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"

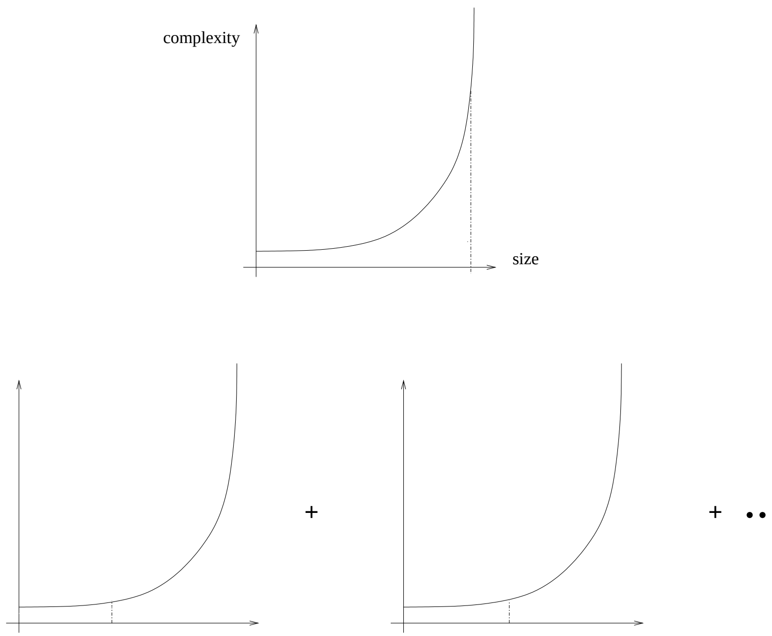
`http://www.math.u-bordeaux.fr/MAB/ODW/`

PART 1: Background

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

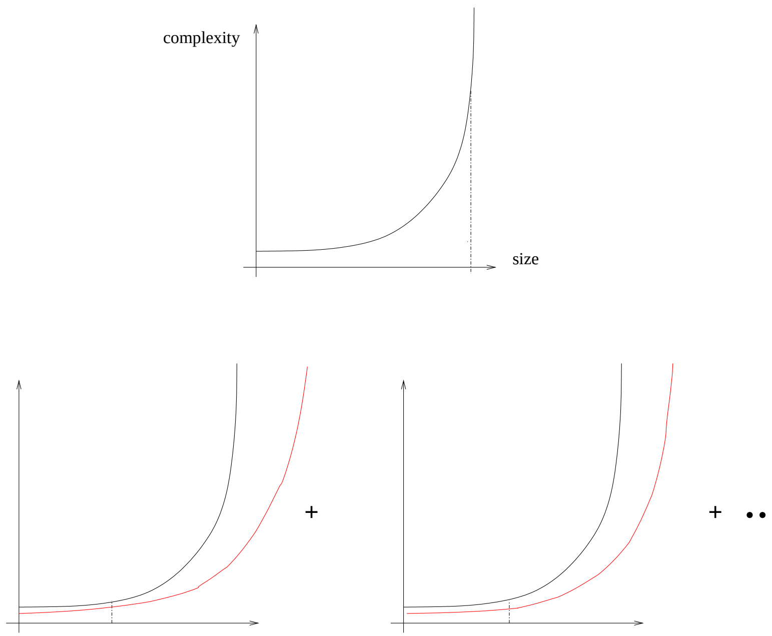
Decomposition: Why

- Divide and Conquer



Decomposition: Why

- Divide and Conquer



- Exploit the structure

Decomposition: When

$$\begin{array}{rcl} \min & c\,x & \\ A\,x & \geq & b \\ x & \in & X \subset \mathbb{N} \end{array}$$

Decomposition: When

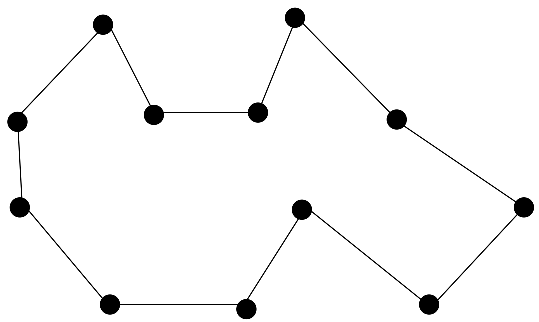
$$\begin{array}{llll} \min & c\,x & & \\ & A\,x & \geq & b \\ & x & \in & X \subset \mathbb{N} \end{array}$$

- Difficult Constraints

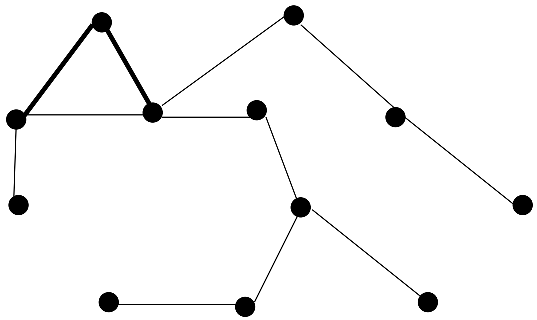
Decomposition: When

$$\begin{array}{lrcl} \min & c\,x & & \\ & A\,x & \geq & b \\ & x & \in & X \subset \mathbb{N} \end{array}$$

- Difficult Constraints **<u>Ex:</u>** The Travelling Salesman Prob.



$$\equiv \left\{ \begin{array}{c} \sum_{e \in \delta(i)} x_e = 2 \quad \forall i \\ + \end{array} \right.$$



Decomposition: When

$$\begin{array}{rrcl} \min & c\,x & & \\ & A\,x & \geq & b \\ & x & \in & X \subset \mathbb{N} \end{array}$$

- Difficult Constraints
- Linking Constraints

$$\begin{pmatrix} A_1 & A_2 & \dots & A_n \\ B & 0 & \dots & 0 \\ 0 & B & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & B \end{pmatrix}$$

Decomposition: When

$$\begin{array}{rrcl} \min & \frac{c}{2}x & + & \frac{c}{2}y \\ & x & = & y \\ & x & \in & X \\ & y & \in & Y \end{array}$$

- Difficult Constraints
- Linking Constraints
- Multiple Sub-Systems (variable splitting)

Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

- Lagrangian relaxation, Lagr. Dual, Subgradient Algorithm

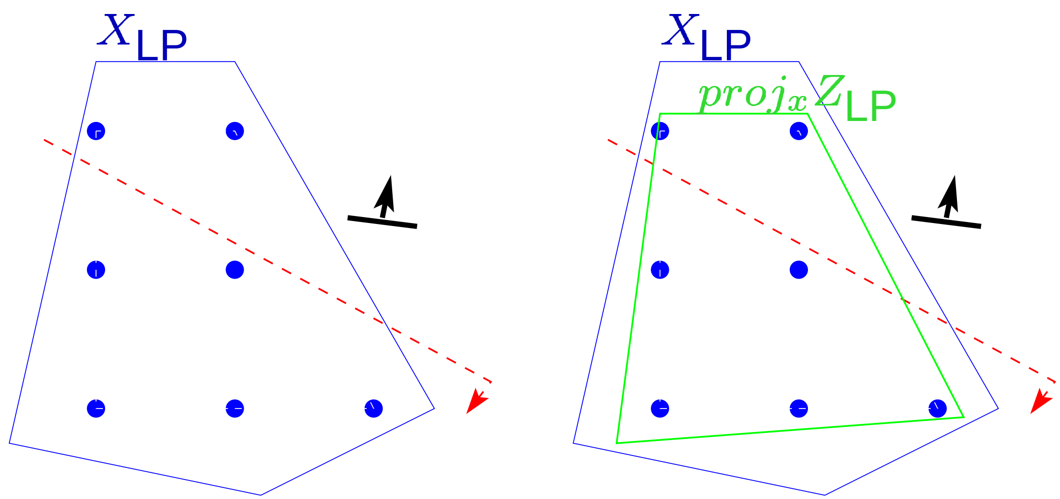
Lagrangian:	$L(x,u)$	$:= c\,x + u\,(b - A\,x)$
Dual function:	$\theta(u)$	$:= \min_{x \in X} \; L(x,u)$
Dual problem:	LD	$:= max_{u \geq 0}\theta(u)$

Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$x \in X \longrightarrow z \in Z : X_{\text{LP}} \subset proj_x Z_{\text{LP}}$$

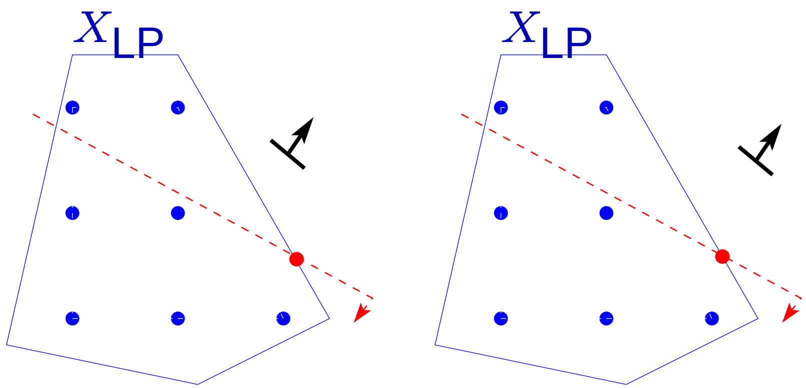


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$\{x \in X_{\mathsf{LP}}\} \longrightarrow \{x \in X_{\mathsf{LP}} \cap \gamma\,x \geq \gamma_0\}$$

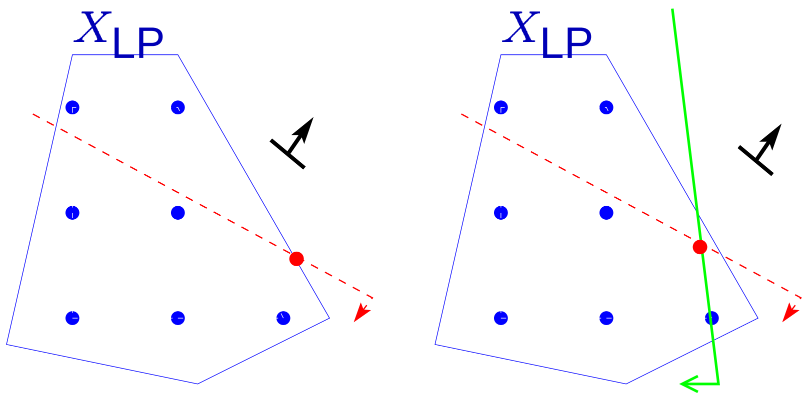


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$\{x \in X_{\mathsf{LP}}\} \longrightarrow \{x \in X_{\mathsf{LP}} \cap \gamma\,x \geq \gamma_0\}$$

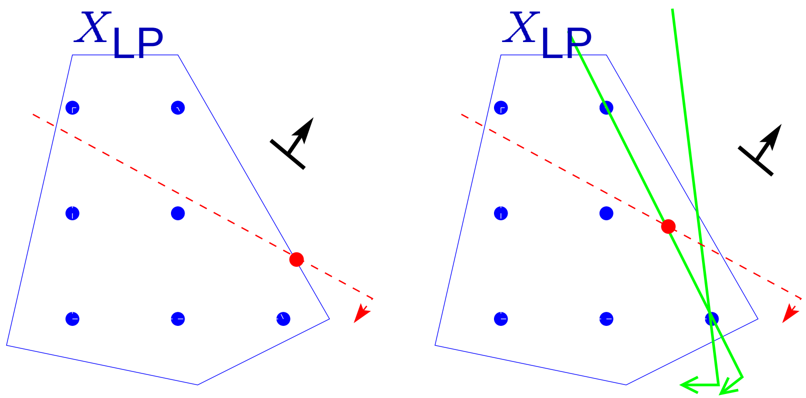


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$\{x \in X_{\mathsf{LP}}\} \longrightarrow \{x \in X_{\mathsf{LP}} \cap \gamma\,x \geq \gamma_0\}$$

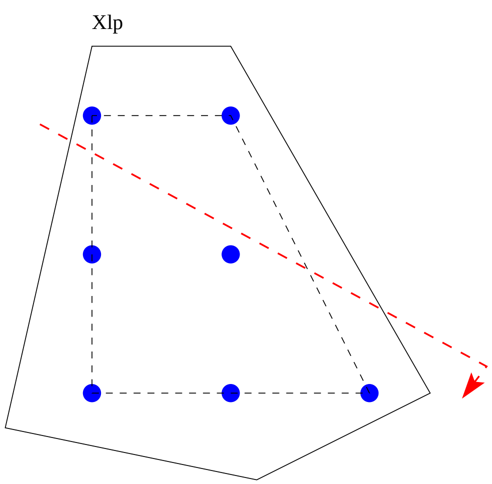


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$x \in X = \{x^i\}_{i \in I} \longrightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0,1\}\}$$

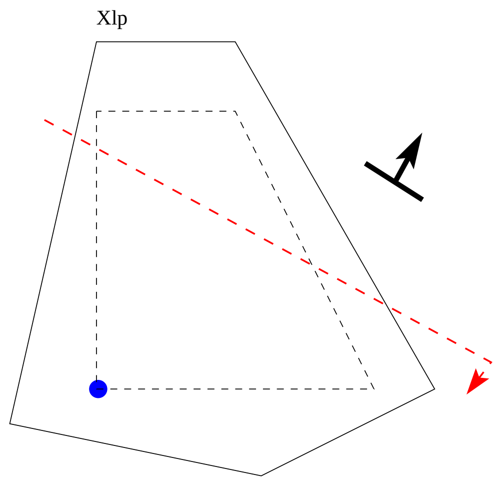


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$x \in X = \{x^i\}_{i \in I} \longrightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0,1\}\}$$

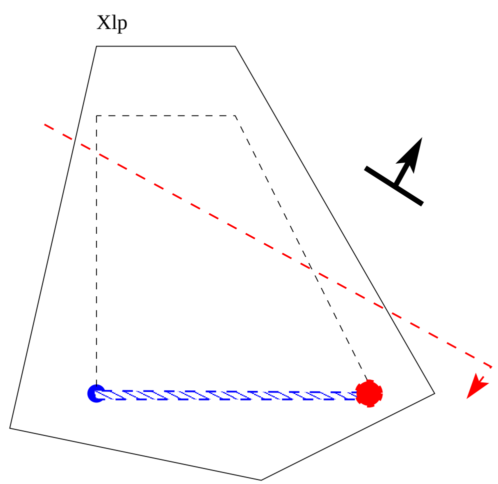


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$x \in X = \{x^i\}_{i \in I} \longrightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0,1\}\}$$

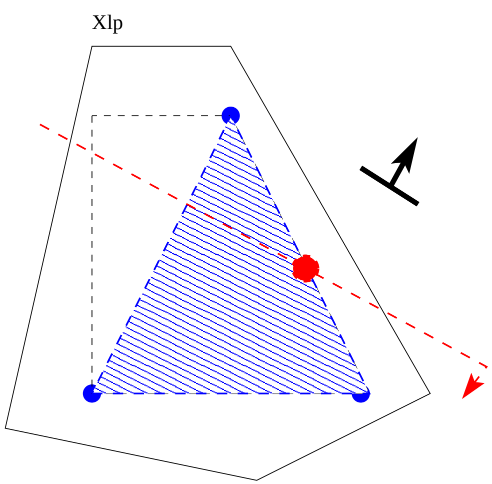


Decomposition: How

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

$$x \in X = \{x^i\}_{i\in I} \longrightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0,1\}\}$$



Decomposition: Output

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

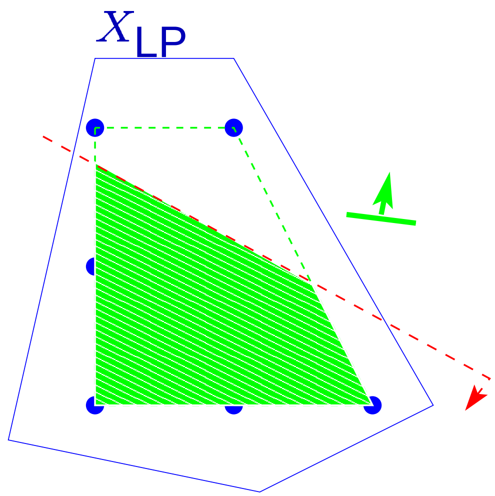
1. A solution method that exploits our hability to "treat" a subproblem
2. A Dual Bound that is typically better than LP bound

Decomposition: Output

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

At best, the **Lagrangian Dual Bound:** $\equiv \min\{cx : \ A\,x \geq b, x \in conv(X)\}$

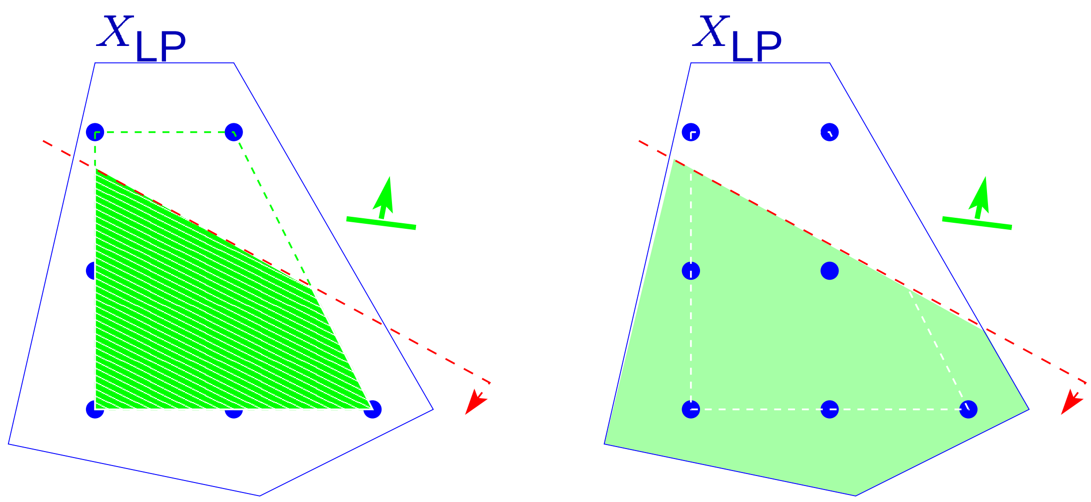


Decomposition: Output

$$\min\{c\,x : \underbrace{A\,x \geq b}_{\text{difficult}},\ \underbrace{x \in X}_{\text{nice}}\}$$

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

At best, the **Lagrangian Dual Bound:** $\equiv \min\{cx :\ A\,x \geq b, x \in conv(X)\}$
While the LP relaxation bound is $\equiv \min\{cx :\ A\,x \geq b, x \in X_{\mathrm{LP}}\}$



Dantzig-Wolfe Decomposition/ Reformulation

Apply the change of variables:

$$x \in X = \{x^i\}_{i \in I} \longrightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0,1\}\}$$

Then,

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

$$\downarrow$$

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

Dantzig-Wolfe Decomposition/ Reformulation

Apply the change of variables:

$$x \in X = \{x^i\}_{i \in I} \longrightarrow \{x = \sum_i x^i \lambda_i : \sum_i \lambda_i = 1, \lambda_i \in \{0,1\}\}$$

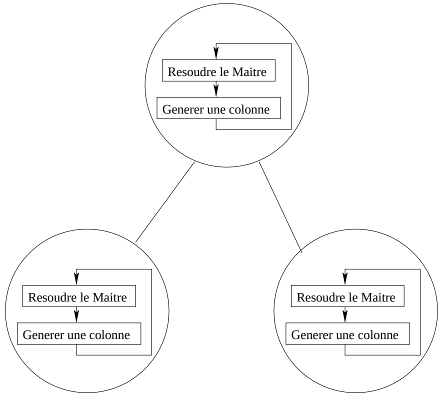
Then,

$$\min\{c\,x : A\,x \geq b,\; x \in X\}$$

$$\downarrow$$

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

to solve by column generation combined with branch-and-bound: i.e. by branch-and-price



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

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
- $\exists$ ‘*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.

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: $\min\{c\,x : A\,x \geq b, x \in X = \{x^i\}_i\}$

$$\Updownarrow$$

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: $\min\{c\,x : \; A\,x \; \geq b, x \in X = \{x^i\}_i\}$

$$\Updownarrow$$

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: $\min\{c\,x : \; A\,x \; \geq b, x \in conv(X)\}$

$$\Updownarrow$$

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: $\min\{c\,x : A\,x \geq b, x \in X = \{x^i\}_i\}$

$$\Updownarrow$$

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: $\min\{c\,x : A\,x \geq b, x \in conv(X)\}$

$$\Updownarrow$$

Master LP: $\min\{\sum_i c\,x^i\,\lambda_i : \sum_i A\,x^i\,\lambda_i \underbrace{\geq}_{u} b,\ \sum_i \lambda_i \underbrace{=}_{r} 1, \lambda_i \geq 0\ \forall i\}$

$$\Updownarrow$$

Dual Master: $\max\{\underbrace{b\,u - r}_{\eta} : uA\,x^i - r \leq c\,x^i\ \forall i,\ u \geq 0\}$

$$\Updownarrow$$

$$\max\{\eta : \eta \leq \underbrace{c\,x^i + u\,(b - A\,x^i)}_{L(x^i,u)}\ \forall i,\ u \geq 0\}$$

$$\Updownarrow$$

Lagrangian D: $\max\{\theta(u) : u \geq 0\}$ with $\theta(u) := \min_i L(x^i, u)$

Standard column generation: primal view

Master Problem

$$
\begin{aligned}
\text{min} \quad & \sum_{i \in \mathbf{I}} (cx^i)\lambda_i \\
\text{s.t.} \quad & \sum_{i \in \mathbf{I}} (Ax^i)\lambda_i \geq b \\
& \sum_{i \in \mathbf{I}} \lambda_i = 1 \\
& \lambda_i \geq 0
\end{aligned}
$$

Standard column generation: primal view

Master Problem

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

Restricted Master Problem (iteration k): $\boxed{I^k \subset I}$

$$\begin{aligned} \text{min} \quad & \sum_{i \in I^k} (cx^i)\lambda_i \\ \text{s.t.} \quad & \sum_{i \in I^k} (Ax^i)\lambda_i \geq b \\ & \sum_{i \in I^k} \lambda_i = 1 \\ & \lambda_i \geq 0 \end{aligned}$$

Standard column generation: primal view

Master Problem

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

Restricted Master Problem

(iteration k): $\boxed{I^k \subset I}$

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

Standard column generation: primal view

Master Problem

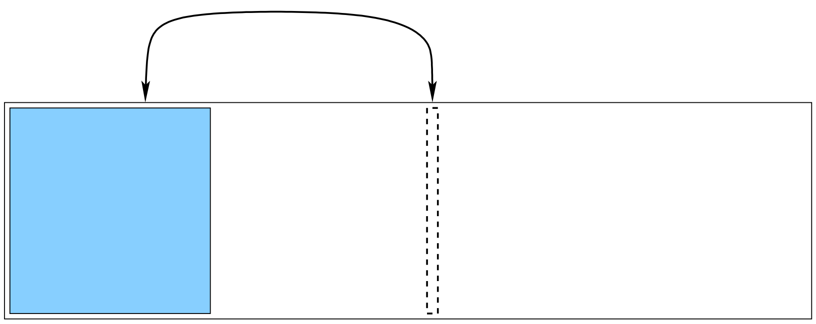
$$\begin{aligned} \min \quad & \sum_{i \in I} (cx^i)\lambda_i \\ \text{s.t.} \quad & \sum_{i \in I} (Ax^i)\lambda_i \geq b \\ & \sum_{i \in I} \lambda_i = 1 \\ & \lambda_i \geq 0 \end{aligned}$$

Restricted Master Problem

(iteration k): $\boxed{I^k \subset I}$

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

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

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

Restricted Master Problem

(iteration k): $\boxed{I^k \subset I}$

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

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: primal view

Master Problem

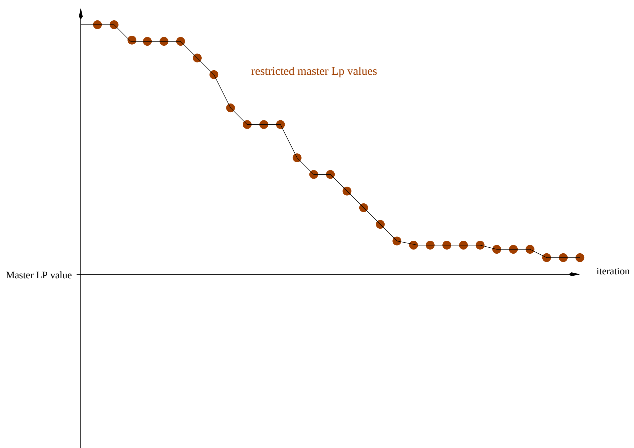
$$\begin{aligned} \min \quad & \sum_{i \in I} (cx^i)\lambda_i \\ \text{s.t.} \quad & \sum_{i \in I} (Ax^i)\lambda_i \geq b \\ & \sum_{i \in I} \lambda_i = 1 \\ & \lambda_i \geq 0 \end{aligned}$$

Restricted Master Problem (iteration k): $\boxed{I^k \subset I}$

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

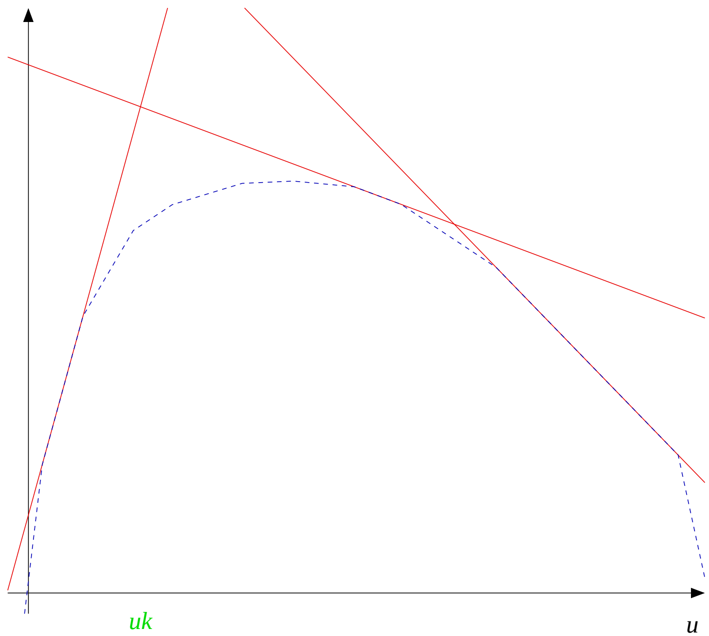
Subproblem (Oracle) : $\min_{i \in I} (c - u^k A)x^i = \min_{x \in X} (c - u^k A)x$

If $((c - u^k A)x^{k+1} - r^k) < 0$, add $k+1$ to I^k. **Otherwise**, STOP.



Standard column generation: dual view

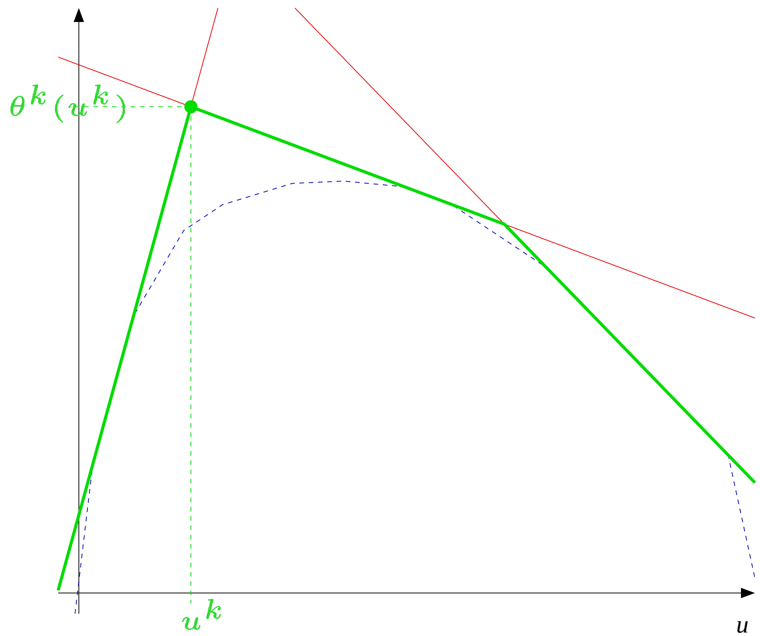
Restricted Master Dual: $\max\{\eta : \eta \leq \underbrace{cx^i + u(b - A\,x^i)}_{L(x^i,u)} \forall i \in I^k,\ u \geq 0\}$



Standard column generation: dual view

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

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

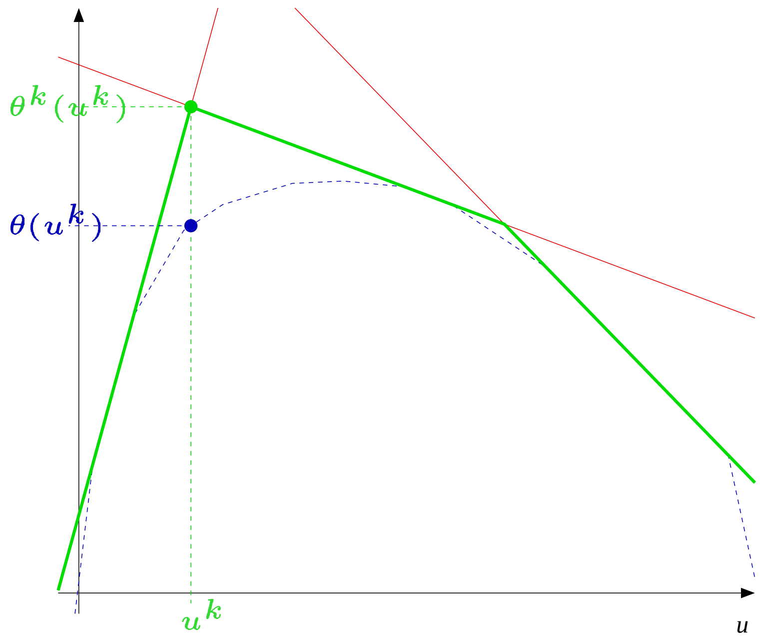


Standard column generation: dual view

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

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

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

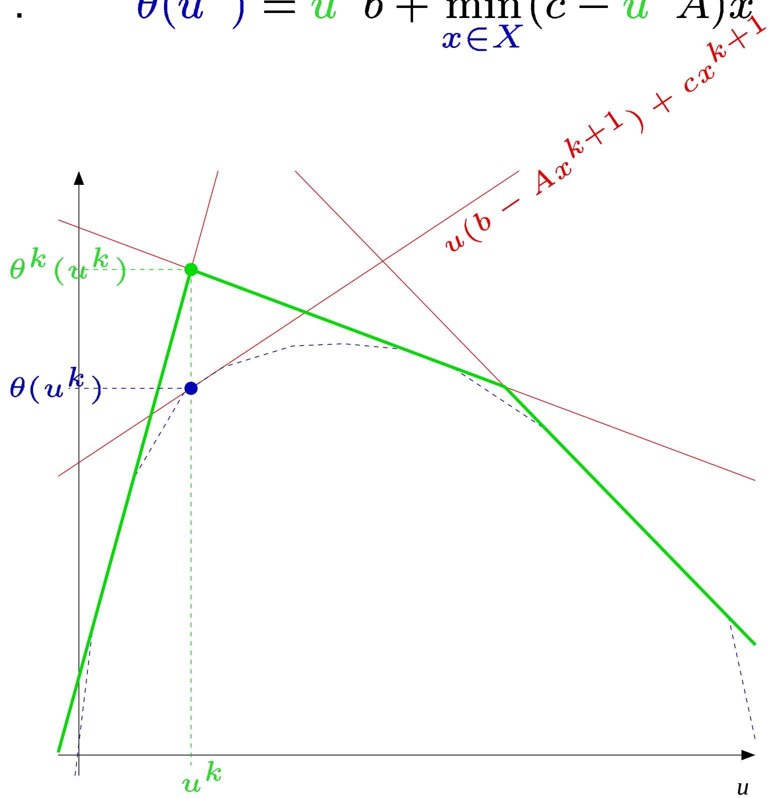


Standard column generation: dual view

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

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

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

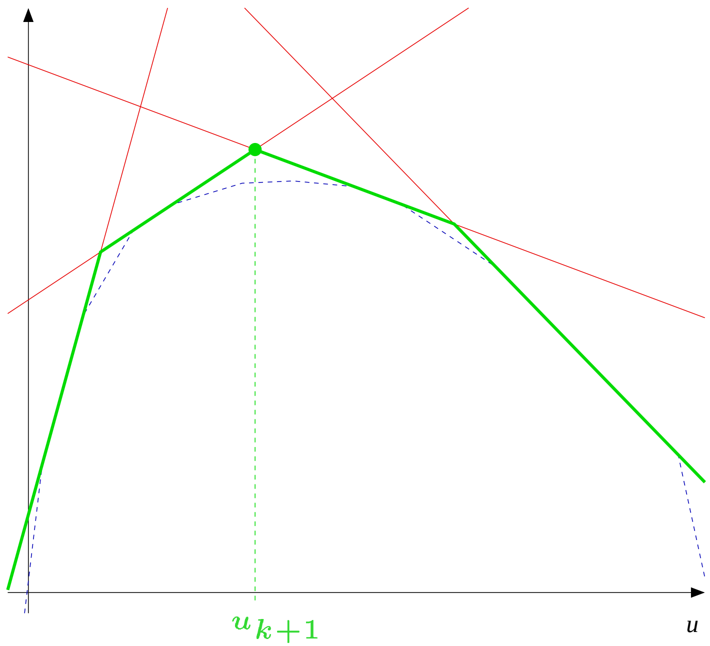


Standard column generation: dual view

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

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

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

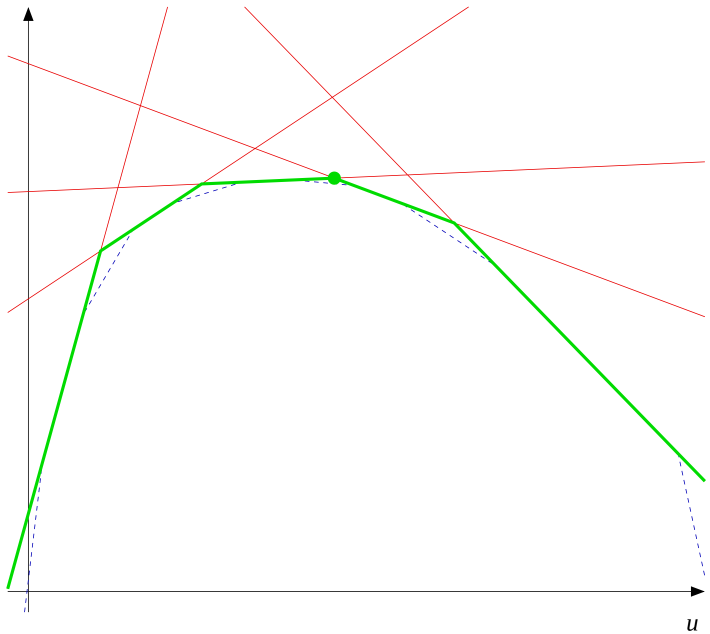


Standard column generation: dual view

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

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

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

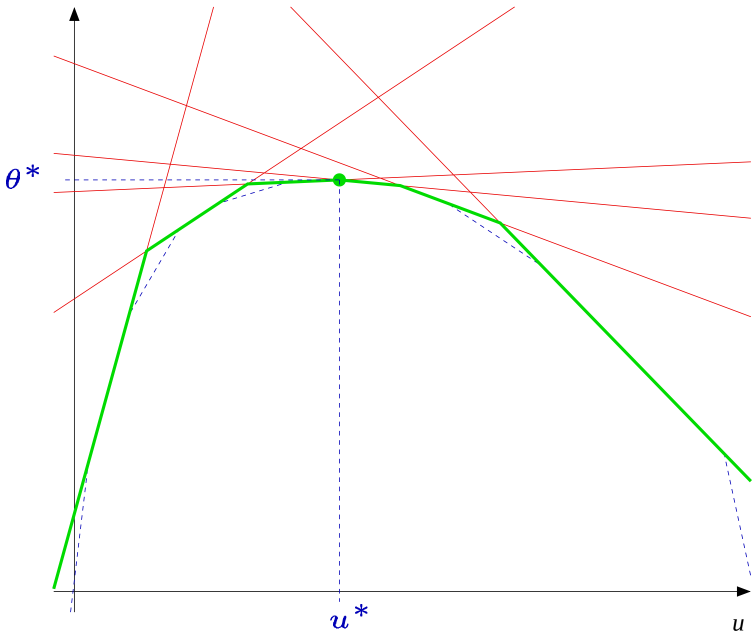


Standard column generation: dual view

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

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

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



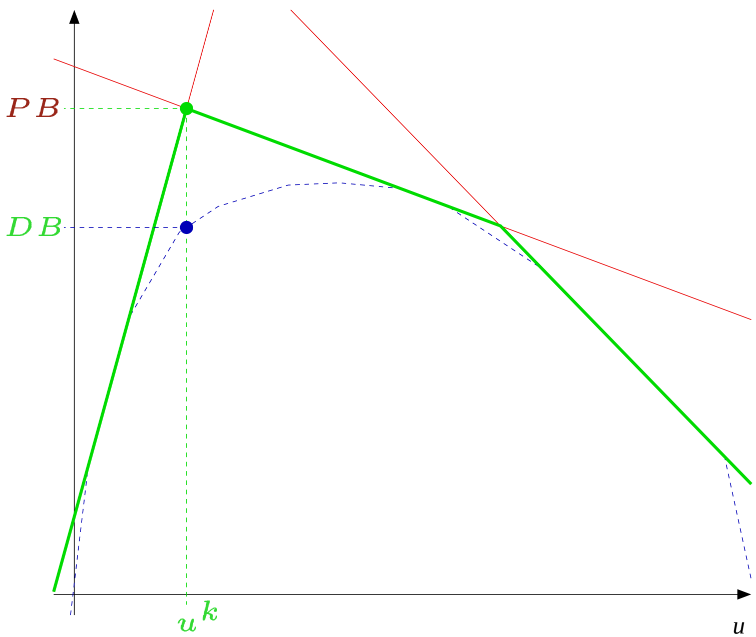
Kelley's cutting plane algorithm

Standard column generation: primal/ dual view

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: $DB \equiv \theta(u^k) = PB - (c - u^k A)x^{k+1}$

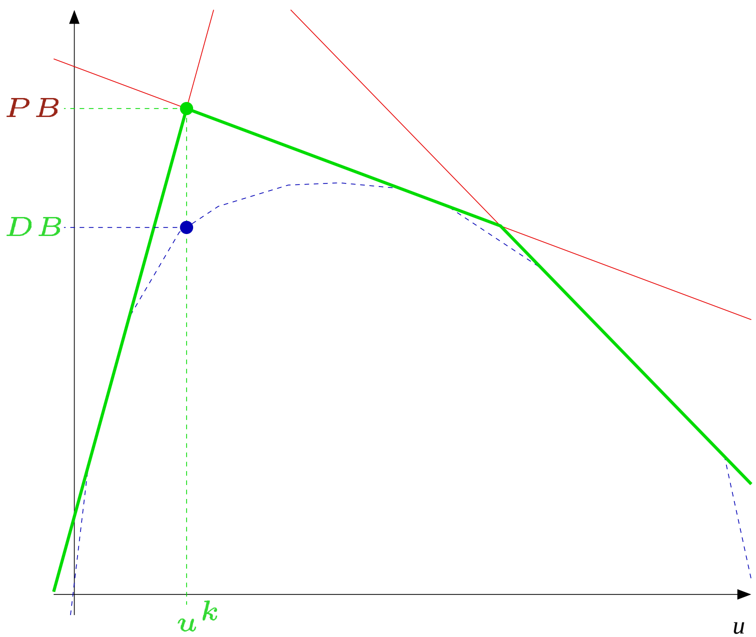


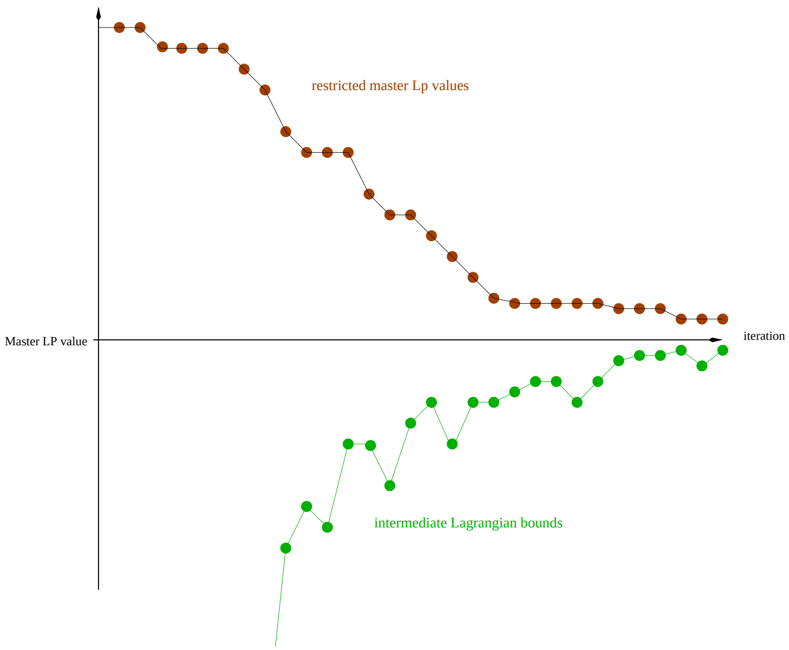
Standard column generation: primal/ dual view

At iteration k, one has

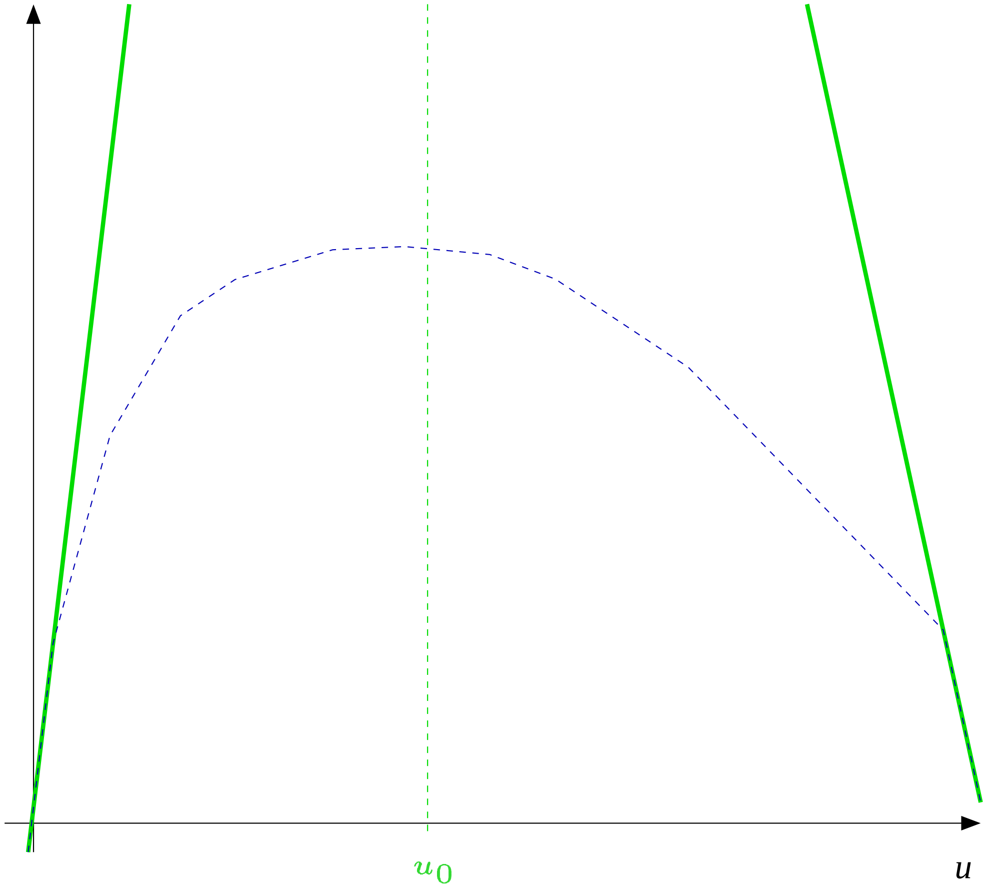
A Primal Bound on the master: $PB \equiv \theta^k(u^k) = \sum_{i \in I^k} cx^i \lambda_i$

A Dual Bound on the master: $DB \equiv \theta(u^k) = PB - (c - u^k A)x^{k+1}$

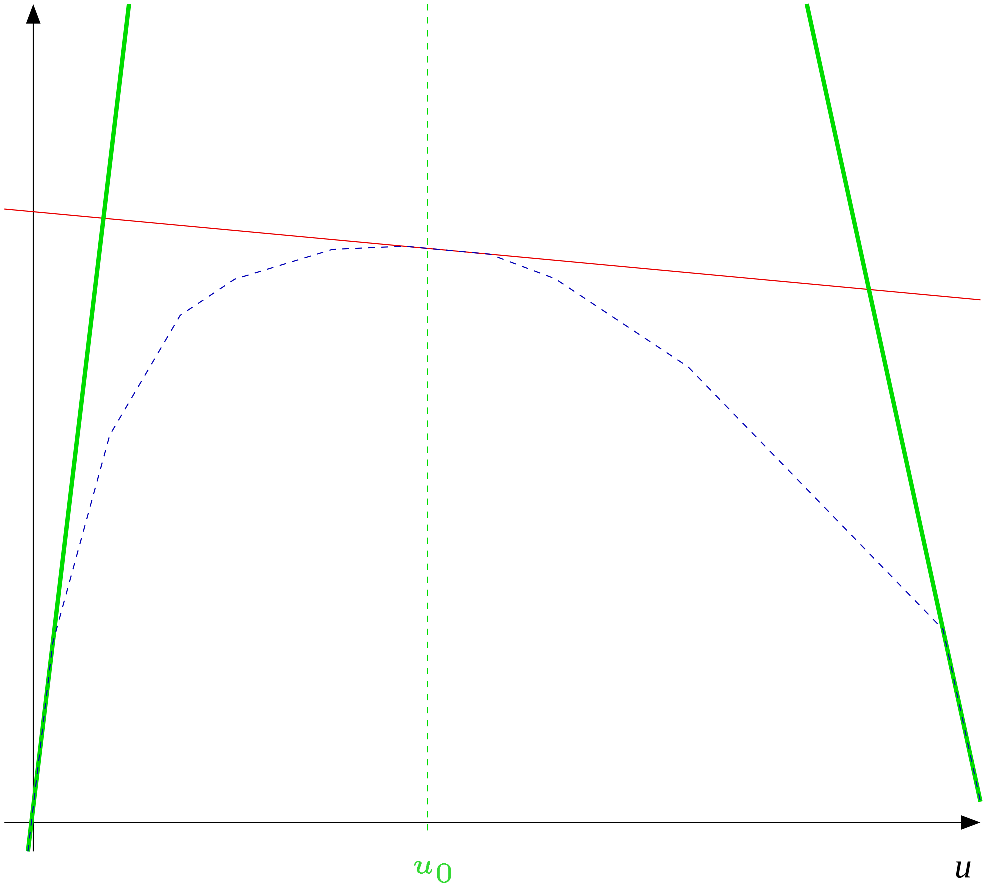




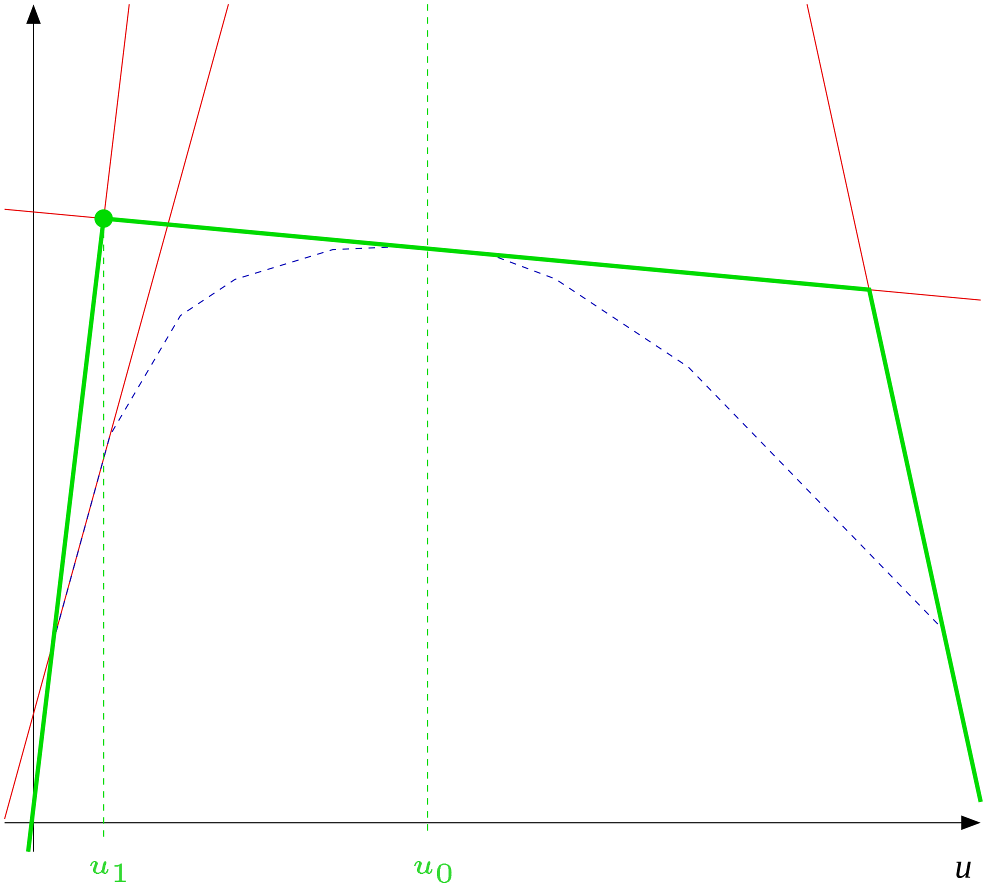
Slow convergence of Kelley's algorithm



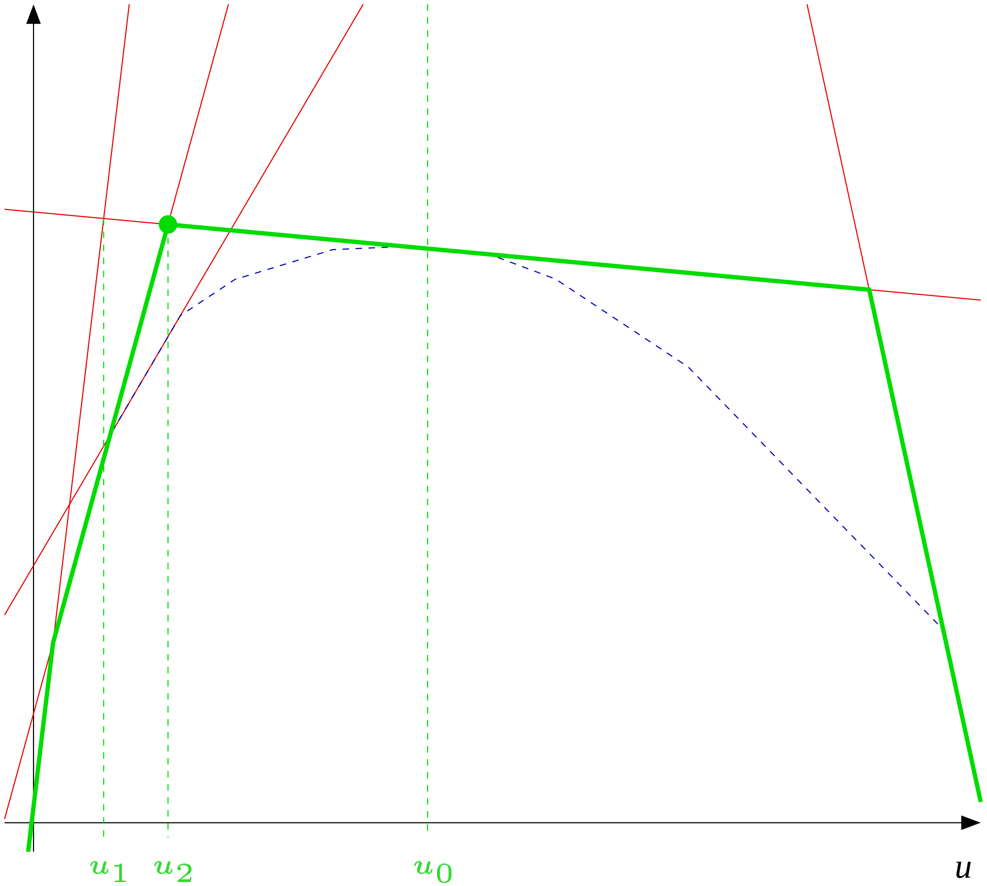
Slow convergence of Kelley's algorithm



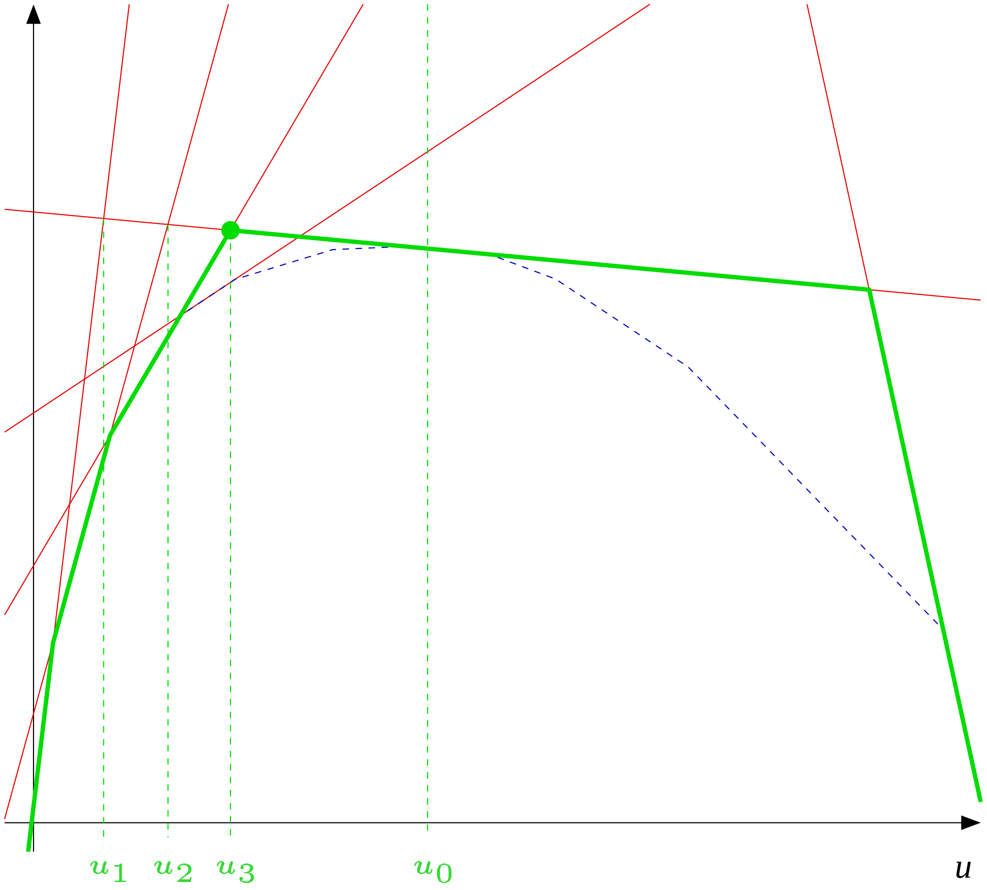
Slow convergence of Kelley's algorithm



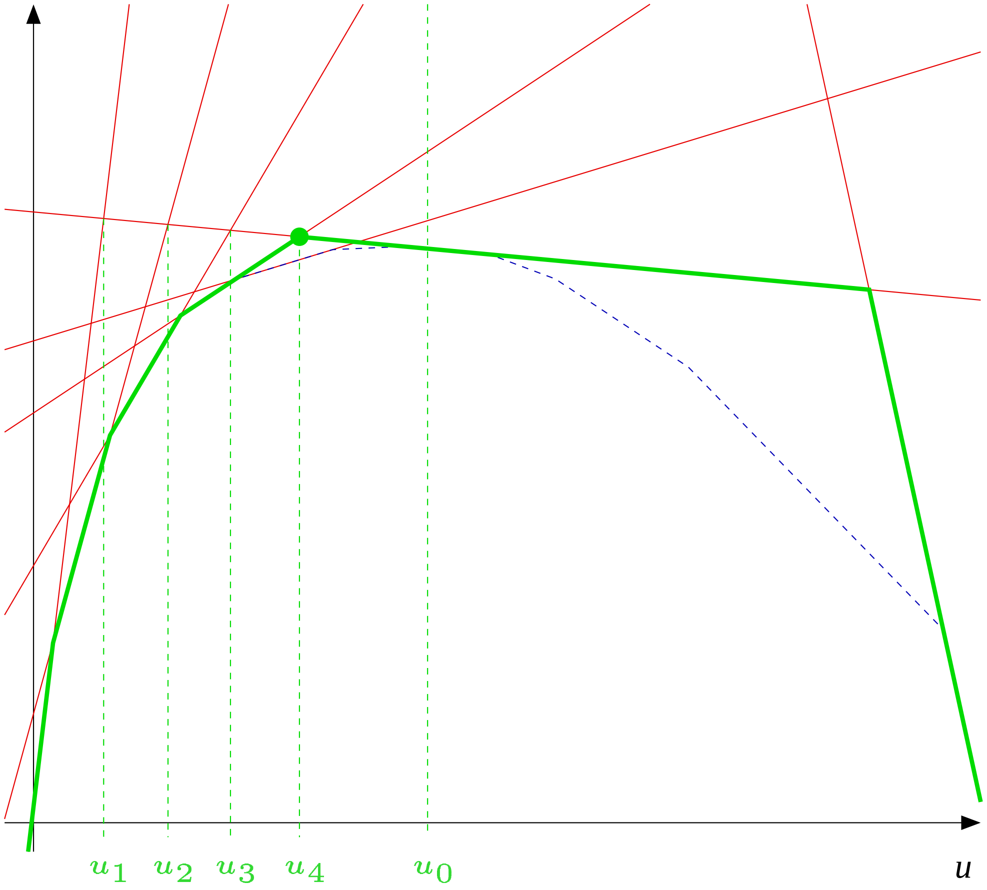
Slow convergence of Kelley's algorithm



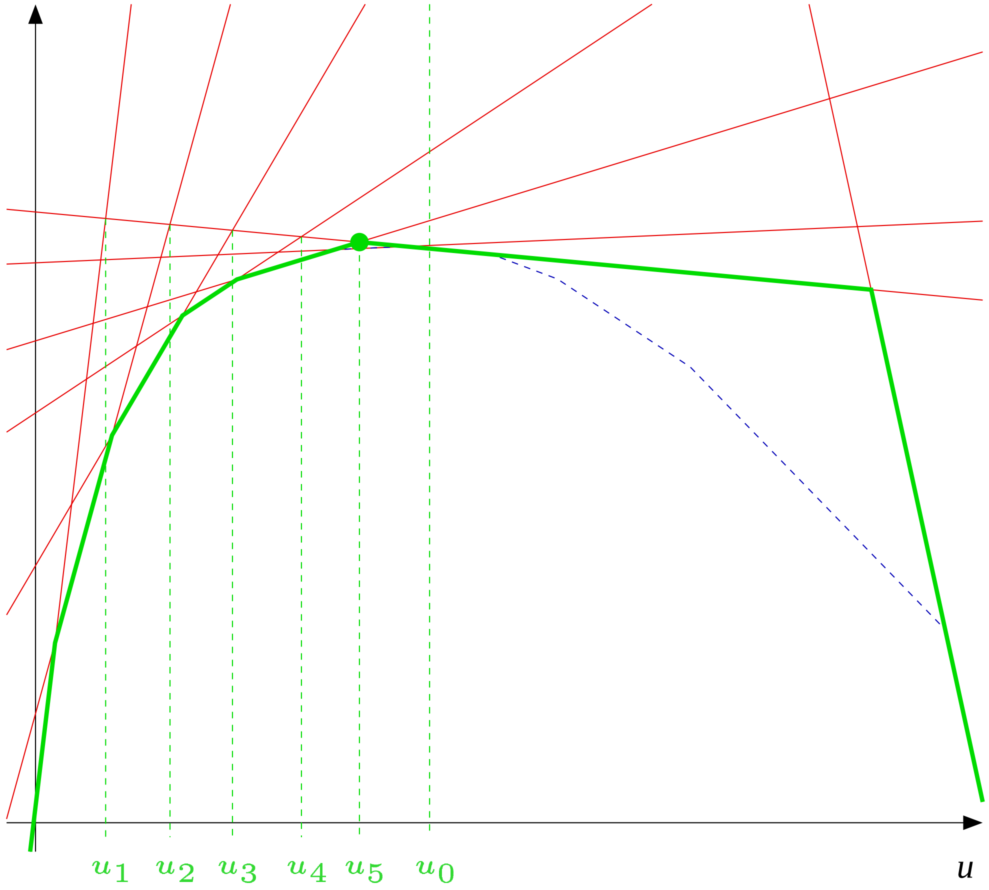
Slow convergence of Kelley's algorithm



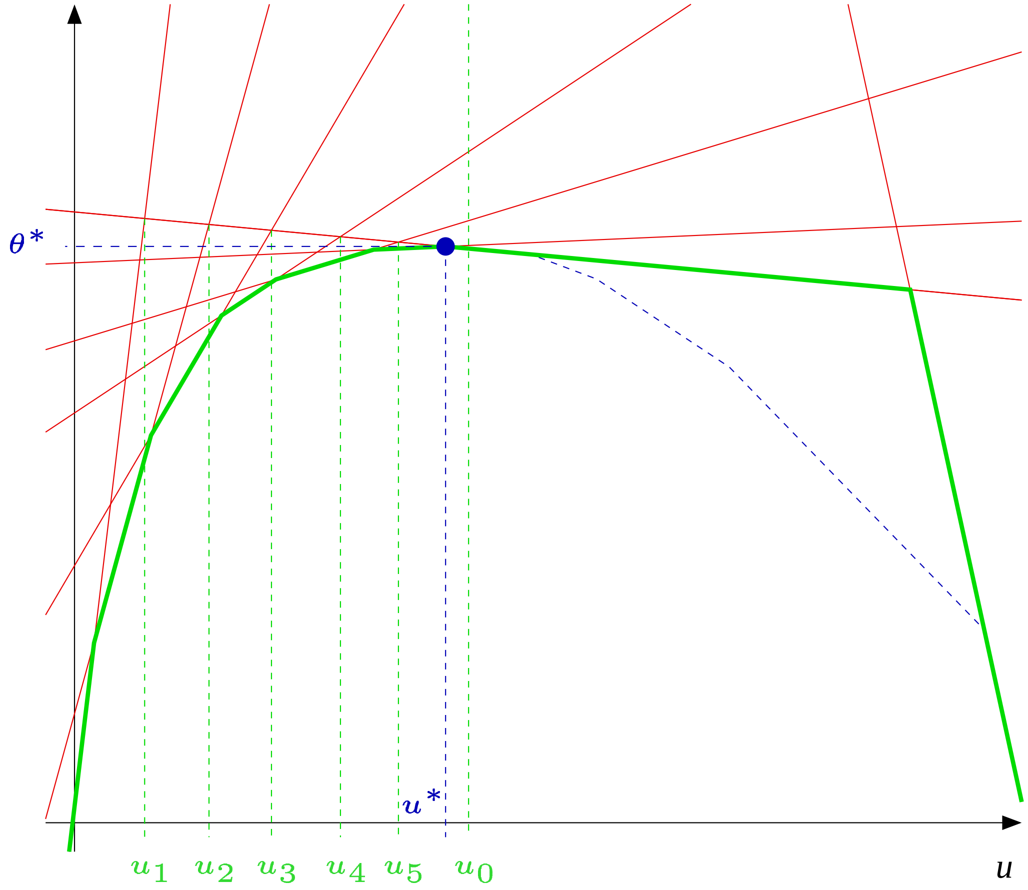
Slow convergence of Kelley's algorithm



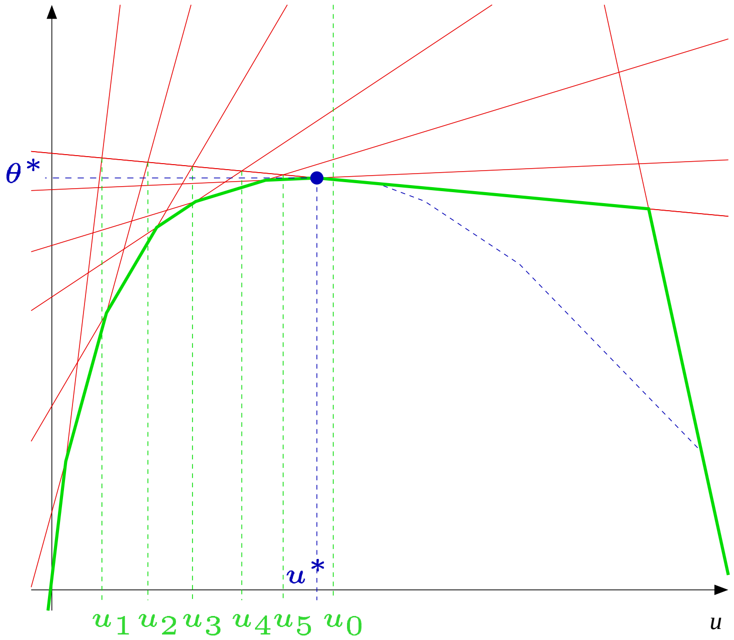
Slow convergence of Kelley's algorithm

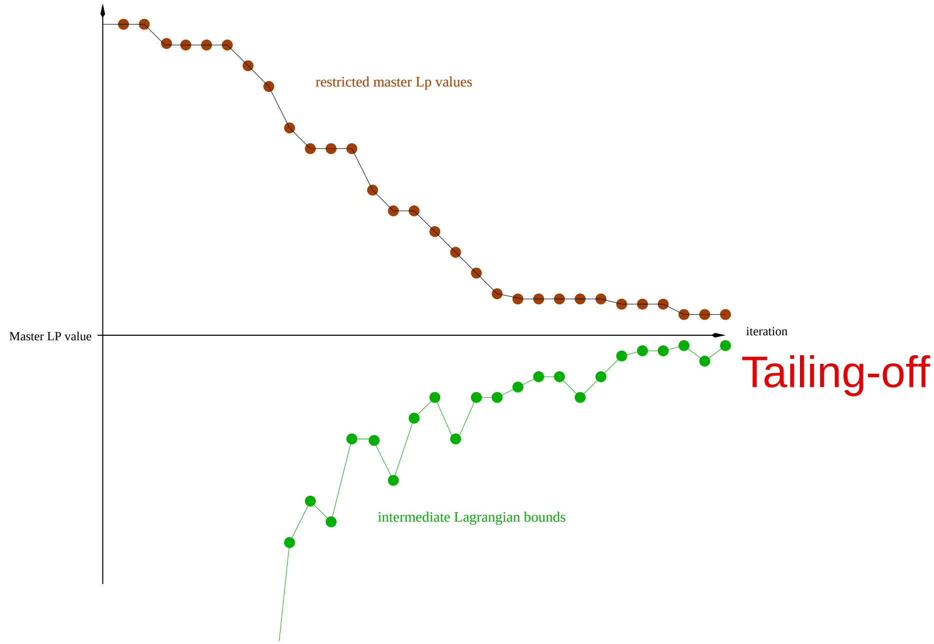


Slow convergence of Kelley's algorithm

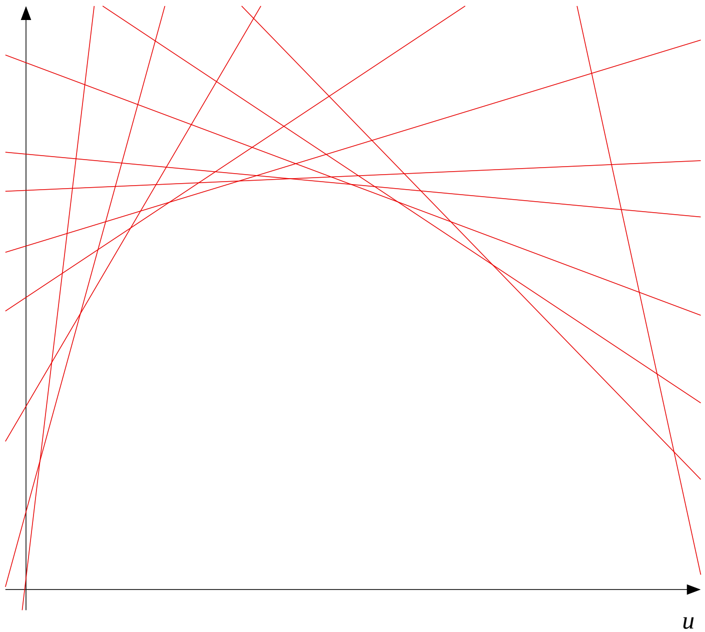


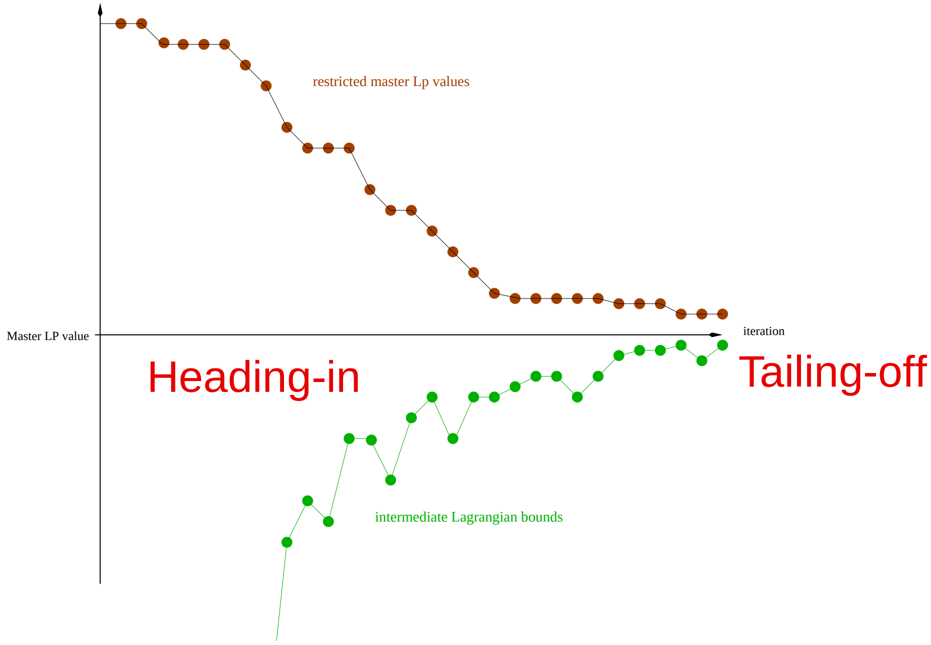
Instability: observed behaviors



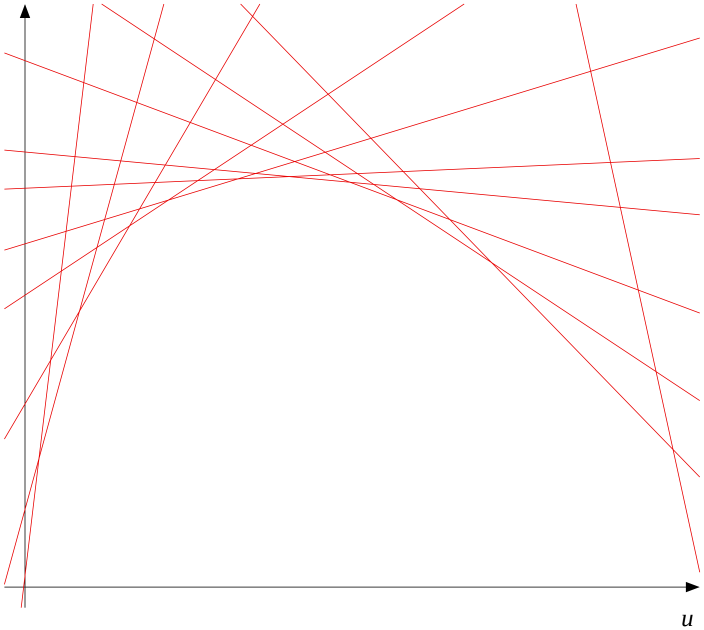


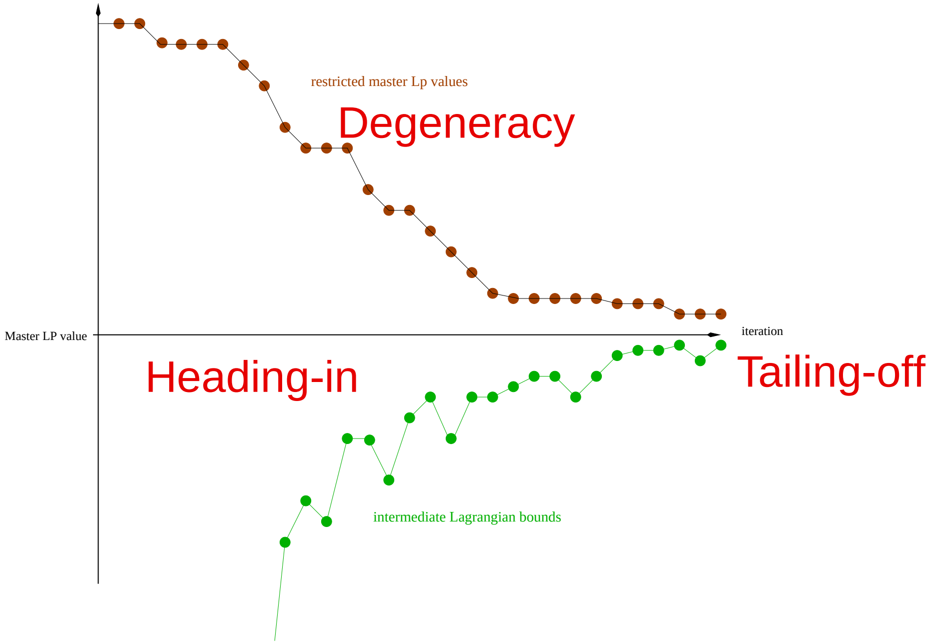
Instability: observed behaviors



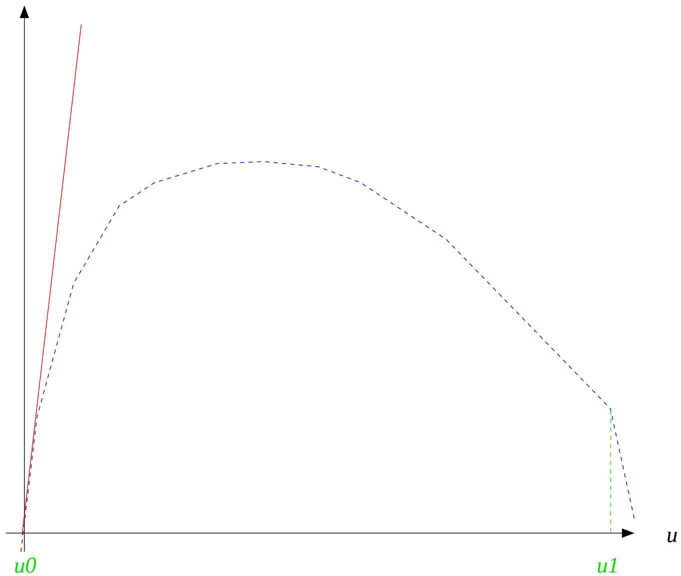


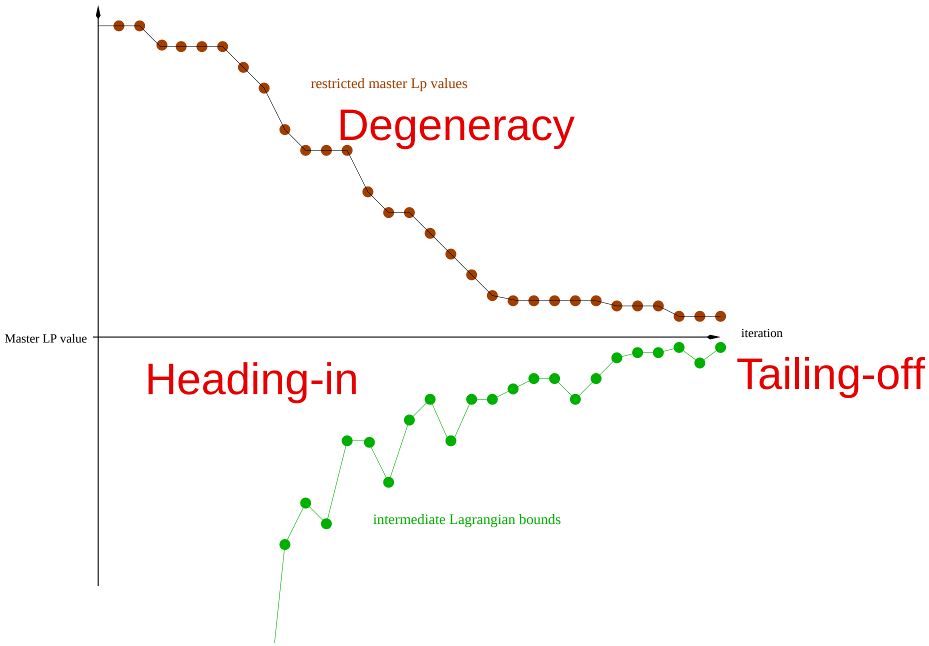
Instability: observed behaviors





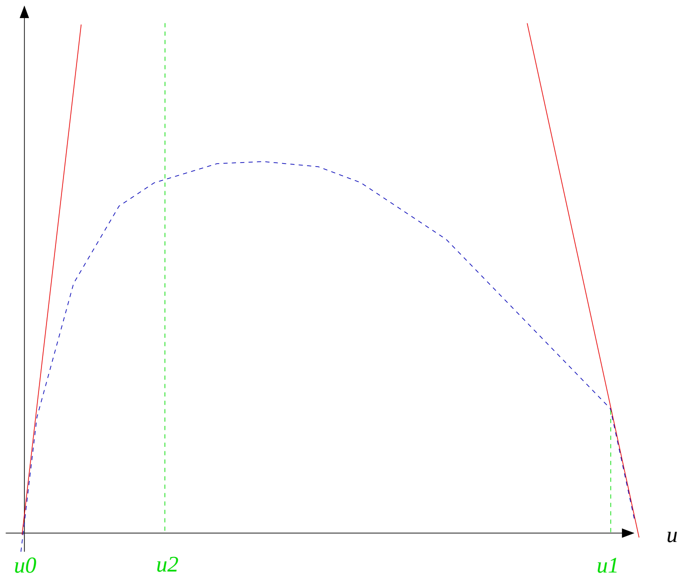
Instability: observed behaviors



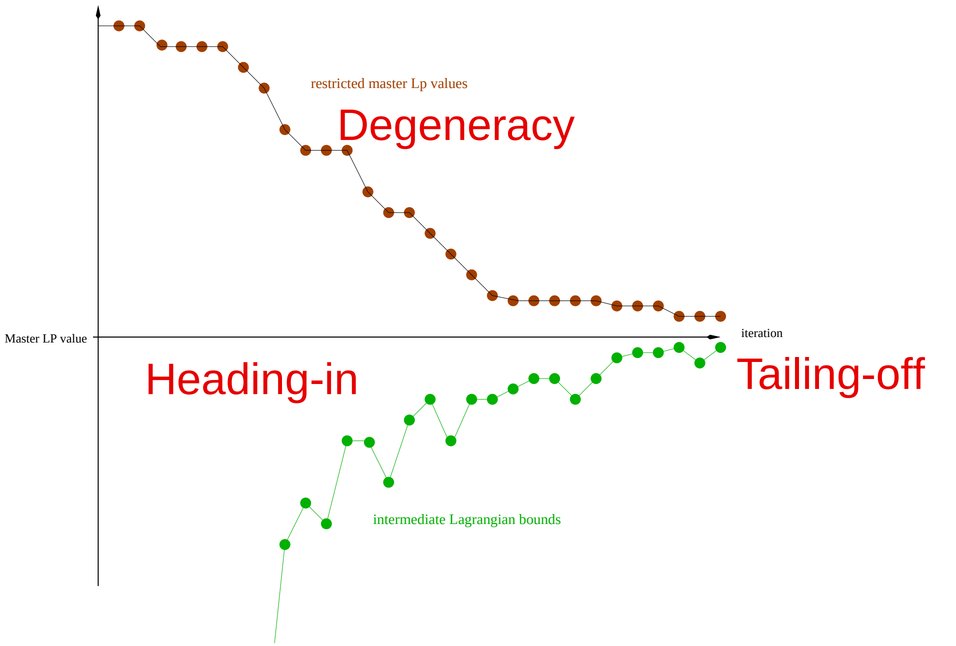


Bang-Bang Behavior

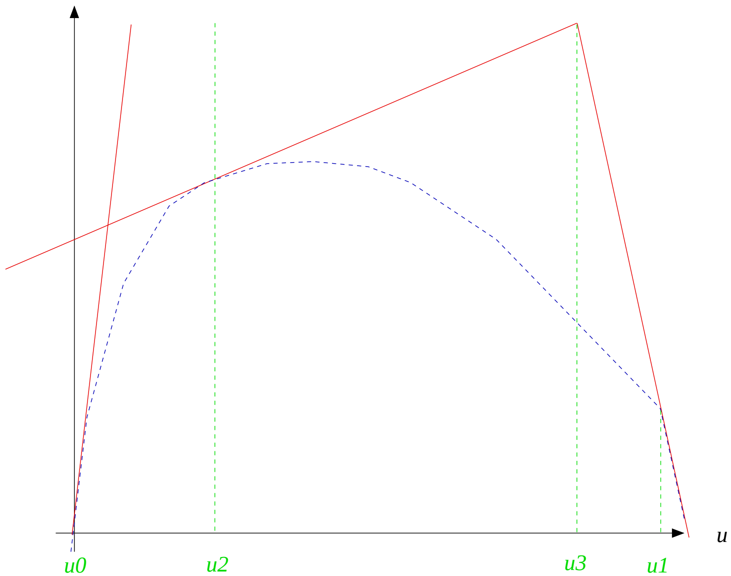
Instability: observed behaviors



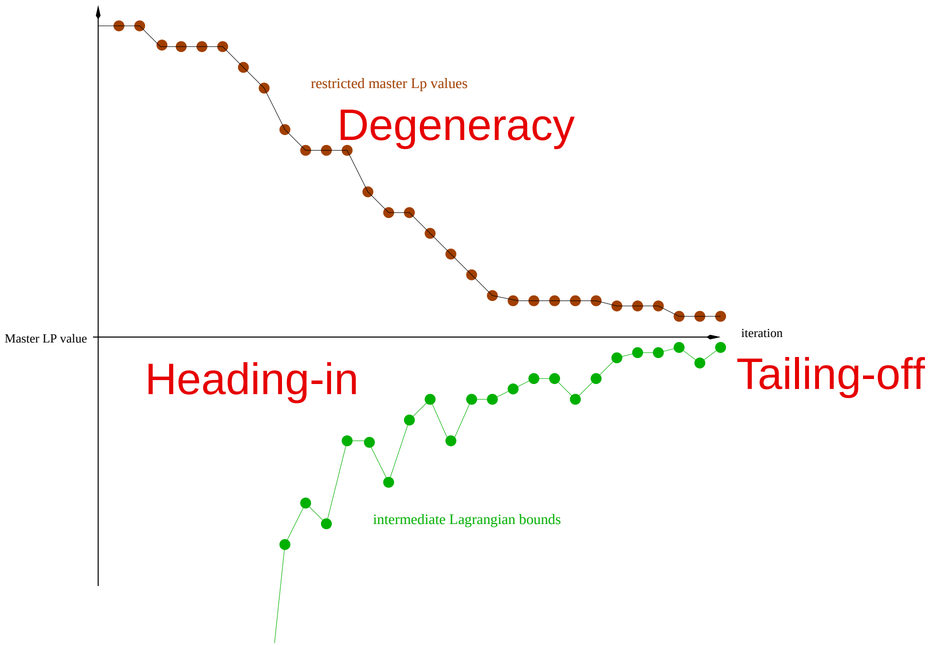
Bang-Bang Behavior



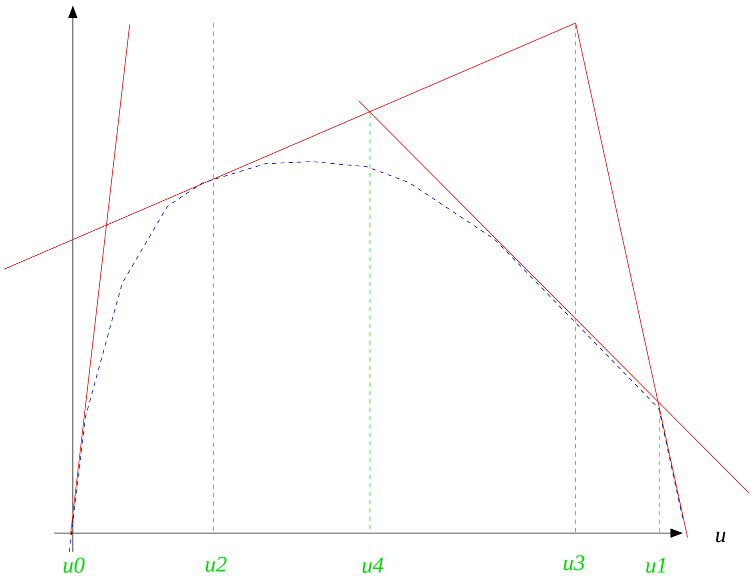
Instability: observed behaviors



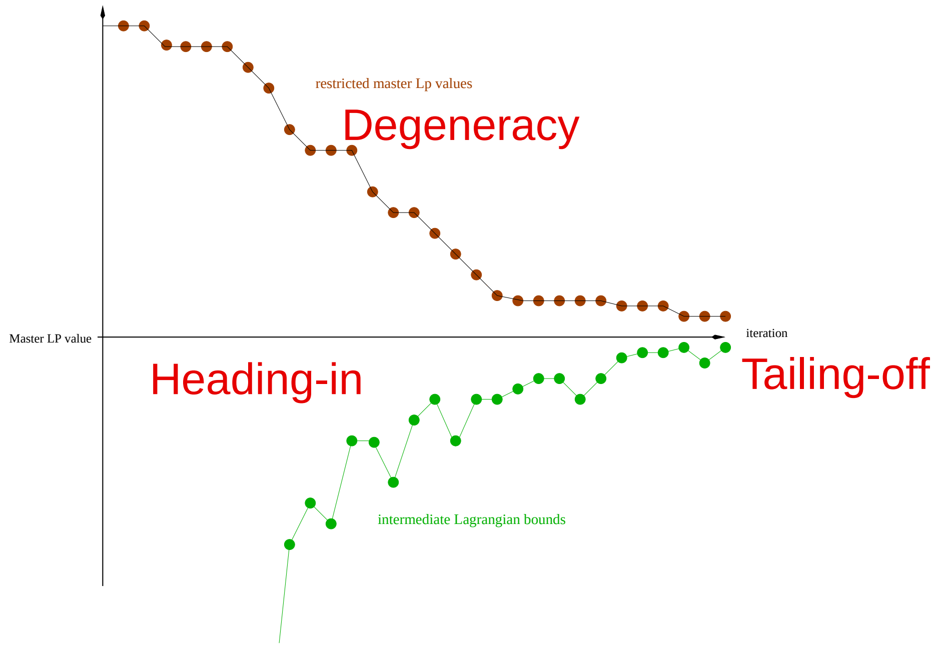
Bang-Bang Behavior



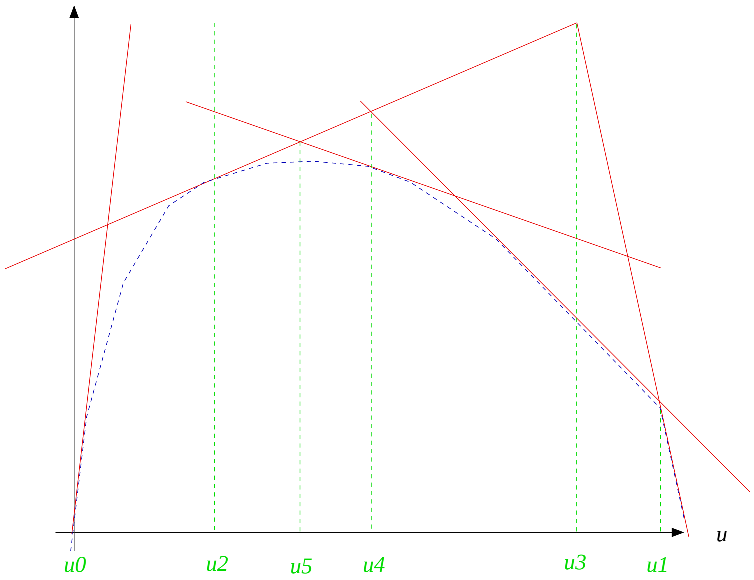
Instability: observed behaviors



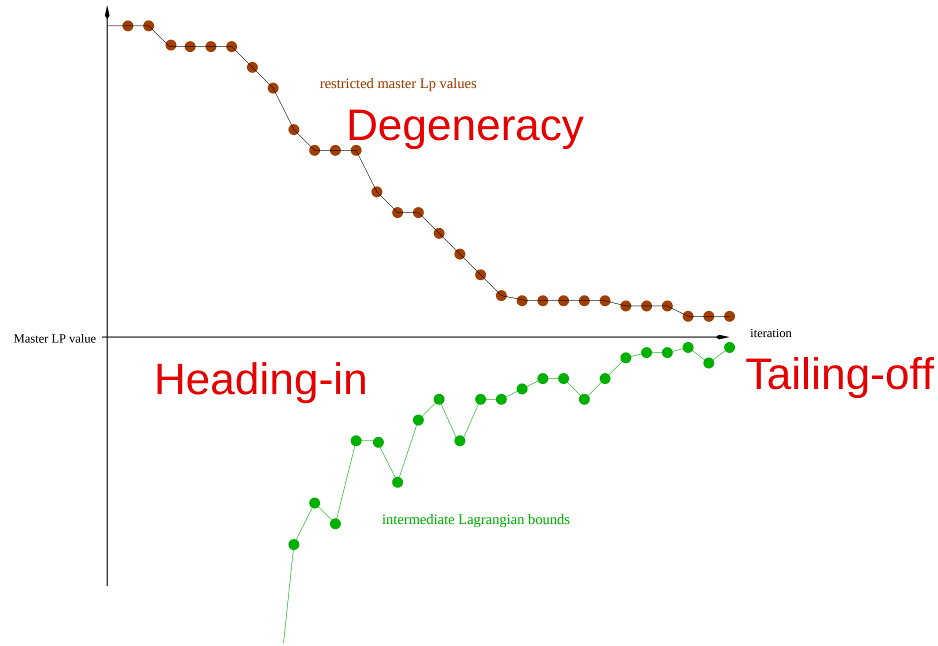
Bang-Bang Behavior



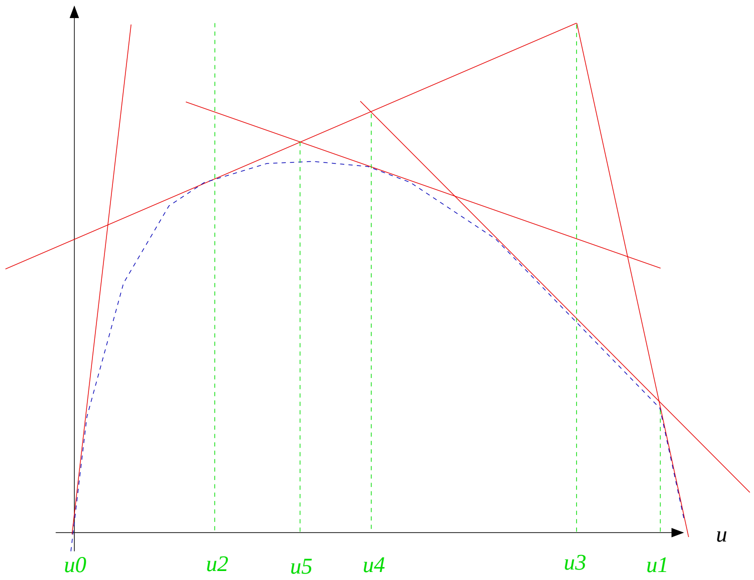
Instability: observed behaviors



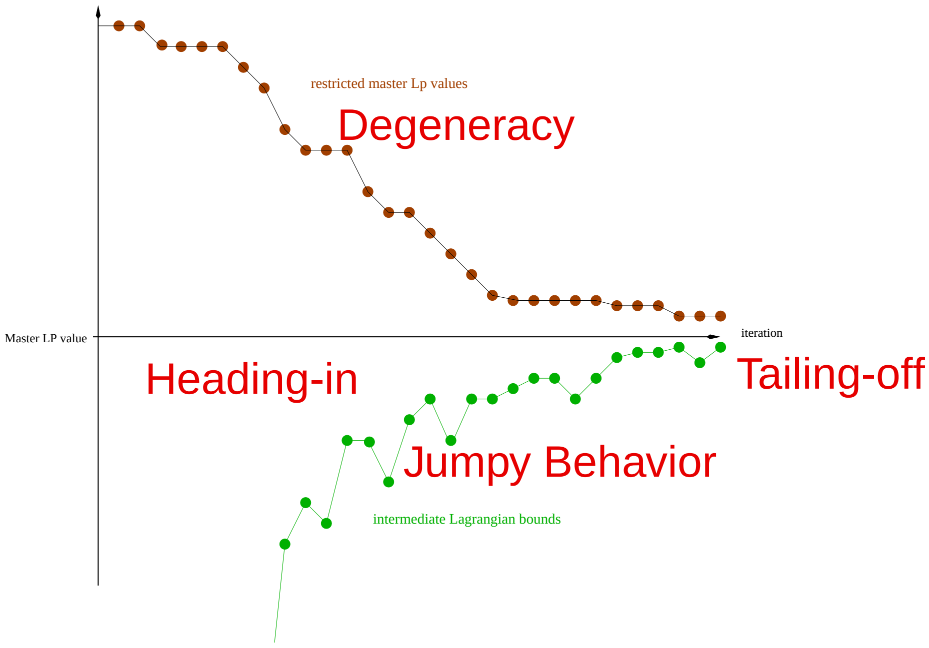
Bang-Bang Behavior



Instability: observed behaviors



Bang-Bang Behavior



Stabilization Methods -

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

$$\hat{u} \rightarrow \text{ best dual bound}$$

Stabilization Methods - $\tilde{\theta}^k(u) = \theta^k(u) - S(u - \hat{u})$

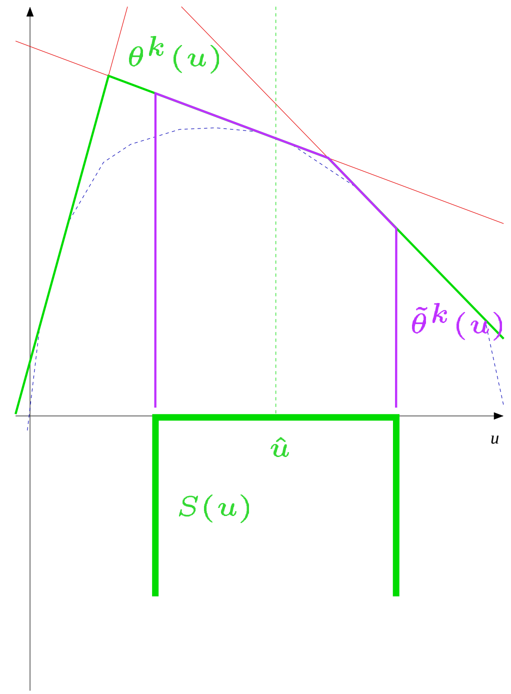
- penalizing distance from a stability center $\hat{u}$

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

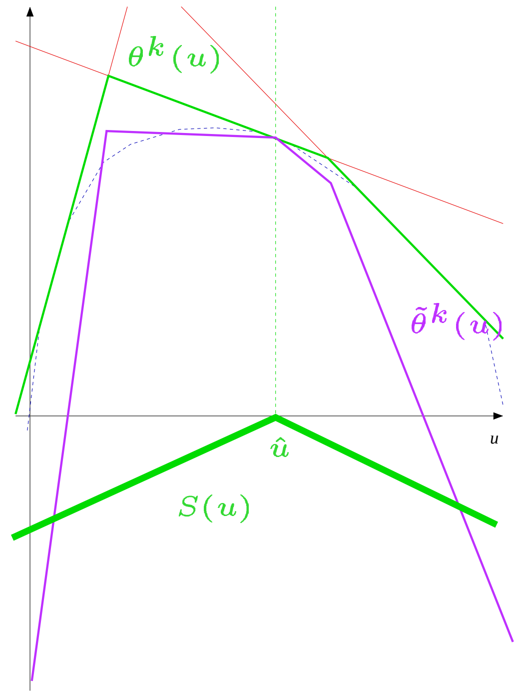
- penalizing distance from a stability center $\hat{u}$

  $S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon)$ (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



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)

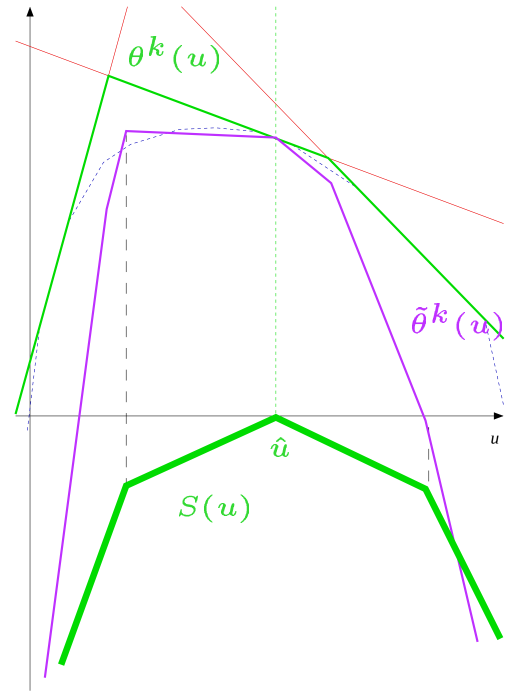
  R.E.Martsen, W.W.Hogan and J.W.Blankenship, 1975

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

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

  $S(u - \hat{u}) = ||u - \hat{u}||_1 +$ 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 $\hat{u}$

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

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

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

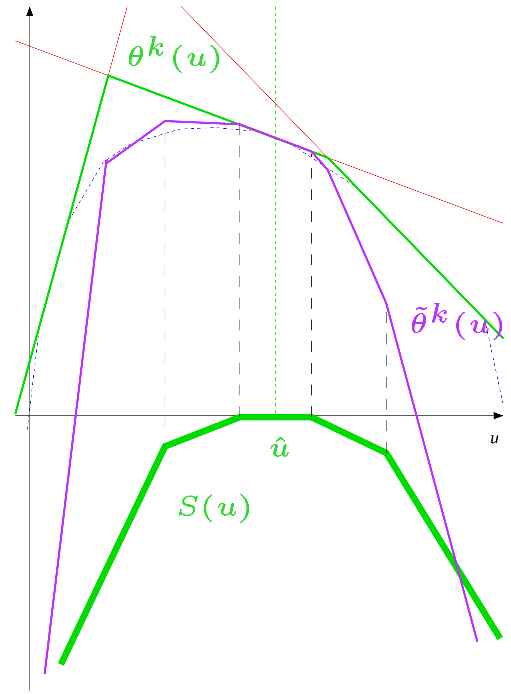
  S.Kim, K.N.Chang and J.Y.Lee, 1995

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

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

  $S(u - \hat{u})$ = 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)

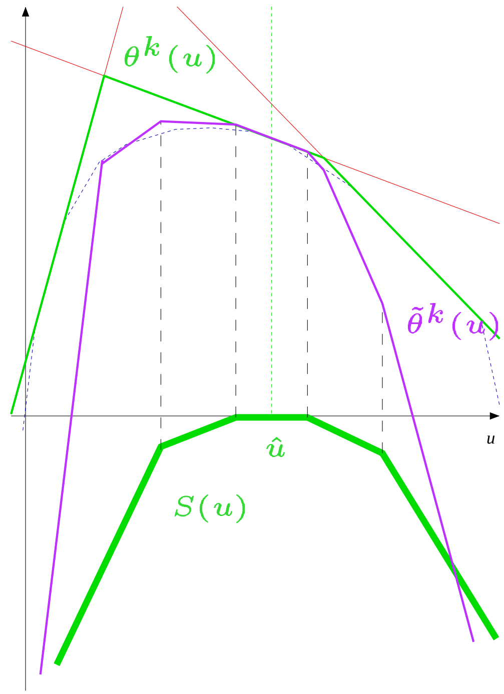
  S.Kim, K.N.Chang and J.Y.Lee, 1995

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

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

  $S(u - \hat{u})$ = 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)

  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

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

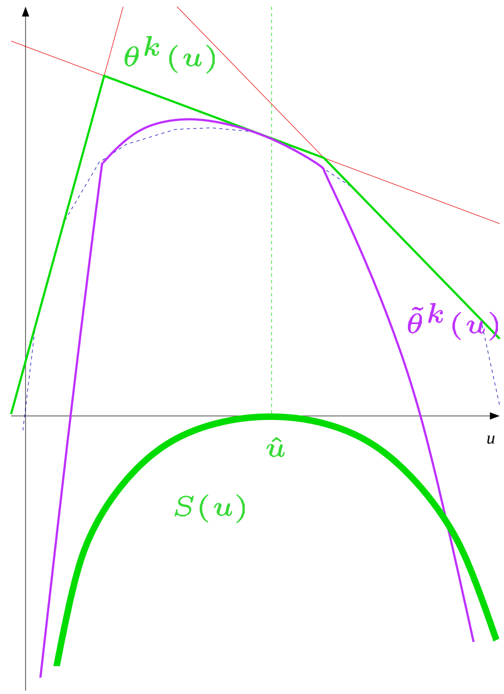
  O.du Merle, D.Villeneuve, J.Desrosiers and P.Hansen, 1999

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

  H.Ben-Amor and J.Desrosiers, 2003

  $S(u - \hat{u}) = \frac{1}{2t}||u - \hat{u}||^2 \Rightarrow$ **Bundle**

  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 $\hat{u}$

  $S(u - \hat{u}) = \delta(||u - \hat{u}||_\infty \leq \epsilon)$ (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

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

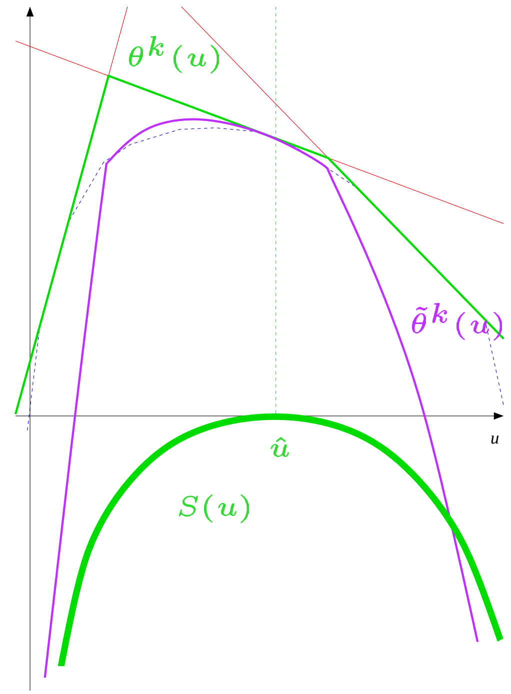
  O.du Merle, D.Villeneuve, J.Desrosiers and P.Hansen, 1999

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

  H.Ben-Amor and J.Desrosiers, 2003

  $S(u - \hat{u}) = \frac{1}{2t}||u - \hat{u}||^2 \Rightarrow$ **Bundle**

  C. Lemaréchal and C. Sagastizábal, 1997



$\Rightarrow$ As $S(u - \hat{u})$ is a quadratic concave function,
we must now use Quadratic solvers *K.C.Kiwiel, 1989, 1994, A.Frangioni, 1996*

Stabilization Methods

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

  P.Wentges, 1997

Stabilization Methods

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

  P.Wentges, 1997

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

Stabilization Methods

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

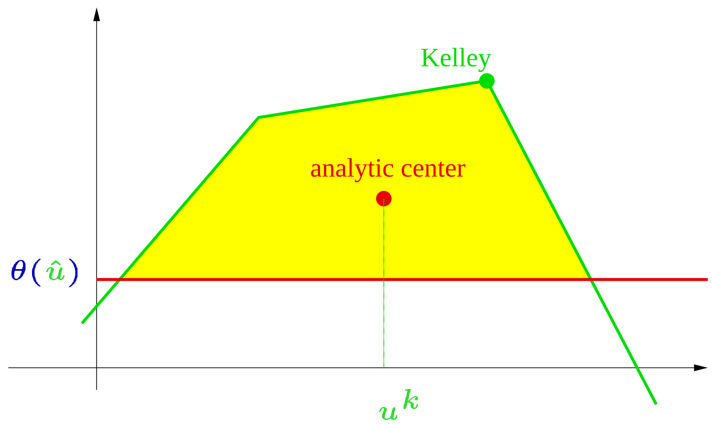
  P.Wentges, 1997

- 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)

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

J.L. Goffin, A.Haurie and J.Ph. Vial, 1992



Bi-Dual: Fenchel duality

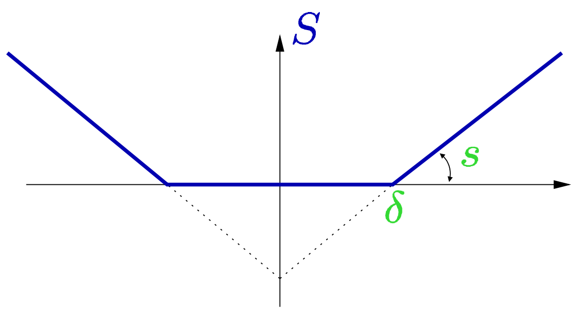
$$
\begin{array}{lcl}
\max \theta^{k}(u) & \equiv & \min\{c\,x : \; A\,x \; \geq b, x \in conv(X^{k})\} \\
\max \theta^{k}(u) - S(u - \hat{u}) & \equiv & \qquad \qquad \ldots
\end{array}
$$

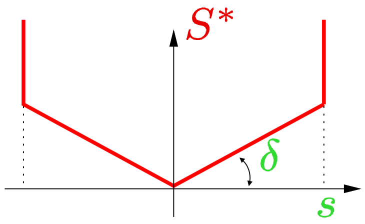
Bi-Dual: Fenchel duality

$$
\begin{aligned}
\max \theta^k(u) &\equiv \min\{c\,x : \ A\,x \ \geq b, x \in conv(X^k)\} \\
\max \theta^k(u) - S(u - \hat{u}) &\equiv \min\{c\,x + \hat{u}\,g + S^*(g) : \\
& \quad A\,x \ + g \geq b, \ x \in conv(X^k), g \in I\!R^m\}
\end{aligned}
$$

Bi-Dual: Fenchel duality

$$\begin{array}{lcl}\max \theta^k(u) & \equiv & \min\{c\,x : \ A\,x \ \geq b, x \in conv(X^k)\} \\ \max \theta^k(u) - S(u-\hat{u}) & \equiv & \min\{c\,x + \hat{u}\,g + S^*(g) : \\ & & A\,x \ + g \geq b, \ x \in conv(X^k), g \in I\!R^m\}\end{array}$$





$$\min \sum_{i=1}^{k} cx^i\lambda_i + \sum_{j=1}^{m}(\delta + \hat{u}_j)g_j^+ + \sum_{j=1}^{m}(\delta - \hat{u}_j)g_j^-$$

$$\begin{array}{rcll}\sum_{i=1}^{k} Ax^i\lambda_i + g_j^+ - g_j^- & \geqslant & b_j & j = 1,\ldots,m\,, \\ s \geqslant g_j^+,\ g_j^- & \geqslant & 0 & j = 1,\ldots,m\,, \\ \sum_{i=1}^{k}\lambda_i = 1,\ \ \lambda_i & \geqslant & 0 & i\,.\end{array}$$

PART 3: Numerical Comparison

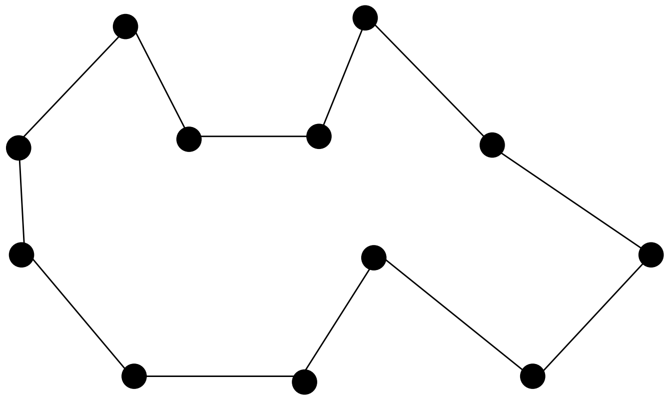
Tested variants

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)

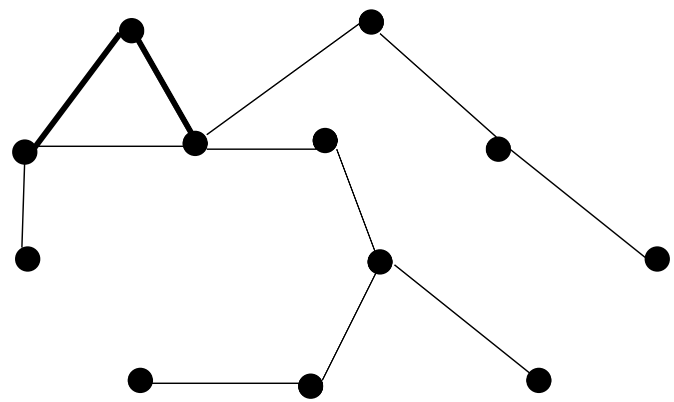
Traveling Salesman Problems (TSP)



$$\equiv$$

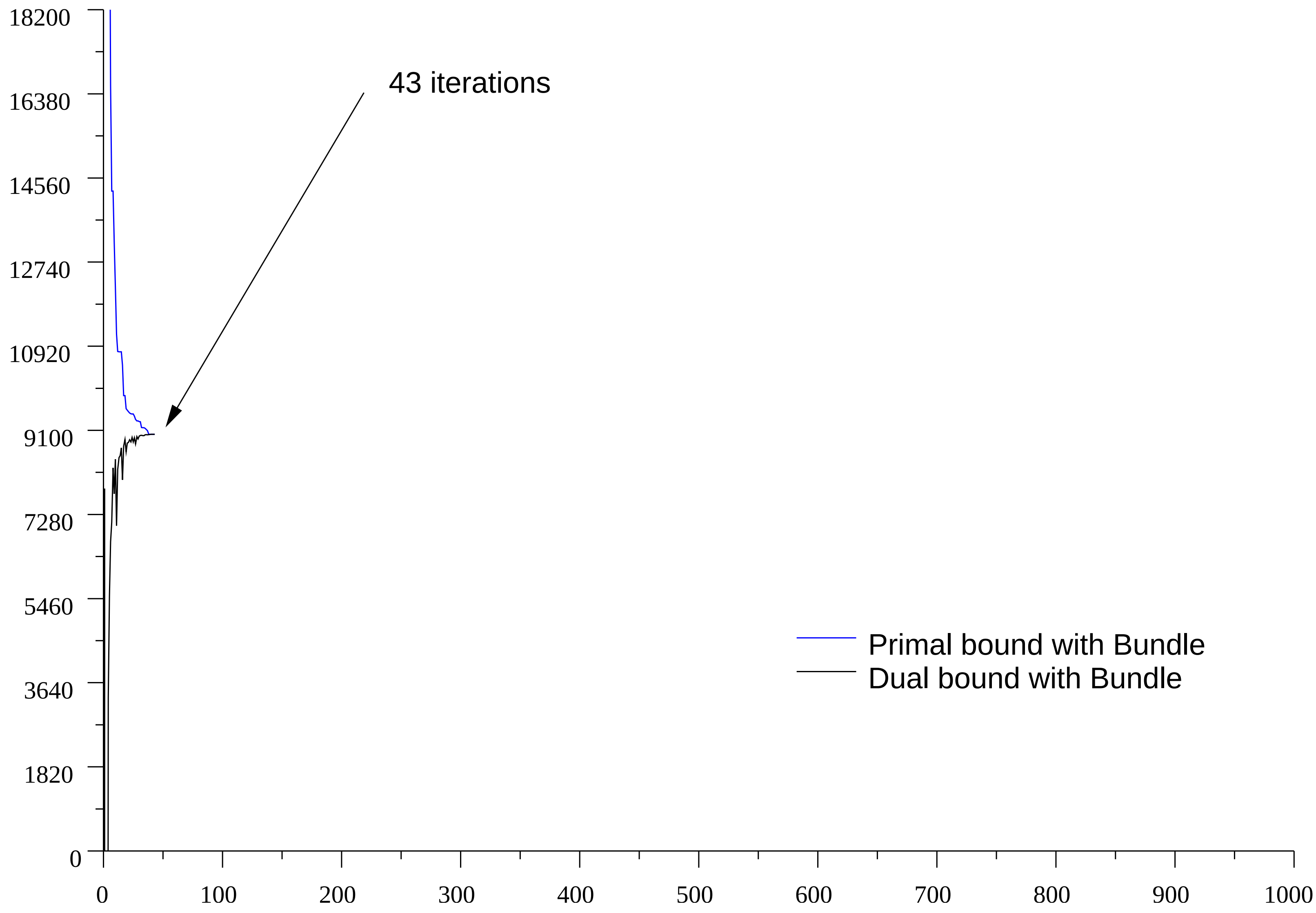
$$\sum_{e\in\delta(i)} x_e = 2 \quad \forall i$$

$$+$$

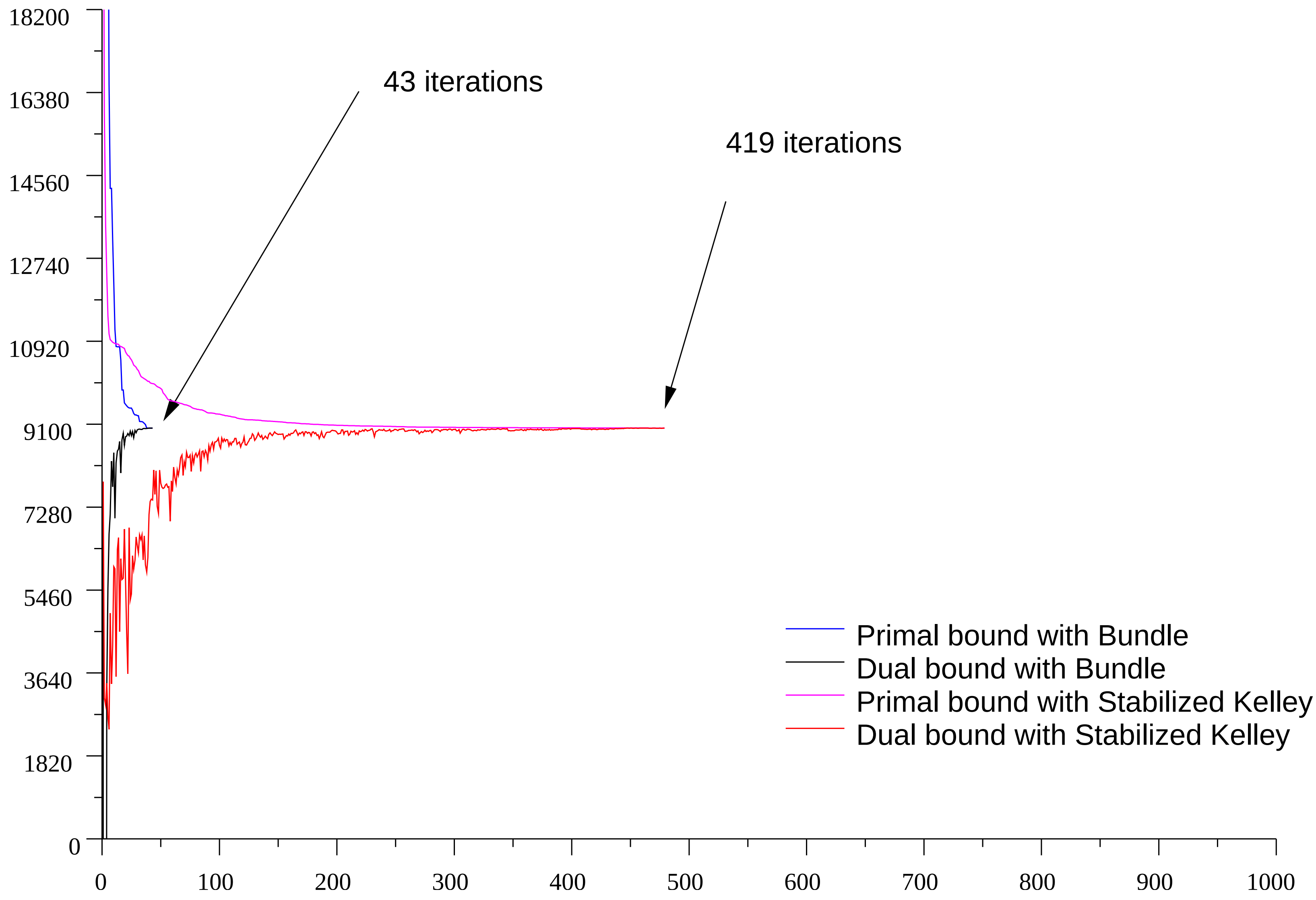


Instances from TSPLIB

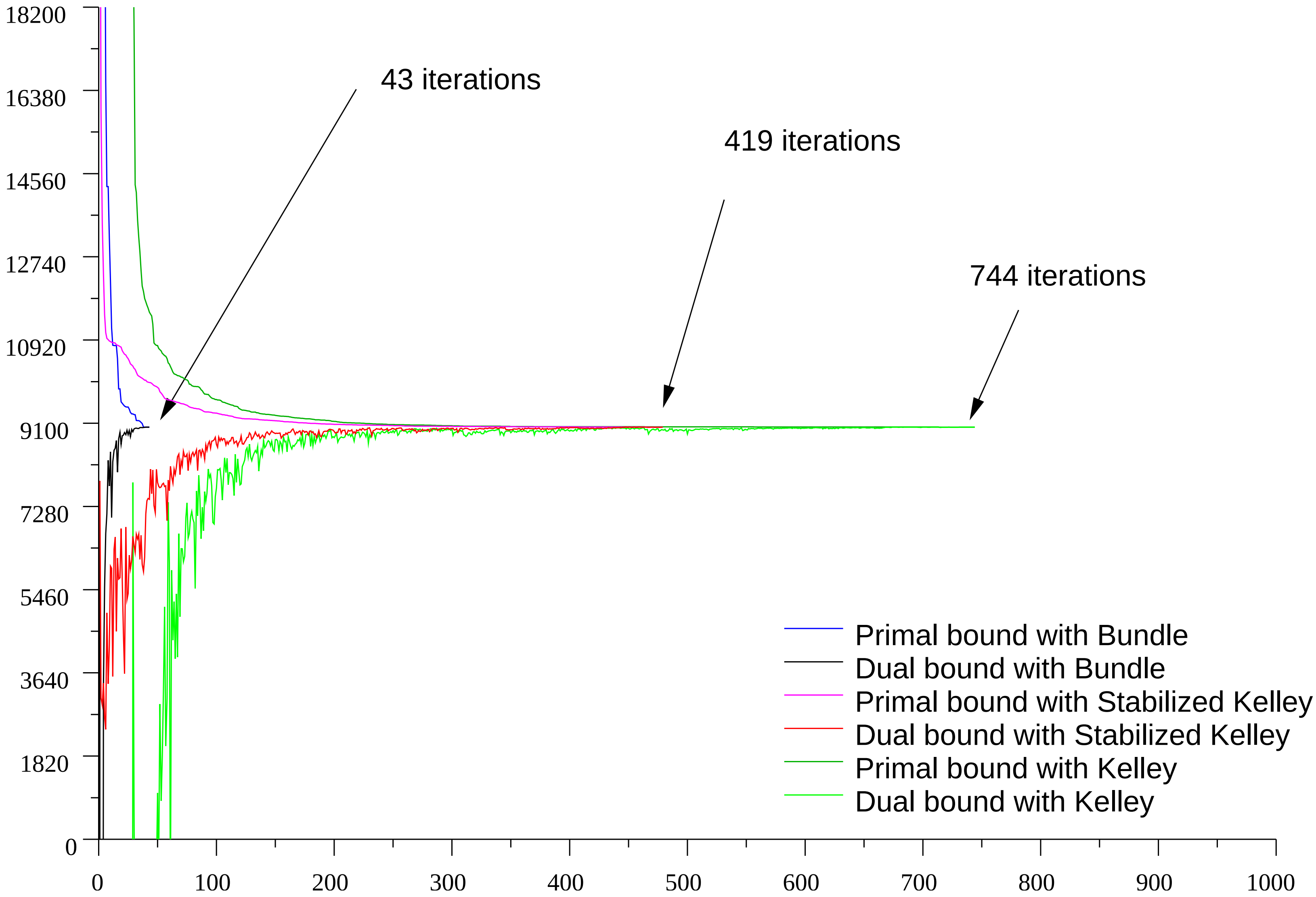
TSP : bays29 : primal and dual bounds



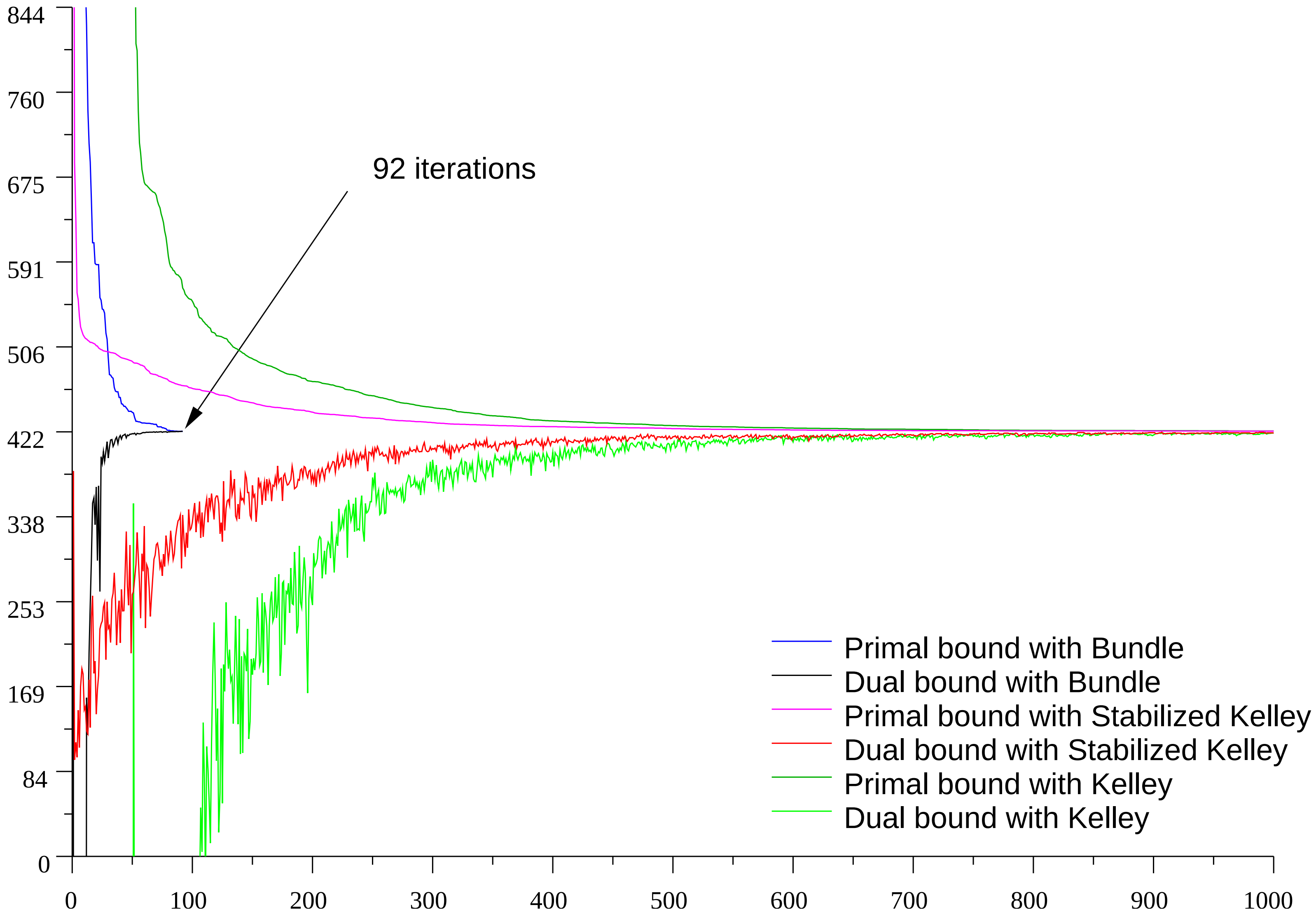
TSP : bays29 : primal and dual bounds



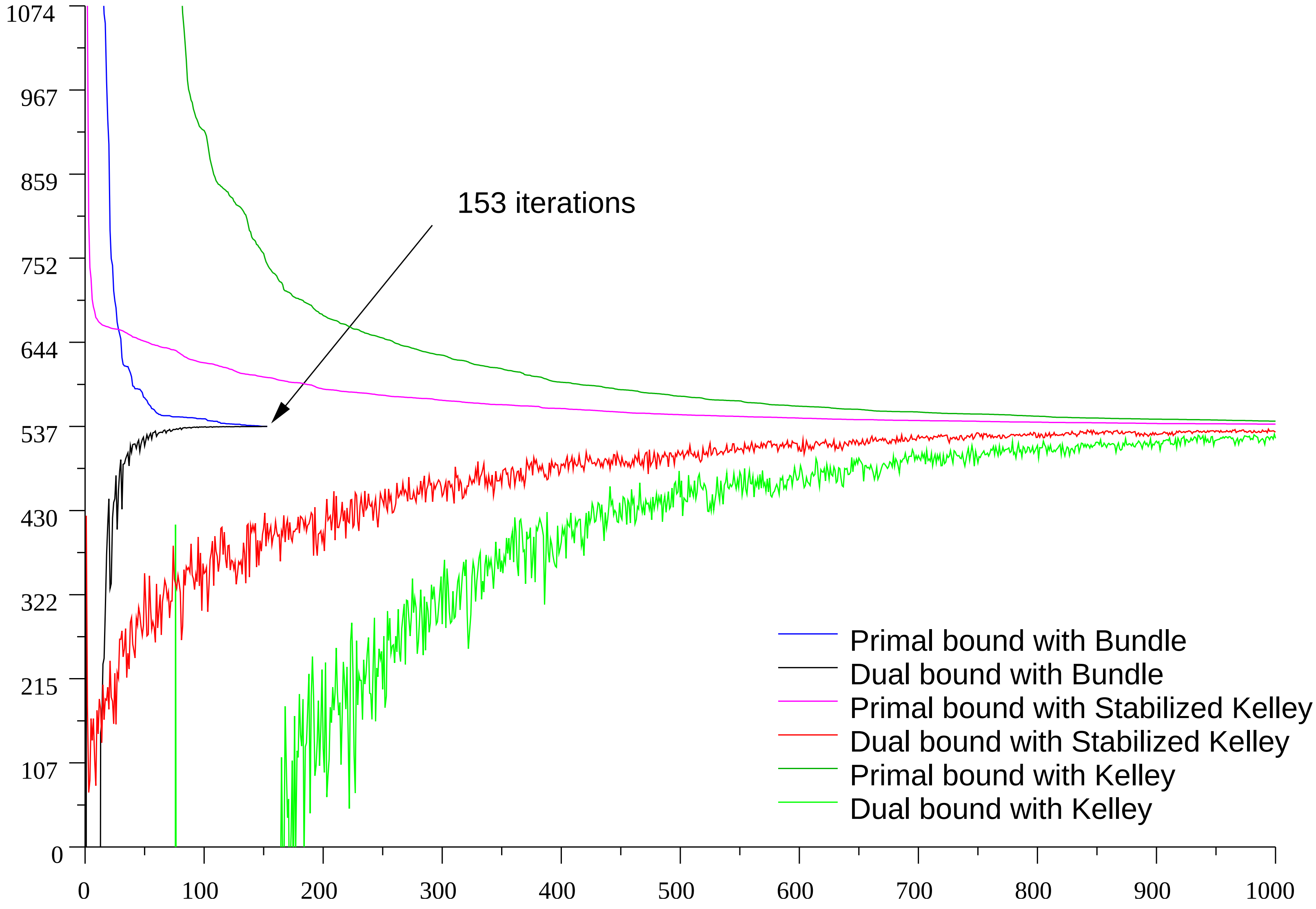
TSP : bays29 : primal and dual bounds



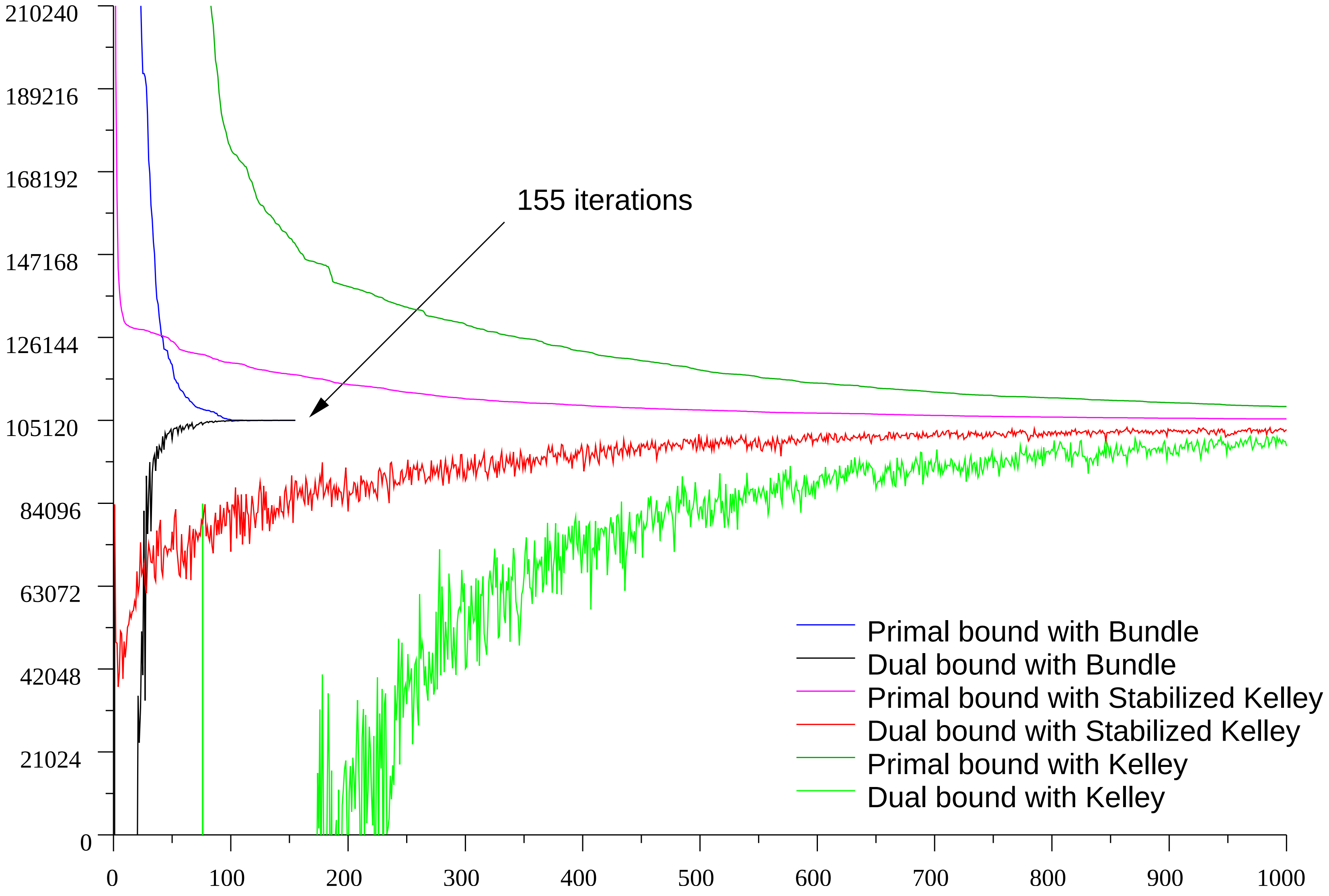
TSP : eil51 : primal and dual bounds



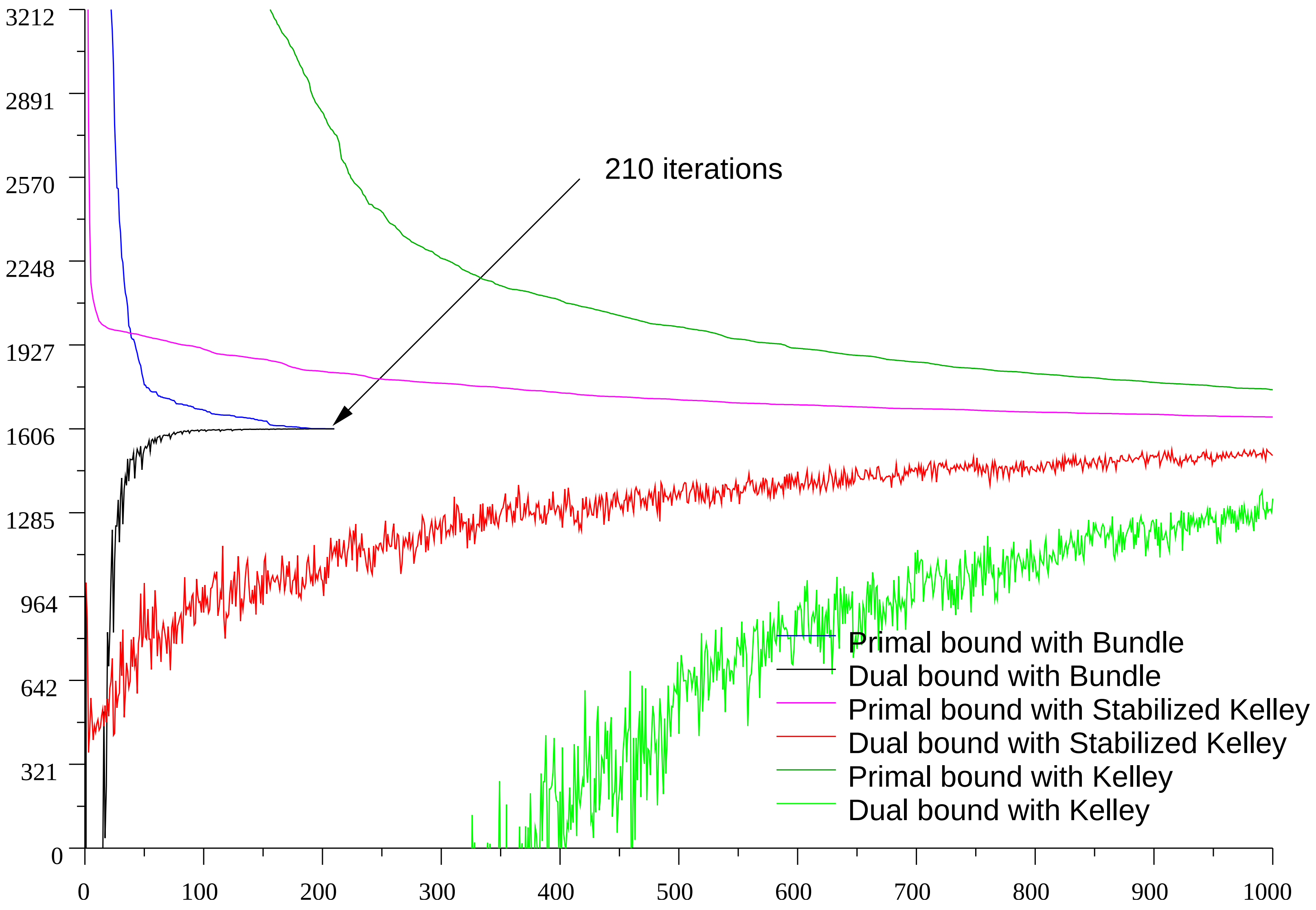
TSP : eil76 : primal and dual bounds



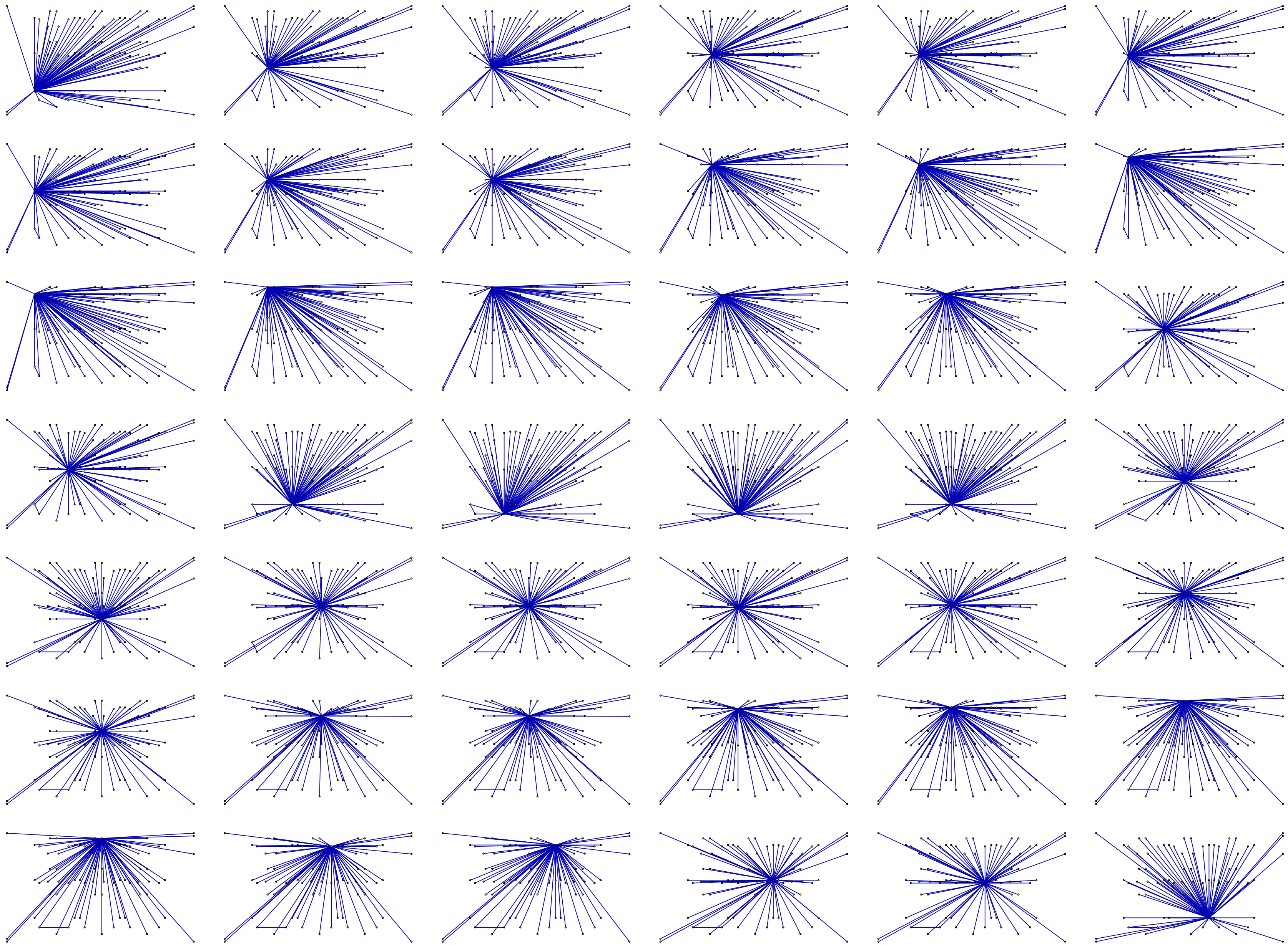
TSP : pr76 : primal and dual bounds



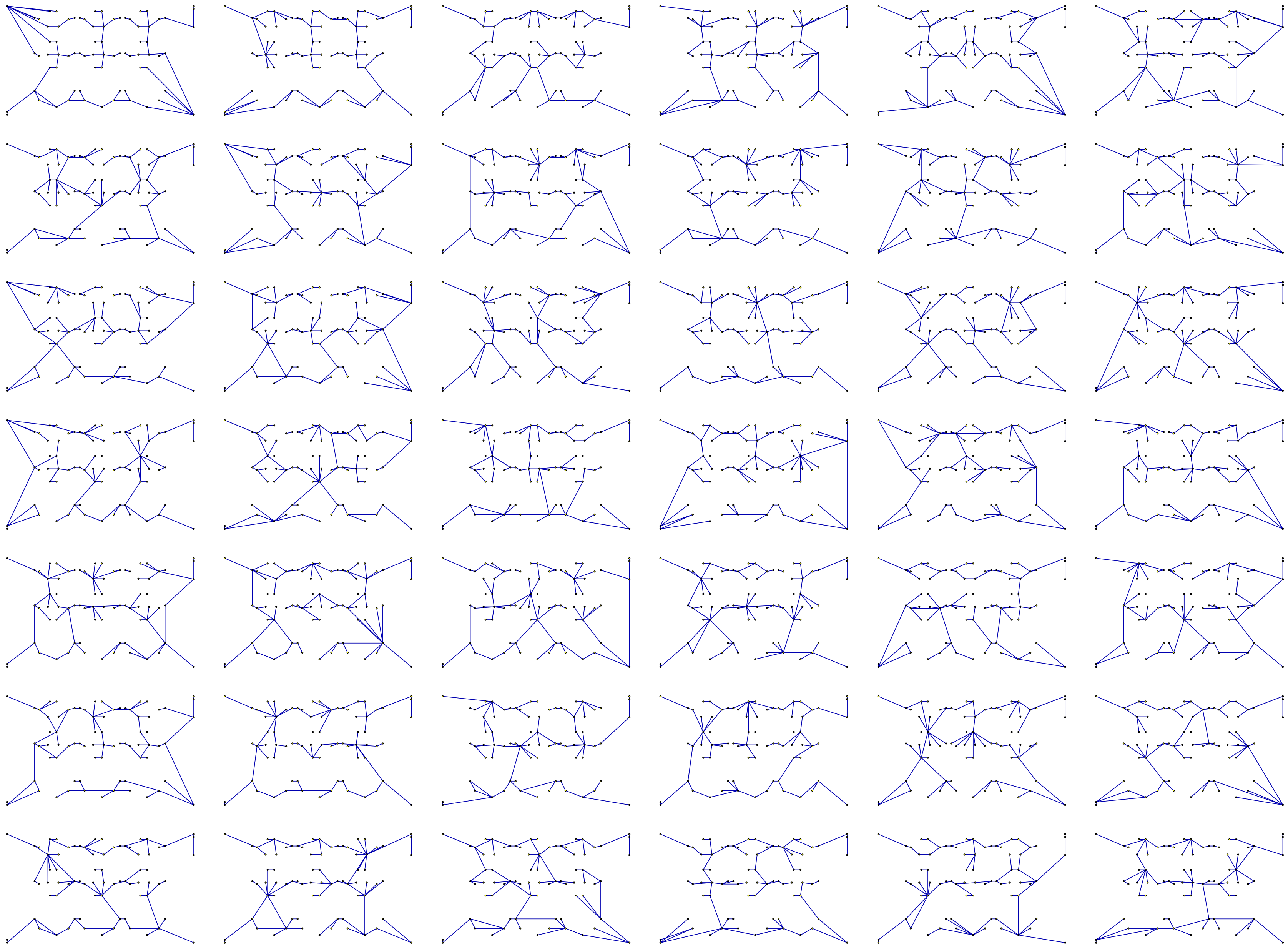
TSP : gr120 : primal and dual bounds



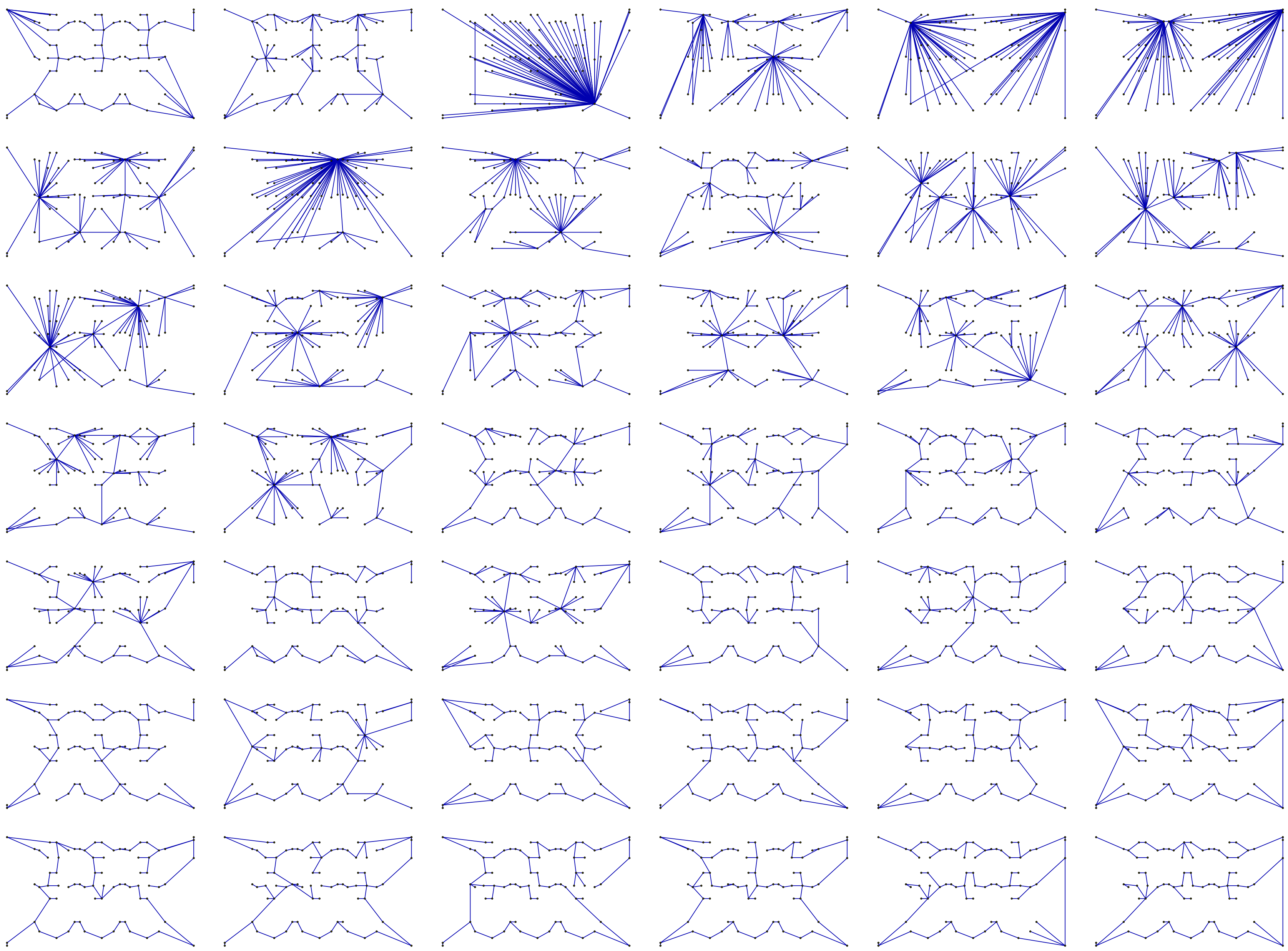
pr76 : first 1-trees with Kelley



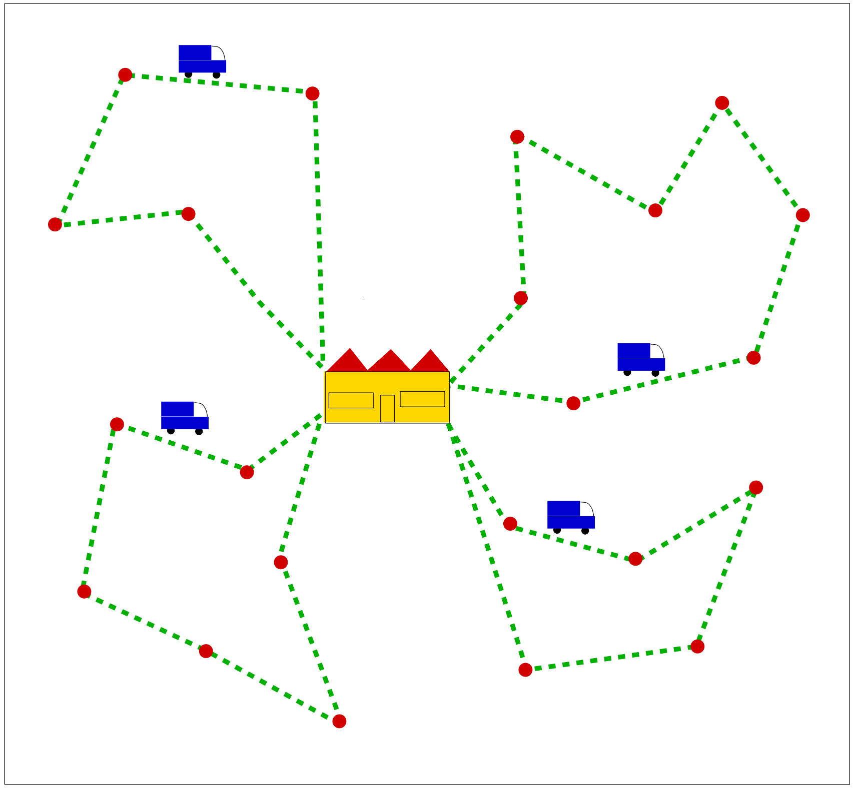
pr76 : first 1-trees with Stabilized Kelley



pr76 : first 1-trees with Bundle



Capacitated Vehicle Routing Problem (CVRP)

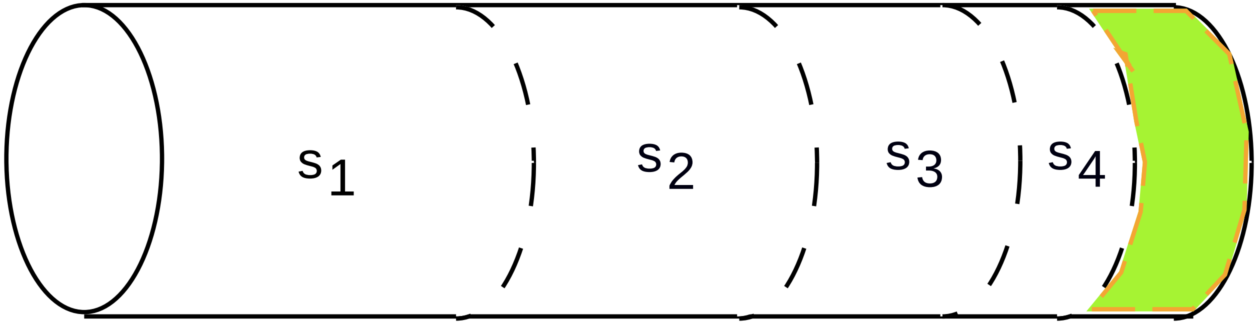


Instances from CVRPLIB

CVRP : Comparative results

		Counters		Timers (ticks)		
Problem	Method	Iter	Col	Oracle	Master	total
n22-k8	kelley poor	53	53	277	12	2s99t
n22-k8	kelley rich	52	35	323	8	3s36t
n22-k8	bundle rich	42	32	483	17	5s08t
n14-k2	kelley poor	54	54	26317	5	4m23s
n14-k2	kelley rich	44	43	30732	3	5m07s
n14-k2	bundle rich	16	14	180509	6	30m05s

Cutting Stock Problem



- $$\min \textbf{ number of rolls} \quad \begin{cases} \min & \sum_k y_k \\ & \sum_k x_{i,k} \geq d_i \quad \forall i \\ & \sum_i s_i\, x_{i,k} \leq y_k \quad \forall k \end{cases}$$

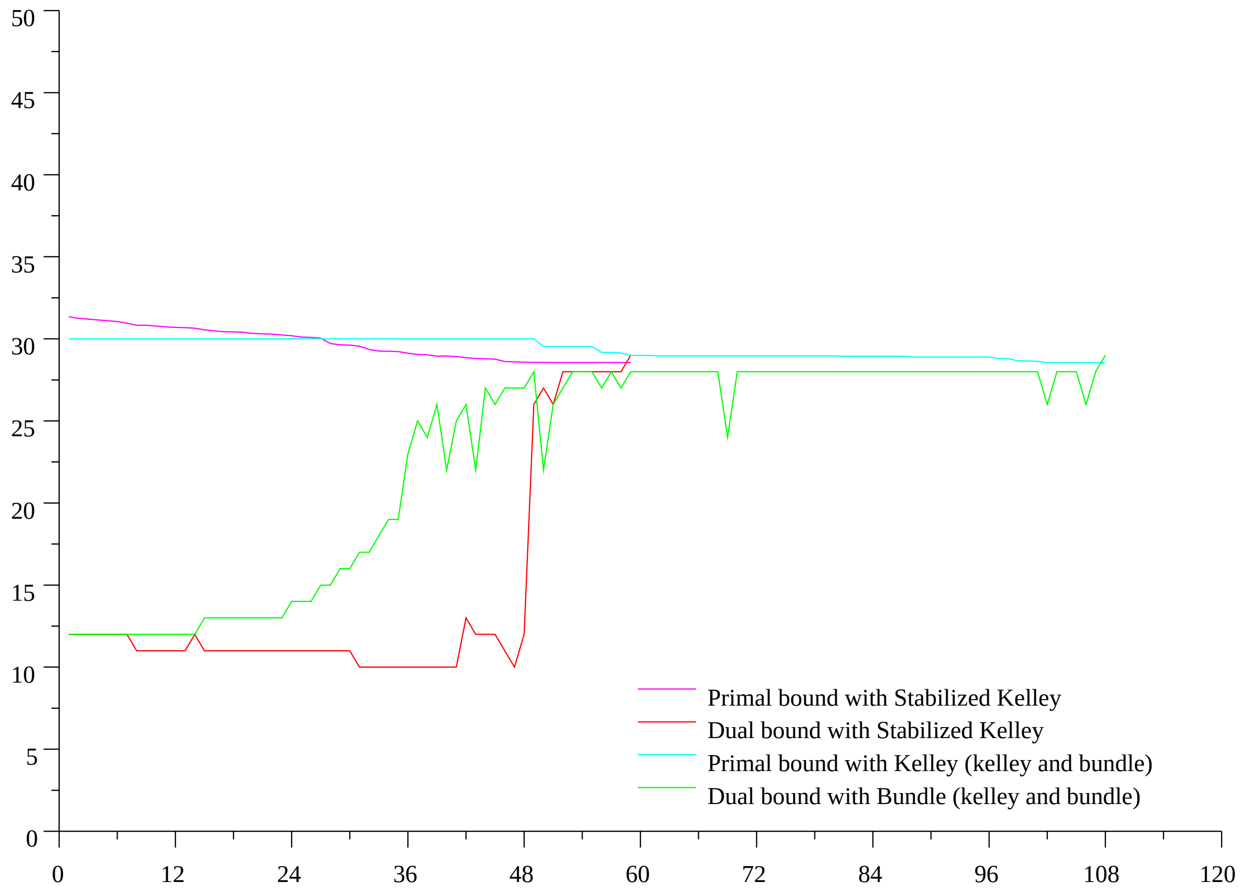
- $$\min \textbf{ waste} \quad \begin{cases} \min & \sum_k (1 - \sum_i s_i x_{i,k}) \\ & \overline{d}_i \geq \sum_k x_{i,k} \geq \underline{d}_i \quad \forall i \\ & \sum_i s_i\, x_{i,k} \leq y_k \quad \forall k \end{cases}$$

CSP number of rolls: Comparative results

average on 10 or 20 instances

		Counters		Timers (ticks)		
Problem	Method	Iter	Col	Oracle	Master	total
realDat20	kelley poor	44	45	54	4	0s65t
realDat20	bundle poor	90	52	772	32	8s17t
realDat20	kelley rich	32	32	434	3	4s42t
realDat20	bundle rich	51	38	1149	20	11s76t
realDat20	bundle + Kelley	39	33	888	14	9s9t
50b100	kelley poor	228	228	612	51	7s4t
50b100	bundle poor	257	190	1633	577	22s56t
50b100	kelley rich	138	138	521	33	5s82t
50b100	bundle rich	145	123	1717	241	19s87t
50b100	bundle + Kelley	124	115	1263	129	14s22t

CSP number of rolls: primal and dual bounds

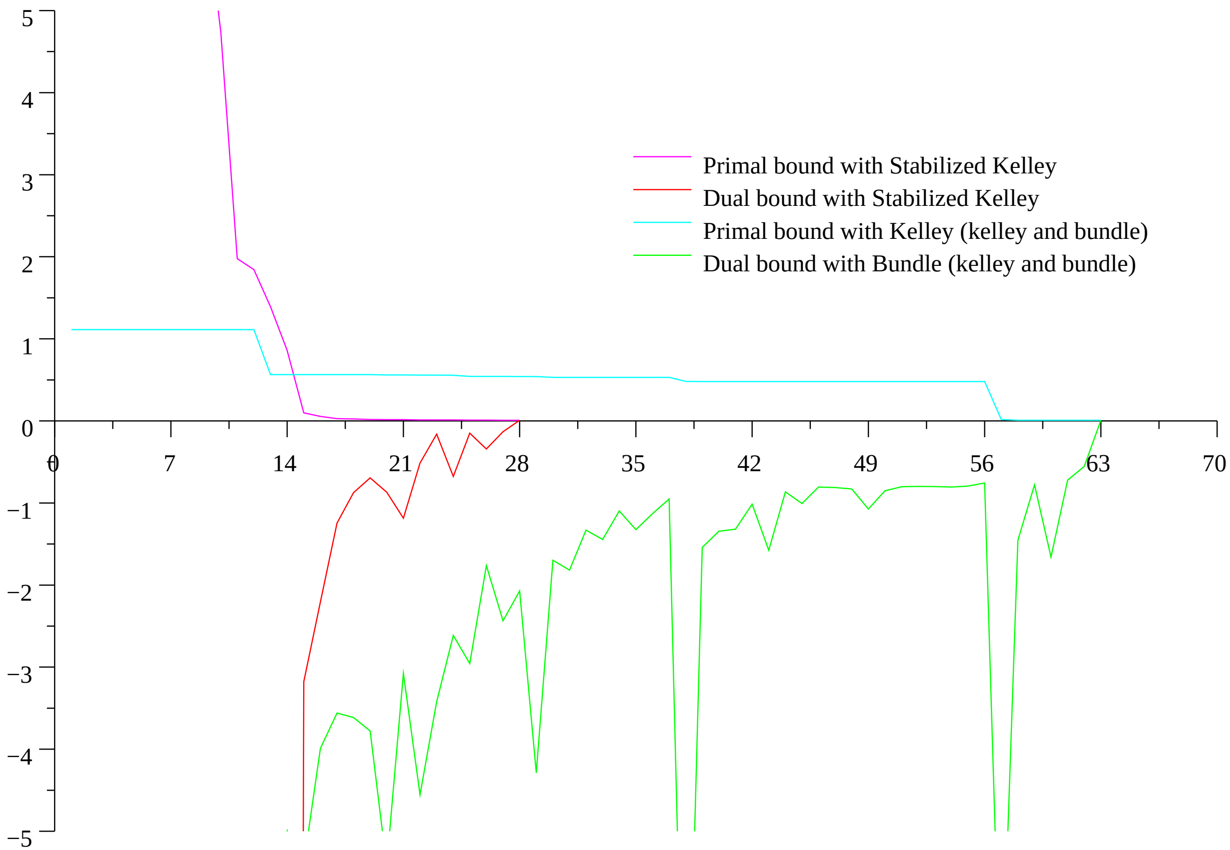


CSP Waste: Comparative results

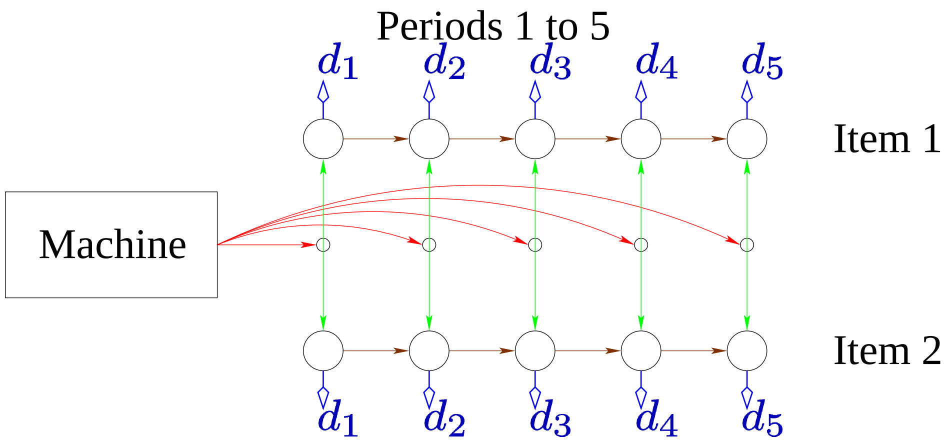
average on 10 or 20 instances

		Counters		Timers (ticks)		
Problem	Method	Iter	Col	Oracle	Master	total
realDat20	kelley poor	50	50	116	74	1s33t
realDat20	kelley rich	47	47	146	8	1s66t
realDat20	bundle rich	71	50	1250	38	13s1t
50b100	kelley poor	146	144	794	40	8s83t
50b100	kelley rich	177	176	700	70	8s24t
50b100	bundle rich	205	149	1815	548	24s18t

CSP Waste: primal and dual bounds



Multi-Item Lot-Sizing : Comparative results



average on 10 instances

Decomposition in SILS

		Counters			Timers (ticks)		
Problem	Method	Iter	SP	Col	Oracle	Master	total
i20-t60	kelley	8	164	117	55	4	4s79t
i20-t60	bundle	66	1326	207	455	92	18s41t

Observations

(based on preliminary results)

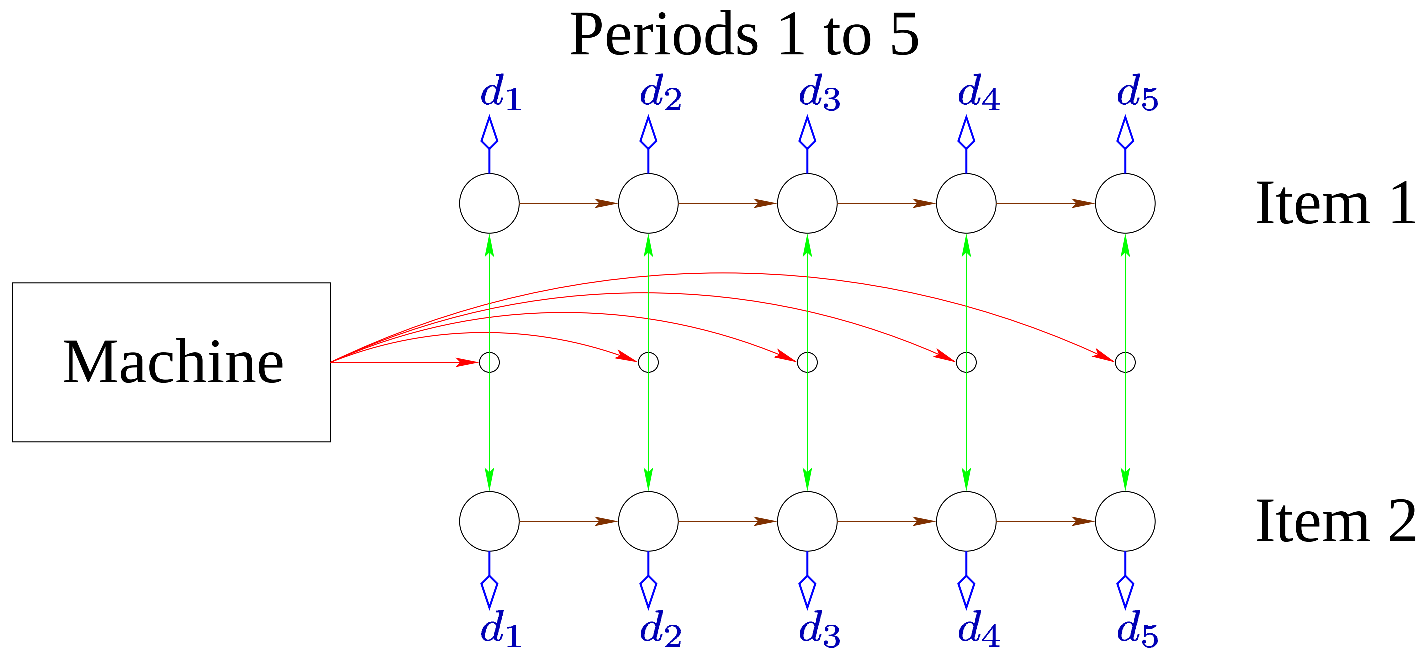
- Bundle takes far fewer iterations for some applications (like TSP), while not much is to win (compared to kelley) in other applications.

  ? Characterization ?

- Stabilization may imply harder oracles. Hence, there is a time tradeoff.
- Kelley finds primal feasible solution earlier.
- Kelley accepts inexact oracles.

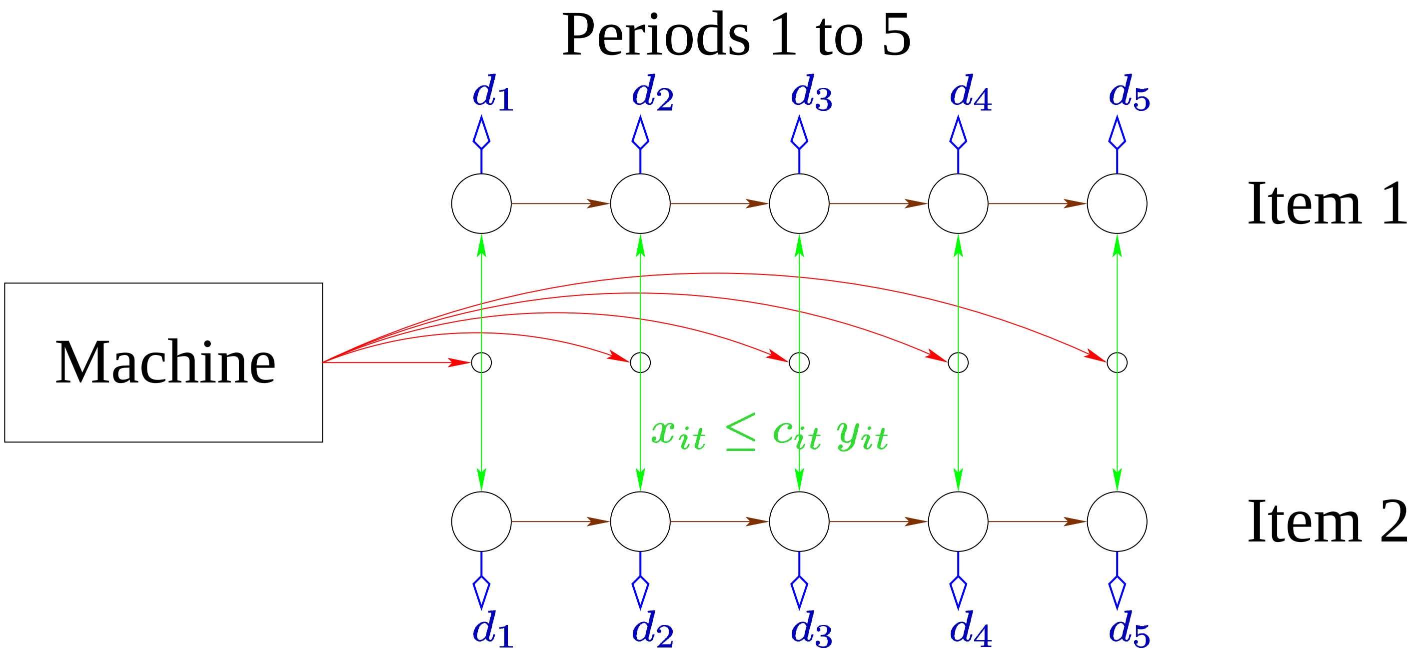
Example: Multi-Item Lot-Sizing

R



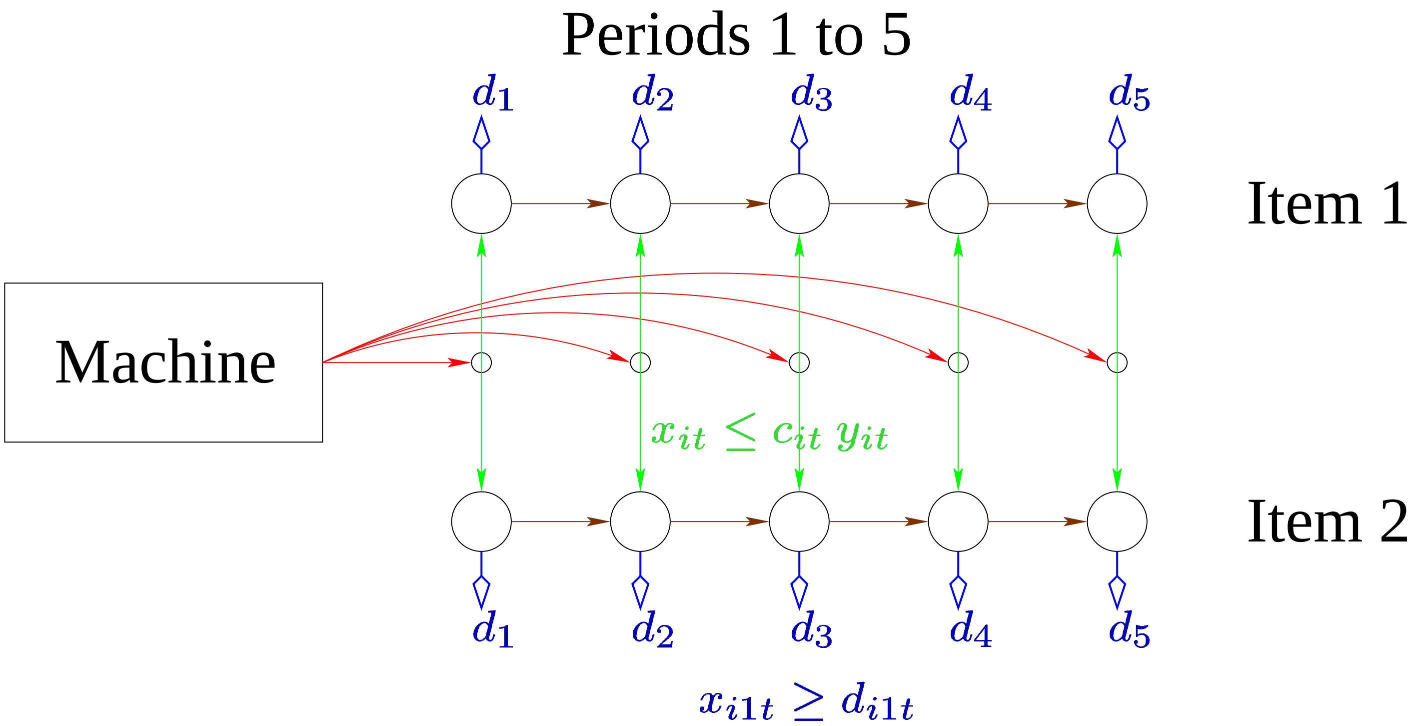
Example: Multi-Item Lot-Sizing

R



Example: Multi-Item Lot-Sizing

R



Example: Multi-Item Lot-Sizing

R