Graph-Theoretical Approaches in Big Optimization

Nurminski E.A.
nurminskiy.ea@dvfu.ru

Far Eastern Federal University, Vladivostok

Optimization Methods and Applications
Irkutsk, June 30 – July 06, 2014
Big Data vs Big Computing

Figure 1. Next-gen sequence data size compared to SPECint.

Optimization dream: to solve $\infty \times \infty$ problem

Successive approximations:

- Megabyte-optimization: $10^6 - 10^8$ variables/constraints;
- Gigabyte-optimization: $10^9 - 10^{11}$ variables/constraints;
- Terabyte-optimization: $10^{12} - 10^{14}$ variables/constraints;
- etc ...
Simple algorithms 1

Coordinate descent:


- etc ...
Simple algorithms 2

Gradient-type algorithms:

- etc . . .
Projection methods.

- D. Henrion, J. Malick Projection methods in conic optimization Optimization Online.
- And many others ...
Step 4. Projection on the polyhedral cone

The final result of the transformations above was to hide all complexities into

\[ \phi_\gamma(u) = \min_{\bar{x}} \left\{ \frac{1}{2} \| \bar{x} \|^2 + (\gamma \bar{A} - ue^{n+1}) \bar{x} \right\} \]

but from the previous slide

\[ \phi_\gamma(u) = \min_{\bar{x}} \left\{ \frac{1}{2} \| \bar{x} \|^2 \right\} = \min_{\bar{x} \in \gamma \bar{A} - ue^{n+1}} \frac{1}{2} \| \bar{x} \|^2 = \min_{\bar{x} \in \gamma \bar{A}} \frac{1}{2} \| \bar{x} - ue^{n+1} \|^2 = \Pi_{\gamma \bar{A}}(ue^{n+1}) \]

and this is very simple problem for large \( \gamma \).
The value of the function $\phi_{\gamma}(u)$ does not depend on further increase of $\gamma$ for a given $u_\star$. 

Nurminski FEFU
Decomposition

We can use the property of the conical hull

\[ K(\mathcal{A}) = \text{Co}\{\bar{A}_i, i = 1, 2, \ldots, m\} = \text{Co}\{\text{Co}\{\bar{A}_k\}, k = 1, 2, \ldots, K\}, \]

where \( \bar{A}_k = \{\bar{A}_i, i \in I_k\} \) and index sets \( I_k \) cover the whole range of rows 1, 2, \ldots, \( m \).

In turn \( \Pi_{K(\bar{A})}(e^{n+1}) \) can be reduced to separate (and parallel) projections \( \Pi_{K(\bar{A}_k)}(e^{n+1}) \) (with slight modifications).
Sketch of Decomposition-Coordination

General idea: iterate between two steps Coordination (C) and Decomposition (D):

**C:** Get proposals $\bar{z}^k, k = 1, 2, \ldots, K$ from each of sub-problems and solve the coordination problem

$$\|\bar{z} - e^{n+1}\|^2 = \min \|z - e^{n+1}\|^2, z \in \text{Co}\{z^k, k = 1, 2, \ldots, K\}$$

**D:** For each of sub-problems modify the feasible cone $\tilde{K}_k = \text{Co}\{\bar{z}, K(\bar{A}_k)\}$ and solve for $k = 1, 2, \ldots, K$ sub-problems

$$\|\bar{z}^k - e^{n+1}\|^2 = \min \|z - e^{n+1}\|^2, z \in \tilde{K}_k$$

to get new proposals $\bar{z}^k, k = 1, 2, \ldots, K$.

To initialize the process one can use of course an arbitrary $\bar{z}^k \in K(\bar{A}_k)$ or to think of smth not that stupid.
Historical remarks

This idea can be traced back at least as far as Demyanov V.F., Malozemov V.N. *Introduction to Minmax*, M.: Nauka, where it was used in its simplest form.
In its current form in was proposed by Nurminski E. (IzVuz, somewhere in ’90).
Some improvements due to Nurminski E, Dolgy D. in Korean Univ publication, 2012.
Probably there are many similar proposals, pls let me know.
Caveats

- $A_k$ should not be too big or ill-structured to complicate solutions of sub-problems.
- $A_k$ should not be too small to make the coordination problem too big or complicated.

**Definition** Vectors $a$ and $b$ from $E^n$ are called structurally orthogonal (or independent) if $a_i b_i = 0$ for all $i = 1, 2, \ldots, n$.

**Definition** An $m \times n$ matrix $A$ is called structurally orthogonal if its rows $A_i, i = 1, 2, \ldots, m$ are structurally orthogonal.

For such matrices the projection problem has linear complexity as $AA'$ is a diagonal matrix.

---

2Of course it is more restrictive than orthogonality (implies it) and less restrictive than complementarity (no sign constraints). Somewhere in between.
Structurally Orthogonal Decomposition

Select $l_k$ such that the corresponding $\bar{A}_k$ are structurally (s-) orthogonal.

Define graph $G_A = (V_A, E_A)$, where

- $V_A$ — the set of rows of $A$,
- $E_A$ — the set of edges $e = (v_1, v_2) \in V_A \times V_A$ such that $v_1$ is NOT $s$-orthogonal to $v_2$.

The set of mutually $s$-orthogonal rows is a set of independent nodes in graph $G$.

**Problem:** Decompose the graph into minimal number of independent components (coloring).

**Specifics:** Graphs are huge, but sparse. Solutions with relatively small (up to $10^4$) uncolored reminders are acceptable.
Heuristics

Greedy:

- try to build from the current graph an independent set as big as possible;
- delete this set from the graph (with incidents edges of course);
- continue with the rest of the graph.

Allows for many variants.
LP-specifics

Simplest cases:

- Sign constraints: $x \geq 0$ — all constraints are s-orthogonal;
- Two-sided constraints: $l \leq x \leq u$ — 2 s-orth sets, $2^n$ variants, any combination of lower-upper constraints;
- Transportation problem: 2 s-orth sets, supply balances and demand balances or any combination;
- Canonical $m \times n$, $m << n$ LP:

$$\min cx$$

$$Ax = b; x \geq 0$$

1 s-ort set (sign constraints), general equality constraints as ”reminder”.

Nurminski FEFU
SHELL_0 (www.netlib.org) 537 cnst, 1775 vars, 4900 nz (0.5%)
SHELL_1 (www.netlib.org) $|IS| = 278$

Graph stat: 257 nodes, 781 arcs
SHELL_2 (www.netlib.org) $|S| = 159$

Graph stat: 96 nodes, 254 arcs
### Shell_3 (www.netlib.org) \(|IS| = 68\)

Graph stat: 28 nodes, 65 arcs
**SHELL (www.netlib.org) Summary of the selection process**

<table>
<thead>
<tr>
<th>Desc</th>
<th>Nodes</th>
<th>Arcs</th>
<th>Av. degree</th>
<th>Indp. set</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHELL_0</td>
<td>537</td>
<td>2210</td>
<td>4.1155</td>
<td>278</td>
</tr>
<tr>
<td>SHELL_1</td>
<td>257</td>
<td>781</td>
<td>3.0389</td>
<td>159</td>
</tr>
<tr>
<td>SHELL_2</td>
<td>96</td>
<td>254</td>
<td>2.6458</td>
<td>68</td>
</tr>
<tr>
<td>SHELL_3</td>
<td>28</td>
<td>65</td>
<td>2.3214</td>
<td></td>
</tr>
</tbody>
</table>
GREENBEA, www.netlib.org 2374x5323x30230 (0.24%)
GREENBEA/20 – the giant core (1105 constraints)
| GREENBEA_1 1313 cnst, $|S| = 1075 , 18148$ ars |
GREENBEA_2 866 cnst, $|IS| = 447,9938$ arcs
GREENBEA_3 685 cnst, $|IS| = 181,7415$ arcs
GREENBEA_4 549 cnst, $|IS| = 136,5490$ arcs
GREENBEA_5 450 cnst, $|IS| = 99,4310$ arcs
<table>
<thead>
<tr>
<th>Projection methods</th>
<th>Least norm problem</th>
<th>Computational issues</th>
<th>Numerical experience</th>
</tr>
</thead>
</table>

GREENBEA\_6 350 cnst, $|I_S| = 99,315$ arcs
GREENBEA\_7 264 cnst, $|\mathcal{IS}| = 81, 2470$ arcs
GREENBEA_8 191 cnst, $|IS| = 67$, 1960 arcs
GREENBEA_9 149 cnst, $|IS| = 40$, 1601 arcs
### Projection methods

#### Least norm problem

#### Computational issues

#### Numerical experience

<table>
<thead>
<tr>
<th>Desc</th>
<th>Nodes</th>
<th>Arcs</th>
<th>Av. degree</th>
<th>Indp. set</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREENBEA_1</td>
<td>2388</td>
<td>34294</td>
<td>6.9633</td>
<td>1075</td>
</tr>
<tr>
<td>GREENBEA_2</td>
<td>1313</td>
<td>18148</td>
<td>7.2350</td>
<td>447</td>
</tr>
<tr>
<td>GREENBEA_3</td>
<td>866</td>
<td>9938</td>
<td>8.7140</td>
<td>181</td>
</tr>
<tr>
<td>GREENBEA_4</td>
<td>685</td>
<td>7415</td>
<td>9.2380</td>
<td>136</td>
</tr>
<tr>
<td>GREENBEA_5</td>
<td>549</td>
<td>5490</td>
<td>10.0000</td>
<td>99</td>
</tr>
<tr>
<td>GREENBEA_6</td>
<td>450</td>
<td>4310</td>
<td>10.4408</td>
<td>99</td>
</tr>
<tr>
<td>GREENBEA_7</td>
<td>350</td>
<td>3135</td>
<td>11.1643</td>
<td>81</td>
</tr>
<tr>
<td>GREENBEA_7</td>
<td>264</td>
<td>2470</td>
<td>10.6883</td>
<td>67</td>
</tr>
<tr>
<td>GREENBEA_9</td>
<td>191</td>
<td>1960</td>
<td>9.7449</td>
<td>40</td>
</tr>
</tbody>
</table>
Convergence of the projection procedure