Combining dual price smoothing and piecewise linear penalty function stabilization in column generation: experimental results

Ruslan Sadykov[1,2] Artur Pessoa[3] Eduardo Uchoa[3]
François Vanderbeck[2,1]

[1] Inria Bordeaux, France

[2] Université Bordeaux, France

[3] Universidade Federal Fluminense, Brazil

ISCO 2014
Lisbon, Portugal, March 5

Contents

Problem decomposition

Assume a bounded integer problem:

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

Assume that subproblem

$$[\mathrm{SP}] \equiv \min\{c\,x : \; x \in Z\} \tag{1}$$

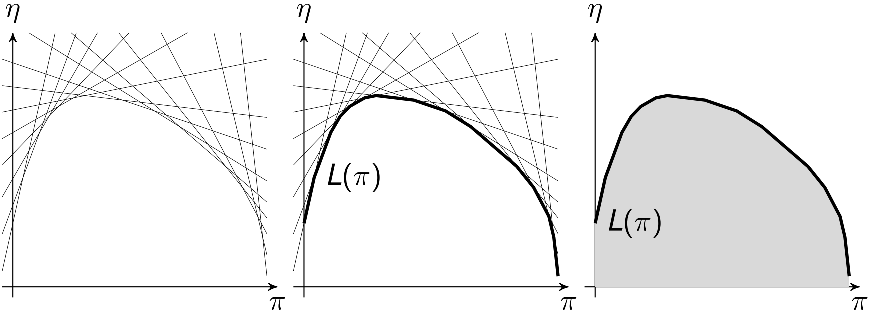
is "relatively easy" to solve compared to problem [F]. Then,

$$
\begin{aligned}
Z &= \{z^q\}_{q \in Q} \\
\mathrm{conv}(Z) &= \{x \in \mathbb{R}^n_+ : \sum_{q \in Q} z^q \lambda_q, \; \sum_{q \in Q} \lambda_q = 1, \; \lambda_q \geq 0 \; q \in Q\}
\end{aligned}
$$

Lagrangian Relaxation & Duality

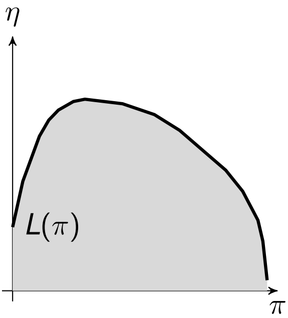
$$L(\pi) := \min_{q \in Q} \{c\, z^q + \pi\, (a - Az^q)\}$$

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



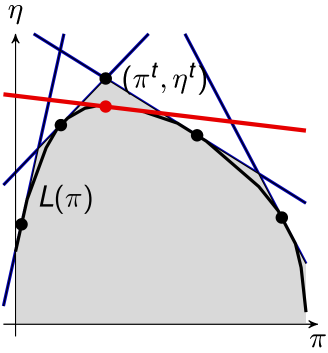
Lagrangian Dual as an LP

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

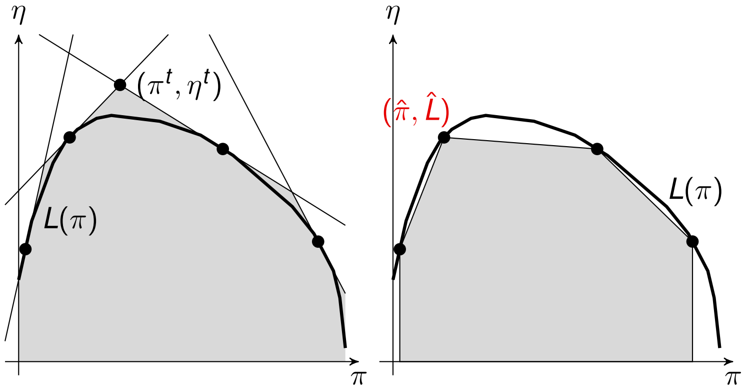


Restricted Master, Dual Polyhedra, & Pricing Oracle

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



Dual Polyhedra: Outer and Inner approximations



Contents

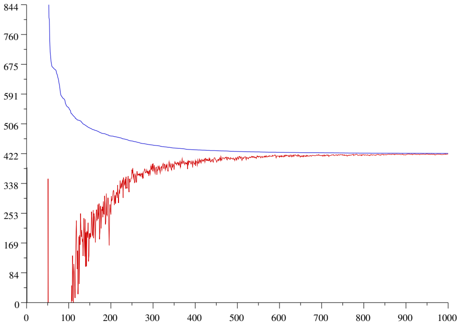
Convergence of Column Generation

- A sequence of candidate **dual solutions**

$$\{\pi^t\}_t \to \pi^*$$

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

$$\{x^t\}_t \to x^*$$



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

Our objectives

Automation

- We are at the point when **generic solvers using column generation** start to appear.
- Stabilization techniques require **critical parameters** to set.
- Our goal is to reduce as much as possible the number of these parameters.

Hybridation

- Several stabilization techniques exist.
- Why not try to combine them?

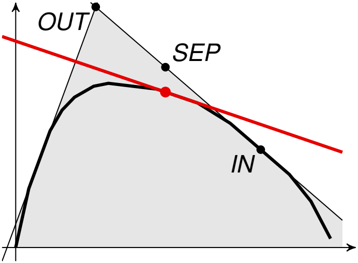
Dual Price Smoothing (I)

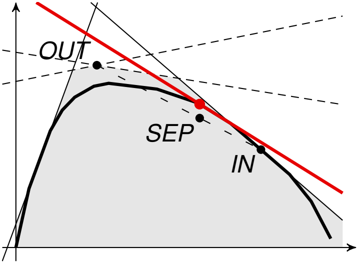
$$\boxed{\tilde{\pi}^t = \alpha\hat{\pi} + (1-\alpha)\pi^t}$$ **[Wentges 97]**

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

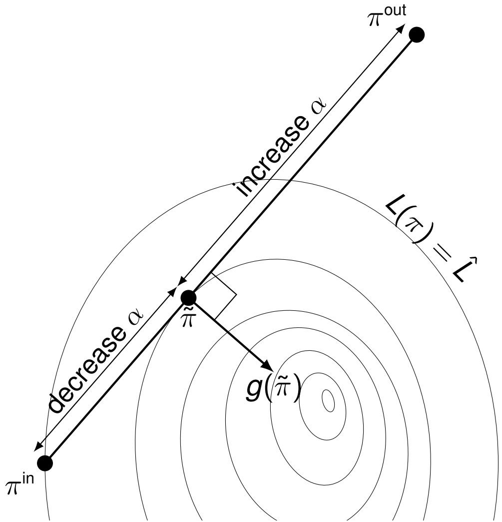
$$(\pi^{\text{out}}, \eta^{\text{out}}) := (\pi^t, \eta^t)$$

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



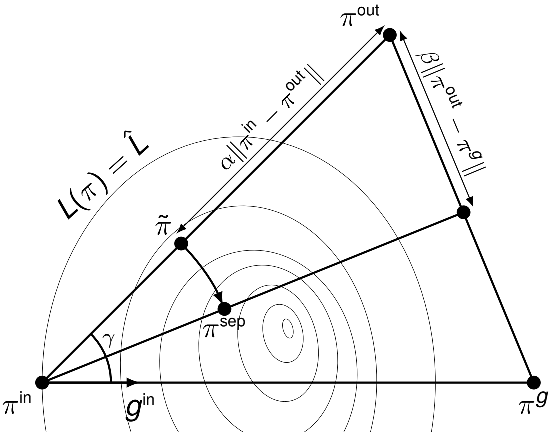


Smoothing: auto-adaptative α-schedule



Directional smoothing

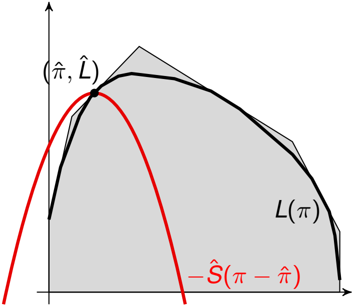
hybridation with ascent methods

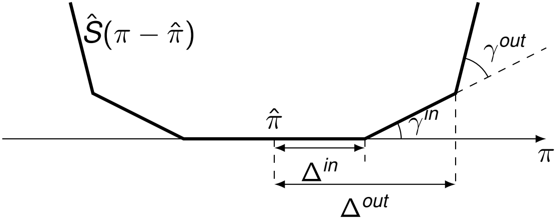


Automatic directional smoothing: $\beta = \cos\gamma$

Penalty functions

$$\pi^t = \underset{\pi \in \mathbb{R}^m_+}{\text{argmax}} \left\{ L^t(\pi) - \hat{S}_t(\pi) \right\}$$





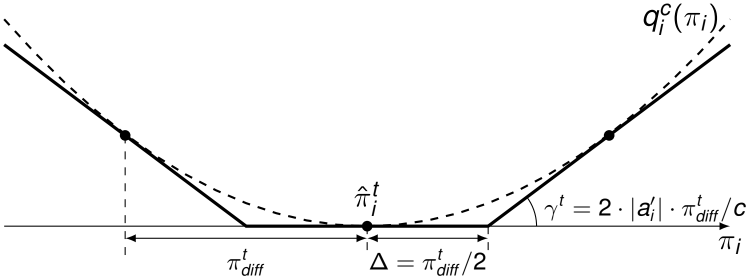
3-pieces: **[du Merle, Villeneuve, Desrosiers, Hansen 99]**

5-pieces: **[Ben Amor, Desrosiers, Frangioni 09]**

“Curvature-based” penalty function

The idea is to **“approximate dynamically” a quadratic function**

$$q_i^c(\pi_i) = |a_i'| \cdot (\pi_i - \hat{\pi}_i)^2 / c.$$

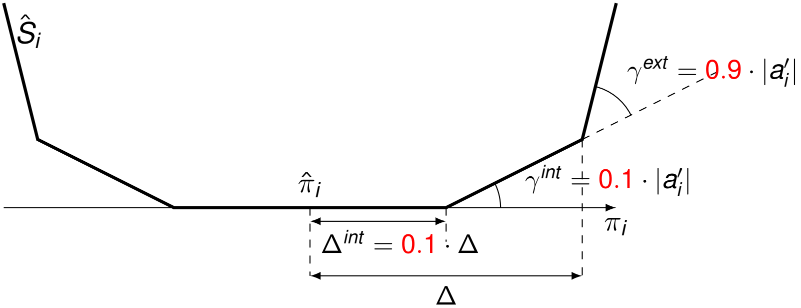


Here:

$$\pi_{diff}^t = \frac{\sum_{i \in M} |\pi_i^t - \hat{\pi}_i^{t-1}|}{|M|}, \quad c = \frac{\pi_{diff}^1}{\kappa}$$

κ is the only parameter.

"Explicit" penalty function



Here:

$$\Delta = \frac{\pi^1_{diff}}{\kappa}.$$

Again, κ is the only parameter.

Contents

Generalized Assignment

Instances

18 OR-Library and similar instances of types D and E with number of agents in $\{5, 10, 20, 40\}$ and jobs in $\{100, 200, 400\}$.

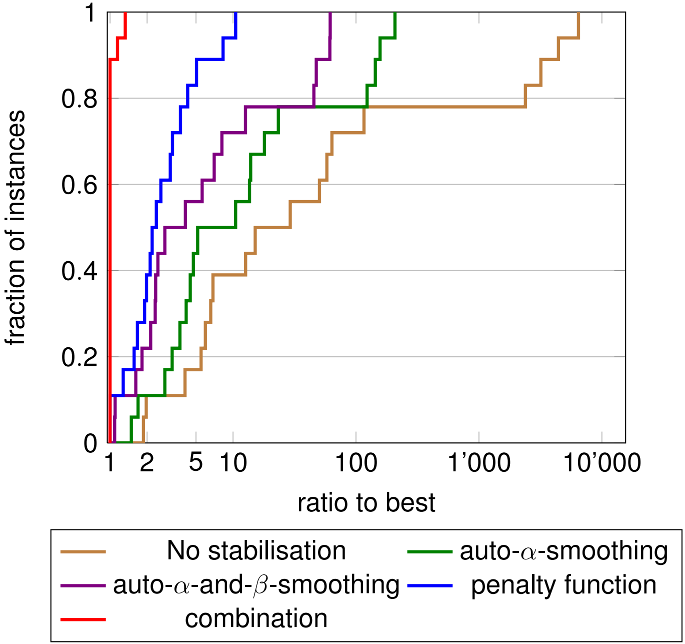
Oracle solver

Pisinger code for the $0-1$ knapsack problem **[Pisinger 97]**.

	It	Sp	Cl	T
Smoothing, best fixed α	3.00	2.72	3.17	4.05
Smoothing, auto α	3.01	2.94	3.32	3.96
Smoothing, auto α, best fixed β	3.82	1.53	6.27	10.06
Auto-smoothing (auto α and β)	3.94	3.58	4.77	7.96
3-pieces curvature-based pen. func. ($\kappa = 80$)	5.32	5.33	5.29	14.57
5-pieces explicit pen. func. ($\kappa = 100$)	4.61	4.62	4.77	17.35
Auto-smooth & 3-p. curv.-bas. pen. func. ($\kappa = 40$)	7.68	6.98	9.54	25.79
Auto-smooth & 3-p. explicit pen. func. ($\kappa = 200$)	6.38	5.94	8.25	43.78

Geometric means of ratios from the “no stabilization” variant are shown

Generalized Assignment: performance profile



Multi-Echelon Small-Bucket Lot-Sizing

Instances

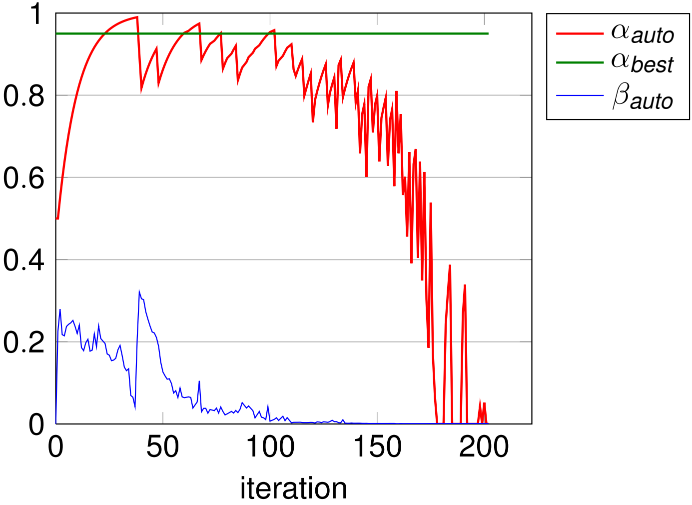
17 randomly generated instances varying by the number of echelons in $\{1, 2, 3, 5\}$, items in $\{10, 20, 40\}$, and periods in $\{50, 100, 200, 400\}$.

Oracle solver

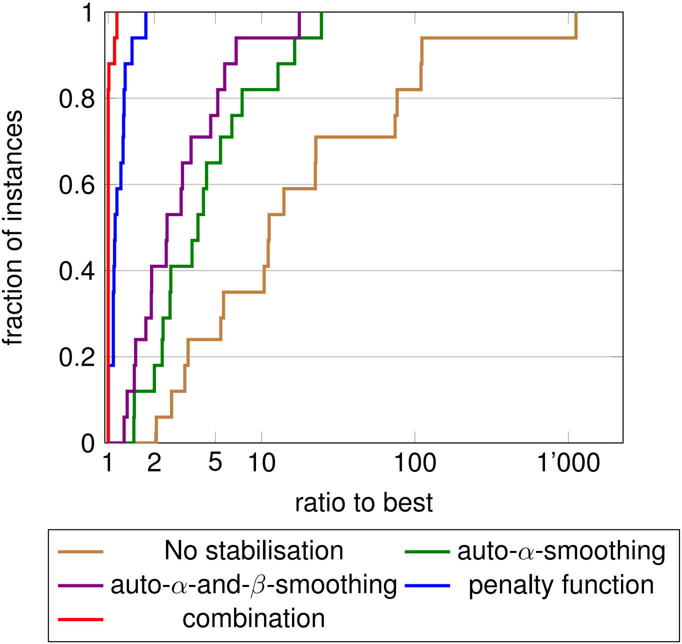
Dynamic programming **[Zangwill 69]**.

	It	Sp	Cl	T
Smoothing, best fixed α	2.26	1.89	3.26	3.41
Smoothing, auto α	2.42	2.26	3.54	4.09
Smoothing, auto α, best fixed β	3.29	2.10	5.10	4.46
Auto-smoothing (auto α and β)	3.29	2.98	4.86	5.99
3-pieces curvature-based pen. func. ($\kappa = 16$)	7.27	7.36	8.02	14.80
5-pieces explicit pen. func. ($\kappa = 10$)	5.63	5.66	6.27	10.96
Auto-smooth & 3-p. curv.-bas. pen. func. ($\kappa = 2$)	8.85	8.49	11.57	17.13
Auto-smooth & 3-p. explicit pen. func. ($\kappa = 10$)	6.71	6.29	9.18	14.68

Multi Echelon Lot Sizing: example of automatic smoothing



Multi Echelon Lot Sizing: performance profile



Parallel machines scheduling

Instances

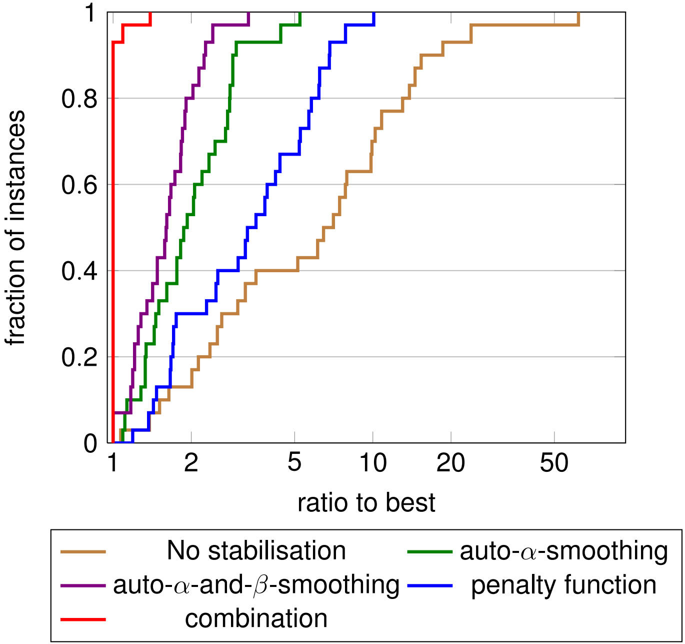
30 OR-Library and similar instances with number of machines in $\{1, 2, 4\}$ and jobs in $\{50, 100, 200\}$.

Oracle solver

Shortest path in an acyclic graph

	It	Sp	Cl	T
Smoothing, best fixed α	2.22	2.12	2.19	2.96
Smoothing, auto α	2.16	2.12	2.14	2.91
Smoothing, auto α, best fixed β	1.33	1.63	2.05	2.42
Auto-smoothing (auto α and β)	2.47	2.40	2.43	3.61
3-pieces curvature-based pen. func. ($\kappa = 4$)	1.61	1.61	1.60	1.77
5-pieces explicit pen. func. ($\kappa = 10$)	1.51	1.51	1.51	2.20
Auto-smooth & 3-p. curv.-bas. pen. func. ($\kappa = 2$)	2.91	2.83	2.87	4.26
Auto-smooth & 3-p. explicit pen. func. ($\kappa = 10$)	2.79	2.69	2.75	5.74

Machine Scheduling: performance profile



Capacitated Vehicle Routing

Instances

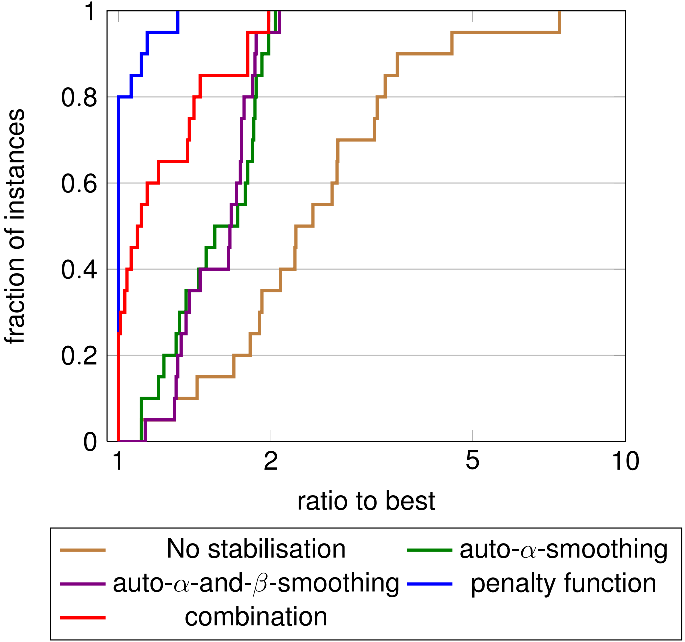
21 widely used instances from the literature of types A, B, E, F, M, P with 50-200 clients and 4-16 vehicles.

Oracle solver

ng-routes **[Baldacci et al. 11]**.

	It	Sp	Cl	T
Smoothing, best fixed α	1.65	1.59	2.01	1.57
Smoothing, auto α	1.67	1.61	2.37	1.55
Smoothing, auto α, best fixed β	1.60	1.47	2.51	1.40
Auto-smoothing (auto α and β)	1.68	1.60	2.56	1.54
3-pieces curvature-based pen. func. ($\kappa = 1.6$)	2.04	2.11	3.34	1.89
5-pieces explicit pen. func. ($\kappa = 1$)	2.59	2.62	3.74	2.34
Auto-α-smooth & 3-p. curv.-b. pen. func. ($\kappa = 0.1$)	2.09	2.06	3.44	1.83
Auto-α-smooth & 3-p. explicit pen. func. ($\kappa = 2$)	2.29	2.17	3.82	2.02

Capacitated Vehicle Routing: performance profile



Multi-Commodity Flow

Instances

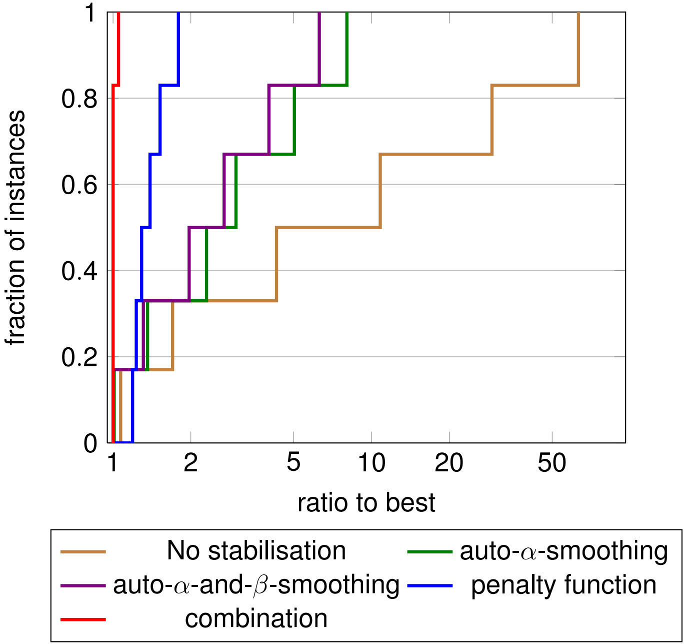
6 practical instances of the freight railcar routing problem **[S. et al. 13]** with 11 commodities, 100-300 thousand nodes, 3.3-10 million arcs, 1'500-5'300 mutual arcs.

Oracle solver

Tree of shortest paths in an acyclic graph (several iterations)

	It	Sp	Cl	T
Smoothing, best fixed α	1.25	1.15	1.32	2.67
Smoothing, auto α	1.57	1.57	1.30	2.73
Smoothing, auto α, best fixed β	1.76	1.76	1.22	3.79
Auto-smoothing (auto α and β)	1.66	1.66	1.24	3.11
3-pieces curvature-based pen. func. ($\kappa = 5$)	1.44	1.48	1.12	4.17
3-pieces explicit pen. func. ($\kappa = 20$)	1.31	1.33	1.07	5.29
Auto-α-smooth & 3-p. curv.-bas. pen. func. ($\kappa = 1$)	1.63	1.66	1.29	5.29
Auto-α-smooth & 3-p. explicit pen. func. ($\kappa = 1$)	1.99	2.01	1.34	7.28

Multi-CommodityFlow: performance profile



Multi-Activity Shift Scheduling

Instances

24 instances from the literature **[Demassey, Pesant, Rousseau 06]** with 20 identical employees, 1 day planning horizon, and 1-10 activities.

Oracle solver

Dynamic programming algorithm **[Côté, Gendron, Rousseau 12]**.

	It	Sp	Cl	T
Smoothing, best fixed α	1.11	1.03	1.41	1.29
Smoothing, auto α	1.19	1.12	1.41	1.28
3-pieces curvature-based pen. func. ($\kappa = 25$)	1.00	1.01	1.04	0.95
5-pieces explicit pen. func. ($\kappa = 0.04$)	1.16	1.4	1.13	1.28
Auto-α-smooth & 3-p. curv.-b. pen. func. ($\kappa = 25$)	1.09	1.08	1.33	1.06
Auto-α-smooth & 3-p. explicit pen. func. ($\kappa = 1$)	1.20	1.14	1.44	1.48

Bin Packing

Instances

12 randomly generated difficult instances with 400-800 items ($\approx$ 2-4 items per bin).

Oracle solver

Pisinger code for the $0-1$ knapsack problem **[Pisinger 97]**.

	It	Sp	Cl	T
Smoothing, best fixed α	1.66	1.64	1.46	2.09
Smoothing, auto α	1.16	1.15	1.13	1.33
Smoothing, auto α, best fixed β	1.64	1.61	1.44	1.88
Auto-smoothing (auto α and β)	1.62	1.60	1.43	1.82

Cutting Stock

Instances

24 randomly generated difficult instances with 250-1'000 items ($\approx$ 2-5 items per bin).

Oracle solver

Pisinger code for the $0-1$ knapsack problem **[Pisinger 97]**.

	`It`	`Sp`	`Cl`	`T`
Smoothing, best fixed α	1.25	1.24	1.14	1.17
Smoothing, auto α	1.12	1.10	1.07	0.93
Smoothing, auto α, best fixed β	1.81	1.77	1.34	1.43
Auto-smoothing (auto α and β)	1.77	1.74	1.33	1.33

Vertex Coloring

Instances

17 DIMACS instances **[Johnson, Trick 93]** ("classic" in the literature).

Oracle solver

Östergård code for the maximum weighted clique problem **[Östergård 01]** or MIP solved by Cplex (depending on the graph sparsity).

	It	Sp	Cl	T
Smoothing, best fixed α	1.45	1.40	1.38	1.22
Smoothing, auto α	1.39	1.31	1.36	1.01

Conclusions

- Automatic smoothing was tested on 8 different problems, and penalty function stabilization on 5 different problems.
- Automatic smoothing reproduces the best (instance-wise!) fixed parameter smoothing performance (except when the subgradient information is very aggregated)
- Directional smoothing gives additional speed-up for some problems.
- Stabilization with a piecewise linear penalty function outperforms smoothing on most problems.
- Combination of two stabilization techniques outperforms these two techniques applied separately on most problems.
- Improvement of 5-piece penalty function over 3-piece is marginal when combined with smoothing or in the “curvature-based” mode.
- Stabilization can make oracle significantly more difficult to solve.