OQNLP: a Scatter Search Multistart Approach for Solving Constrained Non-Linear Global Optimization Problems

Zsolt Ugray, The University of California at Riverside, MIS Dept.
Leon Lasdon, The University of Texas at Austin, MSIS Dept.
John Plummer, The University of Texas at Austin, MSIS Dept.
OQNLP-A Multi-start Search Method

• Intended for nonlinearly constrained, smooth, non-convex NLP’s and MINLP’s
• Combines the OptQuest Callable Library (a scatter search code) of Glover, Laguna, Kelly with a local NLP solver
• Currently uses LSGRG2, a sparse gradient-based GRG code, as the local NLP solver in callable mode
• GAMS version can call any GAMS NLP solver (conopt, minos, snopt, lsgrg2)
• Written in C
Other multistart methods: multi-level single linkage

- (1) Generate N uniformly distributed points, retain p*N best, where 0<p<1.
- (2) Start local solver from each retained point, unless there is a better point within a distance r (decreases).
- (3) add N more uniformly distributed points, go to (1)
- They Prove: total number of local searches finite and each local optimum located with probability one.
- Recent implementation by Fylstra of Frontline Systems for constrained problems, using L1 exact penalty function.
OptQuest (OQ) and OQ Callable Library

- OQ is the Glover-Laguna-Kelly implementation of scatter search
- Option within the **Crystal Ball** Excel add-in for Monte-Carlo simulation, and within **ARENA** for discrete event simulation
- Handles MINLP’s
- OQ callable library:
  - set up problem
  - define variables, bounds, constraints, requirements
  - create initial population
  - for N iterations:
    - get trial solution
    - evaluate objective
    - put trial solution, objective back to database
    - regenerate population if necessary
ex8_1_5 initial population
ex8_1_5 trial points
Search Method Drawbacks

- If feasible region is narrow (e.g. equality constraints) then very difficult for OQ (and other search methods) to find a feasible solution
- Often finds good solution fast, but requires many additional function evaluations to get high accuracy
- Gradient-based NLP solvers (GRG, SQP, SLP) are much better at getting feasible and attaining high accuracy, but find “nearest” local optimum.
When to Start the NLP Solver ($L$)

• An $L$ call is expensive—many function evaluations

• Ideally: start once in the basin of attraction of each local solution

• Don’t start from a trial point if
  – too close to a previously found local solution (*distance filter*)
  – exact penalty function value is too large (*merit filter*)
Distance Filter

• Store max distance traveled to each local optimum

• If the distance from a candidate starting point to any local opt is $< distfactor \times \text{maxdist}$, don’t start $L$ (default $distfactor = 0.75$)

• This assumes regions of attraction are spherical, and maxdist is a good estimate of the radius
Merit Filter

- Use L1 exact penalty, $P(x,w)$, as merit function
  - $P(x,w)=\text{obj}(x)+\sum(i,wi*\text{infeasi}(x))$
  - $wi > \max \text{ abs Lagrange multiplier for constraint } i \text{ over all local solutions}$
- Don’t start $L$ from candidate points which have value for $P > \text{threshold}$
- Initially threshold = best $P$ value over all candidate points so far
- If $P > \text{threshold}$ more than 20 consecutive times, \[ \text{threshold} = \text{threshold} + \text{threshfactor}*(1+\text{abs(\text{threshold}))} \] (default $\text{threshfactor} = 0.2$)
Dynamic Filters

• **Distance Filter**
  – actual basins of attraction partition the hyper-rectangle
  – adjust radii of spherical basin models so that
    • \( r(i) + r(j) \leq d(i,j) \)

• **Merit Filter**
  – at each rejected point \( x(k) \), compute factor, \( f(k) \), such that
    \[ \text{threshold} + f(k) \times (1+\text{abs(threshold)}) = \text{penval}(k) \]
  – if merit filter rejects \( waitcycle \) consecutive points, set factor for
    increasing threshold to
    \[ \text{factor} = \max[\text{user factor, min}(f(k))] \]

• **Effects still being evaluated**
  – somewhat more solver calls
  – harder to measure possible increase in robustness or reduction in
    solver calls to find global solution.
OQNLP Algorithm-Stage 1

INITIALIZE

Read_Problem_Parameters (size, bounds, starting point);
Setup_OptQuest_Parameters (size, iteration limits, population, accuracy, etc);
Initialize_OptQuest_Population;

Call $L$(user starting point, local solution)

STAGE 1: INITIAL OPTQUEST ITERATIONS AND FIRST CALL to $L$

WHILE (stage 1 iterations remain) DO {
    Get (trial solution from OptQuest);
    Evaluate (objective and nonlinear constraint values at trial solution,);
    Put (trial solution, objective and constraint values to OptQuest database); }

ENDDO

Get_Best_Point_from_OptQuest_database (starting point);

Call $L$ (starting point, local solution);

threshold = Penalty value of local solution;
OQNLP-Stage 2

STAGE 2: MAIN ITERATIVE LOOP

WHILE (stopping criteria not met) DO {
    Get (trial solution from OptQuest);
    Evaluate (objective and nonlinear constraint values at trial solution);
    Put (trial solution, objective and constraint values to OptQuest database);
    Calculate_Penalty_Function (trial solution);
    IF (distance and merit filter criteria are satisfied) THEN {
        Call_L (trial solution, local solution);
        Analyze_Solution (L Termination Condition);
        Update_Local_Solutions_Found;
        Update_Largest_Lagrange_Multipliers_Found;
    } ELSE IF (Pen> threshold for waitcycle consecutive iterations) increase threshold END DO
Points where GRG is called-200 initial OQ itns

-0.215

2.104
Why no starting points near other locals

• Best point from 200 initial OQ itns has $f = -.857$, lower than other 2 locals, and in basin of attraction of one of the globals.

• This is point for initial $L$ call, and sets initial threshold

• Even when threshold is increased, no other starting point is near the other locals, despite the effect of the distance filter.

• When no initial OQ itns, same thing happens.

• When threshold raised every 5 itns instead of 20, same thing happens, but $L$ called 22 times rather than 10.
Handling Discrete Variables

• Mode 1:
  – $L$ treats discrete variables as fixed, at values provided by OQ
  – OQ varies both discrete and continuous variables
  – $L$ solutions not returned to OQ
  – Apply distance and merit filters only if discrete variables are the same

• Mode 2:
  – as above, but OQ varies only the discrete variables
  – Obj value returned to OQ is optimal value found by $L$
  – equivalent to OQ solving the discrete problem, where $L$ optimizes over the continuous ones
GAMS Interface

- Motivation: Large Set of Test Problems Coded in GAMS by Floudas et.al., best known sol available
  - downloadable at http://titan.princeton.edu/TestProblems/
  - Most from Chemical process design or operation
  - 142+2N problems

- most small but some with over 100 variables, a few with over 1000.

- Most arise from chemical engineering

- Uses GAMS C Language Library Routines
<table>
<thead>
<tr>
<th>Series</th>
<th>problems</th>
<th>max vars</th>
<th>vars</th>
<th>max linear constraints</th>
<th>max nonlinear constraints</th>
<th>Problem Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX2_1_x</td>
<td>14</td>
<td>24</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>concave QP (min)</td>
</tr>
<tr>
<td>EX3_1_x</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>quad obj and const</td>
</tr>
<tr>
<td>EX4_1_x</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>obj or cons polynomial</td>
</tr>
<tr>
<td>EX5_2_x</td>
<td>2</td>
<td>32</td>
<td>0</td>
<td>8</td>
<td>11</td>
<td>bilinear-pooling</td>
</tr>
<tr>
<td>EX5_3_x</td>
<td>2</td>
<td>62</td>
<td>0</td>
<td>19</td>
<td>34</td>
<td>distillation col sequencing</td>
</tr>
<tr>
<td>EX5_4_x</td>
<td>3</td>
<td>27</td>
<td>0</td>
<td>13</td>
<td>6</td>
<td>heat exch network</td>
</tr>
<tr>
<td>EX6_1_x</td>
<td>4</td>
<td>12</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>gibbs free energy min</td>
</tr>
<tr>
<td>EX6_2_x</td>
<td>10</td>
<td>9</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>gibbs free energy min</td>
</tr>
<tr>
<td>EX7_2_x</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>3</td>
<td>12</td>
<td>gen geometric prog</td>
</tr>
<tr>
<td>EX7_3_x</td>
<td>6</td>
<td>17</td>
<td>0</td>
<td>10</td>
<td>11</td>
<td>robust stability ana.</td>
</tr>
<tr>
<td>EX8_1_x</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>small unconstrained, nl constrained</td>
</tr>
<tr>
<td>EX8_2_x</td>
<td>5</td>
<td>55</td>
<td>0</td>
<td>6</td>
<td>75</td>
<td>batch plant design-uncertainty</td>
</tr>
<tr>
<td>EX8_3_x</td>
<td>14</td>
<td>141</td>
<td>0</td>
<td>43</td>
<td>65</td>
<td>CSTR network synthesis</td>
</tr>
<tr>
<td>EX8_4_x</td>
<td>8</td>
<td>62</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>constrained least squares</td>
</tr>
<tr>
<td>EX8_5_x</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>min tan plane distance</td>
</tr>
<tr>
<td>EX8_6_1</td>
<td>N from 4 to 147</td>
<td>3N</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Lenard-Jones energy min</td>
</tr>
<tr>
<td>EX8_6_2</td>
<td>N from 5 to 80</td>
<td>3N</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Morse energy min</td>
</tr>
<tr>
<td>EX9_1_x</td>
<td>10</td>
<td>29</td>
<td>0</td>
<td>27</td>
<td>5</td>
<td>bilevel LP</td>
</tr>
<tr>
<td>EX9_2_x</td>
<td>9</td>
<td>16</td>
<td>0</td>
<td>11</td>
<td>6</td>
<td>bilevel QP</td>
</tr>
<tr>
<td>EX12_2_x</td>
<td>6</td>
<td>11</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>MINLP</td>
</tr>
<tr>
<td>EX14_1_x</td>
<td>9</td>
<td>10</td>
<td>0</td>
<td>4</td>
<td>17</td>
<td>infinity norm solution of equations</td>
</tr>
<tr>
<td>EX14_2_x</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>infinity norm solution of equations</td>
</tr>
<tr>
<td>Total</td>
<td>142+2N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Floudas Problem Summary
Parameters of base case

• Use 200 stage 1 iterations and 1000 total,
• OQ does not generate trial points which satisfy the linear constraints (greatly increases run times)
• Use LSGRG2 NLP solver for all but largest problems, SNOPT for these (Conopt and Lsgrg2 have infeasibility problems)
• Filter parameters (current defaults): waitcycle = 20, threshfactor = 0.2, distfactor = 0.75)
Floudas Continuous Problems-results

- Best known solution found or improved on in 118 of 128 problems using base case parameters
- Success if “gap” < 1%
- Best OQNLP solution found on first or second Solver call (itn 201) in 99 of 118 solved problems
- All the 10 remaining problems are solved by either loosening the filters or using (1000,5000)
- Solver is called at only 1% to 6% of the trial points where a call is allowed.
### Performance by # of Variables (avg)

<table>
<thead>
<tr>
<th>variable range</th>
<th>problems</th>
<th>variables</th>
<th>constraints</th>
<th>solver calls</th>
<th>locals found</th>
<th>function calls</th>
<th>time to best</th>
<th>total time</th>
<th>failed</th>
<th>first $L$ call</th>
<th>second $L$ call</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 4</td>
<td>32</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>2158</td>
<td>0.1</td>
<td>0.5</td>
<td>1</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td>4 to 7</td>
<td>31</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>4767</td>
<td>0.2</td>
<td>0.6</td>
<td>0</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td>8 to 12</td>
<td>21</td>
<td>9</td>
<td>8</td>
<td>13</td>
<td>3</td>
<td>19698</td>
<td>0.1</td>
<td>0.8</td>
<td>3</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>13 to 20</td>
<td>18</td>
<td>16</td>
<td>12</td>
<td>7</td>
<td>3</td>
<td>5212</td>
<td>0.3</td>
<td>0.7</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>22 to 78</td>
<td>13</td>
<td>39</td>
<td>28</td>
<td>14</td>
<td>3</td>
<td>23078</td>
<td>0.6</td>
<td>2.5</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>110 to 141</td>
<td>13</td>
<td>116</td>
<td>80</td>
<td>24</td>
<td>23</td>
<td>NA</td>
<td>6.6</td>
<td>64.1</td>
<td>0</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>total/avg</td>
<td>128</td>
<td>11</td>
<td>4</td>
<td>7191</td>
<td>0.4</td>
<td>1.7</td>
<td>9</td>
<td>80</td>
<td>19</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Solving 6 Lennard-Jones Problems Using CONOPT and Loose Filters

<table>
<thead>
<tr>
<th>variables</th>
<th>constraints</th>
<th>to best</th>
<th>Solver calls</th>
<th>found</th>
<th>time to best</th>
<th>time</th>
<th>Gap,%</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>10</td>
<td>1</td>
<td>152</td>
<td>39</td>
<td>1.09</td>
<td>21.83</td>
<td>0.00</td>
</tr>
<tr>
<td>24</td>
<td>45</td>
<td>21</td>
<td>130</td>
<td>114</td>
<td>18.56</td>
<td>68.34</td>
<td>0.00</td>
</tr>
<tr>
<td>39</td>
<td>105</td>
<td>6</td>
<td>104</td>
<td>100</td>
<td>13.63</td>
<td>165.09</td>
<td>0.00</td>
</tr>
<tr>
<td>54</td>
<td>190</td>
<td>67</td>
<td>118</td>
<td>118</td>
<td>257.6</td>
<td>396.21</td>
<td>1.12</td>
</tr>
<tr>
<td>69</td>
<td>300</td>
<td>42</td>
<td>94</td>
<td>94</td>
<td>325.8</td>
<td>730.68</td>
<td>1.84</td>
</tr>
<tr>
<td>84</td>
<td>435</td>
<td>16</td>
<td>59</td>
<td>59</td>
<td>134.4</td>
<td>434.56</td>
<td>0.88</td>
</tr>
</tbody>
</table>
Comparison-OQNLP and Random Starts

- **Floudas Series 8_6_1_x - Lennard-Jones Potential Function**

<table>
<thead>
<tr>
<th>atoms</th>
<th>vars</th>
<th>con</th>
<th>calls</th>
<th>nglob</th>
<th>diff</th>
<th>exp calls</th>
<th>oqnlp calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>100</td>
<td>99</td>
<td>2</td>
<td>1.0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>45</td>
<td>100</td>
<td>4</td>
<td>27</td>
<td>25.0</td>
<td>21</td>
</tr>
<tr>
<td>15</td>
<td>39</td>
<td>105</td>
<td>100</td>
<td>3</td>
<td>85</td>
<td>33.3</td>
<td>6</td>
</tr>
<tr>
<td>20</td>
<td>54</td>
<td>190</td>
<td>200</td>
<td>2</td>
<td>167</td>
<td>100</td>
<td>67</td>
</tr>
<tr>
<td>25</td>
<td>69</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42</td>
</tr>
<tr>
<td>30</td>
<td>84</td>
<td>435</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
</tr>
</tbody>
</table>

- nglob = number of times global min found
- diff = number of different local minima found
- pglob = nglob/calls
- exp calls = expected number of random starts until global first found
  \[ = \text{sum}(0=<k<\infty, k*pglob*(1-pglob)^{(k-1)}]=1/pglob \]
- oqnlp calls = number of OQNLP solver calls to find global min
Comparison-OQNLP and Random Starts

**Floudas Series 8_6_2_x - Morse Potential Function**

<table>
<thead>
<tr>
<th>Atoms</th>
<th>vars</th>
<th>con</th>
<th>calls</th>
<th>nglob</th>
<th>diff</th>
<th>best</th>
<th>exp calls to best</th>
<th>oqnlp calls to best</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>9</td>
<td>0</td>
<td>200</td>
<td>13</td>
<td>17</td>
<td>15.4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>0</td>
<td>200</td>
<td>1</td>
<td>107</td>
<td>200</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>39</td>
<td>0</td>
<td>200</td>
<td>9</td>
<td>161</td>
<td>22.2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>54</td>
<td>0</td>
<td>200</td>
<td>10</td>
<td>189</td>
<td>20</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>69</td>
<td>0</td>
<td>200</td>
<td>2</td>
<td>185</td>
<td>100</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>84</td>
<td>0</td>
<td>200</td>
<td>2</td>
<td>188</td>
<td>100</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>114</td>
<td>0</td>
<td>200</td>
<td>3</td>
<td>191</td>
<td>66.7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>144</td>
<td>0</td>
<td>200</td>
<td>3</td>
<td>181</td>
<td>66.7</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>
Solving 13 MINLP’s from Floudas set

- All are small—max binaries=8, max cont=23
- When OQ varies only binaries (subject to linear constraints
  - No stage one
  - All terminate after complete enumeration
  - Average(calls to best/total calls) approx 0.5
  - \( L \) called at every trial point
- When OQ varies all variables
  - Stage 1 helps find better values of binaries for stage 2
  - Average(calls to best all/calls to best bin only)=0.7
Advantages and Disadvantages

• Advantages
  – Finds good or optimal solutions rapidly
  – Can handle problems of general form with hundreds or thousands of continuous variables and/or constraints
  – strong ability to find a feasible solution (due to NLP solver)
  – works very differently from DICOPT or branch and bound

• Disadvantages
  – No guarantee of finding a global solution
  – No way to tell if you have found a global solution
Improving Reliability of NLP Solvers

• Many solver calls from diverse starting points are much more likely to succeed (find a local solution) than a single call
• OQNLP generates the starting points automatically and insures diversity
• OQNLP keeps track of all solutions found, returns best + all others if required.
• NLP solver failures are often due to termination at an infeasible point. Can recast problem as feasibility problem, apply OQNLP
• Feasibility mode: will eliminate need to reformulate problem (in future).
Improving Solver Reliability-Results

- Solved 147 Globallib problems with Conopt and OQNLP
- Conopt failed or ended infeasible in 24
- OQNLP got local solutions to 14 of these with 2 solver calls (58%)
- OQNLP got local solutions to 16 with 1000 iterations, about 40 solver calls per run avg.
GAMS and Callable Library Versions

• GAMS beta test release was 6/03 (see OR/MS today for Ad)
• Full user guide available
• Comparisons with Baron and LGO are planned in ‘03 using the test problems in Globalworld and Minlpworld.
• OQNLP Callable Library also available from OPTTEK Systems or Optimal Methods Inc
Future Work-feeding back local solutions to OQ

• Currently local solutions not returned to OQ
• Could start with initial NLP call at user initial point, pass NLP solution to OQ to include in initial population.
• In subsequent NLP calls, pass some NLP solutions back to OQ (must maintain diversity of its population).
• Criteria for which solutions are returned to OQ not clear.
More Future Work

• Determine effects of varying important OQMS options and parameters
  – basin and merit filter parameters
  – Performance of dynamic filters
  – OQ treats continuous vars as discrete
  – OQ strategies to intensify or diversify search
  – fractional change criterion for overall method

• Comparison with LGO, MLSL of Fylstra, DICOPT

• Improvements to OQMS algorithm motivated by test results
Better starting points for minlp

- If Optquest varies only the discrete variables, can generate all trial points arising from a given reference set at once.
- Choose a “best” point (minimal penval with continuous variables zero), sort the remaining points in increasing order of their distance from the “best” one.
- Solve the NLP subproblems in this order, using the previous optimal solution as the initial values for the continuous variables.
- Effects are being measured, but larger test problems probably needed.