Col. Gen. Stabilization using Dual Smoothing: Theory & Practice

Colgen Workshop, June 2012

J. Han (1), P. Pesneau (2), A. Pessoa (3),
R. Sadykov (1), E. Uchoa (3), F. Vanderbeck (1)

(1) INRIA Bordeaux-Sud-Ouest, team RealOpt

(2) Université de Bordeaux, Institut de Mathématiques

(3) LOGIS , Universidade Federal Fluminense

Outline

Problem decomposition

Assume a bounded integer integer problem:

$$\begin{aligned} [\text{F}] \equiv \min\{cx & \quad : \\ x \in Y \equiv \quad Ax & \quad \geq \quad a \\ x \in Z \equiv \quad Bx & \quad \geq \quad b \\ x & \quad \in \quad N^n \end{aligned}$$

Let $X := Y \cap Z$. Assume that subproblem

$$[\text{SP}] \equiv \min\{cx : x \in Z\} \tag{1}$$

is "relatively easy" to solve compared to problem [F]. Then,

$$\begin{aligned} Z &= \{z^q\}_{q \in Q} \\ &= \{x \in N^n : x = \sum_{q \in Q} z^q \lambda_q, \sum_{q \in Q} \lambda_q = 1; \; \lambda_q \geq 0 \; \forall q \in Q\} \end{aligned}$$

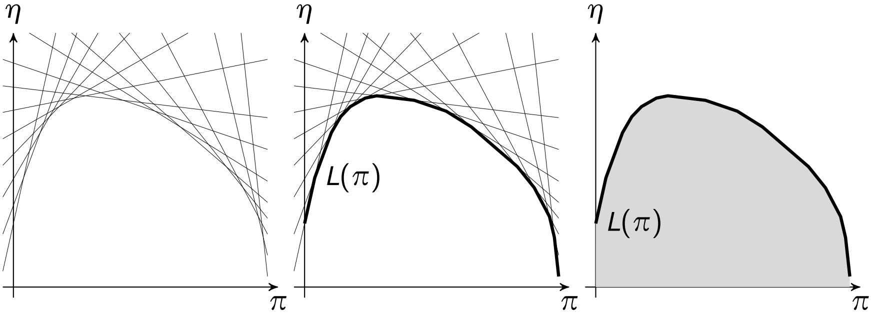
and

$$conv(Z) = \{x \in R^n_+ : \sum_{q \in Q} z^q \lambda_q, \sum_{q \in Q} \lambda_q = 1, \; \lambda_q \geq 0 \; q \in Q\}$$

Lagrangian Relaxation & Duality

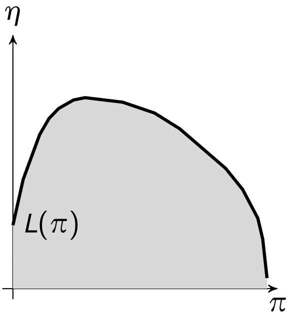
$$L(\pi) := \min_{q \in Q} \{ c\, z^q + \pi(a - Az^q) \}$$

$$[\text{LD}] := \max_{\pi \in R^m_+} \min_{q \in Q} \{ c\, z^q + \pi(a - Az^q) \}$$



Lagrangian Dual as an LP

$$
\begin{aligned}
[\text{LD}] \quad &\equiv \quad \max_{\pi \in R^m_+} \min_{q \in Q} \{\pi\, a + (c - \pi A) z^q\}; \\
&\equiv \quad \max\{\eta, \\
&\qquad \eta \leq c z^q + \pi(a - A z^q) \qquad q \in Q, \\
&\qquad \pi \in R^m_+, \eta \in R^1\}; \\
&\equiv \quad \min\{\sum_{q \in Q} (c z^q) \lambda_q, \\
&\qquad \sum_{q \in Q} (A z^q) \lambda_q \geq a, \\
&\qquad \sum_{q \in Q} \lambda_q = 1, \\
&\qquad \lambda_q \geq 0 \qquad q \in Q\}; \\
&\equiv \quad \min\{cx : Ax \geq a,\ x \in conv(Z)\}.
\end{aligned}
$$



Dantzig-Wolfe Reformulation & Restricted Master

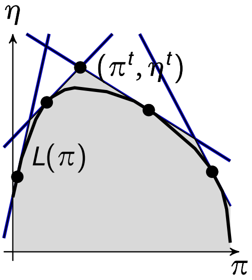
$$\begin{aligned} \min \sum_{q \in Q} (cx)\lambda_q & & \\ \sum_{q \in Q} (Az^q)\lambda_q &\geq & a \\ \sum_{q \in Q} \lambda_q &= & 1 \\ \lambda_q &\in & \{0,1\}\ \forall q \in Q\,. \end{aligned}$$

$$[\mathrm{M}^t] \equiv \min\{\sum_{q \in Q^t} cz^q \lambda_q : \sum_{q \in Q^t} Az^q \lambda_q \geq a; \sum_{q \in Q^t} \lambda_q = 1; \lambda_q \geq 0,\ q \in Q^t\}$$

$$[\mathrm{DM}^t] \equiv \max\{\eta : \pi(Az^q - a) + \eta \leq cz^q,\ q \in Q^t; \pi \in R^m_+; \eta \in R^1\}$$

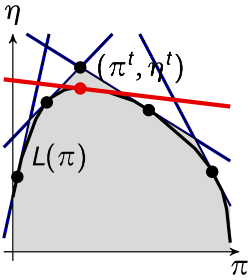
Restricted Master, Dual Polyhedra, & Pricing Oracle

- $[\text{M}^t] \equiv \min$ $\{cx : Ax \geq a, x \in conv(\{z^q\}_{q \in Q^t})\}$.
- $L^t() : \pi \rightarrow L^t(\pi) =$ $\min_{q \in Q^t} \{\pi a + (c - \pi A) z^q\}$;
- **Solving [LSP(π^t)]** yields:
  1. most neg. red. cost col. for $[\text{M}^t]$
  2. most violated constr. for $[\text{DM}^t]$
  3. correct value $L()$ at point π^t

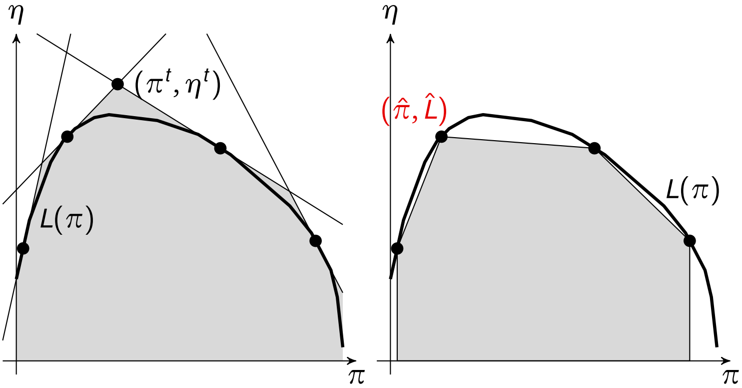


Restricted Master, Dual Polyhedra, & Pricing Oracle

- $[\mathrm{M}^t] \equiv \min \{cx : Ax \geq a, x \in conv(\{z^q\}_{q \in Q^t})\}$.
- $L^t() : \pi \rightarrow L^t(\pi) = \min_{q \in Q^t} \{\pi a + (c - \pi A) z^q\}$;
- **Solving [LSP(π^t)]** yields:
  1. most neg. red. cost col. for $[\mathrm{M}^t]$
  2. most violated constr. for $[\mathrm{DM}^t]$
  3. correct value $L()$ at point π^t



Dual Polyhedra: Outer and Inner approximations



Cut Separation Strategies in the Dual

Column generation for a master program
$\equiv$ cut generation for the dual master

$\Downarrow$

Cutting plane "strategies" translate into in col. gen. **"stabilization"**

- In-Out separation **[BenAmeurNeto07, FischettiSalvagnin10]**
- Central point cutting strategy **[GoffinVial, LeePark11]**
- Lexicographic Simplex **[ZanetteFischettiBalas11]**
- ...

Cut Separation Strategies in the Dual

Column generation for a master program

$\equiv$ cut generation for the dual master

$\Downarrow$

Cutting plane "strategies" translate into in col. gen. **"stabilization"**

- In-Out separation **[BenAmeurNeto07, FischettiSalvagnin10]**
- Central point cutting strategy **[GoffinVial, LeePark11]**
- Lexicographic Simplex **[ZanetteFischettiBalas11]**
- ...

Methods to Solve the Lagrangian Dual

- A sequence of candidate **dual solutions**

$$\{\pi^t\}_t \rightarrow \pi^* \in \Pi^*$$

- A sequence of candidate **primal solutions** (a by-product to prove optimality)

$$\{x^t\}_t \rightarrow x^* \in X^*$$

- **Oracle:** $z^t \leftarrow argmin_{x \in Z}\{(c - \pi^{t-1}A)x\}$.

1. **Ascent methods:**
   - Subgradient
   - Volume
   - Conjugate Sub-gradient
2. **Polyhedral methods:**
   - Kelley
   - Bundle
   - ACCPM

Methods to Solve the Lagrangian Dual

- A sequence of candidate **dual solutions**

$$\{\pi^t\}_t \rightarrow \pi^* \in \Pi^*$$

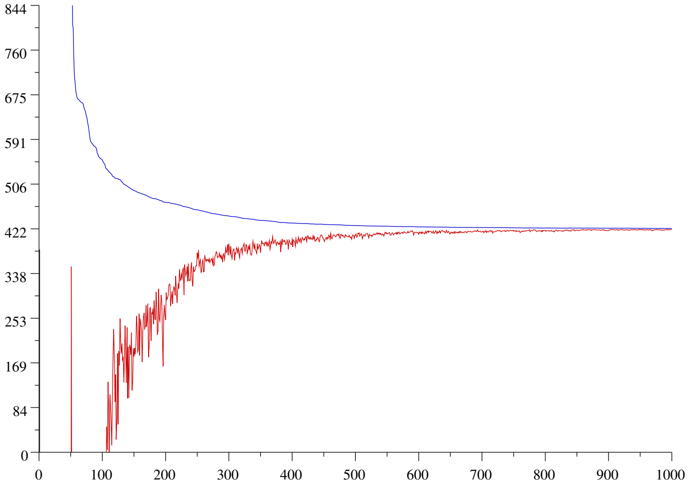
- A sequence of candidate **primal solutions** (a by-product to prove optimality)

$$\{x^t\}_t \rightarrow x^* \in X^*$$

- **Oracle:** $z^t \leftarrow argmin_{x \in Z}\{(c - \pi^{t-1}A)x\}$.

1. **Ascent methods:**
   - Subgradient
   - Volume
   - Conjugate Sub-gradient
2. **Polyhedral methods:**
   - Kelley
   - Bundle
   - ACCPM

Convergence



Convergence

$$\{\pi^t\}_t \rightarrow \pi^* \in \Pi^*$$

- **Dual oscillations:** π^t jump erratically (*bang-bang*), $\{L(\pi^t)\}$ non monotic (*yo-yo*) and possibly

$$||\pi^t - \pi^*|| > ||\pi^{t-1} - \pi^*||$$

- **Tailing-off effect:** towards the end, added inequalities yield marginal improvements / step sizes are very small.
- **Primal degeneracy / multiple dual optima:**
  - fewer non zero λ_q than master constraints;
  - dual system with fewer constraints than variables;
  - cuts co-linear with objective (inherent to cutting plane procedures).

- **Penalty functions** to drive the optimization towards $\hat{\pi}$:

$$\pi^t := argmax_{\pi \in R_+^m} \{L^t(\pi) - \hat{S}(\pi)\}$$

- Dual price **smoothing** (In-Out separation)

$$\begin{aligned} \text{[Wentges97]} \quad \tilde{\pi}^t &= \alpha\hat{\pi} + (1-\alpha)\pi^t \\ \text{[Neame99]} \quad \tilde{\pi}^t &= \alpha\tilde{\pi}^{t-1} + (1-\alpha)\pi^t \end{aligned}$$

- Dual price **centralization** (central point separation)
  - Analytic center (of trust polyhedra)
  - Chebyshev center
  - Optimal face center

Stabilization Techniques

- **Penalty functions** to drive the optimization towards $\hat{\pi}$:

$$\pi^t := argmax_{\pi \in R^m_+} \{L^t(\pi) - \hat{S}(\pi)\}$$

- Dual price **smoothing (In-Out separation)**

$$\textbf{[Wentges97]} \quad \tilde{\pi}^t = \alpha \hat{\pi} + (1-\alpha)\pi^t$$

$$\textbf{[Neame99]} \quad \tilde{\pi}^t = \alpha \tilde{\pi}^{t-1} + (1-\alpha)\pi^t$$

- Dual price **centralization (central point separation)**
  - Analytic center (of trust polyhedra)
  - Chebyshev center
  - Optimal face center

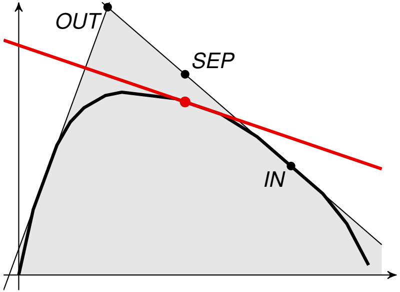
Dual Price Smoothing ≡ In-Out Separation

$$\boxed{\tilde{\pi}^t = \alpha\hat{\pi} + (1-\alpha)\pi^t \longleftrightarrow (\text{Wentges 's rule})}$$

$$(\pi^{\text{in}}, \eta^{\text{in}}) := (\hat{\pi}, \hat{L}) \tag{2}$$

$$(\pi^{\text{out}}, \eta^{\text{out}}) := (\pi^t, \eta^t)\,. \tag{3}$$

$$(\pi^{\text{sep}}, \eta^{\text{sep}}) := \alpha\,(\pi^{\text{in}}, \eta^{\text{in}}) + (1-\alpha)\,(\pi^{\text{out}}, \eta^{\text{out}})\,. \tag{4}$$



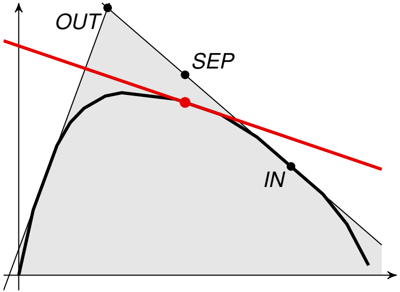
Dual Price Smoothing ≡ In-Out Separation

$$(\pi^{\text{in}}, \eta^{\text{in}}) := \begin{cases} (\hat{\pi}, \hat{L}) & \text{under Wentges 's rule.} \\ (\tilde{\pi}^{t-1}, L(\tilde{\pi}^{t-1})) & \text{under Neame's rule,} \end{cases} \tag{5}$$

$$(\pi^{\text{out}}, \eta^{\text{out}}) := (\pi^t, \eta^t) \,. \tag{6}$$

$$(\pi^{\text{sep}}, \eta^{\text{sep}}) := \alpha\,(\pi^{\text{in}}, \eta^{\text{in}}) + (1-\alpha)\,(\pi^{\text{out}}, \eta^{\text{out}}) \,. \tag{7}$$

Oracle: $z^t \leftarrow argmin_{x \in Z}\{(c - \pi^{\text{sep}} A)x\}$

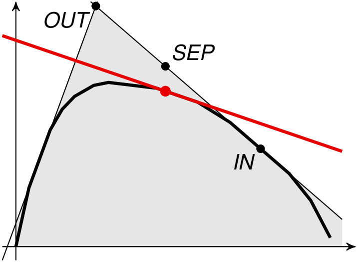


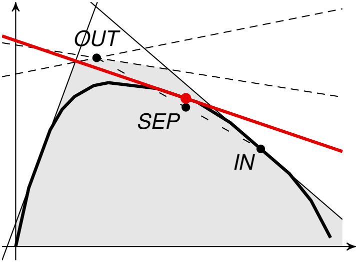
In-Out Separation

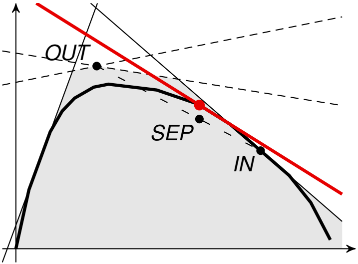
Case A: SEP is cut, so is OUT

Case B: SEP is not cut, but OUT is cut

Case C: neither SEP nor OUT is cut → “mis-price”







Properties

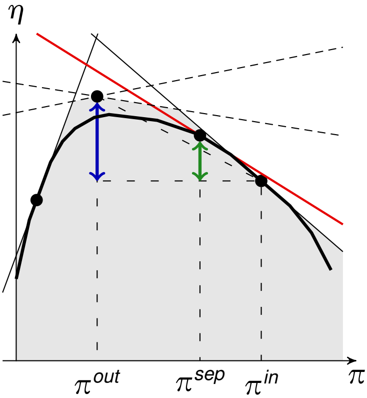
1. If SEP is cut by z^t, OUT is cut.
2. Otherwise,
   a) SEP defines the next IN point.
   b) $(L(\pi^{\text{sep}}) - \eta^{\text{in}}) \geq$ $(1-\alpha)\,(\eta^{\text{out}} - \eta^{\text{in}})$, i.e.,

   $$\eta^{\text{out}} - L(\pi^{\text{sep}}) \leq \alpha\,(\eta^{\text{out}} - \eta^{\text{in}})$$

   c) The OUT point might be cut.
   d) Otherwise, $\pi^t = \pi^{t-1} = \pi^{\text{out}}$ and

   $$\|\tilde{\pi}^t - \pi^{\text{out}}\| = \alpha\,\|\tilde{\pi}^{t-1} - \pi^{\text{out}}\|$$

In case of "mis-price"



Dual Smoothing: Termination of the Algorithm

- The number of columns that can be generated is finite.
- A column, once added, cannot be regenerated: the associated constraint is satisfied by both IN and OUT points (hence by SEP).
- However, there are iterations where no columns are generated, and the OUT point remains unchanged → mis-price
- Thus, on needs to prove convergence of a subsequence of mis-price iterations, starting with $\pi^{in} = \tilde{\pi}^0$. Then, $\tilde{\pi}^{t+1} = \alpha_t \tilde{\pi}^t + (1 - \alpha_t)\pi^{out}$. Hence,

$$||\tilde{\pi}^{t+1} - \pi^{out}|| = \alpha_t \, ||\tilde{\pi}^t - \pi^{out}|| = \ldots = \Pi_{\tau=0}^{t} \alpha_\tau \, ||\tilde{\pi}^0 - \pi^{out}||$$

if $\alpha = 0.8$

At iteration t, $||\tilde{\pi}^0 - \pi^{out}||$ is cut by a factor $(1 - \Pi_{\tau=0}^{t} \alpha_\tau)$.

BEWARE: convergence is only asymptotic for a constant $0 < \alpha < 1$.

Dual Smoothing: Termination of the Algorithm

- The number of columns that can be generated is finite.
- A column, once added, cannot be regenerated: the associated constraint is satisfied by both IN and OUT points (hence by SEP).
- However, there are iterations where no colunms are generated, and the OUT point remains unchanged → mis-price
- Thus, on needs to prove convergence of a subsequence of mis-price iterations, starting with $\pi^{in} = \tilde{\pi}^0$. Then, $\tilde{\pi}^{t+1} = \alpha_t \tilde{\pi}^t + (1 - \alpha_t)\pi^{out}$. Hence,

$$||\tilde{\pi}^{t+1} - \pi^{out}|| = \alpha_t \, ||\tilde{\pi}^t - \pi^{out}|| = \ldots = \Pi_{\tau=0}^{t} \alpha_\tau \, ||\tilde{\pi}^0 - \pi^{out}||$$

if $\alpha = 0.8$

At iteration t, $||\tilde{\pi}^0 - \pi^{out}||$ is cut by a factor $(1 - \Pi_{\tau=0}^{t} \alpha_\tau)$.

BEWARE: convergence is only asymptotic for a constant $0 < \alpha < 1$.

Dual Smoothing: Termination of the Algorithm

- The number of columns that can be generated is finite.
- A column, once added, cannot be regenerated: the associated constraint is satisfied by both IN and OUT points (hence by SEP).
- However, there are iterations where no colunms are generated, and the OUT point remains unchanged → mis-price
- Thus, on needs to prove convergence of a subsequence of mis-price iterations, starting with $\pi^{in} = \tilde{\pi}^0$. Then, $\tilde{\pi}^{t+1} = \alpha_t \tilde{\pi}^t + (1 - \alpha_t)\pi^{out}$. Hence,

$$||\tilde{\pi}^{t+1} - \pi^{out}|| = \alpha_t \, ||\tilde{\pi}^t - \pi^{out}|| = \ldots = \Pi_{\tau=0}^{t} \alpha_\tau \, ||\tilde{\pi}^0 - \pi^{out}||$$

if $\alpha = 0.8$

At iteration t, $||\tilde{\pi}^0 - \pi^{out}||$ is cut by a factor $(1 - \Pi_{\tau=0}^{t} \alpha_\tau)$.

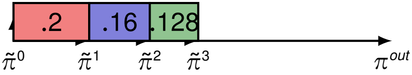
BEWARE: convergence is only asymptotic for a constant $0 < \alpha < 1$.

Dual Smoothing: Termination of the Algorithm

- The number of columns that can be generated is finite.
- A column, once added, cannot be regenerated: the associated constraint is satisfied by both IN and OUT points (hence by SEP).
- However, there are iterations where no colunms are generated, and the OUT point remains unchanged $\rightarrow$ mis-price
- Thus, on needs to prove convergence of a subsequence of mis-price iterations, starting with $\pi^{in} = \tilde{\pi}^0$. Then, $\tilde{\pi}^{t+1} = \alpha_t \tilde{\pi}^t + (1 - \alpha_t)\pi^{out}$. Hence,

$$||\tilde{\pi}^{t+1} - \pi^{out}|| = \alpha_t \, ||\tilde{\pi}^t - \pi^{out}|| = \ldots = \Pi_{\tau=0}^{t} \alpha_\tau \, ||\tilde{\pi}^0 - \pi^{out}||$$

if $\alpha = 0.8$



At iteration t, $||\tilde{\pi}^0 - \pi^{out}||$ is cut by a factor $(1 - \Pi_{\tau=0}^{t} \alpha_\tau)$.

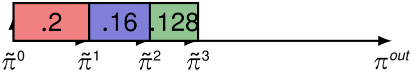
BEWARE: convergence is only asymptotic for a constant $0 < \alpha < 1$.

Dual Smoothing: Termination of the Algorithm

- The number of columns that can be generated is finite.
- A column, once added, cannot be regenerated: the associated constraint is satisfied by both IN and OUT points (hence by SEP).
- However, there are iterations where no colunms are generated, and the OUT point remains unchanged → mis-price
- Thus, on needs to prove convergence of a subsequence of mis-price iterations, starting with $\pi^{in} = \tilde{\pi}^0$. Then, $\tilde{\pi}^{t+1} = \alpha_t \tilde{\pi}^t + (1 - \alpha_t)\pi^{out}$. Hence,

$$||\tilde{\pi}^{t+1} - \pi^{out}|| = \alpha_t \, ||\tilde{\pi}^t - \pi^{out}|| = \ldots = \Pi_{\tau=0}^{t} \alpha_\tau \, ||\tilde{\pi}^0 - \pi^{out}||$$

if $\alpha = 0.8$



At iteration t, $||\tilde{\pi}^0 - \pi^{out}||$ is cut by a factor $(1 - \Pi_{\tau=0}^{t} \alpha_\tau)$.

BEWARE: convergence is only asymptotic for a constant $0 < \alpha < 1$.

Dual Smoothing: Termination “in Theory” vs “in Practice”

In case of **mis-pricing**, one observes a sequence of iterations in which:

- $||\tilde{\pi}^{t+1} - \pi^{out}|| = \alpha_t \, ||\tilde{\pi}^t - \pi^{out}||$
- $||\eta^{t+1} - L(\tilde{\pi}^{t+1})|| \leq \alpha_t \, ||\eta^t - L(\tilde{\pi}^t)||$

Hence, as $(1 - \Pi_{\tau=0}^{t} \alpha_\tau) \rightarrow 1$,

- $\tilde{\pi}^t \rightarrow \pi^{out}$
- $L(\tilde{\pi}^t) \rightarrow \eta^{out}$

After a finite number of iterations, $||\eta^t - L(\tilde{\pi}^t)|| \leq \epsilon$ and the master is considered as optimized. As there is a **finite number of possible values** for $L(\tilde{\pi}^t)$ and η^t,

$$\exists \epsilon \; : \quad ||\eta^t - L(\tilde{\pi}^t)|| \leq \epsilon \quad \Rightarrow \quad \eta^t = L(\tilde{\pi}^t)$$

which proves **finite convergence in theory** **[BenAmeurNeto07, Wentges97]**.

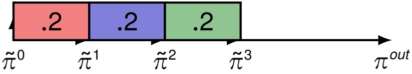
In practice, for finite convergence, one better choose α_τ :

$$(1 - \Pi_{\tau=0}^{t} \alpha_\tau) \geq 1 \quad \text{for a finite } t$$

Algorithm for handling a mis-pricing sequence

Step 0	$k \leftarrow 1$
Step 1	$\beta \leftarrow [1 - k * (1 - \alpha)]^{+}$
Step 2	$\pi^{sep} = \beta\, \pi^{0} + (1 - \beta)\, \pi^{out}$
Step 3	$k \leftarrow k + 1$
Step 4	call the oracle on π^{sep}
Step 5	if mis-pricing occurs, goto Step 1 else, got Step 0.

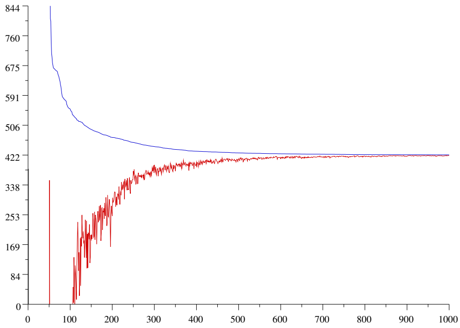
if $\alpha = 0.8$



I.e; we chose α_τ : $(1 - \Pi_{\tau=0}^{k}\alpha_\tau) = k * (1 - \alpha)$. Hence, $(1 - \Pi_{\tau=0}^{k}\alpha_\tau) \geq 1$ after $k = \left\lceil \frac{1}{(1-\alpha)} \right\rceil$ iterations, and smoothing stops with $\beta = 0$.

Dual Smoothing: Implementation in Practice

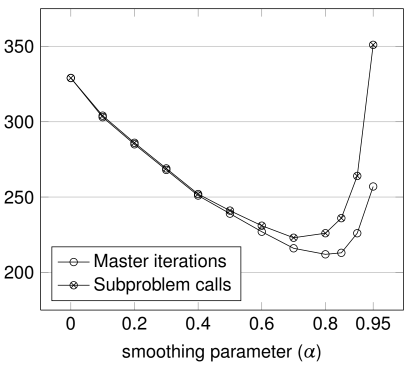
- Turn-off smoothing during HEADING-IN Phase.
- Trust π^{out} during TAILING-OFF Phase: small α.
- Decreasing α as the algorithm converges: f.i., record decreased α-value in mis-pricing sequence.

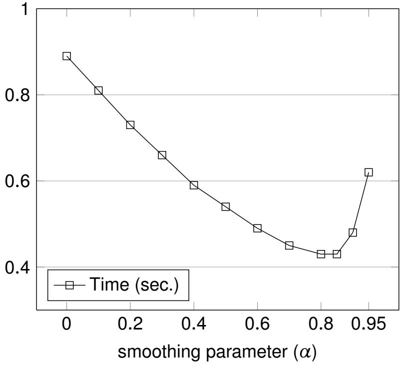


Sensivity to static α: Machine Scheduling

12 jobs per machines

(25 jobs / 2 machines ; 50 jobs / 4 machines)

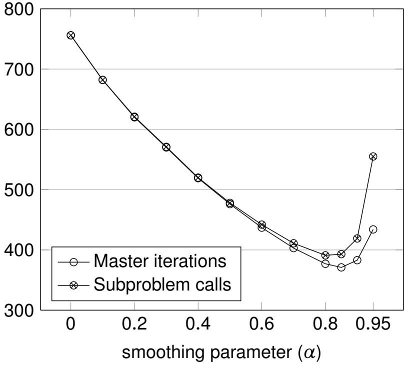


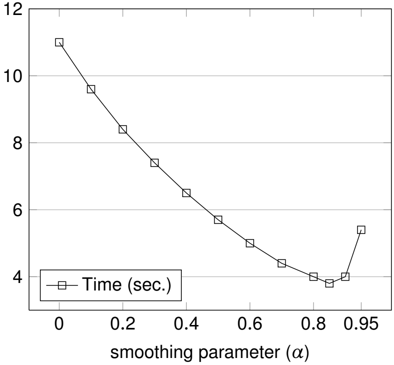


Sensivity to static α: Machine Scheduling

25 jobs per machines

25 jobs / 1 machine ; 50 jobs / 2 machines; 100 jobs / 4 machines

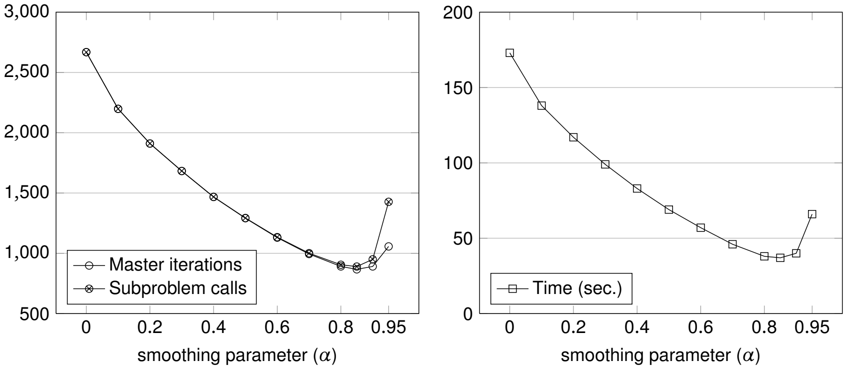




Sensivity to static α: Machine Scheduling

50 jobs per machines

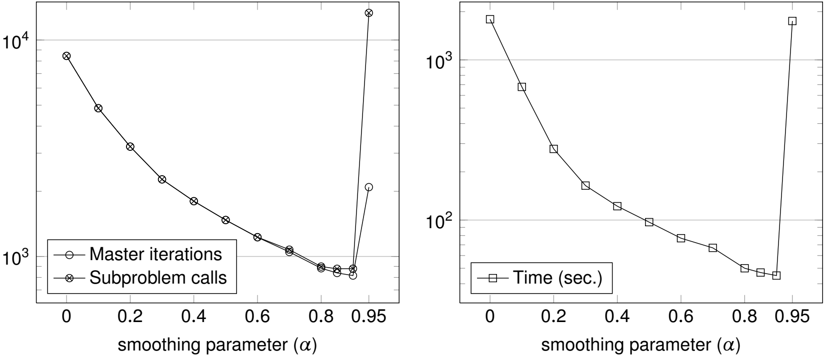
50 jobs / 1 machine ; 100 jobs / 2 machines; 200 jobs / 4 machines



Sensivity to static α: Machine Scheduling

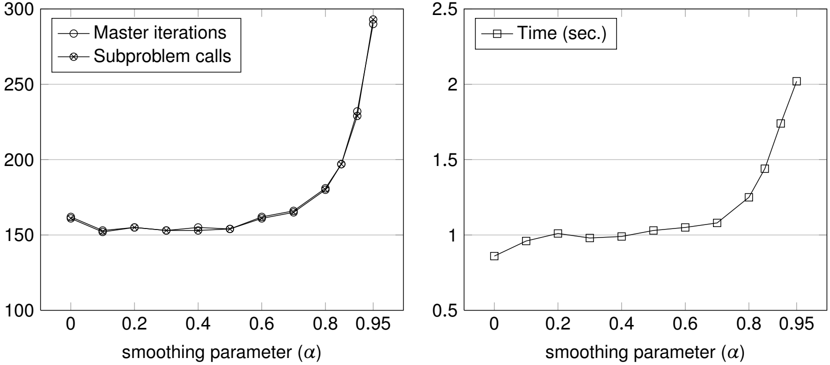
100 jobs per machines

100 jobs / 1 machine

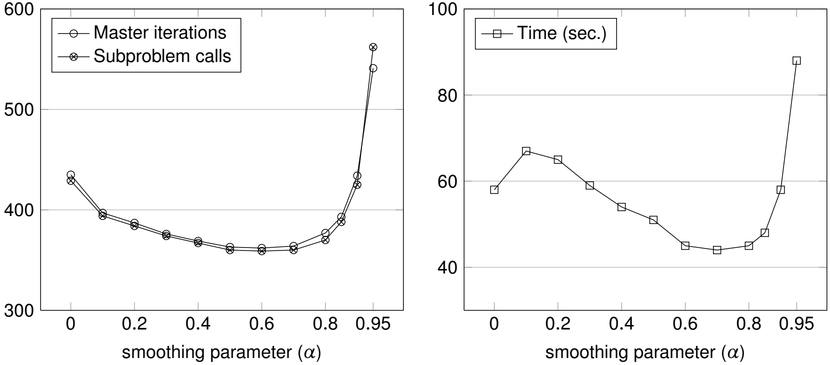


Sensivity to static α: Generalized Assignment

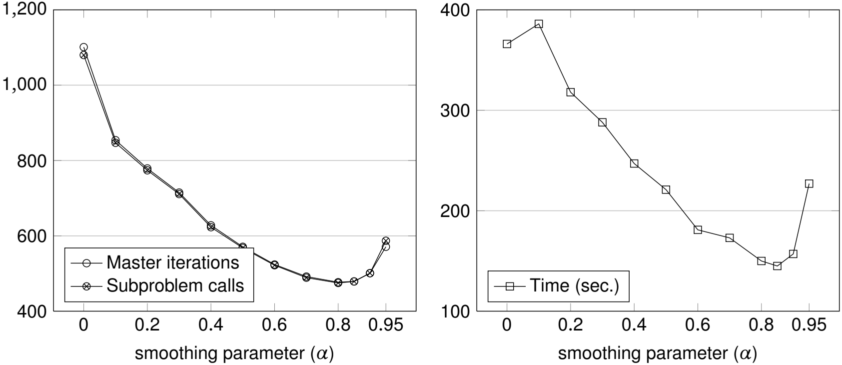
OR-Library C, D, E instances: 5 jobs per machine



OR-Library C, D, E instances: 10 jobs per machine

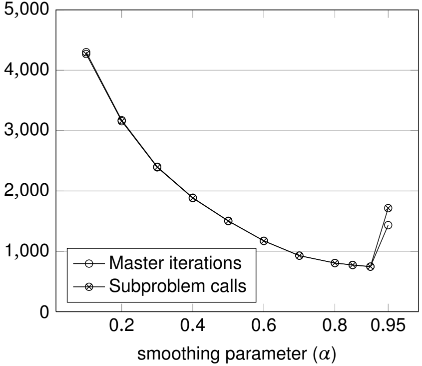


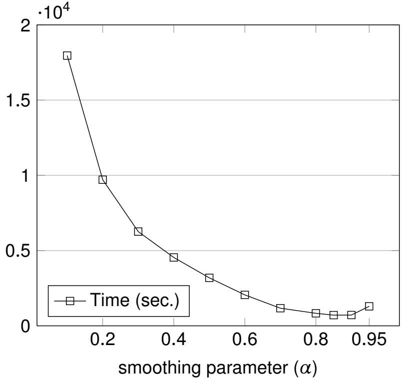
OR-Library C, D, E instances: 20 jobs per machine



Sensivity to static α: Generalized Assignment

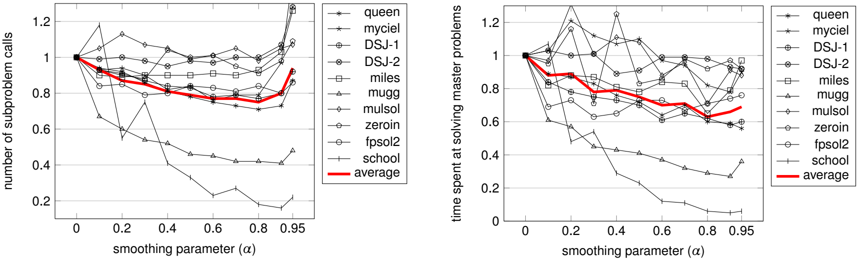
OR-Library C, D, E instances: 40 jobs per machine





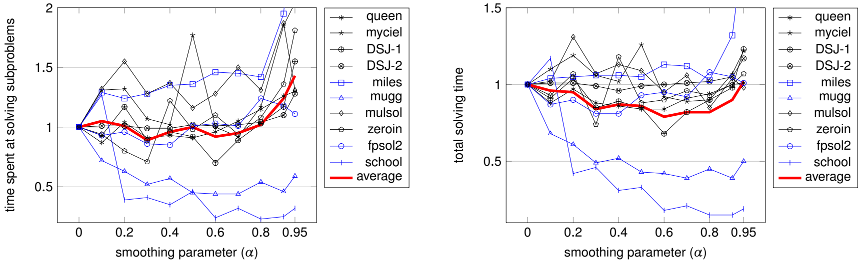
Sensivity to static α: Vertex Coloring

Number of calls to the oracle and Master CPU time



Sensivity to static α: Vertex Coloring

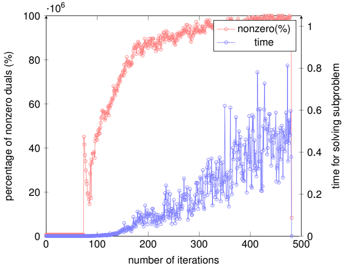
Subproblem CPU time and Overall CPU time

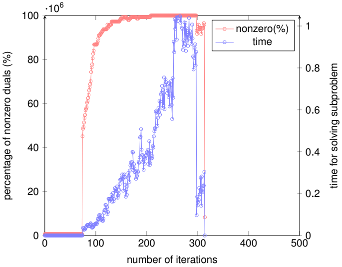


Sensivity to static α: Vertex Coloring

Dual **price vector density** & **subproblem CPU time**

on instance queen12_12, using the oracle of **[Ostergard, 2001]**





Conclusions

- Dual smoothing is the equivalent of IN-OUT separation $\longleftrightarrow$ cross-fertilization with the **cutting-plane community**
- Dual smoothing can have a **large impact on overall performance**.
- Dual smoothing requires a **single parameter**; but it needs to be **carrefully set**.
- **Hard instances** for Kelley seems to be those with many columns (f.i. more jobs per machine). Then, it seems best to apply heavy stabilization f.i. $\boldsymbol{\alpha = 0.85}$.
- **Dynamic and auto-adaptative** α**-schedules** are to be derived.
- **Stabilization** can make the pricing **subproblem harder** and eliminates iterations where pricing is easy.