Acceleration and Stabilization Techniques for Column Generation

Zhouchun Huang
Qipeng Phil Zheng

Department of Industrial Engineering & Management Systems
University of Central Florida

Sep 26, 2014
Outline

1. Introduction to Column Generation
2. Motivation
3. Acceleration and Stabilization Techniques
   - Boxstep method
   - Polyhedral penalty
   - du Merle stabilization
   - Bundle method
   - Interior point stabilization
4. Conclusions
Outline

1 Introduction to Column Generation

2 Motivation

3 Acceleration and Stabilization Techniques
   - Boxstep method
   - Polyhedral penalty
   - du Merle stabilization
   - Bundle method
   - Interior point stabilization

4 Conclusions
Introduction to Column Generation

Consider:

\[ \min \ cx \]
\[ \text{s.t. } \ Ax \geq b \]
\[ \quad x \in X \]

Dantzig-Wolfe’s Decomposition:

\[ \min \ \sum_{i=1}^{Q} (cx^i) \lambda_i \]
\[ \text{s.t. } \sum_{i=1}^{Q} (Ax^i) \lambda_i \geq b \]
\[ \quad \sum_{i=1}^{Q} \lambda_i = 1 \]

\( x^i, i = 1, 2, \ldots, Q \) are extreme points of \( X \), which is nonempty and bounded.
Introduction to Column Generation

With only a subset of columns (EP) included, the D-W reformulation is restricted to \textbf{RMP}:

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

The column that is excluded but will enter the basis of master problem is determined by \textbf{SP} or \textit{oracle}:

\[
\begin{align*}
\min & \quad (c - u^k A)x - v^k \\
\text{s.t.} & \quad x \in X
\end{align*}
\]
Column Generation Algorithm

Initialize: $x^0$, UB = $+\infty$, LB = $-\infty$

$k \leftarrow 0$

§1. $(\lambda^k; u^k, v^k) \leftarrow \text{optimizer}(\text{RMP})$

§2. $\hat{x} \leftarrow \text{oracle}(u^k, v^k)$

§3. IF $z(\text{SP}) \geq 0$

\quad $\lambda^* \leftarrow \lambda^k$

ELSE

\quad $x^{k+1} \leftarrow \hat{x}; k \leftarrow k + 1$

go to §1
Successful Applications

- Cutting stock problems
- Air crew scheduling
- Aircraft fleeting and routing
- Crew rostering
- Vehicle routing
- Global shipping
- Multi-item lot-sizing
- Optical telecommunications network design
- Cancer radiation treatment using IMRT
Outline

1. Introduction to Column Generation

2. Motivation

3. Acceleration and Stabilization Techniques
   - Boxstep method
   - Polyhedral penalty
   - du Merle stabilization
   - Bundle method
   - Interior point stabilization

4. Conclusions
Motivation

‡ Why the CG is ”efficient”
‡. D-W decomposition can reformulate a large-scale linear program into a master problem and a set of subproblems to make the problem solvable.
‡. A compact formulation of a MIP may have a weak LP relaxation, while a D-W reformulation with a huge number of variables could give much better LB.
‡. Most variables will be non-basic and only a subset of variables need to be considered.

‡ Why the CG is ”not efficient”
Multiple dual solutions are associated to each primal solution due to high degeneracy. Dual instability in the behavior of dual variables are frequent when problems get larger. As a result, ”oscillations” are observed and very slow convergence is often obtained while implementing the CG algorithm.
CG seen in the Dual Space

**Note:** The crucial part of CG is choosing dual solutions that drives the algorithm toward convergence.

**Column Generation**

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

\[
\begin{align*}
\text{[SP]} \quad \min & \quad (c - u^k A)x - v^k \\
\text{s.t.} & \quad x \in X
\end{align*}
\]

UB = \(z(\text{RMP})\)
LB = \(z(\text{RMP}) + z(\text{SP})\)

**Lagrangian Method**

\[
\begin{align*}
\text{l(u)} = ub + \min & \quad (c - uA)x \\
\text{s.t.} & \quad x \in X
\end{align*}
\]

\[
\begin{align*}
\text{[LD]} \quad \max & \quad ub + v \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \quad i = 1, 2, \ldots, Q
\end{align*}
\]

\[
\begin{align*}
\text{[RLD]} \quad \max & \quad ub + v \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \quad i = 1, 2, \ldots, k
\end{align*}
\]

UB = \(z(\text{RLD}) = l^k (u^k)\)
LB = \(l(u^k)\)
CG seen in the Dual Space

\[ l^k(u^k) \quad l^{k+1}(u^{k+1}) \]

\[ u^k \quad u^* \quad u^{k+1} \]
Outline

1. Introduction to Column Generation
2. Motivation
3. Acceleration and Stabilization Techniques
   - Boxstep method
   - Polyhedral penalty
   - du Merle stabilization
   - Bundle method
   - Interior point stabilization
4. Conclusions
Acceleration and Stabilization Techniques

Boxstep

Polyhedral penalty

du Merle Stabilization

Bundle

\[ u_c \delta^- + \delta^+ \]

\[ u_c \]

\[ u_c \]

\[ u_c \]

\[ u_c \]

\[ u_c \]

\[ u_c \]

\[ u_c \]

\[ u_c \]

\[ u_c \]
Boxstep method

RECALL:

\[
[RLD] \quad \max \quad ub + v \\
\text{s.t.} \quad v \leq (c - uA)x^i, \quad i = 1, 2, ..., k
\]

The Boxstep method modeled a local problem which is to maximize the Lagrangian function over a box of \([\delta_-, \delta_+]\), so that the distance traveled by the dual variables in the dual space is limited to the box.

\[
\max \quad ub + v \\
\text{s.t.} \quad v \leq (c - uA)x^i, \quad i = 1, 2, ..., k \\
\delta_- \leq u \leq \delta_+
\]

\[
\min \sum_{i=1}^{k} (cx^i)\lambda_i - \delta_- y_- + \delta_+ y_+ \\
\text{s.t.} \quad \sum_{i=1}^{k} (Ax^i)\lambda_i - y_- + y_+ \geq b \\
\sum_{i=1}^{k} \lambda_i = 1
\]

Boxstep method

**Trust Regions / Box Step**

<table>
<thead>
<tr>
<th>$\beta$ (box size)</th>
<th>No. of boxes required</th>
<th>$\bar{N}(\beta)$</th>
<th>$T$ (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>34</td>
<td>12.7</td>
<td>172</td>
</tr>
<tr>
<td>0.5</td>
<td>28</td>
<td>14.2</td>
<td>113</td>
</tr>
<tr>
<td>1.0</td>
<td>13</td>
<td>17.1</td>
<td>104</td>
</tr>
<tr>
<td>2.0</td>
<td>9</td>
<td>17.7</td>
<td>88</td>
</tr>
<tr>
<td>3.0</td>
<td>6</td>
<td>25.0</td>
<td>90</td>
</tr>
<tr>
<td>4.0</td>
<td>4</td>
<td>26.8</td>
<td>76</td>
</tr>
<tr>
<td>5.0</td>
<td>5</td>
<td>33.4</td>
<td>134</td>
</tr>
<tr>
<td>6.0</td>
<td>4</td>
<td>34.3</td>
<td>115</td>
</tr>
<tr>
<td>7.0</td>
<td>3</td>
<td>38.0</td>
<td>119</td>
</tr>
<tr>
<td>20.0</td>
<td>2</td>
<td>37.5</td>
<td>203</td>
</tr>
<tr>
<td>25.0</td>
<td>2</td>
<td>74.0</td>
<td>243</td>
</tr>
<tr>
<td>30.0</td>
<td>1</td>
<td>74.0</td>
<td>128</td>
</tr>
<tr>
<td>1000.0</td>
<td>1</td>
<td>97.0</td>
<td>217</td>
</tr>
</tbody>
</table>

- small box size: too many steps toward the best dual
- large box size: weak stabilization, i.e., hard to get a better dual solution (with better LB)

† Desrosiers J., Ben amore H., Soumis F., Villeneuve D.: IMA workshop: Computational Methods for Large Scale Integer Programs, University of Minnesota, Minneapolis, 14-19 (2002).

Polyhedral penalty

Polyhedral penalty is applied to linearly penalize the distance between the dual variables and a stability center.

**Dual**

\[
\begin{align*}
\text{max} & \quad ub + v - \varepsilon |u - u_c| \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \quad i = 1, 2, \ldots, k \\
\text{max} & \quad ub + v - \varepsilon_- w_- - \varepsilon_+ w_+ \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \quad i = 1, 2, \ldots, k \\
& \quad u + w_- \geq u_c \\
& \quad u - w_+ \leq u_c \\
& \quad u, w_-, w_+ \geq 0
\end{align*}
\]

**Primal**

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{k} (cx^i)\lambda_i - u_c y_- + u_c y_+ \\
\text{s.t.} & \quad \sum_{i=1}^{k} (Ax^i)\lambda_i - y_- + y_+ \geq b \\
& \quad y_- \leq \varepsilon_- \\
& \quad y_+ \leq \varepsilon_+ \\
& \quad \sum_{i=1}^{k} \lambda_i = 1 \\
& \quad \lambda, y_-, y_+ \geq 0
\end{align*}
\]


**du Merle stabilization**

du Merle stabilization is a combination of the **Boxstep method** and **polyhedral penalty**.

**Dual**

\[
\begin{align*}
\text{max} & \quad ub + v - \varepsilon_- w_- - \varepsilon_+ w_+ \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \\
& \quad i = 1, 2, \ldots, k \\
& \quad (u + w_- \geq u_c) \\
& \quad u + w_- \geq \delta_- \\
& \quad (u - w_+ \leq u_c) \\
& \quad u - w_+ \leq \delta_+ \\
& \quad u, w_-, w_+ \geq 0
\end{align*}
\]

**Primal**

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{k} (cx^i) \lambda_i - \delta_- y_- + \delta_+ y_+ \\
\text{s.t.} & \quad \sum_{i=1}^{k} (Ax^i) \lambda_i - y_- + y_+ \geq b \\
& \quad y_- \leq \varepsilon_- \\
& \quad y_+ \leq \varepsilon_+ \\
& \quad \sum_{i=1}^{k} \lambda_i = 1 \\
& \quad \lambda, y_-, y_+ \geq 0
\end{align*}
\]

du Merle stabilization algorithm

Initialize: $x^0, \delta^0, \varepsilon^0$

$k \leftarrow 0$

§1. $(\lambda^k, y^k; u^k, v^k) \leftarrow \text{optimizer(RMP)}$

§2. $\hat{x} \leftarrow \text{oracle}(u^k, v^k)$

§3. \textbf{IF} $z(\text{SP}) \geq 0$ and $y_- = y_+ = 0$

\hspace{1cm} $\lambda^* \leftarrow \lambda^k$

\textbf{ELSE}

\hspace{1cm} $x^{k+1} \leftarrow \hat{x}$

\hspace{1cm} $\delta^{k+1} \leftarrow \text{update} - \delta(k)$

\hspace{1cm} $\varepsilon^{k+1} \leftarrow \text{update} - \varepsilon(k)$

\hspace{1cm} $k \leftarrow k + 1$

\hspace{1cm} go to §1

\begin{align*}
\min_{i=1}^{k} (c^i x^i) \lambda_i - \delta_- y_- + \delta_+ y_+ \\
\text{s.t.} \sum_{i=1}^{k} (A^i x^i) \lambda_i - y_- + y_+ \geq b \\
y_- \leq \varepsilon_- \\
y_+ \leq \varepsilon_+ \\
\sum_{i=1}^{k} \lambda_i = 1 \\
\lambda, y_-, y_+ \geq 0
\end{align*}
Parameters in du Merle stabilization

<table>
<thead>
<tr>
<th>Parameters $(\delta_0,\delta_0)$</th>
<th>Main iterations</th>
<th>CPU time ratio optimizer/oracle</th>
<th>CPU time (s)</th>
<th>Speedup factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(-\infty, \infty)$</td>
<td>433</td>
<td>1.037</td>
<td>1491.0</td>
<td>1.00</td>
</tr>
<tr>
<td>$[-50, \infty)$</td>
<td>440</td>
<td>1.444</td>
<td>1985.3</td>
<td>0.75</td>
</tr>
<tr>
<td>$[0, 100]$</td>
<td>157</td>
<td>0.521</td>
<td>267.6</td>
<td>5.57</td>
</tr>
<tr>
<td>$[50, 100]$</td>
<td>130</td>
<td>0.301</td>
<td>201.2</td>
<td>7.41</td>
</tr>
</tbody>
</table>

$z(\text{SP}) \geq 0$ and $y = 0$

OPTIMAL

$z(\text{SP}) \geq 0$ and $y > 0$

DECREASE $\varepsilon$

UPDATE $\delta^{k+1} \leftarrow \delta^k$ if $l(u^k, v^k) > l(u^{k-1}, v^{k-1})$

$z(\text{SP}) < 0$ and $y = 0$ (*rare to happen*)

ADD columns

$z(\text{SP}) < 0$ and $y > 0$

ADD columns

UPDATE $\delta^{k+1} \leftarrow \delta^k$ if $l(u^k, v^k) > l(u^{k-1}, v^{k-1})$
The quality of dual is estimated via the computation of a lower bound:

\[ l(u^k, v^k) = u^k b + (c - u^k A)x^k \]

since \((u^k, (c - u^k A)x^k)\) is a feasible solution of the LD:

\[
\begin{align*}
\max \quad & ub + v \\
\text{s.t.} \quad & v \leq (c - uA)x^i, \ i = 1, 2, \ldots, Q
\end{align*}
\]
Bundle method differs from polyhedral penalty methods in that it takes infinitely small penalty ranges and infinitely many break points.†

\[
\begin{align*}
\max & \quad ub + v - \frac{1}{2t} ||u - u_c||^2 \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \ i = 1, 2, \ldots, k \\
& \quad u \geq 0
\end{align*}
\]

Its Fenchel bidual is formulated as,

\[
\begin{align*}
\min & \quad \sum_{i=1}^{k} (cx^i)\lambda_i + u_cg + \frac{t}{2}||g||^2 \\
\text{s.t.} & \quad \sum_{i=1}^{k} (Ax^i)\lambda_i + g \geq b \\
& \quad \sum_{i=1}^{k} \lambda_i = 1 \\
& \quad \lambda, g \geq 0
\end{align*}
\]

Interior point stabilization

The idea is to generate a dual solution that is in the **interior** of the convex hull of optimal dual solutions to the master instead of using one of its extreme points.†

\[
\begin{align*}
\text{max} & \quad u(\mu b) + v \\
\text{s.t.} & \quad v \leq (c - uA)x^i, \; i = 1, 2, \ldots, k \\
& \quad u \geq 0
\end{align*}
\]

\[\mu \sim U(0, 1), \text{ i.e. each element of } \mu \text{ is uniformly distributed}\]
\[\text{a set of problems with different } \mu \text{ are solved}\]
\[\text{an interior point is obtained by taking convex combination of optimal solutions}\]
\[\text{the dual of the problem above is,}\]

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

Outline

1. Introduction to Column Generation
2. Motivation
3. Acceleration and Stabilization Techniques
   - Boxstep method
   - Polyhedral penalty
   - du Merle stabilization
   - Bundle method
   - Interior point stabilization
4. Conclusions
Conclusions

† introduction to column generation
† slow convergence of column generation
† stabilization techniques
The End