Computational Studies About Stabilization in Column Generation

Antonio Frangioni [1] Hatem M.T. Ben Amor [2] Jacques Desrosiers [3]

[1]Dipartimento di Informatica, Università di Pisa

[2]Ad-Opt Division, Kronos Canadian Systems

[3]HEC Montréal

Column Generation 2008
Aussois, June 18, 2008

Outline

Outline

Outline

1. Column Generation
2. Stabilized Column Generation
3. Computational results I: it works

Outline

Outline

Column Generation

- $\mathcal{A}$ set of columns, $a \in \mathcal{A} \subset \mathbb{R}^m$, $c_a \in \mathbb{R}$, $b \in \mathbb{R}^m$
- Large-scale primal and dual problems:

$$(P) \quad \begin{array}{lll} \max & \sum_{a \in \mathcal{A}} c_a x_a & \\ & \sum_{a \in \mathcal{A}} a x_a & = b \\ & x_a \geq 0 & a \in \mathcal{A} \end{array} \qquad (D) \quad \begin{array}{lll} \min & \pi b & \\ & \pi a \geq c_a & a \in \mathcal{A} \end{array}$$

- $\mathcal{A}$ too large: impossible (or impractical) to solve at once

- $\mathcal{A}$ set of columns, $a \in \mathcal{A} \subset \mathbb{R}^m$, $c_a \in \mathbb{R}$, $b \in \mathbb{R}^m$
- Large-scale primal and dual problems:

$$(P) \quad \begin{array}{lll} \max & \sum_{a \in \mathcal{A}} c_a x_a & \\ & \sum_{a \in \mathcal{A}} a x_a & = b \\ & x_a \geq 0 & a \in \mathcal{A} \end{array} \qquad (D) \quad \begin{array}{lll} \min & \pi b & \\ & \pi a \geq c_a & a \in \mathcal{A} \end{array}$$

- $\mathcal{A}$ too large: impossible (or impractical) to solve at once
- Column Generation (CG): select $\mathcal{B} \subseteq \mathcal{A}$, solve Master problems

$$(P_{\mathcal{B}}) \quad \begin{array}{lll} \max & \sum_{a \in \mathcal{B}} c_a x_a & \\ & \sum_{a \in \mathcal{B}} a x_a & = b \\ & x_a \geq 0 & a \in \mathcal{B} \end{array} \qquad (D_{\mathcal{B}}) \quad \begin{array}{lll} \min & \pi b & \\ & \pi a \geq c_a & a \in \mathcal{B} \end{array}$$

$\Rightarrow$ primal feasible x^* and dual unfeasible π^*

Column Generation (2)

- Then solve pricing (or separation) problem

$$(P_{\pi^*}) \qquad \max\{\ c_a - \pi^* a \ :\ a \in \mathcal{A}\ \}$$

for some $a \in \mathcal{A}/\mathcal{B}$ or optimality certificate $\pi^* a \geq c_a \ \forall a \in \mathcal{A}$

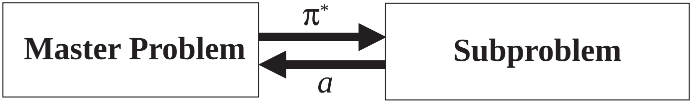
Column Generation (2)

- Then solve pricing (or separation) problem

$$(P_{\pi^*}) \qquad \max\{\ c_a - \pi^* a \ :\ a \in \mathcal{A}\ \}$$

for some $a \in \mathcal{A}/\mathcal{B}$ or optimality certificate $\pi^* a \geq c_a \ \forall a \in \mathcal{A}$

- Very simple idea, very simple implementation (in principle)



... yet surprisingly effective in many applications

... provided that (P_{π^*}) can be efficiently solved

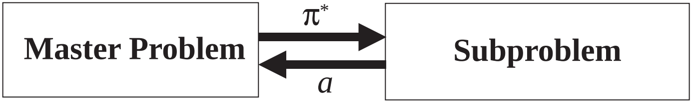
Column Generation (2)

- Then solve pricing (or separation) problem

$$(P_{\pi^*}) \qquad \max\{\ c_a - \pi^* a \ :\ a \in \mathcal{A}\ \}$$

for some $a \in \mathcal{A}/\mathcal{B}$ or optimality certificate $\pi^* a \geq c_a \ \forall a \in \mathcal{A}$

- Very simple idea, very simple implementation (in principle)



. . . yet surprisingly effective in many applications

. . . provided that (P_{π^*}) can be efficiently solved

- x^* feasible $\Rightarrow$ lower bound, but π^* unfeasible $\Rightarrow$ no upper bound

Structure in Column Generation

- In many cases convexity constraint $\sum_{a \in \mathcal{A}} x_a = 1 \Rightarrow$

$$
(D) \quad \begin{array}{ll} \min & \eta + \pi b \\ & \eta \geq c_a - \pi a \quad a \in \mathcal{A} \end{array} \quad \Rightarrow \quad \begin{array}{ll} \min & \pi b + \phi(\pi) \\ & \phi(\pi) = \max\{\, c_a - \pi a \;:\; a \in \mathcal{A} \,\} \end{array}
$$

- ϕ convex, nondifferentiable (polyhedral)

Structure in Column Generation

- In many cases convexity constraint $\sum_{a \in \mathcal{A}} x_a = 1 \Rightarrow$

$$(D) \quad \begin{array}{ll} \min & \eta + \pi b \\ & \eta \geq c_a - \pi a \quad a \in \mathcal{A} \end{array} \quad \Rightarrow \quad \begin{array}{ll} \min & \pi b + \phi(\pi) \\ & \phantom{\pi b + {}} \shortparallel \\ & \max\{\, c_a - \pi a \;:\; a \in \mathcal{A} \,\} \end{array}$$

- ϕ convex, nondifferentiable (polyhedral)
- $(D_{\mathcal{B}}) \Rightarrow \phi_{\mathcal{B}}(\pi) = \max\{\, c_a - \pi a \;:\; a \in \mathcal{B} \,\}$ (cutting plane model of ϕ)
- Each $\phi(\pi)$ provides a valid (Lagrangian) upper bound

Structure in Column Generation

- In many cases convexity constraint $\sum_{a \in \mathcal{A}} x_a = 1 \Rightarrow$

$$(D) \quad \begin{array}{ll} \min & \eta + \pi b \\ & \eta \geq c_a - \pi a \quad a \in \mathcal{A} \end{array} \quad \Rightarrow \quad \begin{array}{ll} \min & \pi b + \phi(\pi) \\ & \phi(\pi) = \max\{\, c_a - \pi a \;:\; a \in \mathcal{A} \,\} \end{array}$$

- ϕ convex, nondifferentiable (polyhedral)
- $(D_{\mathcal{B}}) \Rightarrow \phi_{\mathcal{B}}(\pi) = \max\{\, c_a - \pi a \;:\; a \in \mathcal{B} \,\}$ (cutting plane model of ϕ)
- Each $\phi(\pi)$ provides a valid (Lagrangian) upper bound
- General case: k disjoint convexity constraints, $\mathcal{A} = \mathcal{A}_0 \cup \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_k$

$$\min \{\, \pi b + \phi(\pi) \;:\; \pi \in \Pi \,\} \quad \text{where} \quad \Pi = \{\, \pi \;:\; \pi a \geq c_a \;,\; a \in \mathcal{A}_0 \,\}$$

$$\phi(\pi) = \sum_h \left(\phi^h(\pi) = \max\{\, c_a - \pi a \;:\; a \in \mathcal{A}_h \,\} \right)$$

... minimizing convex polyhedral function over convex polyhedral set

Structure in Column Generation (2)

- The (dual) Master problem

$$(D_{\mathcal{B}}) \qquad \min \left\{ \pi b + \sum_h \phi^h_{\mathcal{B}}(\pi) \; : \; \pi \in \Pi_{\mathcal{B}} \right\}$$

  - $\phi^h_{\mathcal{B}}$ cutting-plane model of ϕ^h
  - $\Pi_{\mathcal{B}} \supseteq \Pi$ outer approximation

[1] J.E. Kelley "The Cutting-Plane Method for Solving Convex Programs" J. of the SIAM 8, 1960

Structure in Column Generation (2)

- The (dual) Master problem

$$(D_{\mathcal{B}}) \qquad \min \left\{ \pi b + \sum_h \phi^h_{\mathcal{B}}(\pi) \ : \ \pi \in \Pi_{\mathcal{B}} \right\}$$

  - $\phi^h_{\mathcal{B}}$ cutting-plane model of ϕ^h
  - $\Pi_{\mathcal{B}} \supseteq \Pi$ outer approximation

- $k + 1$ pricing problems (sometimes only k)
  - $\phi^h(\pi^*)$ give optimality cuts (subgradients of ϕ^h)
  - $\phi^0(\pi^*)$ gives feasibility cuts (faces of Π)

  (sometimes, faces are extreme rays of unbounded pricing problems)

- Something as old as Kelley's cutting plane approach[1]

[1] J.E. Kelley "The Cutting-Plane Method for Solving Convex Programs" J. of the SIAM 8, 1960

Structure in Column Generation (2)

- The (dual) Master problem

$$(D_{\mathcal{B}}) \qquad \min \left\{ \pi b + \sum_h \phi^h_{\mathcal{B}}(\pi) \ : \ \pi \in \Pi_{\mathcal{B}} \right\}$$

  - $\phi^h_{\mathcal{B}}$ cutting-plane model of ϕ^h
  - $\Pi_{\mathcal{B}} \supseteq \Pi$ outer approximation

- $k + 1$ pricing problems (sometimes only k)
  - $\phi^h(\pi^*)$ give optimality cuts (subgradients of ϕ^h)
  - $\phi^0(\pi^*)$ gives feasibility cuts (faces of Π)

  (sometimes, faces are extreme rays of unbounded pricing problems)

- Something as old as Kelley's cutting plane approach[1]

- A well-known drawback: instability

[1] J.E. Kelley "The Cutting-Plane Method for Solving Convex Programs" J. of the SIAM 8, 1960

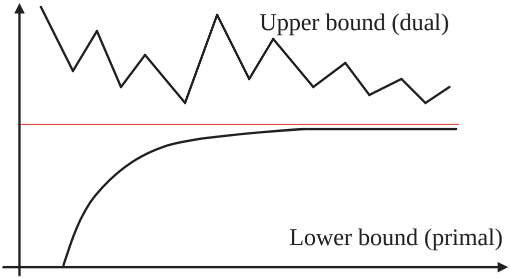
Instability

- $(P_{\mathcal{B}})$ empty $\equiv$ $(D_{\mathcal{B}})$ unbounded $\Rightarrow$ Phase 0 / Phase 1 approach

[2] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, F. Vanderbeck "Comparison of bundle and classical column generation" Math. Prog. 2006

Instability

- $(P_{\mathcal{B}})$ empty $\equiv$ $(D_{\mathcal{B}})$ unbounded $\Rightarrow$ Phase 0 / Phase 1 approach
- More in general: the sequence $\{\pi^*\}$ has no locality properties[2]
  - frequent oscillations of dual values

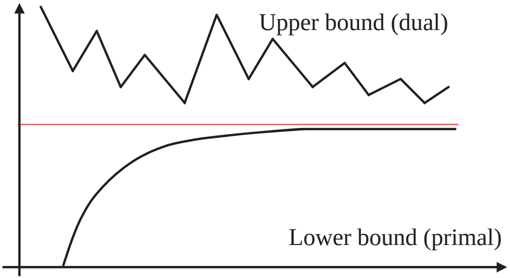


  - "bad quality" of generated columns

[2] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, F. Vanderbeck "Comparison of bundle and classical column generation" Math. Prog. 2006

Instability

- $(P_{\mathcal{B}})$ empty $\equiv$ $(D_{\mathcal{B}})$ unbounded $\Rightarrow$ Phase 0 / Phase 1 approach
- More in general: the sequence $\{\pi^*\}$ has no locality properties[2]
  - frequent oscillations of dual values



  - "bad quality" of generated columns

$\Rightarrow$ tailing off, slow convergence

[2] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, F. Vanderbeck "Comparison of bundle and classical column generation" Math. Prog. 2006

Instability (2)

- In other words: a good estimate of dual optimum is useless!

[3] H. Ben Amor, J. Desrosiers, F. Soumis "Recovering an optimal LP basis from an optimal dual solution" O.R. Lett. 2006

Instability (2)

- In other words: a good estimate of dual optimum is useless!
- . . . even a perfect one. Conceptual experiment on (MDVS)[3]: compute dual optimum, re-solve + dual box constraint of given width

	cpu(s)		CG iter.		SP cols.		MP itrs.	
width		%		%		%		%
∞	4178.4		509		37579		926161	
200.0	835.5	20.0	119	23.4	9368	24.9	279155	30.1
20.0	117.9	2.8	35	6.9	2789	7.4	40599	4.4
2.0	52.0	1.2	20	3.9	1430	3.8	8744	0.9
0.2	47.5	1.1	19	3.7	1333	3.5	8630	0.9

- Convergence speed does not improve near the optimum

[3] H. Ben Amor, J. Desrosiers, F. Soumis "Recovering an optimal LP basis from an optimal dual solution" O.R. Lett. 2006

Instability (2)

- In other words: a good estimate of dual optimum is useless!
- ... even a perfect one. Conceptual experiment on (MDVS)[3]: compute dual optimum, re-solve + dual box constraint of given width

	cpu(s)		CG iter.		SP cols.		MP itrs.	
width		%		%		%		%
∞	4178.4		509		37579		926161	
200.0	835.5	20.0	119	23.4	9368	24.9	279155	30.1
20.0	117.9	2.8	35	6.9	2789	7.4	40599	4.4
2.0	52.0	1.2	20	3.9	1430	3.8	8744	0.9
0.2	47.5	1.1	19	3.7	1333	3.5	8630	0.9

- Convergence speed does not improve near the optimum
- Stabilization is useful

[3] H. Ben Amor, J. Desrosiers, F. Soumis "Recovering an optimal LP basis from an optimal dual solution" O.R. Lett. 2006

Stabilized Column Generation

- CG $\equiv$ CP algorithm $\Rightarrow$ stabilization $\equiv$ (generalized) bundle methods

Stabilized Column Generation

- CG $\equiv$ CP algorithm $\Rightarrow$ stabilization $\equiv$ (generalized) bundle methods
- Current center $\bar{\pi}$, stabilizing term $\mathcal{D} : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$

Stabilized Column Generation

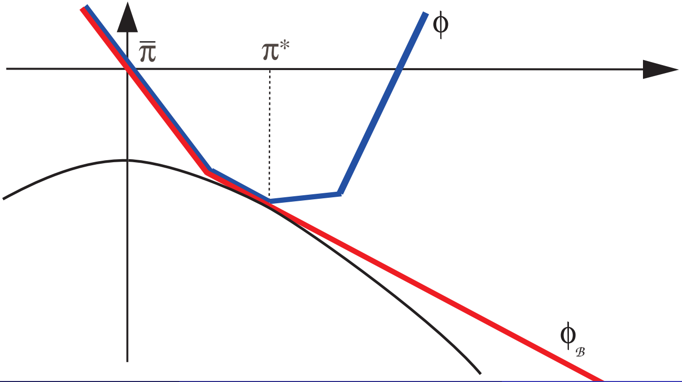
- CG $\equiv$ CP algorithm $\Rightarrow$ stabilization $\equiv$ (generalized) bundle methods
- Current center $\bar{\pi}$, stabilizing term $\mathcal{D} : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$
- Stabilized dual master problem

$$(D_{\mathcal{B},\bar{\pi},\mathcal{D}}) \qquad \min\{\ \phi_{\mathcal{B}}(\pi) + \mathcal{D}(\pi - \bar{\pi})\ :\ \pi \in \Pi_{\mathcal{B}}\ \}$$

Stabilized Column Generation

- CG $\equiv$ CP algorithm $\Rightarrow$ stabilization $\equiv$ (generalized) bundle methods
- Current center $\bar{\pi}$, stabilizing term $\mathcal{D} : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$
- Stabilized dual master problem

$$(D_{\mathcal{B},\bar{\pi},\mathcal{D}}) \qquad \min\{\ \phi_{\mathcal{B}}(\pi) + \mathcal{D}(\pi - \bar{\pi})\ :\ \pi \in \Pi_{\mathcal{B}}\ \}$$



Stabilized Column Generation: Primal View

- Stabilized primal master problem ($k = 1$)

$$(P_{\mathcal{B},\bar{\pi},\mathcal{D}}) \quad \begin{array}{ll} \max & \sum_{a \in \mathcal{B}} c_a x_a + \bar{\pi}\left(b - \sum_{a \in \mathcal{B}} a x_a \right) - \mathcal{D}^*\left(\sum_{a \in \mathcal{B}} a x_a - b \right) \\ & \sum_{a \in \mathcal{B}_1} x_a = 1 \quad , \quad x_a \geq 0 \quad a \in \mathcal{B} \end{array}$$

a (possibly nonquadratic) augmented Lagrangian of $(P_{\mathcal{B}})$

Stabilized Column Generation: Primal View

- Stabilized primal master problem ($k = 1$)

$$(P_{\mathcal{B},\bar{\pi},\mathcal{D}}) \quad \begin{array}{ll} \max & \sum_{a\in\mathcal{B}} c_a x_a + \bar{\pi}\Big(b - \sum_{a\in\mathcal{B}} a x_a\Big) - \mathcal{D}^*\Big(\sum_{a\in\mathcal{B}} a x_a - b\Big) \\ & \sum_{a\in\mathcal{B}_1} x_a = 1 \quad , \quad x_a \geq 0 \quad a \in \mathcal{B} \end{array}$$

  a (possibly nonquadratic) augmented Lagrangian of $(P_{\mathcal{B}})$

- $\bar{\pi}$ = "first-order" Lagrangian term

- $\mathcal{D}^*$ = Fenchel's conjugate of $\mathcal{D}$ = "second-order" term

- A "distance-like" function with a (hard to tune) proximity weight

$$\left(\frac{1}{2t}\|\cdot\|_2^2\right)^* = \frac{1}{2}t\|\cdot\|_2^2 \qquad \left(I_{B_\infty(t)}\right)^* = t\|\cdot\|_1 \qquad \left(\frac{1}{t}\|\cdot\|_1\right)^* = I_{B_\infty(1/t)}$$

Stabilized Column Generation Algorithm

⟨ Initialize $\bar{\pi}$ and $\mathcal{D}$ ⟩
⟨ solve $P_{\bar{\pi}}$, initialize $\mathcal{B}$ with the resulting columns ⟩
repeat
    ⟨ solve $(D_{\mathcal{B},\bar{\pi},\mathcal{D}})/(P_{\mathcal{B},\bar{\pi},\mathcal{D}})$ for π^* and x^* ⟩
    if($\sum_{a \in \mathcal{B}} c_a x_a^* \approx \phi(\bar{\pi})$ **and** $\sum_{a \in \mathcal{B}} a x_a^* \approx b$)
    then stop
    else ⟨ solve P_{π^*}, i.e., compute $\phi(\pi^*)$ ⟩
        ⟨ possibly add some of the resulting columns to $\mathcal{B}$ ⟩
        ⟨ possibly remove columns from $\mathcal{B}$ ⟩
        if($\phi(\pi^*)$ is "substantially lower" than $\phi(\bar{\pi})$)
        then $\bar{\pi} = \pi^*$ /* Serious Step */
        ⟨ possibly update $\mathcal{D}$ ⟩
while(not stop)

- Generic SCG algorithm, allowing many variants
- General and flexible convergence theory

Convergence of Stabilized Column Generation

- Conditions for convergence (not the most general ones)[4]:
  i) $\mathcal{D} \geq 0$ convex, $\mathcal{D}(0) = 0$, $S_\delta(\mathcal{D})$ compact and full-dimensional $\forall \delta > 0$ (these hold for $\mathcal{D}$ if and only if they hold for $\mathcal{D}^*$)
  ii) $\mathcal{D}$ differentiable in 0 $\iff$ $\mathcal{D}^*$ strictly convex in 0
  iii) $\mathcal{D}$ is "steep enough" $\Rightarrow$ $(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ is always bounded
  iv) for some $m \in (0, 1]$, $\phi(\pi^*)$ is "substantially lower" than $\phi(\bar{\pi})$ if
  $$\phi(\pi^*) - \phi(\bar{\pi}) \leq m(\ v(D_{\mathcal{B},\bar{\pi},\mathcal{D}}) - \phi(\bar{\pi})\ )$$
  but SS can be postponed finitely many times, e.g. to "flatten" $\mathcal{D}$
  v) in a sequence of consecutive NS, $\mathcal{D}$ changes finitely many times
  vi) $\mathcal{D} \leq \bar{\mathcal{D}}$ for some $\bar{\mathcal{D}}$ as in i)
  vii) no two removals from $\mathcal{B}$ unless $v(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ increases by $\varepsilon > 0$
- Ensures finite termination
- If $\mathcal{D}^*$ differentiable $|\mathcal{B}|$ can be kept bounded by aggregation
- Allows many variants (linesearch, scatter search, curved search . . .)

[4] F. "Generalized Bundle Methods", SIOPT, 2002

The Proximal Point Case

- No convexity constraint ⇒ only feasibility cuts ⇒ no upper bound ...
  until "full" stabilized primal and dual problems solved

$$(P_{\bar\pi,\mathcal{D}}) \quad \begin{array}{ll} \max & \sum_{a\in\mathcal{A}} c_a x_a + \bar\pi\big(b - \sum_{a\in\mathcal{A}} a x_a\big) - \mathcal{D}^*\big(\sum_{a\in\mathcal{A}} a x_a - b\big) \\ & x_a \geq 0 \quad , \quad a \in \mathcal{A} \end{array}$$

$$(D_{\bar\pi,\mathcal{D}}) \qquad \min\left\{\, \phi(\pi) + \mathcal{D}(\pi - \bar\pi) \;:\; \pi \in \Pi \,\right\}$$

  only at this point $\bar\pi$ can be updated

- Something as old as a (nonquadratic[5]) Proximal Point approach[6]

[5] M. Teboulle "Convergence of Proximal-like Algorithms" SIAM J. on Opt. 7, 1997

[6] R.T. Rockafellar "Monotone Operators and the Proximal Point Algorithm" SIAM J. on Contr. and Opt., 1976

The Proximal Point Case

- No convexity constraint ⇒ only feasibility cuts ⇒ no upper bound . . . until "full" stabilized primal and dual problems solved

$$(P_{\bar\pi,\mathcal{D}}) \quad \begin{array}{ll} \max & \sum_{a\in\mathcal{A}} c_a x_a + \bar\pi\big(b - \sum_{a\in\mathcal{A}} a x_a\big) - \mathcal{D}^*\big(\sum_{a\in\mathcal{A}} a x_a - b\big) \\ & x_a \geq 0 \quad , \quad a \in \mathcal{A} \end{array}$$

$$(D_{\bar\pi,\mathcal{D}}) \qquad \min\left\{\ \phi(\pi) + \mathcal{D}(\pi - \bar\pi)\ :\ \pi \in \Pi\ \right\}$$

  only at this point $\bar\pi$ can be updated

- Something as old as a (nonquadratic[5]) Proximal Point approach[6]
- Large effort to solve $(D_{\bar\pi,\mathcal{D}})$ to optimality even for "bad" $\bar\pi$
- SCG = Bundle method = proximal point with early stop rule

  Bundle method + $m = 1$ ⇒ proximal point method
- Introduce "artificial" convexity constraints if possible

[5] M. Teboulle "Convergence of Proximal-like Algorithms" SIAM J. on Opt. 7, 1997

[6] R.T. Rockafellar "Monotone Operators and the Proximal Point Algorithm" SIAM J. on Contr. and Opt., 1976

Choosing the Stabilizing Term

- Reasonable choices: piecewise-linear functions $\Rightarrow (P_{\mathcal{B},\bar{\pi},\mathcal{D}})$ is a LP
- Something as old as the BoxStep method[7]

$$(D_{\bar{\pi},M}) \qquad \min \left\{ \, \phi(\pi) \; : \; \|\pi - \bar{\pi}\| \leq M \; , \; \pi \in \Pi \, \right\}$$

[7] R.E. Marsten, W.W. Hogan, J.W. Blankenship "The BOXSTEP Method for Large-scale Optimization" Op. Res. 1975

Choosing the Stabilizing Term

- Reasonable choices: piecewise-linear functions $\Rightarrow (P_{\mathcal{B},\bar{\pi},\mathcal{D}})$ is a LP
- Something as old as the BoxStep method[7]

$$(D_{\bar{\pi},M}) \qquad \min \left\{ \phi(\pi) \ : \ \|\pi - \bar{\pi}\| \leq M \ , \ \pi \in \Pi \right\}$$

- The stabilizing term must:
  - have at least one parameter controlling the stabilization
  - make $(P_{\mathcal{B},\bar{\pi},\mathcal{D}})/(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ easy to solve
  - provide a good rate of convergence in practice

[7] R.E. Marsten, W.W. Hogan, J.W. Blankenship "The BOXSTEP Method for Large-scale Optimization" Op. Res. 1975

Choosing the Stabilizing Term

- Reasonable choices: piecewise-linear functions $\Rightarrow$ $(P_{\mathcal{B},\bar{\pi},\mathcal{D}})$ is a LP
- Something as old as the BoxStep method[7]

$$(D_{\bar{\pi},M}) \qquad \min \{ \phi(\pi) \ : \ \|\pi - \bar{\pi}\| \leq M \ , \ \pi \in \Pi \}$$

- The stabilizing term must:
  - have at least one parameter controlling the stabilization
  - make $(P_{\mathcal{B},\bar{\pi},\mathcal{D}})/(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ easy to solve
  - provide a good rate of convergence in practice
- Balancing master problem cost vs convergence speed is nontrivial:
  - master problem cost is often, but not always, predominant
  - less iterations $\Rightarrow$ smaller $\mathcal{B}$ $\Rightarrow$ more efficient

[7] R.E. Marsten, W.W. Hogan, J.W. Blankenship "The BOXSTEP Method for Large-scale Optimization" Op. Res. 1975

Choosing the Stabilizing Term

- Reasonable choices: piecewise-linear functions $\Rightarrow$ $(P_{\mathcal{B},\bar{\pi},\mathcal{D}})$ is a LP
- Something as old as the BoxStep method[7]

$$(D_{\bar{\pi},M}) \qquad \min \{ \, \phi(\pi) \; : \; \|\pi - \bar{\pi}\| \leq M \, , \; \pi \in \Pi \, \}$$

- The stabilizing term must:
  - have at least one parameter controlling the stabilization
  - make $(P_{\mathcal{B},\bar{\pi},\mathcal{D}})/(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ easy to solve
  - provide a good rate of convergence in practice
- Balancing master problem cost vs convergence speed is nontrivial:
  - master problem cost is often, but not always, predominant
  - less iterations $\Rightarrow$ smaller $\mathcal{B}$ $\Rightarrow$ more efficient
- Does BoxStep achieve all this? Not quite

[7] R.E. Marsten, W.W. Hogan, J.W. Blankenship "The BOXSTEP Method for Large-scale Optimization" Op. Res. 1975

Choosing the (Right) Stabilizing Term

- Why BoxStep fails? Scylla / Charybdis situation (a.k.a. Catch 22):
  - M small $\Rightarrow \pi_i^* = \bar{\pi}_i \pm M$, small and independent on problem's data
  - M large $\Rightarrow$ no stabilization

[8] S. Kim, K.N. Chang, J.Y. Lee "A Descent Method with L.P. Subproblems for Nondiff. Convex Opt." Math. Prog. 1995

[9] O. du Merle, D. Villeneuve, J. Desrosiers, P. Hansen "Stabilized Column Generation" Disc. Math. 1999

[10] C. Lemaréchal "Bundle Methods in Nonsmooth Optimization" in *Nonsmooth Optimization*, 1978

[11] M.C. Pinar, S.A. Zenios "On Smoothing Exact Penalty Functions for Convex Constrained Optimization" SIOPT, 1994

[12] M.D. Grigoriadis, L.G. Kahchiyan "An exponential-function reduction method for block-angular convex programs" Networks, 1995

Choosing the (Right) Stabilizing Term

- Why BoxStep fails? Scylla / Charybdis situation (a.k.a. Catch 22):
  - M small $\Rightarrow \pi_i^* = \bar{\pi}_i \pm M$, small and independent on problem's data
  - M large $\Rightarrow$ no stabilization
- The theory allows a lot, in particular mixing penalty with trust region

[8] S. Kim, K.N. Chang, J.Y. Lee "A Descent Method with L.P. Subproblems for Nondiff. Convex Opt." Math. Prog. 1995
[9] O. du Merle, D. Villeneuve, J. Desrosiers, P. Hansen "Stabilized Column Generation" Disc. Math. 1999
[10] C. Lemaréchal "Bundle Methods in Nonsmooth Optimization" in *Nonsmooth Optimization*, 1978
[11] M.C. Pinar, S.A. Zenios "On Smoothing Exact Penalty Functions for Convex Constrained Optimization" SIOPT, 1994
[12] M.D. Grigoriadis, L.G. Kahchiyan "An exponential-function reduction method for block-angular convex programs" Networks, 1995

Choosing the (Right) Stabilizing Term

- Why BoxStep fails? Scylla / Charybdis situation (a.k.a. Catch 22):
  - M small $\Rightarrow \pi_i^* = \bar{\pi}_i \pm M$, small and independent on problem's data
  - M large $\Rightarrow$ no stabilization
- The theory allows a lot, in particular mixing penalty with trust region
- Thus, increase the number of pieces. But how many?
  - Two[8]? (too few)
  - Three[9]? (still too few)
  - Infinitely many[10,2]? (perhaps too many?)
- Piecewise-quadratic[11] or exponential[12] also possible (but why?)

---

[8] S. Kim, K.N. Chang, J.Y. Lee "A Descent Method with L.P. Subproblems for Nondiff. Convex Opt." Math. Prog. 1995

[9] O. du Merle, D. Villeneuve, J. Desrosiers, P. Hansen "Stabilized Column Generation" Disc. Math. 1999

[10] C. Lemaréchal "Bundle Methods in Nonsmooth Optimization" in *Nonsmooth Optimization*, 1978

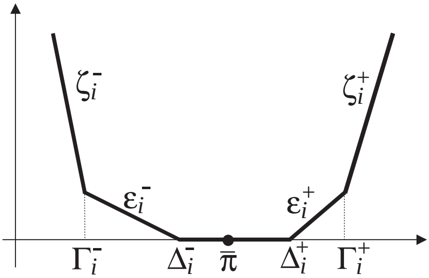
[11] M.C. Pinar, S.A. Zenios "On Smoothing Exact Penalty Functions for Convex Constrained Optimization" SIOPT, 1994

[12] M.D. Grigoriadis, L.G. Kahchiyan "An exponential-function reduction method for block-angular convex programs" Networks, 1995

A 5-piecewise-linear Function

$\mathcal{D}(u) = \sum_{i=1}^{m} \mathcal{D}_i(u_i)$ where

$$\mathcal{D}_i(u_i) = \begin{cases} -(\zeta_i^- + \varepsilon_i^-)\ (u_i + \Gamma_i^-) - \zeta_i^- \Delta_i^- & -\infty \leq u_i \leq -\Gamma_i^- - \Delta_i^- \\ -\varepsilon_i^-(\ u_i - \Delta_i^-) & -\Gamma_i^- - \Delta_i^- \leq u_i \leq -\Delta_i^- \\ 0 & -\Delta_i^- \leq u_i \leq \Delta_i^+ \\ +\varepsilon_i^+(\ u_i - \Delta_i^+) & \Delta_i^+ \leq u_i \leq \Delta_i^+ + \Gamma_i^+ \\ +(\varepsilon_i^+ + \zeta_i^+)\ (u_i - \Gamma_i^+) - \zeta_i^+ \Delta_i^+ & \Delta_i^+ + \Gamma_i^+ \leq u_i \leq +\infty \end{cases}$$



Trust region on $\bar{\pi}$ + small penalty close + much larger penalty far

Why A 5-piecewise-linear Function

- Many parameters: interval widths $\Gamma^{\pm}/\Delta^{\pm}$, penalty costs $\zeta^{\pm}/\varepsilon^{\pm}$

Why A 5-piecewise-linear Function

- Many parameters: interval widths $\Gamma^{\pm}/\Delta^{\pm}$, penalty costs $\zeta^{\pm}/\varepsilon^{\pm}$
- Different roles for small and large penalty costs:
  - large penalties $\zeta^{\pm}$ to make $(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ bounded $\Rightarrow$ no Phase 0
  - small penalties $\varepsilon^{\pm}$ take into account problem's data

Why A 5-piecewise-linear Function

- Many parameters: interval widths $\Gamma^{\pm}/\Delta^{\pm}$, penalty costs $\zeta^{\pm}/\varepsilon^{\pm}$
- Different roles for small and large penalty costs:
  - large penalties $\zeta^{\pm}$ to make $(D_{\mathcal{B},\bar{\pi},\mathcal{D}})$ bounded $\Rightarrow$ no Phase 0
  - small penalties $\varepsilon^{\pm}$ take into account problem's data
- Its Fenchel's conjugate: $\mathcal{D}^*(y) = \sum_{i=1}^m \mathcal{D}_i^*(y_i)$ where

$$\mathcal{D}_i^*(y_i) = \begin{cases} +\infty & y_i < -(\zeta_i^- + \varepsilon_i^-) \\ -(\Gamma_i^- + \Delta_i^-)\ y_i - \Gamma_i^-\varepsilon_i^- & -\zeta_i^- - \varepsilon_i^- \leq y_i \leq -\varepsilon_i^- \\ -\Delta_i^-\ y_i & -\varepsilon_i^- \leq y_i \leq 0 \\ +\Delta_i^+\ y_i & 0 \leq y_i \leq \varepsilon_i^- \\ +(\Gamma_i^+ + \Delta_i^+)\ y_i - \Gamma_i^+\varepsilon_i^+ & \varepsilon_i^+ \leq y_i \leq (\zeta_i^+ + \varepsilon_i^+) \\ +\infty & y_i > (\zeta_i^+ + \varepsilon_i^+) \end{cases}$$

4-piecewise-linear, outer box $[-(\Gamma_i^- + \Delta_i^-), -(\Gamma_i^- + \Delta_i^-)]$,
interval widths $\Longleftrightarrow$ penalties

The Corresponding Master Problems

- Notation: $\gamma^{\pm} = \bar{\pi} \pm \Delta^{\pm} \pm \Gamma^{\pm}$, $\delta^{\pm} = \bar{\pi} \pm \Delta^{\pm}$

The Corresponding Master Problems

- Notation: $\gamma^{\pm} = \bar{\pi} \pm \Delta^{\pm} \pm \Gamma^{\pm}$, $\delta^{\pm} = \bar{\pi} \pm \Delta^{\pm}$
- Stabilized dual master problem (case $\mathcal{A} = \mathcal{A}_0$):

$$
(D_{\mathcal{B},\bar{\pi},\mathcal{D}}) \quad
\begin{array}{ll}
\min & \pi b + \zeta^- v^- + \varepsilon^- u^- + \varepsilon^+ u^+ + \zeta^+ v^+ \\
 & -u^- + \delta^- \leq \pi \leq \delta^+ + u^+ \\
 & -v^- + \gamma^- \leq \pi \leq \gamma^+ + v^+ \\
 & \pi a \leq c_a \quad , \quad a \in \mathcal{B} \\
 & v^-, u^-, u^+, v^+ \geq 0
\end{array}
$$

The Corresponding Master Problems

- Notation: $\gamma^{\pm} = \bar{\pi} \pm \Delta^{\pm} \pm \Gamma^{\pm}$, $\delta^{\pm} = \bar{\pi} \pm \Delta^{\pm}$
- Stabilized dual master problem (case $\mathcal{A} = \mathcal{A}_0$):

$$
(D_{\mathcal{B},\bar{\pi},\mathcal{D}}) \quad
\begin{array}{ll}
\min & \pi b + \zeta^- v^- + \varepsilon^- u^- + \varepsilon^+ u^+ + \zeta^+ v^+ \\
& -u^- + \delta^- \leq \pi \leq \delta^+ + u^+ \\
& -v^- + \gamma^- \leq \pi \leq \gamma^+ + v^+ \\
& \pi a \leq c_a \quad , \quad a \in \mathcal{B} \\
& v^-, u^-, u^+, v^+ \geq 0
\end{array}
$$

- Stabilized primal master problem:

$$
(P_{\mathcal{B},\bar{\pi},\mathcal{D}}) \quad
\begin{array}{ll}
\max & \sum_{a \in \mathcal{B}} c_a x_a + \gamma^- z^- + \delta^- y^- - \delta^+ y^+ - \gamma^+ z^+ \\
& \sum_{a \in \mathcal{B}} a x_a + z^- + y^- - y^+ - z^+ = b \\
& z^- \leq \zeta^- \quad , \quad y^- \leq \varepsilon^- \quad , \quad y^+ \leq \varepsilon^+ \quad , \quad z^+ \leq \zeta^+ \\
& z^-, \ y^-, \ y^+, \ z^+ \ \geq 0 \quad x_a \geq 0 \quad a \in \mathcal{B}
\end{array}
$$

as many constraints as $(P_{\mathcal{B}})$, 4 variables for each stabilized constraint

A 5-to-3-pieces Variant

- Large penalties inactive near optimum ⇒ dynamic 5-to-3-pieces

	CG	PP-5	5-3	CG	PP-5	5-3	CG	PP-5	5-3
	p1			*p3*			*p5*		
time(s)	204	43	37	285	45	49	3562	306	256
mp(s)	126	14	12	181	20	18	2676	142	114
itr	149	52	48	196	53	58	422	80	72
	p6			*p7*			*p8*		
time(s)	4178	596	501	2883	1224	1068	1429	837	757
mp(s)	3149	216	166	1641	611	593	779	480	272
itr	509	196	160	380	178	145	259	145	134

A 5-to-3-pieces Variant

- Large penalties inactive near optimum ⇒ dynamic 5-to-3-pieces

	CG	PP-5	5-3	CG	PP-5	5-3	CG	PP-5	5-3
	p1			*p3*			*p5*		
time(s)	204	43	37	285	45	49	3562	306	256
mp(s)	126	14	12	181	20	18	2676	142	114
itr	149	52	48	196	53	58	422	80	72
	p6			*p7*			*p8*		
time(s)	4178	596	501	2883	1224	1068	1429	837	757
mp(s)	3149	216	166	1641	611	593	779	480	272
itr	509	196	160	380	178	145	259	145	134

- Master problem cost decreases, iterations count does not increase
- Pure Proximal test, but extends to Bundle
- Per-constraint parameter handling (flexibility but complexity)

Computational results: MDVS

- Multiple-Depot Vehicle Scheduling:
  - two-components route costs: fixed vehicle cost + arc costs
  - number of tasks $m \in \{400, 800, 1000, 1200\}$
  - type $T \in \{A, B\}$ (location of depots)
  - number of depots $d \in \{4, 5\}$
- $m + d$ constraints in the master problem, d subproblems
- Initialization by MCF $\Rightarrow$ initial solution and good initial $\bar{\pi}$
- High fixed cost + initial solution $\Rightarrow$ "artificial" convexity constraint

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$
T	A	A	B	B	A	B	A	B	A	B
m	400	400	400	400	800	800	1000	1000	1200	1200
d	4	4	4	4	4	4	5	5	4	4
Arcs	2.1e5	2.1e5	2.1e5	2.0e5	7.9e5	8.2e5	1.3e6	9.7e5	1.5e6	1.1e6

Computational results: MDVS (2)

		p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
cpu	CG	139	177	235	159	3138	3966	3704	1742	3685	3065
	PP	31	36	38	28	482	335	946	572	1065	2037
	BM	26	28	35	21	295	257	639	352	545	1505
mp	CG	88	125	165	105	1679	2004	1955	925	1984	1743
	PP	13	16	17	10	189	128	428	257	542	1326
	BM	10	14	15	10	100	70	329	206	334	983
itr	CG	117	149	200	165	408	524	296	186	246	247
	PP	47	47	49	45	93	64	98	83	86	150
	BM	37	43	44	36	57	53	59	49	51	101

- both PP and BM improve upon CG both in iterations count and time
- BM better that PP on large instances where initial $\bar{\pi}$ is worse
- master problem time for BM slightly larger (more SSs)

Computational results: VCS

- Simultaneous Vehicle & Crew Scheduling: cover trips $\rightarrow$ segments $\rightarrow$ duties (with deadheading), set departure times from parkings

Computational results: VCS

- Simultaneous Vehicle & Crew Scheduling: cover trips $\longrightarrow$ segments $\longrightarrow$ duties (with deadheading), set departure times from parkings
- Subproblems:
  - constrained shortest paths with up to 7 resources $\Rightarrow$ expensive, or
  - many subnetworks (one for each departure time) $\Rightarrow$ less resources

Computational results: VCS

- Simultaneous Vehicle & Crew Scheduling: cover trips → segments → duties (with deadheading), set departure times from parkings
- Subproblems:
  - constrained shortest paths with up to 7 resources ⇒ expensive, or
  - many subnetworks (one for each departure time) ⇒ less resources
- Solve only a small subset (10-20) of networks at all iterations . . .
  . . . but the last one for proving optimality

Problem	Cov	Flow	Net	Nodes	Arcs
*p*199	199	897	822	1528	3653
*p*204	204	919	829	1577	3839
*p*206	206	928	835	1569	3861
*p*262	262	1180	973	1908	4980
*p*315	315	1419	1039	2180	6492
*p*344	344	1549	1090	2335	7210
*p*463	463	2084	1238	2887	9965

Computational results: VCS (2)

		p199	p204	p206	p262	p315	p344	p463
cpu(min)	CG	26	26	30	68	142	238	662
	BM	12	13	14	40	73	163	511
mp(min)	CG	13	9	14	35	43	90	273
	BM	3	3	4	7	19	20	93
sp(min)	CG	13	17	16	33	99	148	389
	BM	9	10	10	33	54	143	418
itr	CG	167	129	245	263	239	303	382
	BM	116	119	173	160	213	201	333

- Pure Proximal worse than CG: proving optimality too many times
- Bundle Method improves upon CG both in iterations and time
- ...even if subproblem time sometimes increases
- "rough" BM with no "flattening" ever of $\mathcal{D}$

Guidelines for Choosing the Stabilizing Term

- Study the (combined) impact of three factors:
  - “shape” of the stabilizing term
  - “steepness” of the stabilizing term
  - quality of initial dual estimate

Guidelines for Choosing the Stabilizing Term

- Study the (combined) impact of three factors:
  - "shape" of the stabilizing term
  - "steepness" of the stabilizing term
  - quality of initial dual estimate
- As few parameters as possible (symmetric, uniform, normalized):
  - Quadratic ST (Q): $t = 10^j$ for $j \in T = \{7, 5, 3, 2, 1\}$
  - Boxstep ST (1P): $\Delta \in \{1000, 500, 100, 10, 1\}$
  - 3-pieces ST (3P): Δ, $t\varepsilon = 2\Delta$ (tangent to Q)
  - 5-pieces linear ST (5P): for each (Δ, ε) in 3P, two sub-intervals of width $\Delta/2$, slopes 1 and $\varepsilon - 1.0$ (outer, inner)
- 5 Q, 5 1P algorithms, $\leq$ 25 3P and 5P ("useless" variants removed)

Guidelines for Choosing the Stabilizing Term

- Study the (combined) impact of three factors:
  - "shape" of the stabilizing term
  - "steepness" of the stabilizing term
  - quality of initial dual estimate
- As few parameters as possible (symmetric, uniform, normalized):
  - Quadratic ST (Q): $t = 10^j$ for $j \in T = \{7, 5, 3, 2, 1\}$
  - Boxstep ST (1P): $\Delta \in \{1000, 500, 100, 10, 1\}$
  - 3-pieces ST (3P): Δ, $t\varepsilon = 2\Delta$ (tangent to Q)
  - 5-pieces linear ST (5P): for each (Δ, ε) in 3P, two sub-intervals of width $\Delta/2$, slopes 1 and $\varepsilon - 1.0$ (outer, inner)
- 5 Q, 5 1P algorithms, $\leq$ 25 3P and 5P ("useless" variants removed)
- No dynamic adjustment, favors "simple" approaches

- Initial dual points: given dual optimal solution $\tilde{\pi}$
  - α-points: $\alpha\tilde{\pi}$ for $\alpha \in \{0.9, 0.75, 0.5, 0.25, 0.0\}$ (feasible as 0 is)
  - random points such that $\delta_1 \leq \|\pi - \tilde{\pi}\|_\infty \leq \delta_2$ for $(\delta^1, \delta^2) \in \{(0, 0.5), (0, 1), (0.5, 1)\}$.

(α-points are better structured than random ones)

[13] H. Ben Amor, J. Desrosiers "A Proximal Trust Region Algorithm for Col. Gen. Stabilization" Comput. & O.R. 2006

[14] D. Carpaneto, M. Dall'Amico, M. Fischetti, P. Toth "A Branch and Bound Algorithm for the Multiple Depot Vehicle Scheduling Problem" Networks 1989

[15] A. Oukil, H. Ben Amor, J. Desrosiers "Stabilized Column Generation for Highly Degenerate Multiple-Depot Vehicle Scheduling Problems" Comput. & O.R. 2006

Guidelines for . . .

- Initial dual points: given dual optimal solution $\tilde{\pi}$
  - α-points: $\alpha\tilde{\pi}$ for $\alpha \in \{0.9, 0.75, 0.5, 0.25, 0.0\}$ (feasible as 0 is)
  - random points such that $\delta_1 \leq \|\pi - \tilde{\pi}\|_\infty \leq \delta_2$ for $(\delta^1, \delta^2) \in \{(0, 0.5), (0, 1), (0.5, 1)\}$.

  (α-points are better structured than random ones)

- The Test Instances:
  - MDVS[13] (easy)
  - resource-constrained Urban Bus Scheduling[14] (hard)
  - Long-Horizon (weekly) MDVS[15] (harder)

- Stop: $\leq 10^{-4}$/1500 iterations ($\leq 10^{-7}$/700 iterations for MDVS)

[13] H. Ben Amor, J. Desrosiers "A Proximal Trust Region Algorithm for Col. Gen. Stabilization" Comput. & O.R. 2006

[14] D. Carpaneto, M. Dall'Amico, M. Fischetti, P. Toth "A Branch and Bound Algorithm for the Multiple Depot Vehicle Scheduling Problem" Networks 1989

[15] A. Oukil, H. Ben Amor, J. Desrosiers "Stabilized Column Generation for Highly Degenerate Multiple-Depot Vehicle Scheduling Problems" Comput. & O.R. 2006

MDVS: Comparing *k*P, Using Initial Dual α-points

Δ		1000					500					100				
alg	CG	1P	3P		5P		1P	3P		5P		1P	3P		5P	
t			10^5	10^3	10^5	10^3		10^5	10^3	10^5	10^3		10^3	10^2	10^3	10^2
p1	134	85	110	79	110	72	80	118	72	120	65	122	74	58	75	58
p2	151	92	117	92	114	84	84	115	83	111	77	119	84	81	80	83
p3	183	117	162	114	161	99	109	163	96	158	89	129	103	80	94	81
p4	137	83	124	83	115	80	79	123	76	120	76	115	81	75	78	74
p5	592	343	481	256	468	222	260	498	223	493	194	200	274	241	278	279
p6	505	195	394	177	384	150	158	423	149	422	126	151	177	125	181	121
p7	287	152	271	125	275	110	125	284	110	281	107	181	164	141	160	159
p8	192	126	172	108	178	96	107	190	97	187	94	184	139	115	142	118
p9	258	161	224	127	215	109	140	225	110	223	101	179	130	140	130	170
p10	298	214	244	144	238	120	177	242	128	257	120	254	146	163	147	207
0.9	274	139	227	133	226	113	112	236	113	238	91	68	122	84	119	83
0.75	274	157	228	130	224	109	130	236	111	236	99	100	128	100	132	112
0.5	274	167	231	126	225	115	132	240	115	232	105	174	137	125	141	149
0.25	274	161	227	131	225	118	142	237	116	240	109	229	144	147	146	161
0	274	159	236	133	230	117	145	242	118	240	120	247	155	155	146	170

MDVS: Comparing kP, Using Initial Dual α-points (2)

Δ		100					10							1				
alg	CG	1P	3P		5P		1P	3P			5P			1P	3P		5P	
t			10^3	10^2	10^3	10^2		10^3	10^2	10	10^3	10^2	10		10^2	10	10^2	10
p1	134	122	74	58	75	58	408	115	73	110	113	73	111	255	123	96	126	97
p2	151	119	84	81	80	83	387	118	88	130	112	104	145	306	133	110	126	107
p3	183	129	103	80	94	81	473	156	97	122	150	106	150	442	168	122	166	125
p4	137	115	81	75	78	74	396	122	79	99	128	81	111	271	131	101	127	100
p5	592	200	274	241	278	279	544	474	346	388	475	332	406	620	481	336	489	335
p6	505	151	177	125	181	121	556	368	192	205	369	194	239	599	386	203	377	205
p7	287	181	164	141	160	159	554	289	185	244	275	183	286	601	286	204	282	193
p8	192	184	139	115	142	118	466	183	150	174	181	152	179	508	189	156	188	154
p9	258	179	130	140	130	170	505	222	178	235	219	179	233	531	242	178	232	183
p10	298	254	146	163	147	207	566	232	160	302	231	151	329	655	244	176	246	176
0.9	274	68	122	84	119	83	214	218	139	155	215	139	179	541	231	155	235	156
0.75	274	100	128	100	132	112	380	221	148	187	222	147	205	305	237	160	231	161
0.5	274	174	137	125	141	149	561	225	154	217	230	163	223	516	239	167	232	168
0.25	274	229	144	147	146	161	650	230	160	215	228	166	240	518	237	175	238	171
0	274	247	155	155	146	170	623	245	173	232	233	163	248	514	248	185	245	181

Comments

- First half: average on α, instance-wise (robustness to initial point)
- Second half: average on instance, α-wise (dependence to initial point)

Comments

- First half: average on α, instance-wise (robustness to initial point)
- Second half: average on instance, α-wise (dependence to initial point)

- "Weak" penalties already better than CG (is $+\infty$ weak?)

Comments

- First half: average on α, instance-wise (robustness to initial point)
- Second half: average on instance, α-wise (dependence to initial point)

- "Weak" penalties already better than CG (is $+\infty$ weak?)
- An "intermediate" stabilization is better:
  - intermediate Δ values are best (not too large or too small)
  - something similar happens for t (better seen next)

Comments

- First half: average on α, instance-wise (robustness to initial point)
- Second half: average on instance, α-wise (dependence to initial point)

- "Weak" penalties already better than CG (is $+\infty$ weak?)
- An "intermediate" stabilization is better:
  - intermediate Δ values are best (not too large or too small)
  - something similar happens for t (better seen next)
- 1P has best overall performance for $\alpha = 0.9$ and $\Delta = 100$
  however performance quickly degrades farther from $\tilde{\pi}$
- 3P and 5P are much more robust, both on α and "extreme" Δ, t

Comments

- First half: average on α, instance-wise (robustness to initial point)
- Second half: average on instance, α-wise (dependence to initial point)

- "Weak" penalties already better than CG (is $+\infty$ weak?)
- An "intermediate" stabilization is better:
  - intermediate Δ values are best (not too large or too small)
  - something similar happens for t (better seen next)
- 1P has best overall performance for $\alpha = 0.9$ and $\Delta = 100$ however performance quickly degrades farther from $\tilde{\pi}$
- 3P and 5P are much more robust, both on α and "extreme" Δ, t
- For each Δ either 3P or 5P outperforms 1P for at least one t
- Most often 5P better than 3P despite the very "rigid" shape

MDVS: Comparing Q, Using Initial Dual α-points

t		10^7		10^5				10^3							
alg	CG	1P	Q	1P	3P	5P	Q	1P	3P			5P			Q
Δ		1000		500	1000	1000		100	1000	500	100	1000	500	100	
p1	134	85	100	80	110	110	88	122	79	72	74	72	65	75	63
p2	151	92	112	84	117	114	103	119	92	83	84	84	77	80	86
p3	183	117	146	109	162	161	125	129	114	96	103	99	89	94	103
p4	137	83	115	79	124	115	98	115	83	76	81	80	76	78	73
p5	592	343	422	260	481	468	288	200	256	223	274	222	194	278	213
p6	505	195	318	158	394	384	199	151	177	149	177	150	126	181	109
p7	287	152	223	125	271	275	155	181	125	110	164	110	107	160	118
p8	192	126	162	107	172	178	126	184	108	97	139	96	94	142	112
p9	258	161	213	140	224	215	152	179	127	110	130	109	101	130	132
p10	298	214	204	177	244	238	148	254	144	128	146	120	120	147	153
0.9	274	139	200	112	227	226	149	68	133	113	122	113	91	119	84
0.75	274	157	200	130	228	224	148	100	130	111	128	109	99	132	94
0.5	274	167	203	132	231	225	147	174	126	115	137	115	105	141	114
0.25	274	161	202	142	227	225	149	229	131	116	144	118	109	146	139
0.0	274	159	203	145	236	230	149	247	133	118	155	117	120	146	151

MDVS: Comparing Q, Using Initial Dual α-points (2)

t		10^2						10					
alg	CG	1P	3P		5P		Q	1P	3P		5P		Q
Δ		10	100	10	100	10		1	10	1	10	1	
p1	134	408	58	73	58	73	188	255	110	96	111	97	423
p2	151	387	81	88	83	104	243	306	130	110	145	107	385
p3	183	473	80	97	81	106	261	442	122	122	150	125	494
p4	137	396	75	79	74	81	235	271	99	101	111	100	499
p5	592	544	241	346	279	332	336	620	388	336	406	335	583
p6	505	556	125	192	121	194	300	599	205	203	239	205	610
p7	287	554	141	185	159	183	316	601	244	204	286	193	568
p8	192	466	115	150	118	152	212	508	174	156	179	154	513
p9	258	505	140	178	170	179	263	531	235	178	233	183	539
p10	298	566	163	160	207	151	339	655	302	176	329	176	594
0.9	274	214	84	139	83	139	134	541	155	155	179	156	269
0.75	274	380	100	148	112	147	199	305	187	160	205	161	431
0.5	274	561	125	154	149	163	286	516	217	167	223	168	585
0.25	274	650	147	160	161	166	338	518	215	175	240	171	662
0.0	274	623	155	173	170	163	390	514	232	185	248	181	658

Comments (2)

- Weak penalties: 1P better than Q, Q better than 3P and 5P (Q has no $+\infty \Rightarrow$ too weak, 3P and 5P weaker than Q)

Comments (2)

- Weak penalties: 1P better than Q, Q better than 3P and 5P
  (Q has no $+\infty \Rightarrow$ too weak, 3P and 5P weaker than Q)
- Mid penalties: Q better than 1P, often outperforms 3P and 5P
- Q more competitive w.r.t. 1P as α decreases
- 3P and 5P more competitive w.r.t. Q as α decreases, better for low α

Comments (2)

- Weak penalties: 1P better than Q, Q better than 3P and 5P (Q has no $+\infty \Rightarrow$ too weak, 3P and 5P weaker than Q)
- Mid penalties: Q better than 1P, often outperforms 3P and 5P
- Q more competitive w.r.t. 1P as α decreases
- 3P and 5P more competitive w.r.t. Q as α decreases, better for low α
- Strong penalties: 3P and 5P better than Q, all the more if α low

Comments (2)

- Weak penalties: 1P better than Q, Q better than 3P and 5P (Q has no $+\infty \Rightarrow$ too weak, 3P and 5P weaker than Q)
- Mid penalties: Q better than 1P, often outperforms 3P and 5P
- Q more competitive w.r.t. 1P as α decreases
- 3P and 5P more competitive w.r.t. Q as α decreases, better for low α
- Strong penalties: 3P and 5P better than Q, all the more if α low
- Summing up:
  - for very good dual estimate and good Δ, 1P wins
  - Q much more robust w.r.t. dual estimate, very good for proper t
  - 3P and 5P even better for proper Δ (same t), even more robust

Comments (2)

- Weak penalties: 1P better than Q, Q better than 3P and 5P
  (Q has no $+\infty \Rightarrow$ too weak, 3P and 5P weaker than Q)
- Mid penalties: Q better than 1P, often outperforms 3P and 5P
- Q more competitive w.r.t. 1P as α decreases
- 3P and 5P more competitive w.r.t. Q as α decreases, better for low α
- Strong penalties: 3P and 5P better than Q, all the more if α low
- Summing up:
  - for very good dual estimate and good Δ, 1P wins
  - Q much more robust w.r.t. dual estimate, very good for proper t
  - 3P and 5P even better for proper Δ (same t), even more robust
- Morale: more parameters help

MDVS: Random Initial Dual Points

δ^1—δ^2			0.0—0.5			0.0—1.0			0.5—1.0		
t	alg	Δ	md1	md2	md3	md1	md2	md3	md1	md2	md3
	CG		151	549	259	151	549	259	151	549	259
10^7	1P	1000	632	700	700	611	700	700	529	600	514
	Q		115	367	207	114	376	198	114	361	200
10^5	1P	500	414	527	543	520	421	393	550	627	345
	3P	500	126	447	222	131	434	226	126	437	221
	5P	500	127	437	218	125	434	225	122	440	230
	Q		101	240	144	101	246	145	104	259	143
10^3	1P	100	212	255	176	331	274	285	293	337	261
	3P	1000	96	232	123	94	216	121	94	221	121
		500	83	189	103	85	182	107	80	185	107
		100	85	203	130	85	201	123	85	206	132
	5P	1000	84	181	107	87	182	109	84	184	110
		500	74	156	99	74	155	97	74	156	95
		100	82	193	130	86	199	137	84	203	139
	Q		54	118	71	87	157	111	111	141	113

MDVS: Random Initial Dual Points (2)

δ^1—δ^2			0.0—0.5			0.0—1.0			0.5—1.0		
t	alg	Δ	md1	md2	md3	md1	md2	md3	md1	md2	md3
	CG		151	549	259	151	549	259	151	549	259
10^2	1P	10	300	464	468	312	471	503	300	517	544
	3P	100	63	129	92	71	137	94	65	147	93
		10	80	211	142	82	207	138	82	221	146
	5P	100	58	122	97	72	123	94	60	136	97
		10	79	203	140	82	218	149	86	215	155
	Q		184	190	193	369	386	447	352	396	468
10	1P	1	320	651	509	363	683	487	294	700	508
	3P	10	96	226	198	118	213	174	97	213	192
		1	91	221	160	89	224	161	88	237	167
	5P	10	122	364	214	117	256	190	117	254	202
		1	92	235	156	84	225	163	92	248	161
	Q		456	673	618	531	700	675	491	700	688

Comments (3)

- 1P strongly dependent on good initial estimate (much worse here)
- Q much better than 1P everywhere except for very strong penalties
- 3P and especially 5P better than Q with proper choice of Δ
- 3P and 5P even more insensitive to dual estimate than Q

LH-MDVS (Random Initial Dual Points)

δ^1—δ^2			0.0—0.5				0.0—1.0				0.5—1.0			
t	alg	Δ	lh1	lh2	lh3	slv	lh1	lh2	lh3	slv	lh1	lh2	lh3	slv
	CG		629	1866	3588	6	629	1866	3588	6	629	1866	3588	6
10^7	1P	1000	1500	1500	1500	0	1500	1500	1500	0	1500	1500	1500	0
	Q		448	1283	1500	9	446	1247	1500	8	458	1249	1500	9
10^5	1P	500	1208	1500	1500	1	1500	1500	1500	0	1224	1500	1500	1
	3P	500	494	1265	1500	8	494	1283	1500	8	510	1306	1500	8
	5P	500	483	1231	1484	9	483	1251	1486	9	489	1227	1498	9
	Q		331	880	1314	12	343	882	1316	12	330	878	1331	12
10^3	1P	100	624	1500	1500	4	476	1374	1500	7	452	1382	1500	8
	3P	1000	370	1443	1500	6	384	1394	1500	7	382	1442	1500	6
		500	298	1161	1500	9	314	1186	1487	9	315	1183	1500	8
		100	245	651	1155	13	249	696	1189	13	258	688	1209	13
	5P	1000	298	1203	1484	9	293	1172	1450	9	314	1166	1473	10
		500	240	882	1377	11	246	900	1362	11	253	885	1347	11
		100	233	528	867	14	244	559	948	14	239	566	931	14
	Q		136	283	396	14	155	323	457	14	152	317	460	14

LH-MDVS (2)

δ^1—δ^2			0.0—0.5				0.0—1.0				0.5—1.0			
t	alg	Δ	lh1	lh2	lh3	slv	lh1	lh2	lh3	slv	lh1	lh2	lh3	slv
	CG		629	1866	3588	6	629	1866	3588	6	629	1866	3588	6
10^2	1P	10	505	1284	1500	8	611	1484	1500	5	546	1435	1500	5
	3P	100	191	578	1085	13	199	755	915	10	200	626	1131	13
		10	216	418	573	14	222	469	717	14	226	466	713	14
	5P	100	164	436	793	14	170	519	822	14	175	481	797	14
		10	217	396	518	14	220	447	685	14	225	444	641	14
	Q		282	293	676	14	617	864	1249	10	608	777	1362	12
10	1P	1	969	1500	1500	4	1133	1500	1500	2	1106	1500	1500	3
	3P	10	205	308	448	14	248	571	610	14	232	575	540	14
		1	224	434	526	14	247	529	757	14	297	501	702	14
	5P	10	234	303	614	14	270	636	651	14	250	618	491	14
		1	249	442	532	14	257	608	880	14	263	485	682	14
	Q		838	1486	1500	5	1233	1500	1500	2	1163	1500	1500	3

Comments (4)

- 1P never solves all, only occasionally better than pure CG
- 3P better than CG, quite good most of the time
- 5P remarkably better than 3P (much more so than in MDVS)
- Q even better than 5P (except for very large penalty)
- More difficult problem $\Rightarrow$ more pieces?

UBS

t	alg	Δ	u5s0	u5s1	u7s0	u7s1	u10s0	u10s1	u12s0	u12s1	u15s0	u15s1	u20s0	u20s1
	CG		106	132	158	169	321	300	371	506	858	785	1004	989
10^7	1P	1000	1500	285	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500
	Q		98	97	143	152	331	304	361	471	681	694	871	1105
10^5	1P	500	1373	209	1500	1020	1500	1500	1500	1500	1500	664	231	729
	3P	500	99	125	169	181	458	321	394	590	944	1012	1251	1375
	5P	500	111	113	164	193	404	362	441	602	920	816	1230	1490
	Q		81	92	111	110	188	166	206	220	339	271	298	357
10^3	1P	100	336	167	350	328	675	417	953	315	690	286	260	391
	3P	1000	75	85	107	98	169	149	181	188	277	243	233	335
		500	72	75	90	88	141	133	155	167	243	191	181	230
		100	77	80	101	102	171	156	178	203	342	270	279	399
	5P	1000	71	74	97	87	154	131	150	159	248	191	194	239
		500	62	68	82	78	122	107	130	129	199	139	158	186
		100	77	76	99	99	166	155	175	196	317	304	297	412
	Q		107	55	108	80	171	171	119	93	195	163	106	108

UBS (2)

t	alg	Δ	u5s0	u5s1	u7s0	u7s1	u10s0	u10s1	u12s0	u12s1	u15s0	u15s1	u20s0	u20s1
	CG		106	132	158	169	321	300	371	506	858	785	1004	989
10^2	1P	10	259	458	349	461	645	534	698	725	811	787	763	950
	3P	100	55	60	77	69	107	95	114	116	199	124	150	245
		10	106	125	99	131	167	154	176	274	309	336	391	448
	5P	100	52	59	70	83	109	89	104	209	187	118	159	227
		10	107	80	102	140	171	154	178	268	388	380	372	499
	Q		364	261	384	327	427	451	452	337	529	388	296	404
10	1P	1	361	391	559	462	729	748	879	1098	1397	1472	1500	1500
	3P	10	134	140	183	172	198	297	213	320	278	270	362	370
		1	111	119	128	143	182	171	210	254	407	372	368	516
	5P	10	147	136	199	186	250	328	265	305	301	370	393	389
		1	104	113	103	131	192	189	194	291	371	361	355	501
	Q		506	486	634	877	1352	1273	1485	1125	1500	1276	995	1276

Comments (5)

- Again, 1P not much (if any) better than pure CG
- 3P and 5P significantly better than 1P
- 5P most often better than 3P
  (more than in MDVS, less than in LH-MDVS)
- Q often the best, except for large penalty
  (more than in MDVS, less than in LH-MDVS)

Comments (5)

- Again, 1P not much (if any) better than pure CG
- 3P and 5P significantly better than 1P
- 5P most often better than 3P
  (more than in MDVS, less than in LH-MDVS)
- Q often the best, except for large penalty
  (more than in MDVS, less than in LH-MDVS)
- Trend confirmed: the more difficult problem, the more pieces needed

Comments (5)

- Again, 1P not much (if any) better than pure CG
- 3P and 5P significantly better than 1P
- 5P most often better than 3P
  (more than in MDVS, less than in LH-MDVS)
- Q often the best, except for large penalty
  (more than in MDVS, less than in LH-MDVS)
- Trend confirmed: the more difficult problem, the more pieces needed
- Reminder: very "rigid" 3P and 5P, not exploiting all their parameters
- Reminder: static parameters, less important for Q (infinitely-sloped)

Conclusions

- Generic framework for column generation stabilization
- Various piecewise-linear and quadratic stabilizing terms

Conclusions

- Generic framework for column generation stabilization
- Various piecewise-linear and quadratic stabilizing terms
- Pure Proximal versus (rough) Bundle variants

Conclusions

- Generic framework for column generation stabilization
- Various piecewise-linear and quadratic stabilizing terms
- Pure Proximal versus (rough) Bundle variants
- Tests on large-scale real-world problems
- Lessons:
  - stabilization adds complexity, but not unmanageable
  - performances significantly improve
  - having upper bound to follow is important
  - best shape/stiffness of stabilization depends on several parameters
  - general guidelines seem to exist ⇒ applicable in practice