Finding Good Starting Points For Solving Nonlinear Constrained Optimization Problems by Parallel Decomposition

Soomin Lee and Benjamin Wah

10/29/2008
Mexican International Conference on AI
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory

UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

Definition: Constrained Optimization Problems (COPs)

\[(I') \quad \min_{x} f(x) \quad \text{subject to } h(x) = 0 \text{ and } g(x) < 0, \]

where \(x = (x, y)\): Variables, \(x \in \mathbb{R}^p, y \in \mathbb{D}^p\)
\(f(x)\): objective function
\(h(x) = (h_1(x), \ldots, h_m(x))^T\): Equality constraints
\(g(x) = (g_1(x), \ldots, g_r(x))^T\): Inequality constraints

- NLP (Nonlinear Programming) and MINLP (Mixed Integer Nonlinear Programming)
- We focus on finding CLM (Constrained Local Minima).
Motivation: Good Starting Points

- Ex: HAGER4 from CUTEr
  - An NLP with 10,000 variables and 5,000 constraints
  - A large problem that cannot be solved by an existing solver MINOS from the default starting point.

\[
\begin{align*}
\min_{x,u} & \quad \sum_{i=1}^{n} (C_1 z_{i-1} x_i^2 + C_2 z_{i-1} x_i (x_{i-1} - x_i) \\
& + C_3 z_{i-1} (x_{i-1} - x_i)^2 + C_4 u_i^2) \\
\text{subject to} & \quad h_i(x, u) = 0 \\
\text{where} & \quad h_i(x, u) = (n - 1)x_i - nx_{i-1} - \exp(t_i) u_i, \text{ for } 1 \leq i \leq 5000
\end{align*}
\]

- Very well structured
- Constraints are all in the same format
  - Good constraint locality wrt variables

50 constraints
250 constraints
500 constraints

Generated starting points (by extrapolations and interpolations)

The final solution generated by other solver

Now, MINOS can solve the problem in 3.13 sec with this starting point.
Constraint Localities in AI Problems

- Rovers-Propositional-P18 from IPC-5.
  - 2,207 variables, 104,941 constraints

Constraint localities allow starting points to be generated

Problem Statement

- Exploit the localities of constraints in constrained optimization problems
- In order to generate much simpler subproblems
- Whose solutions can be generalized to form good starting points to the original problems
- That allow the original problems to be efficiently solved
Approach: Generating Good Starting Points: Overview

1. Analyze the structure of the original problem
2. Generate small problems (relaxed/easier versions) that contain all the features of the original problem
3. Solve those small problems
4. Use the outputs to generalize to the starting point of the original problem

- How do we generate those small problems?
  - constraint relaxation

(A) With a Few Common Variables (1/2)

- Constraint sampling and interpolations
  - All the constraints share a few common variables
  - Example:

\[
(\sum_{i=1}^{5} x[i] z[i])^2 = 0, \quad 1 \leq i \leq 5000
\]
(A) With a Few Common Variables (2/2)

Interpolate variables $x[i]$ and $y[i]$ from the extrapolated variables

With a Few Common Variables (2/2)

5 samples

10 samples

Variable $z_1$, $z_2$ and $z_3$ with various number of sampled constraints

Extrapolate variables $x[i]$ and $y[i]$ from subproblem with sampled constraints

20 samples

100 samples

(B) With Coupled Constraints (1/3)

- Constraint sampling and extrapolation
  - Constraints are related to their neighborhood constraints
  - Example: \[
  (n - 1)x_i - nx_{i-1} - \exp(i)u_i = 0, \text{ for } 1 \leq i \leq 5000
  \]
  \[
  (n - 1)x_1 - nx_0 - \exp(t_1)u_1 = 0
  \]
  \[
  (n - 1)x_2 - nx_1 - \exp(t_2)u_2 = 0
  \]
  \[
  (n - 1)x_3 - nx_2 - \exp(t_3)u_3 = 0
  \]
  \[
  (n - 1)x_4 - nx_3 - \exp(t_4)u_4 = 0
  \]
  \[
  (n - 1)x_{5000} - nx_{4999} - \exp(t_{5000})u_{5000} = 0
  \]
  - Take a subset of contiguous constraints and extrapolate
    \[
    (n - 1)x_i - nx_{i-1} - \exp(i)u_i = 0, \text{ for } 1 \leq i \leq 500
    \]
(B) With Coupled Constraints (2/3)

Example: \((n - 1)x_i - nx_{i-1} - \exp(t_i)u_i = 0, \text{ for } 1 \leq i \leq 500^C\)

\[ \begin{align*}
\text{Simplified Problem #1} \\
1 \leq i \leq 50 \\
\text{Variable Index (i)}
\end{align*} \]

\[ \begin{align*}
\text{Simplified Problem #2} \\
1 \leq i \leq 250 \\
\text{Variable Index (i)}
\end{align*} \]

\[ \begin{align*}
\text{Simplified Problem #3} \\
1 \leq i \leq 500 \\
\text{Variable Index (i)}
\end{align*} \]

\[
\begin{align*}
\text{Example: A set of variables } x[1..N^2], \ N = 5^6
\end{align*}
\]

\[ \begin{align*}
\text{Simplified Problem #1} \\
N = 19
\end{align*} \]

\[ \begin{align*}
\text{Simplified Problem #2} \\
N = 29
\end{align*} \]

\[ \begin{align*}
\text{Simplified Problem #3} \\
N = 39
\end{align*} \]
(C) With Mixed Integer Constraints

- **Constraint relaxation for MINLPs**
  - Solve an MINLP as an NLP without the integrality of integer variables.
  - Apply (A) and (B) if MINLP is still too large to be handled by existing NLP solvers.

### Illustration on Performance of Approach

<table>
<thead>
<tr>
<th>Name</th>
<th>Nc</th>
<th>Nv</th>
<th>Ns</th>
<th>Tech</th>
<th>Solver</th>
<th>Default Time</th>
<th>Default Sol</th>
<th>Proposed Sptime</th>
<th>Proposed Time</th>
<th>Proposed Sol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex4</td>
<td>3717 (3481)</td>
<td>7198</td>
<td>514</td>
<td>B</td>
<td>Conopt</td>
<td>f</td>
<td>f</td>
<td>5.61</td>
<td>27.26</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ipopt</td>
<td>1.36</td>
<td>0.08</td>
<td>1.71</td>
<td>1.77</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Knitro</td>
<td>1.42</td>
<td>0.08</td>
<td>1.64</td>
<td>1.17</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Snopt</td>
<td>130.40</td>
<td>0.08</td>
<td>8.31</td>
<td>8.16</td>
<td>0.08</td>
</tr>
<tr>
<td>cont5_</td>
<td>40200 (200)</td>
<td>40400</td>
<td>131</td>
<td>B</td>
<td>Conopt</td>
<td>152.80</td>
<td>0.07</td>
<td>3.02</td>
<td>55.36</td>
<td>0.07</td>
</tr>
<tr>
<td>2_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ipopt</td>
<td>208.18</td>
<td>0.07</td>
<td>2.90</td>
<td>148.26</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Knitro</td>
<td>483.15</td>
<td>0.07</td>
<td>29.78</td>
<td>257.17</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Snopt</td>
<td>f</td>
<td>f</td>
<td>2.96</td>
<td>2180.97</td>
<td>0.07</td>
</tr>
</tbody>
</table>

For SNOPT, even a good starting point is not good enough.
Handling Large-Scale COPs by Parallel Decomposition [CP2005, Wah/Chen]

Locality Analysis → Optimal # of partitions (granularity) → Solving subproblems → Handling CVs

Complicating variables (CVs)
Subproblem → Overhead (Time)
# of partitions

Complicating parts...

Again, Need Good Starting Points in Parallel Decomposition

- Fast convergence
  - Subproblems are solved with the knowledge of others.

Subproblems
z1=0
z2=0
z3=0
4.9
0
0
5.0
1.5
0
...
5.0
0
0

- Handling complicating variables (CVs)

Need to enforce consistency of complicating variables in all subproblems.

Ex) Make them fixed throughout the solving process
Need good values for z1, z2 and z3
### Illustration on Parallel Decomposition with Good Starting Points [CP2005, Wah/Chen]

<table>
<thead>
<tr>
<th>Name</th>
<th>Nc</th>
<th>Nv</th>
<th>Ns</th>
<th>Tech</th>
</tr>
</thead>
<tbody>
<tr>
<td>orthrgds</td>
<td>5000(5000)</td>
<td>10003</td>
<td>5,10,20</td>
<td>A</td>
</tr>
<tr>
<td>lukvl7</td>
<td>4(4)</td>
<td>50000</td>
<td>52,502</td>
<td>B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Solver</th>
<th>w/o partitioning</th>
<th>Proposed w/o partitioning</th>
<th>Proposed w/ partitioning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Sol</td>
<td>sp</td>
<td>Time</td>
</tr>
<tr>
<td>orthrgds</td>
<td>Conopt</td>
<td>439.41</td>
<td>5585.15</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>Ipopt</td>
<td>3.15</td>
<td>1523.90</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>Knitro</td>
<td>9.19</td>
<td>1776.23</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>Snopt</td>
<td>f</td>
<td>f</td>
<td>1.12</td>
</tr>
<tr>
<td>lukvl7</td>
<td>Conopt</td>
<td>35.71</td>
<td>62521.38</td>
<td>7.13</td>
</tr>
<tr>
<td></td>
<td>Ipopt</td>
<td>7.57</td>
<td>-18633.85</td>
<td>4.01</td>
</tr>
<tr>
<td></td>
<td>Knitro</td>
<td>43.49</td>
<td>-18633.85</td>
<td>5.10</td>
</tr>
<tr>
<td></td>
<td>Snopt</td>
<td>1966.72</td>
<td>-18633.85</td>
<td>139.74</td>
</tr>
</tbody>
</table>

### No Enumerations! - Problem Identification

- Each dot shows an instance.
- The best method for solving that particular instance.
- Ex) Knitro

Good starting points are important for large problems

- The same behavior applies to the other solvers as well.
Conclusions/Future Work

- We were able to solve all the problems using one of the methods.
- We can determine what technique to use before actually solving a problem.
- Integrate the process of generating good starting points as a preprocessor to existing NLP and MINLP benchmarks.