Approche décomposition de Dantzig-Wolfe en programmation entière: apport de l’optimisation convexe

Université Bordeaux 1
Olivier Briant
Philippe Meurdesoif
Krystel Monneris
Sophie Michel
Nancy Perrot
François Vanderbeck

INRIA - Rhones-Aples
Claude Lemaréchal

Université de Genève
Claude Tadonki
Jean-Philippe Vial
Cesar Beltran

Funded by Inria New Investigation Grant
“Convex Optimization and Dantzig-Wolfe Decomposition”
http://www.math.u-bordeaux.fr/MAB/ODW/
PART 1: Background

- Dantzig-Wolfe Decomposition
- *BaPCod*: our projet of building a generic code
Decomposition: Why

- Divide and Conquer
Decomposition: Why

- Divide and Conquer

- Exploit the structure
Decomposition: When

\[ \min c x \]
\[ A x \geq b \]
\[ x \in X \subset \mathbb{N} \]
Decomposition: When

\[ \min \ c \ x \]

\[ A \ x \ \geq \ b \]

\[ x \ \in \ X \subset \mathbb{N} \]

Difficult Constraints
Decomposition: When

\[
\min c x \\
A x \geq b \\
x \in X \subset \mathbb{N}
\]

Difficult Constraints **Ex:** The Travelling Salesman Prob.

\[
\sum_{e \in \delta(i)} x_e = 2 \ \forall i
\]
Decomposition: When

\[ \min_c c \mathbf{x} \]

\[ A \mathbf{x} \geq b \]

\[ \mathbf{x} \in X \subset \mathbb{N} \]

Difficult Constraints

Linking Constraints

\[
\begin{pmatrix}
A_1 & A_2 & \ldots & A_n \\
B & 0 & \ldots & 0 \\
0 & B & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 0 & \ldots & B
\end{pmatrix}
\]
Decomposition: When

\[ \min \frac{c}{2} x + \frac{c}{2} y \]

\[ x = y \]

\[ x \in X \]

\[ y \in Y \]

- Difficult Constraints
- Linking Constraints
- **Multiple Sub-Systems** (variable splitting)
Decomposition: How

\[
\min \{ c^T x : A x \geq b, \quad x \in X \}
\]

difficult

nice
Decomposition: How

\[
\min \{ c \ x : A \ x \geq b, \ x \in X \}
\]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm

Lagrangian: \( L(x, u) := c \ x + u \ (b - A \ x) \)

Dual function: \( \theta(u) := \min_{x \in X} L(x, u) \)

Dual problem: \( LD := \max_{u \geq 0} \theta(u) \)
Decomposition: How

\[
\min \{ c \ x : \ A \ x \geq b, \ x \in X \} \tag{difficult}
\]
\[
\min \{ c \ x : \ A \ x \geq b, \ x \in X \} \tag{nice}
\]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)

\[
x \in X \rightarrow z \in Z : X_{LP} \subseteq \text{proj}_x Z_{LP}
\]
Decomposition: How

\[
\min \{ c^T x : A x \geq b, \quad x \in X \}
\]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)

\[
\{ x \in X_{LP} \} \longrightarrow \{ x \in X_{LP} \cap \gamma x \geq \gamma_0 \}
\]
Decomposition: How

\[ \min \{ c \ x : A \ x \geq b, \ x \in X \} \]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)

\[ \{ x \in X_{\text{LP}} \} \rightarrow \{ x \in X_{\text{LP}} \cap \gamma \ x \geq \gamma_0 \} \]
Decomposition: How

\[
\min \{ c \, x : A \, x \geq b, \ x \in X \} \tag{difficult}
\]

\[
\min \{ c \, x : A \, x \geq b, \ x \in X \} \tag{nice}
\]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)

\[
\{ x \in X_{LP} \} \rightarrow \{ x \in X_{LP} \cap \gamma \, x \geq \gamma_0 \}
\]
Decomposition: How

\[
\min \{ c \, x : \begin{array}{c} A \, x \geq b, \\ x \in X \end{array} \}
\]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)
- Dynamic Column Generation (Optimization Sub-Problem)

\[ x \in X = \{ x^i \}_{i \in I} \longrightarrow \{ x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\} \} \]
Decomposition: How

\[ \min \{ c \, x : A \, x \geq b, \quad x \in X \} \]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)
- Dynamic Column Generation (Optimization Sub-Problem)

\[ x \in X = \{ x^i \}_{i \in I} \rightarrow \{ x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\} \} \]
Decomposition: How

\[ \min \{ c \mathbf{x} : A \mathbf{x} \geq b, \ x \in X \} \]

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)
- Dynamic Column Generation (Optimization Sub-Problem)

\[ x \in X = \{ x^i \}_{i \in I} \rightarrow \{ x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\} \} \]
Decomposition: How

\[
\min \{ c \mathbf{x} : A \mathbf{x} \geq b, \quad \mathbf{x} \in X \}
\]

difficult \hspace{1cm} nice

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm
- Reformulation (Variable Redefinition)
- Dynamic Cut Generation (Separation Sub-Problem)
- Dynamic Column Generation (Optimization Sub-Problem)

\[
x \in X = \{ x^i \}_{i \in I} \rightarrow \{ x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{ 0, 1 \} \}
\]
Decomposition: Output

\[
\min \{ c \ x : \ A \ x \geq b, \ x \in X \} \\
\]  
\text{difficult} \quad \text{nice}

1. A solution method that exploits our hability to “treat” a subproblem

2. A Dual Bound that is typically better than LP bound
Decomposition: Output

\[ \min \{c x : A x \geq b, \ x \in X\} \]

1. A solution method that exploits our hablility to “treat” a subproblem

2. A Dual Bound that is typically better than LP bound

At best, the Lagrangian Dual Bound: \( \equiv \min \{c x : A x \geq b, x \in \text{conv}(X)\} \)
Decomposition: Output

\[
\min\{c x : A x \geq b, \ x \in X\}
\]

1. A solution method that exploits our hability to “treat” a subproblem
2. A Dual Bound that is typically better than LP bound

At best, the Lagrangian Dual Bound: \(\equiv \min\{cx : A x \geq b, x \in \text{conv}(X)\}\)
While the LP relaxation bound is \(\equiv \min\{cx : A x \geq b, x \in X_{\text{LP}}\}\)
Dantzig-Wolfe Decomposition/ Reformulation

Apply the change of variables:

\[ x \in X = \{x^i\}_{i \in I} \rightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\}\} \]

Then,

\[ \min \{c^T x : A x \geq b, \ x \in X\} \]

\[ \downarrow \]

\[ \min \{\sum_i c x^i \lambda_i : \sum_i A x^i \lambda_i \geq b, \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\}\} \]
Dantzig-Wolfe Decomposition/ Reformulation

Apply the change of variables:

\[ x \in X = \{x^i\}_{i \in I} \rightarrow \{ x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\} \} \]

Then,

\[ \min\{c \ x : A \ x \geq b, \ x \in X\} \]

\[ \downarrow \]

\[ \min\{\sum_i c \ x^i \lambda_i : \sum_i A \ x^i \lambda_i \geq b, \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\}\} \]

...to solve by column generation combined with branch-and-bound: i.e. by branch-and-price
The Column Generation approach has proved very successful in practice.

However, restricted to experts due to the difficulty of implementing it.
BaPCod : a generic Branch-And-Price Code

- The Column Generation approach has proved very successful in practice.
- However, restricted to experts due to the difficulty of implementing it.
- ∃ ‘tool-box’ codes : MINTO, ABACUS, BCP, or MAESTRO
- Our aim is to develop a ‘black-box’ implementation (in the same way as Cplex or Xpress provide implementation of Branch-and-Bound).
BaPCod : a generic Branch-And-Price Code

- The Column Generation approach has proved very successful in practice
- However, restricted to experts due to the difficulty of implementing it
- ∃ ‘tool-box” codes : MINTO, ABACUS, BCP, or MAESTRO
- Our aim is to develop a ‘black-box” implementation (in the same way as Cplex or Xpress provide implementation of Branch-and-Bound)
  - The user provides a MIP formulation of its problem
  - The user specifies what constraints make a subproblem
  - The code reformulates the problem for column generation
  - The code solves it with a default branch-and-price implementation
**BaPCod**: motivations

**Practical:**
- Provides a starting point to try out a column generation approach.
- Allows to test different decompositions: just change the subproblem definition.

**Research Wise:**
- Demands to generalize ad-hoc features to turn them into generic tools: initialization, preprocessing, branching, . . .
- Any algorithmic development is made available for all applications
- Permits to test algorithmic features across applications

**Future Outcome:**
Paves the way for future integration of column generation techniques in commercial MIP solvers
BaPCod: motivations

Practical:
- Provides a starting point to try out a column generation approach.
- Allows to test different decompositions: just change the subproblem definition.

Research Wise:
- Demands to generalize ad-hoc features to turn them into generic tools: initialization, preprocessing, branching, . . .
- Any algorithmic development is made available for all applications
- Permits to test algorithmic features across applications
BaPCod: motivations

Practical:

- Provides a starting point to try out a column generation approach.
- Allows to test different decompositions: just change the subproblem definition.

Research Wise:

- Demands to generalize ad-hoc features to turn them into generic tools: initialization, preprocessing, branching, . . .
- Any algorithmic development is made available for all applications
- Permits to test algorithmic features across applications

Future Outcome:

- Paves the way for future integration of column generation techniques in commercial MIP solvers
PART 2: Column Generation

- Standard Algorithm
- Primal/Dual point of view
- Stabilization
Dantzig-Wolfe Reformulation: LP relax. and dual

Original P: \[ \min \{ c \, x : A \, x \geq b, \, x \in X = \{ x^i \}_i \} \]

\[ \Downarrow \]

Master IP: \[ \min \{ \sum_i c \, x^i \, \lambda_i : \sum_i A \, x^i \, \lambda_i \geq b, \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\} \ \forall i \} \]
Dantzig-Wolfe Reformulation: LP relax. and dual

Original P: \[
\min \{ c \ x : \ A \ x \geq b, \ x \in X = \{ x^i \}_i \}
\]
\[\uparrow\]
Master IP: \[
\min \{ \sum_i c \ x^i \lambda_i : \sum_i A \ x^i \lambda_i \geq b, \ \sum_i \lambda_i = 1, \ \lambda_i \in \{0, 1\} \ \forall i \}
\]

Relaxed P: \[
\min \{ c \ x : \ A \ x \geq b, \ x \in \text{conv}(X) \}
\]
\[\uparrow\]
Master LP: \[
\min \{ \sum_i c \ x^i \lambda_i : \sum_i A \ x^i \lambda_i \geq b, \ \sum_i \lambda_i = 1, \ \lambda_i \geq 0 \ \forall i \} \]
Dantzig-Wolfe Reformulation: LP relax. and dual

Original P: \[ \min \{ c x : \ A x \geq b, \ x \in X = \{ x^i \}_i \} \]

\[ \Downarrow \]

Master IP: \[ \min \{ \sum_i c x^i \lambda_i : \sum_i A x^i \lambda_i \geq b, \ \sum_i \lambda_i = 1, \lambda_i \in \{0, 1\} \ \forall i \} \]

\[ \Downarrow \]

Relaxed P: \[ \min \{ c x : \ A x \geq b, \ x \in \text{conv}(X) \} \]

\[ \Downarrow \]

Master LP: \[ \min \{ \sum_i c x^i \lambda_i : \sum_i A x^i \lambda_i \geq b, \sum_i \lambda_i = 1, \lambda_i \geq 0 \ \forall i \} \]

\[ \Downarrow \]

Dual Master: \[ \max \{ b u - r : \ u A x^i - r \leq c x^i \ \forall i, \ u \geq 0 \} \]

\[ \Downarrow \]

\[ \max \{ \eta : \eta \leq c x^i + u (b - A x^i) \ \forall i, \ u \geq 0 \} \]

\[ \Downarrow \]

Lagrangian D: \[ \max \{ \theta(u) : \ u \geq 0 \} \text{ with } \theta(u) := \min \limits_i L(x^i, u) \]
Master Problem

\[
\min \sum_{i \in I} (c x^i) \lambda_i \\
\text{s.t.} \quad \sum_{i \in I} (A x^i) \lambda_i \geq b \\
\quad \sum_{i \in I} \lambda_i = 1 \\
\quad \lambda_i \geq 0
\]
Standard column generation: primal view

Master Problem

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I} (c^i x^i) \lambda_i \\
\text{s.t.} & \quad \sum_{i \in I} (A x^i) \lambda_i \geq b \\
& \quad \sum_{i \in I} \lambda_i = 1 \\
& \quad \lambda_i \geq 0
\end{align*}
\]

Restricted Master Problem (iteration \(k\)): \(I^k \subset I\)

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I^k} (c^i x^i) \lambda_i \\
\text{s.t.} & \quad \sum_{i \in I^k} (A x^i) \lambda_i \geq b \\
& \quad \sum_{i \in I^k} \lambda_i = 1 \\
& \quad \lambda_i \geq 0
\end{align*}
\]
Standard column generation: primal view

Master Problem

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I} (c x^i) \lambda_i \\
\text{s.t.} & \quad \sum_{i \in I} (A x^i) \lambda_i \geq b \\
& \quad \sum_{i \in I} \lambda_i = 1 \\
& \quad \lambda_i \geq 0
\end{align*}
\]

Restricted Master Problem (iteration \(k\)):

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I^k} (c x^i) \lambda_i \\
\text{s.t.} & \quad \sum_{i \in I^k} (A x^i) \lambda_i \geq b \quad \rightarrow u^k \\
& \quad \sum_{i \in I^k} \lambda_i = 1 \quad \rightarrow r^k \\
& \quad \lambda_i \geq 0
\end{align*}
\]
Standard column generation: primal view

Master Problem

$$\min \sum_{i \in I} (c x^i) \lambda_i$$

s.t. $$\sum_{i \in I} (A x^i) \lambda_i \geq b$$

$$\sum_{i \in I} \lambda_i = 1$$

$$\lambda_i \geq 0$$

Restricted Master Problem (iteration $k$):

$$\min \sum_{i \in I^k} (c x^i) \lambda_i$$

s.t. $$\sum_{i \in I^k} (A x^i) \lambda_i \geq b \rightarrow u^k$$

$$\sum_{i \in I^k} \lambda_i = 1 \rightarrow r^k$$

$$\lambda_i \geq 0$$

Subproblem (Oracle):

$$\min_{i \in I} (c - u^k A) x^i = \min_{x \in X} (c - u^k A) x \rightarrow x^{k+1}$$
Standard column generation: primal view

Master Problem

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I}(cx^i)\lambda_i \\
\text{s.t.} & \quad \sum_{i \in I}(Ax^i)\lambda_i \geq b \\
& \quad \sum_{i \in I}\lambda_i = 1 \\
& \quad \lambda_i \geq 0
\end{align*}
\]

Restricted Master Problem (iteration \(k\)):

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I^k}(cx^i)\lambda_i \\
\text{s.t.} & \quad \sum_{i \in I^k}(Ax^i)\lambda_i \geq b \rightarrow u^k \\
& \quad \sum_{i \in I^k}\lambda_i = 1 \rightarrow r^k \\
& \quad \lambda_i \geq 0
\end{align*}
\]

Subproblem (Oracle) : \(\min_{i \in I}(c - u^k A)x^i = \min_{x \in X}(c - u^k A)x\)

If \(((c - u^k A)x^{k+1} - r^k) < 0\), add \(k + 1\) to \(I^k\). Otherwise, \text{STOP.}
Standard column generation: primal view

Master Problem

\[
\begin{align*}
\min & \quad \sum_{i \in I} (cx^i)\lambda_i \\
\text{s.t.} & \quad \sum_{i \in I} (Ax^i)\lambda_i \geq b \\
& \quad \sum_{i \in I} \lambda_i = 1 \\
& \quad \lambda_i \geq 0
\end{align*}
\]

Restricted Master Problem (iteration \(k\)):

\[
\begin{align*}
\min & \quad \sum_{i \in I^k} (cx^i)\lambda_i \\
\text{s.t.} & \quad \sum_{i \in I^k} (Ax^i)\lambda_i \geq b \rightarrow u^k \\
& \quad \sum_{i \in I^k} \lambda_i = 1 \rightarrow r^k \\
& \quad \lambda_i \geq 0
\end{align*}
\]

Subproblem (Oracle): \(\min_{i \in I}(c - u^k A)x^i = \min_{x \in X}(c - u^k A)x\)

If \(((c - u^k A)x^{k+1} - r^k) < 0\), add \(k + 1\) to \(I^k\). Otherwise, STOP.
Standard column generation: dual view

Restricted Master Dual: \[
\max \{ \eta : \eta \leq cx^i + u(b - Ax^i) \forall i \in I^k, \ u \geq 0 \} \\
L(x^i, u)
\]
Standard column generation: dual view

Restricted Master Dual: \[ \max \{ \eta : \eta \leq cx^i + u(b - Ax^i) \forall i \in I^k, u \geq 0 \} \rightarrow (u^k) \]

Approxim Dual Function: \[ \theta^k(u) = ub + \min_{i \in I^k} (c - uA)x^i \]
Standard column generation: dual view

Restricted Master Dual: \[ \max \{ \eta : \eta \leq cx^i + u(b - A x^i) \forall i \in I^k, \; u \geq 0 \} \rightarrow (u^k) \]

Approxim Dual Function: \[ \theta^k(u) = ub + \min_{i \in I^k} (c - uA)x^i \]

Subproblem (Oracle): \[ \theta(u^k) = u^k b + \min_{x \in X} (c - u^k A)x \]
Standard column generation: dual view

Restricted Master Dual: \[
\max \{ \eta : \eta \leq cx^i + u(b - A x^i) \forall i \in I^k, \ u \geq 0 \} \rightarrow (u^k)
\]

Approxim Dual Function: \[
\theta^k(u) = ub + \min_{i \in I^k} (c - uA)x^i
\]

Subproblem (Oracle): \[
\theta(u^k) = u^kb + \min_{x \in X} (c - u^kA)x \rightarrow (x^{k+1})
\]
### Standard column generation: dual view

**Restricted Master Dual:**

\[
\max\{\eta : \eta \leq cx^i + u(b - Ax^i) \forall i \in I^k, \ u \geq 0\} \rightarrow (u^k)
\]

**Approxim Dual Function:**

\[
\theta^k(u) = ub + \min_{i \in I^k} (c - uA)x^i
\]

**Subproblem (Oracle):**

\[
\theta(u^k) = u^kB + \min_{x \in X} (c - u^kA)x \rightarrow (x^{k+1})
\]
Standard column generation: dual view

Restricted Master Dual: \( \max \{ \eta : \eta \leq cx^i + u(b - Ax^i) \forall i \in I^k, \ u \geq 0 \} \rightarrow (u^k) \)

Approxim Dual Function: \( \theta^k(u) = ub + \min_{i \in I^k} (c - uA)x^i \)

Subproblem (Oracle): \( \theta(u^k) = u^k b + \min_{x \in X} (c - u^k A)x \rightarrow (x^{k+1}) \)
Standard column generation: dual view

Restricted Master Dual: \[
\max \{ \eta : \eta \leq cx^i + u(b - Ax^i) \forall i \in I^k, \; u \geq 0 \} \rightarrow (u^k)
\]

Approxim Dual Function: \[
\theta^k(u) = ub + \min_{i \in I^k} (c - uA)x^i
\]

Subproblem (Oracle): \[
\theta(u^k) = u^k b + \min_{x \in X} (c - u^k A)x \rightarrow (x^{k+1})
\]

Kelley’s cutting plane algorithm
At iteration $k$, one has

A Primal Bound on the master: \[ PB \equiv \theta^k(u^k) = \sum_{i \in I^k} cx^i \lambda_i \]

A Dual Bound on the master: \[ DB \equiv \theta(u^k) = PB - (c - u^k A)x^{k+1} \]
At iteration $k$, one has

**A Primal Bound** on the master:

$$PB \equiv \theta^k(u^k) = \sum_{i \in I^k} cx^i \lambda_i$$

**A Dual Bound** on the master:

$$DB \equiv \theta(u^k) = PB - (c - u^kA)x^{k+1}$$
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Slow convergence of Kelley’s algorithm
Instability: observed behaviors

[Image of a graph with labeled axes and markers indicating "restricted master LP values," "intermediate Lagrangian bounds," and "Tailing-off".]
Instability: observed behaviors

- Restricted master LP values
- Intermediate Lagrangian bounds

Heading-in

Tailing-off
Instability: observed behaviors

- Instability: observed behaviors
- Restricted master LP values
- Intermediate Lagrangian bounds
- Master LP value iteration
- Degeneracy
- Heading-in
- Tailing-off
Instability: observed behaviors

Bang-Bang Behavior

Degeneracy

Heading-in

Tailing-off

restricted master LP values

intermediate Lagrangian bounds
Instability: observed behaviors

Bang-Bang Behavior

Degeneracy

Heading-in

Tailing-off

restricted master Lp values

intermediate Lagrangian bounds

Master LP value
Instability: observed behaviors

Bang-Bang Behavior
Instability: observed behaviors

Bang-Bang Behavior
Instability: observed behaviors

Bang-Bang Behavior
Instability: observed behaviors

Bang-Bang Behavior

Degeneracy

Heading-in

Jumpy Behavior

Tailing-off
Stabilization Methods -

- penalizing distance from a stability center \( \hat{u} \)

\( \hat{u} \rightarrow \) best dual bound
Stabilization Methods - \( \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \)

- penalizing distance from a stability center \( \hat{u} \)
Stabilization Methods - \[ \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \]

- penalizing distance from a stability center \( \hat{u} \)

\[ S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon) \] (box step)  

*R.E. Martsen, W.W. Hogan and J.W. Blankenship, 1975*
Stabilization Methods - \( \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \)

- penalizing distance from a stability center \( \hat{u} \)

\[
S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon) \quad \text{(box step)}
\]

*R.E.Martsen, W.W.Hogan and J.W.Blankenship, 1975*

\[
S(u - \hat{u}) = ||u - \hat{u}||_1 \quad \text{(1 breakpoint)}
\]

*S.Kim, K.N.Chang and J.Y.Lee, 1995*
Stabilization Methods -

\[ \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \]

- penalizing distance from a stability center \( \hat{u} \)

\[ S(u - \hat{u}) = \delta(||u - \hat{u}||_{\infty} \leq \epsilon) \] (box step)

R.E. Martsen, W.W. Hogan and J.W. Blankenship, 1975

\[ S(u - \hat{u}) = ||u - \hat{u}||_1 \] (1 breakpoint)

S. Kim, K.N. Chang and J.Y. Lee, 1995

\[ S(u - \hat{u}) = ||u - \hat{u}||_1 + \text{penalization outside boxstep} \] (3 breakpoints)

O. du Merle, D. Villeneuve, J. Desrosiers and P. Hansen, 1999
Stabilization Methods - \( \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \)

- Penalizing distance from a stability center \( \hat{u} \)
  \[
  S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon) \text{ (box step)}
  \]

R.E. Martsen, W.W. Hogan and J.W. Blankenship, 1975

- \( S(u - \hat{u}) = ||u - \hat{u}||_1 \) (1 breakpoint)

S. Kim, K.N. Chang and J.Y. Lee, 1995

- \( S(u - \hat{u}) = ||u - \hat{u}||_1 + \text{penalization outside boxstep (3 breakpoints)} \)

O. du Merle, D. Villeneuve, J. Desrosiers and P. Hansen, 1999

- \( S(u - \hat{u}) = \text{penalization outside 2 boxsteps (4 breakpoints)} \)

H. Ben-Amor and J. Desrosiers, 2003
Stabilization Methods -

\[ \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \]

- penalizing distance from a stability center \( \hat{u} \)

\[ S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon) \] (box step)

R.E. Martsen, W.W. Hogan and J.W. Blankenship, 1975

\[ S(u - \hat{u}) = ||u - \hat{u}||_1 \] (1 breakpoint)

S.Kim, K.N. Chang and J.Y. Lee, 1995

\[ S(u - \hat{u}) = ||u - \hat{u}||_1 + \text{penalization outside boxstep (3 breakpoints)} \]

O.du Merle, D. Villeneuve, J. Desrosiers and P. Hansen, 1999

\[ S(u - \hat{u}) = \text{penalization outside 2 boxsteps (4 breakpoints)} \]

H. Ben-Amor and J. Desrosiers, 2003

\( \Rightarrow \) As \( S(u - \hat{u}) \) is a concave and piecewise linear function, we can still use LP solvers.
Stabilization Methods -

\[ \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \]

- penalizing distance from a stability center \( \hat{u} \)

\[ S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon) \] (box step)

\[ S(u - \hat{u}) = ||u - \hat{u}||_1 \] (1 breakpoint)

\[ S(u - \hat{u}) = ||u - \hat{u}||_1 + \text{penalization outside boxstep} \] (3 breakpoints)

\[ S(u - \hat{u}) = \text{penalization outside 2 boxsteps} \] (4 breakpoints)

\[ S(u - \hat{u}) = \frac{1}{2t}||u - \hat{u}||^2 \] ⇒ Bundle

R.E. Martsen, W.W. Hogan and J.W. Blankenship, 1975

S. Kim, K.N. Chang and J.Y. Lee, 1995

O. du Merle, D. Villeneuve, J. Desrosiers and P. Hansen, 1999

H. Ben-Amor and J. Desrosiers, 2003

C. Lemaréchal and C. Sagastizábal, 1997
Stabilization Methods - \( \tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u}) \)

- penalizing distance from a stability center \( \hat{u} \)
  \[ S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon) \quad \text{(box step)} \]
  \[ S(u - \hat{u}) = ||u - \hat{u}||_1 \quad \text{(1 breakpoint)} \]
  \[ S(u - \hat{u}) = ||u - \hat{u}||_1 + \text{penalization outside boxstep} \quad \text{(3 breakpoints)} \]
  \[ S(u - \hat{u}) = \text{penalization outside 2 boxsteps} \quad \text{(4 breakpoints)} \]

- Bundle
  \[ S(u - \hat{u}) = \frac{1}{2t} ||u - \hat{u}||^2 \Rightarrow \text{Bundle} \]

- As \( S(u - \hat{u}) \) is a quadratic concave function, we must now use Quadratic solvers

*References:
  - R.E. Martsen, W.W. Hogan and J.W. Blankenship, 1975
  - S. Kim, K.N. Chang and J.Y. Lee, 1995
  - O. du Merle, D. Villeneuve, J. Desrosiers and P. Hansen, 1999
  - C. Lemaréchal and C. Sagastizábal, 1997
Stabilization Methods

- smoothing dual value: 
  \[ u^k = \alpha u^{Kelley} + (1 - \alpha) u^{k-1} \]

*P. Wentges, 1997*
Stabilization Methods

- smoothing dual value: $u^k = \alpha u^{K_{elley}} + (1 - \alpha) u^{k-1}$
  
  P.Wentges, 1997

- Using interior-point-like technique to generate dual solution
Stabilization Methods

- smoothing dual value: \( u^k = \alpha u^{Kelley} + (1 - \alpha) u^{k-1} \)
  
  *P.Wentges, 1997*

- Using interior-point-like technique to generate dual solution
  
  *Kelley solved by interior point method*

  *L-M Rousseau, M.Gendreau and D.Feillet, 2003*

- Analytic Center Cutting-Plane Method (ACCPM)

  \[
  \tilde{\theta}^k(u) = \sum_{i \in I^k} \log(c^i x^i - u a^i x^i + r) + k \log(b u - r - \theta(\hat{u}))
  \]

  *J.L. Goffin, A.Haurie and J.Ph. Vial, 1992*
Bi-Dual: Fenchel duality

\[
\begin{align*}
\max \theta^k(u) & \equiv \min \{ c \, x : A \, x \geq b, \, x \in \text{conv}(X^k) \} \\
\max \theta^k(u) - S(u - \hat{u}) & \equiv \ldots
\end{align*}
\]
Bi-Dual: Fenchel duality

$$\max \theta^k(u) \equiv \min \{ c x : A x \geq b, x \in \text{conv}(X^k) \}$$
$$\max \theta^k(u) - S(u - \hat{u}) \equiv \min \{ c x + \hat{u} g + S^*(g) : A x + g \geq b, x \in \text{conv}(X^k), g \in \mathbb{R}^m \}$$
Bi-Dual: Fenchel duality

\[
\begin{align*}
\max \theta^k(u) & \equiv \min \{ c x : A x \geq b, x \in \text{conv}(X^k) \} \\
\max \theta^k(u) - S(u - \hat{u}) & \equiv \min \{ c x + \hat{u} g + S^*(g) : \\
& \quad A x + g \geq b, x \in \text{conv}(X^k), g \in \mathbb{R}^m \}
\end{align*}
\]

\[
\begin{align*}
\min \sum_{i=1}^{k} c x^i \lambda_i + \sum_{j=1}^{m} (\delta + \hat{u}_j) g^+_j + \sum_{j=1}^{m} (\delta - \hat{u}_j) g^-_j \\
\sum_{i=1}^{k} A x^i \lambda_i + g^+_j - g^-_j & \geq b_j \quad j = 1, \ldots, m, \\
s & \geq g^+_j, g^-_j \geq 0 \quad j = 1, \ldots, m, \\
\sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0 \quad i.
\end{align*}
\]
PART 3: Numerical Comparison
Tested variants

1. “Poor” initialization
   - Kelley initialized with 1 artificial column
   - Bundle initialized with $u_0 = 0$

2. “Rich” initialization: heuristic $\hat{u}$
   - Kelley initialized with $m$ artificial columns of cost $\hat{u}$
   - Bundle initialized with $u_0 = \hat{u}$

3. Primal / Dual approach:
   - Kelley (for primal bounds) + “rich” bundle (for dual bounds)
Traveling Salesman Problems (TSP)

Instances from TSPLIB

\[ \sum_{e \in \delta(i)} x_e = 2 \quad \forall i \]

12èmes journées du groupe MODE, Le Havre, 2004. – p.29/47
TSP : bays29 : primal and dual bounds

43 iterations

Primal bound with Bundle
Dual bound with Bundle

0 100 200 300 400 500 600 700 800 900 1000
0
1820
3640
5460
7280
9100
10920
12740
14560
16380
18200

Dual bound with Kelley
Primal bound with Kelley
Dual bound with Stabilized Kelley
Primal bound with Stabilized Kelley
Dual bound with Bundle
Primal bound with Bundle
TSP : bays29 : primal and dual bounds
TSP : bays29 : primal and dual bounds

- 43 iterations
- 419 iterations
- 744 iterations

Dual bound with Kelley
Primal bound with Kelley
Dual bound with Stabilized Kelley
Primal bound with Stabilized Kelley
Dual bound with Bundle
Primal bound with Bundle
TSP : eil51 : primal and dual bounds

92 iterations

Primal bound with Bundle
Dual bound with Bundle
Primal bound with Stabilized Kelley
Dual bound with Stabilized Kelley
Primal bound with Kelley
Dual bound with Kelley
TSP : eil76 : primal and dual bounds

153 iterations

12èmes journées du groupe MODE, Le Havre, 2004. – p.32/47
TSP : pr76 : primal and dual bounds

155 iterations
TSP : gr120 : primal and dual bounds

210 iterations

Dual bound with Kelley
Primal bound with Kelley
Dual bound with Stabilized Kelley
Primal bound with Stabilized Kelley
Dual bound with Bundle
Primal bound with Bundle

12èmes journées du groupe MODE, Le Havre, 2004. – p.34/47
pr76 : first 1-trees with Kelley
pr76 : first 1-trees with Stabilized Kelley
pr76 : first 1-trees with Bundle
Instances from CVRPLIB
## CVRP: Comparative results

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Iter</th>
<th>Col</th>
<th>Oracle</th>
<th>Master</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>n22-k8</td>
<td>kelley poor</td>
<td>53</td>
<td>53</td>
<td>277</td>
<td>12</td>
<td>2s99t</td>
</tr>
<tr>
<td>n22-k8</td>
<td>kelley rich</td>
<td>52</td>
<td>35</td>
<td>323</td>
<td>8</td>
<td>3s36t</td>
</tr>
<tr>
<td>n22-k8</td>
<td>bundle rich</td>
<td>42</td>
<td>32</td>
<td>483</td>
<td>17</td>
<td>5s08t</td>
</tr>
<tr>
<td>n14-k2</td>
<td>kelley poor</td>
<td>54</td>
<td>54</td>
<td>26317</td>
<td>5</td>
<td>4m23s</td>
</tr>
<tr>
<td>n14-k2</td>
<td>kelley rich</td>
<td>44</td>
<td>43</td>
<td>30732</td>
<td>3</td>
<td>5m07s</td>
</tr>
<tr>
<td>n14-k2</td>
<td>bundle rich</td>
<td>16</td>
<td>14</td>
<td>180509</td>
<td>6</td>
<td>30m05s</td>
</tr>
</tbody>
</table>

Cutting Stock Problem

\[ \begin{align*}
\min \text{number of rolls} & \quad \left\{ \begin{array}{l}
\min \sum_k y_k \\
\sum_k x_{i,k} \geq d_i \quad \forall i \\
\sum_i s_i x_{i,k} \leq y_k \quad \forall k
\end{array} \right.
\]

\[ \begin{align*}
\min \text{waste} & \quad \left\{ \begin{array}{l}
\min \sum_k (1 - \sum_i s_i x_{i,k}) \\
\bar{d}_i \geq \sum_k x_{i,k} \geq d_i \quad \forall i \\
\sum_i s_i x_{i,k} \leq y_k \quad \forall k
\end{array} \right.
\]
CSP number of rolls: Comparative results

average on 10 or 20 instances

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Counters</th>
<th>Timers (ticks)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Iter</td>
<td>Col</td>
</tr>
<tr>
<td>realDat20</td>
<td>kelley poor</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>realDat20</td>
<td>bundle poor</td>
<td>90</td>
<td>52</td>
</tr>
<tr>
<td>realDat20</td>
<td>kelley rich</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>realDat20</td>
<td>bundle rich</td>
<td>51</td>
<td>38</td>
</tr>
<tr>
<td>realDat20</td>
<td>bundle + Kelley</td>
<td>39</td>
<td>33</td>
</tr>
<tr>
<td>50b100</td>
<td>kelley poor</td>
<td>228</td>
<td>228</td>
</tr>
<tr>
<td>50b100</td>
<td>bundle poor</td>
<td>257</td>
<td>190</td>
</tr>
<tr>
<td>50b100</td>
<td>kelley rich</td>
<td>138</td>
<td>138</td>
</tr>
<tr>
<td>50b100</td>
<td>bundle rich</td>
<td>145</td>
<td>123</td>
</tr>
<tr>
<td>50b100</td>
<td>bundle + Kelley</td>
<td>124</td>
<td>115</td>
</tr>
</tbody>
</table>
CSP number of rolls: primal and dual bounds
## CSP Waste: Comparative results

average on 10 or 20 instances

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Counters</th>
<th>Timers (ticks)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Iter</td>
<td>Col</td>
</tr>
<tr>
<td>realDat20</td>
<td>kelley poor</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>realDat20</td>
<td>kelley rich</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>realDat20</td>
<td>bundle rich</td>
<td>71</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50b100</td>
<td>kelley poor</td>
<td>146</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50b100</td>
<td>kelley rich</td>
<td>177</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50b100</td>
<td>bundle rich</td>
<td>205</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12èmes journées du groupe MODE, Le Havre, 2004. – p.43/47
CSP Waste: primal and dual bounds

Primal bound with Stabilized Kelley
Dual bound with Stabilized Kelley
Primal bound with Kelley (kelley and bundle)
Dual bound with Bundle (kelley and bundle)
### Multi-Item Lot-Sizing: Comparative results

**Diagram:**
- **Machine** connected to **Item 1** and **Item 2** for periods 1 to 5.

**Table:**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Iter</th>
<th>SP</th>
<th>Col</th>
<th>Oracle</th>
<th>Master</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>i20-t60</td>
<td>kelley</td>
<td>8</td>
<td>164</td>
<td>117</td>
<td>55</td>
<td>4</td>
<td>4s79t</td>
</tr>
<tr>
<td>i20-t60</td>
<td>bundle</td>
<td>66</td>
<td>1326</td>
<td>207</td>
<td>455</td>
<td>92</td>
<td>18s41t</td>
</tr>
</tbody>
</table>

**Note:** Average on 10 instances. Decomposition in SILS.
Observations

(based on preliminary results)

- Bundle takes far fewer iterations for some applications (like TSP), while not much is to win (compared to kelley) in other applications.

  ? Characterization ?

- Stabilization may imply harder oracles. Hence, there is a time tradeoff.

- Kelley finds primal feasible solution earlier.

- Kelley accepts inexact oracles.
Example: Multi-Item Lot-Sizing

Periods 1 to 5

Machine

Item 1

Item 2
Example: Multi-Item Lot-Sizing

Periods 1 to 5

Machine

Item 1

Item 2

\[ x_{it} \leq c_{it} y_{it} \]
Example: Multi-Item Lot-Sizing

Periods 1 to 5

Machine

$\text{Item 1}$

$\text{Item 2}$

$d_1$ $d_2$ $d_3$ $d_4$ $d_5$

$x_{it} \leq c_{it} y_{it}$

$x_{i1t} \geq d_{i1t}$
Example: Multi-Item Lot-Sizing

Periods 1 to 5

Machine

Item 1

\[ \sum_i (x_{it} + s_i y_{it}) \leq C_t \]

\[ x_{it} \leq c_{it} y_{it} \]

Item 2

\[ x_{i1t} \geq d_{i1t} \]