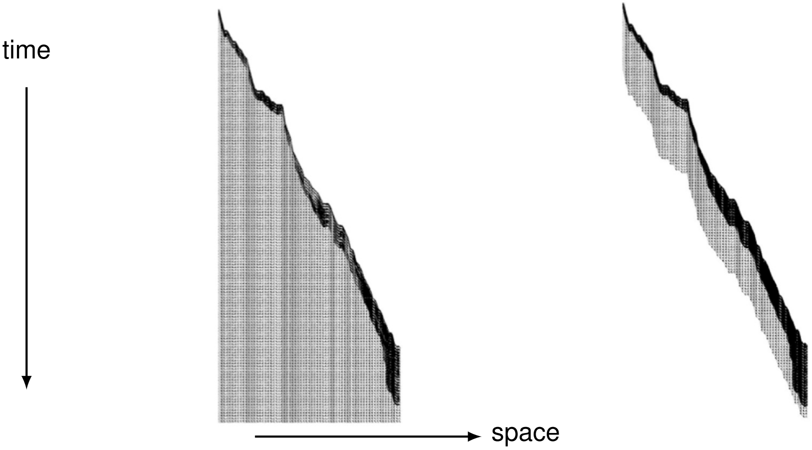
[F. Fischer, C. Helmberg, MP2012
Dynamic Graph Generation for Shortest Path in Time Expanded Networks]



Train Timetabling

Coping with the Subproblem: Related work

[F. Fischer, C. Helmberg, MP2012
Dynamic Graph Generation for Shortest Path in Time Expanded Networks]



Train Timetabling

Coping with the Subproblem: Related work

[F. Fischer, C. Helmberg, MP2012
Dynamic Graph Generation for Shortest Path in Time Expanded Networks]

Assumption

Capacity as only linking constraints ⇒ reduced cost $= \bar{c}_a \geq c_a \forall a \in A$.

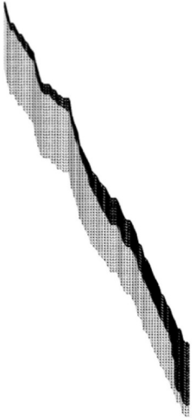
Proposition

Given a **restricted graph** $\overline{G} \subset G$ and **its augmentation** G^+:

$$G^+ = \overline{G} \cup \delta(\overline{G}) \cup \mathrm{SP}(\delta(\overline{G}))$$

Let $\hat{c}_a = \bar{c}_a$ for $a \in \overline{G}$, and c_a otherwise.
Let $\boldsymbol{P^* = \mathrm{argmin}\{\hat{c}(P_{st}) : P_{st} \in G^+\}}$.

If $P^* \in \overline{G}$, then $P^* = \mathrm{argmin}\{\bar{c}(P_{st}) : P_{st} \in G\}$.
Otherwise, $\overline{G} \leftarrow \overline{G} \cup P^*$.



Practical issues

1. Coping with the size of the Subproblem

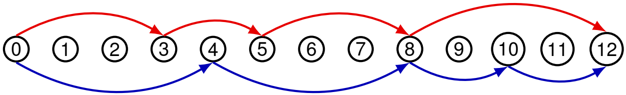
Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master

    $\longrightarrow$ Preprocessing
    $\longrightarrow$ Master cleanup
    $\longrightarrow$ Disaggregate only if it yields recombinations

Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master
   - → Preprocessing
   - → Master cleanup
   - → Disaggregate only if it yields recombinations

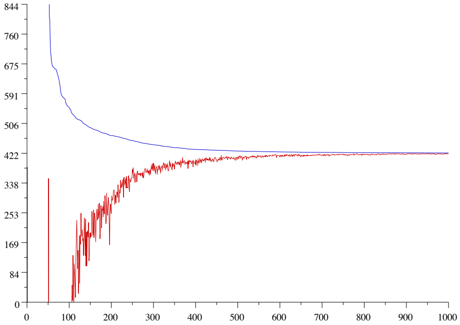


Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master
   - → Preprocessing
   - → Master cleanup
   - → Disaggregate only if it yields recombinations
3. Acceleration of column generation convergence

Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master
   - → Preprocessing
   - → Master cleanup
   - → Disaggregate only if it yields recombinations
3. Acceleration of column generation convergence
   - Stabilization techniques



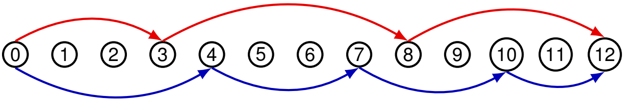
- **Dual oscillations**
- **Tailing-off effect**
- **Primal degeneracy**

Practical issues

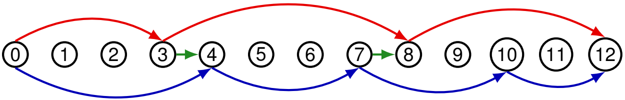
1. Coping with the size of the Subproblem
2. Coping with the size of the Master
   - → Preprocessing
   - → Master cleanup
   - → Disaggregate only if it yields recombinations
3. Acceleration of column generation convergence
   - Stabilization techniques
     - → Penalty functions & Smoothing

Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master
   - → Preprocessing
   - → Master cleanup
   - → Disaggregate only if it yields recombinations
3. Acceleration of column generation convergence
   - Stabilization techniques
     - → Penalty functions & Smoothing
     - → Disaggregations/Recombinations



1. Coping with the size of the Subproblem
2. Coping with the size of the Master
   - → Preprocessing
   - → Master cleanup
   - → Disaggregate only if it yields recombinations
3. Acceleration of column generation convergence
   - Stabilization techniques
     - → Penalty functions & Smoothing
     - → Disaggregations/Recombinations (add waiting arcs)



Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master

   → Preprocessing
   → Master cleanup
   → Disaggregate only if it yields recombinations

3. Acceleration of column generation convergence
   - Stabilization techniques
     → Penalty functions & Smoothing
     → Disaggregations/Recombinations (add waiting arcs)
   - Strategies for column generation
     → Build a global solution to the master at each iteration

Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master

   → Preprocessing
   → Master cleanup
   → Disaggregate only if it yields recombinations

3. Acceleration of column generation convergence
   - Stabilization techniques

     → Penalty functions & Smoothing
     → Disaggregations/Recombinations (add waiting arcs)
   - Strategies for column generation

     → Build a global solution to the master at each iteration
     → Stage-by-stage approach: decreasing restriction/relaxation level

Practical issues

1. Coping with the size of the Subproblem
2. Coping with the size of the Master

   → Preprocessing
   → Master cleanup
   → Disaggregate only if it yields recombinations

3. Acceleration of column generation convergence
   - Stabilization techniques

     → Penalty functions & Smoothing
     → Disaggregations/Recombinations (add waiting arcs)

   - Strategies for column generation

     → Build a global solution to the master at each iteration
     → Stage-by-stage approach: decreasing restriction/relaxation level

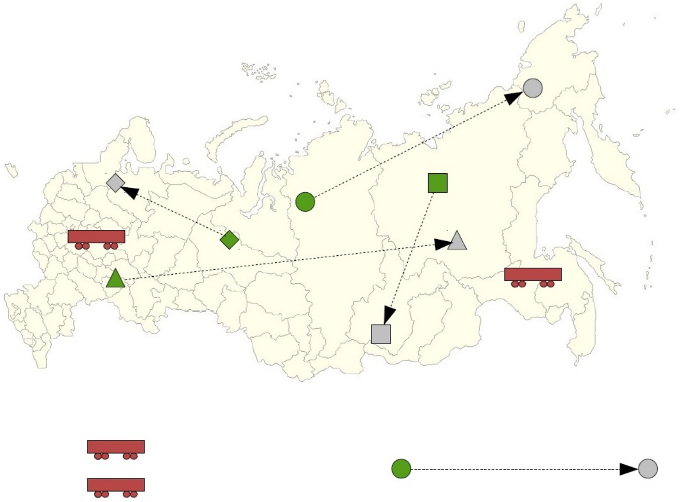
4. Combination with cut generation

   → Lifting added variables

Outline

The freight car routing problem

[R.Sadykov et al, 2013]

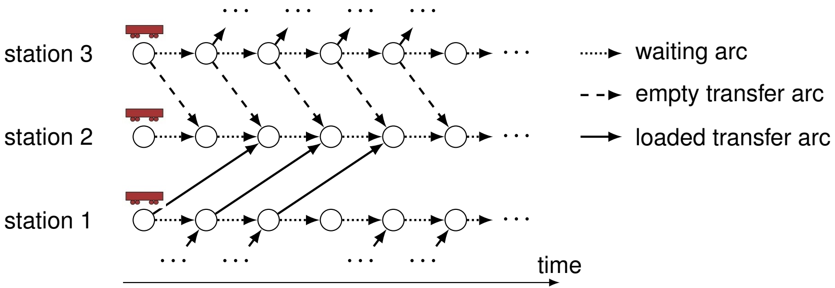


initial car distribution

transportation demand

Time-Expanded-Network

Each **type of railcar** defines a commodity $c \in C$



Multi-commodity flow formulation

Variables

- $x_a \in \mathbb{N}$ — nb of cars using arc $a \in A_c$, $c \in C$
- $y_d \in \{0,1\}$ — demand d is accepted or not

$$
\begin{aligned}
\max \sum_{c \in C} \sum_{a \in A_c} p_a \, x_a & \\
\sum_{c \in C_q} \sum_{a \in A_{cd}} x_a \geq n_d^{\min} \, y_d & \quad \forall d \\
\sum_{c \in C_q} \sum_{a \in A_{cd}} x_a \leq n_d^{\max} \, y_d & \quad \forall d \\
\sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a = b_v & \quad \forall c \in C, v \in V_c \\
x_a \in \mathbb{N} & \quad \forall c \in C, a \in V_c \\
y_d \in \{0,1\} & \quad \forall d
\end{aligned}
$$

LP-Solution approaches

- **Direct**: solving a multi-commodity flow problem using *Clp*
  (specifically modified)
- **Standard Column Generation**: a column is

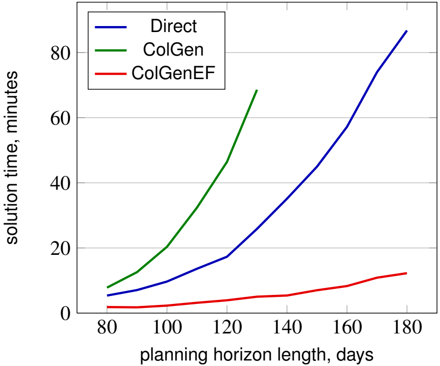
  Option A: A full planning for a type of car
  (decomposition per commodity)

  Option B: A in-tree into a sink
  (decomposition per sink)

  Option C: A path for origin to destination
  (decomposition per pair o-d)

- **Column Generation for Extended Formulation**: using option A.

1'025 stations, up to 6'800 demands, 11 car types, 12'651 cars, and 8'232 sources → 300 thousands nodes and 10 millions arcs.



Horizon	Direct	ColGenEF
80	5m24s	1m52s
90	7m05s	1m47s
100	9m42s	2m19s
110	13m38s	3m11s
120	17m19s	3m57s
130	25m52s	5m03s
140	35m08s	5m25s
150	44m58s	7m02s
160	57m11s	8m19s
170	1h13m58s	10m53s
180	1h26m46s	12m16s

$\leq$ 15 iterations, about 3% of the arc variables have been generated

Take away messages

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by combining ideas of
  - Restriction / Relaxation
  - Benders projection
  - Dantzig-Wolfe dynamic generation.
- With dynamic row-and-column generation, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for efficiency enhancement features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Take away messages

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by combining ideas of
  - Restriction / Relaxation
  - Benders projection
  - Dantzig-Wolfe dynamic generation.
- With dynamic row-and-column generation, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for efficiency enhancement features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Perspectives

Take away messages

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by **combining generic** ideas of
  - Restriction / Relaxation **[Soumis et al]**
  - Benders projection **[Van Vyve & Wolsey, EJCO, 2013]**
  - Dantzig-Wolfe dynamic generation.
- With dynamic row-and-column generation, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for efficiency enhancement features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Perspectives

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by combining ideas of
  - Restriction / Relaxation
  - Benders projection
  - Dantzig-Wolfe dynamic generation.
- With **dynamic row-and-column generation** **[M. Goycoolea et al 2013]**, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for efficiency enhancement features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Perspectives

Take away messages

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by combining ideas of
  - Restriction / Relaxation
  - Benders projection
  - Dantzig-Wolfe dynamic generation.
- With dynamic row-and-column generation, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for **efficiency enhancement** features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Perspectives

An approach based on an extended formulation

- An EASY WAY to bring-in combinatorial structure.
- Its size can be coped with by combining ideas of
  - Restriction / Relaxation
  - Benders projection
  - Dantzig-Wolfe dynamic generation.
- With dynamic row-and-column generation, a small % of variables and constraints are needed; hence it scales up to real-life applications.
- Is well suited for efficiency enhancement features: cuts on lifted variables, Dynamic Progr. state-space-relax., red.-cost-fixing.

Perspectives