Column Generation with the Primal-Dual Interior Point Method

Jacek Gondzio

joint work with

Pablo González-Brevis and Pedro Munari

Buzios, 23 May 2016
Outline

• IPMs for Optimization
  → central path, perturbed complementarity
• Column Generation/Cutting Planes with IPM
• Warmstarting IPMs
  → theory and practice
• Applications (numerous examples)
  → summary results
• Conclusions
What is the best method to solve LPs/QPs:

- the Simplex Method (active-set method) or
- the Interior Point Method?

Maths question: How to cross a polytope?

... let us ask the expert
Brazil view of LP/QP:

How to solve LP/QP problems?

If we asked Neymar Jr, the likely answer would be: “go through the interior of the polytope”.
Interior Point Methods
Primal-Dual Pair of Quadratic Programs

Primal

\[
\begin{align*}
\text{min} & \quad c^T x + \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \geq 0;
\end{align*}
\]

Dual

\[
\begin{align*}
\text{max} & \quad b^T y - \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad A^T y + s = c, \\
& \quad s \geq 0.
\end{align*}
\]

Lagrangian

\[L(x, y) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - s^T x.\]

Optimality Conditions

\[
\begin{align*}
Ax &= b, \\
A^T y + s - Q x &= c, \\
X S e &= 0, \quad \text{(i.e., } x_j \cdot s_j = 0 \quad \forall j), \\
(x, s) &\geq 0,
\end{align*}
\]

\[X = \text{diag}\{x_1, \cdots, x_n\}, \quad S = \text{diag}\{s_1, \cdots, s_n\}, \quad e = (1, \cdots, 1) \in \mathbb{R}^n.\]
First Order Opt Conditions for QP

\[ Ax = b, \]
\[ A^Ty + s - Qx = c, \]
\[ XSe = 0, \]
\[ (x, s) \geq 0, \]

First Order Opt Conditions for Barrier QP

\[ Ax = b, \]
\[ A^Ty + s - Qx = c, \]
\[ XSe = \mu e, \]
\[ (x, s) > 0, \]
Complementarity in the Interior Point Method

The first order optimality conditions (FOC)

\[ \begin{align*}
Ax & = b, \\
A^T y + s - Qx & = c, \\
XSe & = \mu e, \\
x, s & \geq 0,
\end{align*} \]

where \( X = \text{diag}\{x_j\} \), \( S = \text{diag}\{s_j\} \) and \( e = (1, \ldots, 1) \in \mathbb{R}^n \).

**Analytic centre (\( \mu \)-centre):** a (unique) point \((x(\mu), y(\mu), s(\mu))\), \(x(\mu) > 0 \), \(s(\mu) > 0\) that satisfies FOC.

The interior point method gradually reduces the complementarity products

\[ x_j \cdot s_j \approx \mu \rightarrow 0 \quad \forall j = 1, 2, \ldots, n. \]
Attractive features

IPMs:

• excell on large-scale problems
• can easily control the distance to optimality
• are indifferent to degeneracy
• are able to warm start

Decomposition Approaches

What IPMs can offer?

- Use IPM to solve the Master
- Use IPM to solve the Subproblem early termination with $\varepsilon$-optimality
  $\rightarrow \varepsilon$-subgradients (“on-demand accuracy”)


The First Order Optimality Conditions

\[ Ax = b, \]
\[ -Qx + A^T y + s = c, \]
\[ XSe = \mu e, \]
\[ (x, s) > 0. \]

Assume primal-dual feasibility:

\[ Ax = b \quad \text{and} \quad -Qx + A^T y + s = c \]

Parameter \( \mu \) controls the distance to optimality.

\[
(c^T x + \frac{1}{2} x^T Q x) - (b^T y - \frac{1}{2} x^T Q x) = x^T s = n \mu.
\]

On-demand accuracy is readily available.

Buzios, 23 May 2016
Primal-Dual Column Generation Method:

Heading-in problems:
terminate RMP solution early:
→ get stable dual solution $\bar{u}$

Tailing-off problems:
use IPM to solve the RMP:
→ no degeneracy issues
Primal-Dual
Column Generation Method
Column Generation (CG)

Consider an LP, called the master problem (MP):

\[ z^* := \min \sum_{j \in N} c_j \lambda_j, \]

\[ \text{s.t. } \sum_{j \in N} a_j \lambda_j = b, \]

\[ \lambda_j \geq 0, \quad \forall j \in N. \]

- \( N \) is too big;
- The columns \( a_j \) are implicit elements of \( \mathcal{A} \);
- We know how to generate them!
CG: Restricted master problem (RMP): $\overline{N} \subset N$

$$z_{RMP} := \min \sum_{j \in \overline{N}} c_j \lambda_j,$$

s.t. $\sum_{j \in \overline{N}} a_j \lambda_j = b,$

$$\lambda_j \geq 0, \quad \forall j \in \overline{N}.$$

- Optimal $\bar{\lambda}$ for the RMP $\Rightarrow$ feasible $\hat{\lambda}$ for the MP;
- $\hat{\lambda}_j = \bar{\lambda}_j, \forall j \in \overline{N}$, and $\hat{\lambda}_j = 0$ otherwise;
- Hence, $z^* \leq z_{RMP} = UB$ (Upper Bound).
- How to know it is optimal?
  - Call the oracle!
CG:

- Oracle: check the feasibility of the dual \( \overline{u} \);
- Reduced costs: \( s_j = c_j - \overline{u}^T a_j, \forall j \in N \);
- But the columns are not explicit and, hence,
  \[ z_{SP} := \min \{ c_j - \overline{u}^T a_j | a_j \in A \} \]
- (we reset \( z_{SP} := 0 \), if \( z_{SP} > 0 \));
- Lower Bound: \( LB = z_{RMP} + \kappa z_{SP} \leq z^* \), where
  \[ \kappa \geq \sum_{i \in N} \lambda_i^* \]
- If \( z_{SP} < 0 \), then new columns are generated;
- Otherwise, an optimal solution of the MP was found!
Appealing features of IPMs:

- Use IPM to solve the RMP:
  → no degeneracy issues

- Terminate RMP solution early:
  → use reliable estimate of duality gap
  → get stable dual solution $\bar{u}$
**PDCGM Algorithm**  Parameters: $\varepsilon_{\text{max}}, D > 1, \delta, \kappa$

1. set $\text{LB} = -\infty$, $\text{UB} = \infty$, gap = $\infty$, $\varepsilon = 0.5$;
2. while (gap $> \delta$) do
3. find a well-centred $\varepsilon$-opt $(\tilde{\lambda}, \tilde{u})$ of the RMP;
4. $\text{UB} = \min\{\text{UB}, \tilde{z}_{RMP}\}$;
5. call the oracle with the query point $\tilde{u}$;
6. $\text{LB} = \max\{\text{LB}, \kappa\tilde{z}_{SP} + b^T\tilde{u}\}$;
7. gap = $(\text{UB} - \text{LB})/(1 + |\text{UB}|)$;
8. $\varepsilon = \min\{\varepsilon_{\text{max}}, \text{gap}/D\}$;
9. if $(\tilde{z}_{SP} < 0)$ then add new columns to the RMP;
10. end (while)
Theorem

Let $z^*$ be the optimal solution of the (MP). Given $\delta > 0$, the primal-dual column generation method converges in a finite number of steps to a feasible solution $\hat{\lambda}$ of the MP with objective value $\hat{z}$ that satisfies

$$\hat{z} - z^* \leq \delta(1 + |z^*|).$$

Oscillation in a real instance

Changes of dual variables between iterations $j$ and $j+1$, $||u^j - u^{j+1}||_2$ on VRPTW instance (Solomon C207):
Warmstarting IPMs
A need to solve a sequence of similar problems

- column generation
- cutting plane methods
- subproblems in the block-angular LPs (Dantzig-Wolfe decomp., Benders decomp.)
- B&B, (and B&Cut, B&Cut&Price, etc)
- SQP
- any sequence of similar problems
  example: computing efficient frontier in Markowitz portfolio optimization
Warm Starts  Which method should be used?

- Simplex Method, or
- Interior Point Method.

\[ x_j s_j = 0, \forall j \]

When is the Simplex Method better?
→ few indices change optimal partition
B & B, adding one cut in CPM, etc.

When is the Interior Point Method better?
→ many indices change optimal partition
adding many cuts in CPM,
dealing with a general change of problem data, etc.

Conjecture:
The more changes in the (large) problem
the more attractive IPM-based warm starts are.
Difficulty of IPM Warm Starts

Original Problem

Modified Problem
Warmstarting Heuristic

Idea: Start close to the (new) central path, not close to the (old) solution

\textbf{JG}, Warm start of the primal-dual method applied in CPM, \textit{Mathematical Programming} 83 (1998) 125–143
Warm Start in PDCGM context: new results

Theorem

Let $n$ and $n+k$ be the dims of RMP and newRMP, resp. Let a $\mu^0$-centre $(x^0, y^0, z^0) \in N_S(\gamma)$ of RMP be given. The warm start solution $(x^w, \bar{x}^w, y^w, z^w, \bar{z}^w) \in N_S(\beta\gamma)$ produced by the new warmstarting algorithm satisfies

$$(x^w)^T s^w + (\bar{x}^w)^T \bar{s}^w \leq \frac{6}{\gamma} \frac{n+k}{n} (x^0)^T s^0.$$

JG and González-Brevis,
Applications of PDCGM

- UFLP: Uncapacitated Facility Location Problem
- MKL: Multiple Kernel Learning problem
- TSSP: Two-Stage Stochastic Problem
- MCNF: MultiCommodity (MinCost) Network Flow problem
- CSP: Cutting Stock Problem
- CLSPST: Capacitated Lot-Sizing Problem with Setup Times
- VRPTW: Vehicle Routing Pb with Time Windows

PDCGM Software available at:
http://www.maths.ed.ac.uk/~gondzio/software/pdcgm.html
**Number of iterations** (relative to PDCGM)

<table>
<thead>
<tr>
<th></th>
<th>CSP</th>
<th>VRPTW</th>
<th>CLSPST</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCGM</td>
<td>1.52</td>
<td>1.33</td>
<td>1.60</td>
</tr>
<tr>
<td>ACCPM</td>
<td>2.41</td>
<td>4.86</td>
<td>1.26</td>
</tr>
</tbody>
</table>
CPU time [s] (relative to PDCGM)

<table>
<thead>
<tr>
<th></th>
<th>CSP</th>
<th>VRPTW</th>
<th>CLSPST</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCGM</td>
<td>3.50</td>
<td>1.95</td>
<td>1.26</td>
</tr>
<tr>
<td>ACCPM</td>
<td>8.97</td>
<td>4.01</td>
<td>1.27</td>
</tr>
</tbody>
</table>
Conclusions

A completely new perspective is needed to exploit the insight offered by IPMs in a number of **combinatorial optimization** applications:

- column generation
- cutting plane methods
- B & B, (and B & Cut, B & Cut & Price, etc)

**Warmstarting works well** in the CG/CPM context: problems are re-optimized in **3-5** IPM iterations
References


http://www.maths.ed.ac.uk/~gondzio/software/pdcm.html

Buzios, 23 May 2016
Vehicle Routing Problem with Time Windows

A company delivers goods to customers $i \in C$. The company has vehicles $k \in V$ and each of them starts at a depot, travels to several customers and returns to the depot. The visit of vehicle $k$ to customer $i$ needs to take place in a specific time window: $a_i \leq s_{ik} \leq b_i$, where $s_{ik}$ is the time when vehicle $k$ reaches customer $i$.

Objective: Minimize the total cost of delivery.

Define binary variable $x_{ijk}$ which takes value 1 if vehicle $k$ travels from customer $i$ to customer $j$ ($k \in V, i, j \in C$) and takes value zero otherwise.
Vehicle Routing Problem with Time Windows

Constraints:

Exactly one vehicle leaves customer $i$:

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C$$

Vehicle capacity constraint:

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq q, \quad \forall k \in V$$

Each vehicle leaves the depot and returns to it:

$$\sum_{j \in N} x_{0jk} = 1 \quad \text{and} \quad \sum_{j \in N} x_{i(n+1)k} = 1, \quad \forall k \in V$$
VRPTW: Constraints (continued)

Time-window constraint

\[ s_{ik} + t_{ij} - M(1 - x_{ijk}) \leq s_{jk}, \quad \forall i, j \in N, \forall k \in V. \]

Since \( x_{ijk} \) is binary the above constraint has the following meaning: If \( x_{ijk} = 1 \) (vehicle \( k \) travels from customer \( i \) to customer \( j \)) then

\[ s_{ik} + t_{ij} \leq s_{jk} \]

that is, the arrival time of vehicle \( k \) to customer \( j \) is greater than or equal the sum of time when vehicle \( k \) arrives to customer \( i \) and the time \( t_{ij} \) it takes to travel from \( i \) to \( j \).

Otherwise (if \( x_{ijk} = 0 \)) the presence of “big” \( M \) guarantees that the constraint is always inactive.
\textbf{VRPTW} \quad \text{min}\quad \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}

\text{s.t.}\quad \sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C,

\sum_{i \in C} \sum_{j \in N} d_i x_{ijk} \leq q, \quad \forall k \in V,

\sum_{j \in N} x_{0jk} = 1, \quad \sum_{i \in N} x_{i(n+1)k} = 1, \quad \forall k \in V,

\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{jhk} = 0, \quad \forall h \in C, \forall k \in V,

s_{ik} + t_{ij} - M(1 - x_{ijk}) \leq s_{jk}, \quad \forall i, j \in N, \forall k \in V,

a_i \leq s_{ik} \leq b_i, \quad \forall i \in N, \forall k \in V,

x_{ijk} \in \{0, 1\}, \quad \forall i, j \in N, \forall k \in V.