SBB: A New Solver for Mixed Integer Nonlinear Programming

Michael R. Bussieck
GAMS Development Corp.

Arne Drud
ARKI Consulting & Development A/S
Overview

- Introduction: The MINLP Model
- The B&B Algorithm
- Global/Local Solutions: What can go wrong
- The SBB Design Criteria and Implementation
- Some Results and Comparison with DICOPT
- Conclusions and Future Work
The MINLP Model

- Min or max $f(x,y)$
  
  s.t. $g(x,y) = b$
  
  $l \leq x \leq u$ continuous
  $l_i \leq y \leq u_i$ integer

- For convenience we only discuss “Min”
- SBB handles also: SOS1, SOS2, SemiContinuous, and SemiInteger Variables (not included in this talk)
Some Definitions

- When the Integrality Constraints on $y$ are removed we get the corresponding RMINLP model.
- A Node is the RMINLP model in which the bounds on $y$ may be tightened.
The B&B Algorithm

Solve RMINLP

Initialize/Store In a Node Table

Is the Node Table empty?
Yes

Select a Node, remove from Table, and Solve

Infeasible or Obj worse than Best?

Yes

Select a fractional discrete Variable. Create/store 2 nodes with Tighter Infeasible Bounds

Integer Solution?
Yes

Report and Stop

Unbounded
Infeasible
Integer

Report and Stop

Store Solution
Update Best
Remove Nodes With worse Obj

SBB: A new solver for MINLP

Michael R. Bussieck, Arne Drud
Assumptions Needed to Get a Global Optimum

- If a Node is declared “Infeasible” then it is indeed Globally Infeasible
- The objective for a Node is a valid lower bound for all integer solutions satisfying the bounds of the Node
- These assumptions are satisfied if the RMINLP model is convex in both $x$ and $y$. 
Non-Convex Models: Potential Problems

- The nodes are solved using an NLP Local Optimization Algorithm. Potential Problems are:
  - Branches or the Search Tree can incorrectly be cut off because:
    - A node is declared “Infeasible” even though it is only “locally infeasible” and there exist feasible regions.
    - A node is fathomed because the objective for the node is a poor approximation for the global optimum for the node.
  - In addition: Problems will occur if the NLP solver fails.
Desirable Properties of MINLP Models

• The feasible region for the RMINLP is connected and bound tightening in the nodes does not change this property.

• The nodes are not likely to be caught in bad local optima.
Design Objectives for SBB

- Should be able to use any existing GAMS NLP solver
- Should be able to handle solver failures
- Should have an option that can help against incorrect infeasibilities
- Should report information on observed non-convexities.
The SBB Implementation
The Concept

GAMS Process

Model Files

Solution Files

Solver Process

SBB Process

Revised Model Files

NLP Solver Process

RMINLP Solution Files
Efficient SBB Implementation

- GAMS Process
  - Model Files
  - Solution Files
- Solver Process
- SBB Process
  - Bound Revision File
  - RMINLP Solution Files
- NLP Solver Process
  - Node File with Primal and Dual Solution
Making a GAMS NLP Solver SBB-aware

• The GAMS Link is revised so
  - It knows whether it is called by SBB or not
  - It can read the Bound Revision File
  - It can get Bounds of parent plus initial primal and dual values from a node file and it can store the solution in the Node File

• Most of this is handled in the GAMS interface library

• Currently, no special solver changes
Solving the NLP Submodels

Parent Node with known Optimal Solution

Add One or a few bounds

Child Node with few primal Infeasibilities, Dual Feasible

• Solution Approaches:
  - Standard Phase 1 – Phase 2
  - Specialized Dual-Type Parametric Programming
Some SBB Options

- failseq solver1.opt1 solver2.opt2
  - Try solver1 with options defined by opt1. If it fails, try solver2 with options defined by opt2, etc. If all fail, ignore the node and continue the search
- infeasseq level solver1.opt1 solver2.opt2
  - If a node with depth ≤ level is locally infeasible then try solver1 with options defined by opt1, etc.
- rootsolver solver.opt
  - Use a special solver/option pair for the root node
- subiter/subres max
  - Avoid that one node uses all resources
67 Test Models

<table>
<thead>
<tr>
<th>Name</th>
<th>Rows</th>
<th>Columns</th>
<th>Discrete</th>
<th>NonZeros</th>
<th>NL</th>
<th>NNZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>4stufen</td>
<td>99</td>
<td>150</td>
<td>48</td>
<td>319</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>alanbar</td>
<td>9</td>
<td>10</td>
<td>4</td>
<td>26</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>batchdesb</td>
<td>21</td>
<td>21</td>
<td>9</td>
<td>55</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>beuster</td>
<td>115</td>
<td>158</td>
<td>52</td>
<td>398</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>cecil_13ba</td>
<td>900</td>
<td>842</td>
<td>180</td>
<td>2814</td>
<td>360</td>
<td></td>
</tr>
<tr>
<td>contvar</td>
<td>285</td>
<td>297</td>
<td>87</td>
<td>1281</td>
<td>530</td>
<td></td>
</tr>
<tr>
<td>deb10bar</td>
<td>131</td>
<td>184</td>
<td>22</td>
<td>694</td>
<td>432</td>
<td></td>
</tr>
<tr>
<td>deb6</td>
<td>508</td>
<td>476</td>
<td>20</td>
<td>2342</td>
<td>1432</td>
<td></td>
</tr>
<tr>
<td>deb7</td>
<td>898</td>
<td>814</td>
<td>10</td>
<td>4116</td>
<td>2816</td>
<td></td>
</tr>
<tr>
<td>deb8</td>
<td>898</td>
<td>824</td>
<td>10</td>
<td>4136</td>
<td>2816</td>
<td></td>
</tr>
<tr>
<td>deb9</td>
<td>918</td>
<td>814</td>
<td>10</td>
<td>4156</td>
<td>2816</td>
<td></td>
</tr>
<tr>
<td>detf1</td>
<td>6206</td>
<td>4408</td>
<td>400</td>
<td>26610</td>
<td>15400</td>
<td></td>
</tr>
<tr>
<td>ecop</td>
<td>1547</td>
<td>1536</td>
<td>19</td>
<td>3932</td>
<td>759</td>
<td></td>
</tr>
<tr>
<td>ecop2</td>
<td>927</td>
<td>937</td>
<td>2</td>
<td>2215</td>
<td>440</td>
<td></td>
</tr>
<tr>
<td>enpro48ba</td>
<td>216</td>
<td>155</td>
<td>92</td>
<td>744</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>enpro56ba</td>
<td>193</td>
<td>129</td>
<td>73</td>
<td>653</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>ex1222</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ex1223</td>
<td>14</td>
<td>12</td>
<td>4</td>
<td>40</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>ex1223a</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>32</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>ex1224bar</td>
<td>9</td>
<td>13</td>
<td>8</td>
<td>33</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>ex1225bar</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td>29</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ex1233bar</td>
<td>66</td>
<td>54</td>
<td>12</td>
<td>223</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>ex1243bar</td>
<td>98</td>
<td>70</td>
<td>16</td>
<td>331</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>ex1244bar</td>
<td>131</td>
<td>97</td>
<td>23</td>
<td>471</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>ex1252bar</td>
<td>45</td>
<td>41</td>
<td>15</td>
<td>120</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>ex1263aba</td>
<td>37</td>
<td>26</td>
<td>24</td>
<td>155</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>ex1263bar</td>
<td>57</td>
<td>94</td>
<td>72</td>
<td>243</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>ex1264aba</td>
<td>37</td>
<td>26</td>
<td>24</td>
<td>155</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>ex1264bar</td>
<td>57</td>
<td>90</td>
<td>68</td>
<td>239</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>ex1265aba</td>
<td>46</td>
<td>37</td>
<td>35</td>
<td>224</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>ex1265bar</td>
<td>76</td>
<td>132</td>
<td>100</td>
<td>349</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>ex1266aba</td>
<td>55</td>
<td>50</td>
<td>48</td>
<td>304</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>ex1266bar</td>
<td>97</td>
<td>182</td>
<td>138</td>
<td>478</td>
<td>72</td>
<td></td>
</tr>
</tbody>
</table>

SBB: A new solver for MINLP

Michael R. Bussieck, Arne Drud
Capability Issues

- Although discrete Variables in non-linear terms work in principal with DICOPT it usually results in a bad problem formulations. Therefore, it is disabled in GAMS/DICOPT. We recommend not to use discrete non-linear variables but permit their use.
- DICOPT does not support SOS and Semi Continuous Variables.
### Solution Status for 67 Models

<table>
<thead>
<tr>
<th></th>
<th>O</th>
<th>2</th>
<th>R</th>
<th>N</th>
<th>C</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>O</strong></td>
<td>36</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td><strong>U</strong></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td><strong>R</strong></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td><strong>N</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>36</td>
<td>3</td>
<td>1</td>
<td>12</td>
<td>15</td>
<td>67</td>
</tr>
</tbody>
</table>

- **Optimal**
- Unfinished (Int. Sol)
- Root Node Failure
- No Integer Solution
- Capability Issues

Subsolvers: CONOPT2, CPLEX6.6.
No options and no Alternatives at failure/infeasibility

---

SBB: A new solver for MINLP

Michael R. Bussieck, Arne Drud
## Comparison of 36 “Optimal” Models

<table>
<thead>
<tr>
<th></th>
<th>Objective</th>
<th>Solution Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Same</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>SBB Better</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>DICOPT Better</td>
<td>0</td>
<td>21</td>
</tr>
</tbody>
</table>
Future Work

• Today MINLP is at the stage where MIP was 10-15 years ago.

• Model Formulations:
  - Recommendations for good formulations and warnings against bad formulations
  - Preprocessing

• Algorithmic/Theoretical Work:
  - Preprocessing, Probing
  - Cut Generation, Pseudo Cost, ...

• System Work
  - Parallel NLP Solvers