Best practices for using the multistart algorithm in AIMMS

Marcel Hunting
AIMMS Optimization Specialist

Webinar, October 21, 2015
Overview

> Introduction: Global versus local optimization

> The AIMMS multistart algorithm visualized

> Parallel multistart

> Other changes in AIMMS 4.9

> Remarks & references
Global versus local optimum

- Global optimum
- Local optimum
Convex problems...

...local solver will do
Global nonlinear solvers (+ drawbacks)

> BARON

- Can “only” handle problems up to 10,000 variables and constraints
- Cannot handle problems with goniometric functions, external functions, if-then-else, $x^y$

> CPLEX

- Can solve non-convex quadratic programming problems
- Option Solution Target

BARON and CPLEX can also handle problems with integer variables
Multistart parameters

> MulStart::IterationLimit
> MulStart::UsePresolver
> MulStart::UseInitialPoint
> MulStart::ShrinkFactor
> MulStart::NumberOfBestSolutions
> MulStart::ShowSolverProgress
> MulStart::ThreadLimit
> MulStart::UseOpportunisticAlgorithm
Parallel multistart

> Controlled by the parameter MulStart::ThreadLimit
  • Default 0: Use as much threads as possible

> Amount of threads used is limited by:
  • Number of cores
  • AIMMS license & Solver

> Uses one solver
  • Not possible: a CONOPT solve in parallel with an IPOPT solve

> Can be **deterministic** or **opportunistic**
  • Deterministic means that multiple runs with the same model using the same parameter settings and the same solver on the same computer will produce the same results
## Parallel multistart – Results

**Model:** Elec  
**Minimization,** 601 vars, 201 cons, 1201 nz

**Iterations:** 2  
**Selected sample points:** 8 (from 16)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Deterministic</th>
<th>Opportunistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads</td>
<td>1  2  4  8</td>
<td>1  2  4  8</td>
</tr>
<tr>
<td># Solves</td>
<td>16 16 16 16</td>
<td>16 16 16 16</td>
</tr>
<tr>
<td># Solutions</td>
<td>16 16 16 16</td>
<td>16 16 16 16</td>
</tr>
<tr>
<td>Objective</td>
<td>18439 18439 18439 18439</td>
<td>18439 18439 18439 18439</td>
</tr>
<tr>
<td>Time</td>
<td>131.6 80.3 53.4 42.1</td>
<td>129.3 73.5 51.7 41.7</td>
</tr>
<tr>
<td>Speed up</td>
<td>1.00 1.64 2.47 3.13</td>
<td>1.02 1.76 2.50 3.10</td>
</tr>
</tbody>
</table>
Parallel multistart – Results (cont’d)

Model: Trainf Minimization, 20001 vars, 10003 cons, 60001 nz

Iterations: 5  Selected sample points: 8 (from 16)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Deterministic</th>
<th>Opportunistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads</td>
<td>1  2  4  8</td>
<td>1  2  4  8</td>
</tr>
<tr>
<td># Solves</td>
<td>7  6  8  8</td>
<td>7  8  7  8</td>
</tr>
<tr>
<td># Solutions</td>
<td>1  1  1  1</td>
<td>1  1  1  1</td>
</tr>
<tr>
<td>Time</td>
<td>115.8 85.1 80.6 73.9</td>
<td>115.9 77.1 69.1 73.0</td>
</tr>
<tr>
<td>Speed up</td>
<td>1.00 1.36 1.44 1.57</td>
<td>1.00 1.50 1.68 1.59</td>
</tr>
</tbody>
</table>

Remark: First solve takes much longer than rest
Parallel multistart – Results (cont’d)

Model: South  Maximization, 82 vars, 67 cons, 379 nz

Iterations: 100  Selected sample points: 32 (from 64)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Deterministic</th>
<th>Opportunistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads</td>
<td>1  2  4  8</td>
<td>1  2  4  8</td>
</tr>
<tr>
<td># Solves</td>
<td>75 89 95 95</td>
<td>75 90 94 97</td>
</tr>
<tr>
<td># Solutions</td>
<td>62 72 78 78</td>
<td>62 73 77 79</td>
</tr>
<tr>
<td>Objective</td>
<td>596.34 601.83 601.83 601.83</td>
<td>596.34 601.83 601.83 601.83</td>
</tr>
<tr>
<td>Time</td>
<td>15.6 14.6 14.5 13.8</td>
<td>15.1 13.7 14.0 13.7</td>
</tr>
<tr>
<td>Speed up</td>
<td>1.00 1.07 1.08 1.13</td>
<td>1.03 1.10 1.08 1.10</td>
</tr>
</tbody>
</table>
Parallel multistart – Results (cont’d)

Model: Blotter  Maximization, 24491 vars, 1180 cons, 126261 nz

Iterations: 3  Selected sample points: 32 (from 64)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Deterministic</th>
<th></th>
<th>Opportunistic</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Threads</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td># Solves</td>
<td>77</td>
<td>78</td>
<td>78</td>
<td>78</td>
</tr>
<tr>
<td># Solutions</td>
<td>77</td>
<td>78</td>
<td>78</td>
<td>78</td>
</tr>
<tr>
<td>Objective</td>
<td>847.09</td>
<td>847.09</td>
<td>847.09</td>
<td>847.09</td>
</tr>
<tr>
<td>Time</td>
<td>9689.9</td>
<td>6851.3</td>
<td>5160.9</td>
<td>4087.6</td>
</tr>
<tr>
<td>Speed up</td>
<td>1.00</td>
<td>1.41</td>
<td>1.88</td>
<td>2.37</td>
</tr>
</tbody>
</table>

Remark: With 8 selected sample points objective is 844.85
Solvers suitable for parallel multistart

- Solver has to be thread-safe
  - CONOPT
  - KNITRO

- Not thread-safe are: MINOS, SNOPT, (BARON)

- Special case: IPOPT
  - Thread-safety depends on Linear Solver (used to solve sparse symmetric indefinite linear systems)
    - MUMPS (not thread-safe) (Only one available in AIMMS)
    - MKL Pardiso (thread-safe)
    - HSL (thread-safe)

Build yourself. For instructions see: https://projects.coin-or.org/AIMMSlinks
Multistart changes in AIMMS 4.9

> Parallel multistart

> Use AIMMS Presolver by default

> Improved documentation in the Language Reference

> Progress window

MulStart::ShowSolverProgress
Some remarks

> KNITRO has its own parallel multistart algorithm
  • No clustering
  • No extra license requirements

> GMP-AOA (MINLP) can be used in combination with multistart

In procedure `SolveNLPSubProblem`, replace

```plaintext
GMP::SolverSession::Execute( ssNLP ) ;
GMP::Solution::RetrieveFromSolverSession( ssNLP, SolNumb ) ;
GMP::Solution::SendToModel( GNLP, SolNumb ) ;
```

by

```plaintext
MulStart::DoMultiStart( GNLP, 10, 5 ) ; ! You can play with the input values
GMP::Solution::RetrieveFromModel( GNLP, SolNumb ) ;
GMP::Solution::SendToSolverSession( ssNLP, SolNumb ) ;
```
References

Language Reference

• The AIMMS multistart algorithm: Chapter 17.2 (AIMMS 4)