A Problem Instance Analyzer for Optimization Services

Robert Fourer

Industrial Engineering & Management Sciences
Northwestern University, Evanston, IL, USA

`4er@iems.northwestern.edu`

Dominique Orban

Mathématiques et Génie Industriel
École Polytechnique Montréal, Canada

`dominique.orban@polymtl.ca`

INFORMS Annual Meeting
San Francisco — Tuesday, November 15, 2005
TC44.2, Optimization Tools and Modeling Languages

Abstract

We describe new developments in the design and testing of Dr. AMPL, a collection of utilities for determining properties of optimization problem instances generated from the AMPL modeling language. Dr. AMPL's problem analyzer checks properties ranging from size and sparsity to linearity and convexity, then compares the results to a database of solver characteristics to produce a list of recommended solvers. This information can then be fed to optimization services such as NEOS and OSxL.

Outline

Example 1: Nonlinear output from AMPL

Problem analysis

- Information included with problem instance
- Characteristics readily determined by analyzer
- Convexity *(with Arnold Neumaier & Hermann Schichl)*

Example 2: Analysis of a nonlinear problem

Solver choice

- Relational database
- Database queries

Example 2 (continued): Choice of a solver

Context . . .

Example 1

Nonlinear Output from AMPL

Transportation with nonlinear costs

```
set ORIG;    # origins
set DEST;    # destinations

param supply {ORIG} >= 0;    # amounts available at origins
param demand {DEST} >= 0;    # amounts required at destinations

param rate {ORIG,DEST} >= 0;    # base shipment costs per unit
param limit {ORIG,DEST} > 0;    # limit on units shipped

var Trans {i in ORIG, j in DEST}
   >= 1e-10, <= .9999 * limit[i,j], := limit[i,j]/2;

minimize Total_Cost:
   sum {i in ORIG, j in DEST}
      rate[i,j] * Trans[i,j]^0.8 / (1 - Trans[i,j]/limit[i,j]);

subject to Supply {i in ORIG}:
   sum {j in DEST} Trans[i,j] = supply[i];

subject to Demand {j in DEST}:
   sum {i in ORIG} Trans[i,j] = demand[j];
```

Example 1

Nonlinear Output *(cont'd)*

Transportation data

```
param: ORIG:  supply :=
        GARY   1400
        CLEV   2600
        PITT   2900 ;

param: DEST:  demand :=
        FRA     900    STL    1700
        DET    1200    FRE    1100
        LAN     600    LAF    1000
        WIN     400 ;

param rate :  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY   39   14   11   14   16   82    8
        CLEV   27    9   12    9   26   95   17
        PITT   24   14   17   13   28   99   20 ;

param limit :  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY   500 1000 1000 1000  800  500 1000
        CLEV   500  800  800  800  500  500 1000
        PITT   800  600  600  600  500  500  900 ;
```


Example 1

Nonlinear Output *(cont'd)*

AMPL's .nl file: Summary information in header

```
0 1        # nonlinear constraints, objectives
0 0        # network constraints: nonlinear, linear
0 21 0     # nonlinear vars in constraints, objectives, both
0 0 0 1    # linear network vars; functions; arith, flags
0 0 0 0 0  # discrete vars: binary, integer, nonlinear (b,c,o)
42 21      # nonzeros in Jacobian, gradients
0 0        # max name lengths: constraints, variables
0 0 0 0 0  # common exprs: b,c,o,c1,o1
```

. . . AMPL does all the work here

Example 1

Nonlinear Output *(cont'd)*

AMPL's .nl file: Nonlinear expressions

```
O0 0   #Total_Cost
o54    #sumlist
21
o3     #/
o2     #*
n39
o5     #^
v0     #Trans['GARY','FRA']
n0.8
o1     # -
n1
o3     #/
v0     #Trans['GARY','FRA']
n500
o3     #/
o2     #*
n14
o5     #^
........
```


Problem Analysis

Information included in .nl file header

- Size
- Differentiability
- Linearity
- Sparsity

Features readily deduced from expression trees

- Quadraticity
- Smoothness

Convexity . . .

Problem analysis

Convexity

Significance

- For an optimization problem of the form

$$\begin{array}{ll} \text{Minimize} & f(x_1,\ldots,x_n) \\ \text{Subject to} & g_i(x_1,\ldots,x_n) \ge 0, \quad i = 1,\ldots,r \\ & h_i(x_1,\ldots,x_n) = 0, \quad i = 1,\ldots,s \end{array}$$

  a local minimum is global provided
  - ⋆ f is convex
  - ⋆ each g_i is convex
  - ⋆ each h_i is linear
- Many physical problems are naturally convex if formulated properly

Analyses . . .

- Disproof of convexity
- Proof of convexity

Problem analysis

Disproof of Convexity

Find any counterexample

- Sample in feasible region
- Test any characterization of convex functions

Sampling along lines

- Look for $f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) > \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2)$
- See implementation in mProbe (John Chinneck, www.sce.carleton.ca/faculty/chinneck.html)

Sampling at points

- Look for $\nabla^2 f(\mathbf{x})$ not positive semi-definite
- Implemented in Dr. AMPL . . .

Problem analysis

Disproof of Convexity *(cont'd)*

Sampling

- Choose points $\mathbf{x}_0$
  such that $x_{01}, \ldots, x_{0n}$ are within inferred bounds

Testing

- Apply GLTR (galahad.rl.ac.uk/galahad-www/doc/gltr.pdf) to

$$\min_{\mathbf{d}} \; \nabla f(\mathbf{x}_0)\mathbf{d} + \tfrac{1}{2}\mathbf{d}\nabla^2 f(\mathbf{x}_0)\mathbf{d}$$
$$\text{s.t.} \quad \|\mathbf{d}\|_2 \leq \max\{10, \|\nabla f(x_0)\|/10\}$$

- Declare ***nonconvex*** if GLTR's Lanczos method
  finds a direction of negative curvature
- Declare ***inconclusive*** if GLTR
  reaches the trust region boundary
  without finding a direction of negative curvature

Problem analysis

Proof of Convexity

Recursively assess each expression tree node for

- Bounds
- Monotonicity
- Convexity / Concavity

Apply properties of functions

- $\|\mathbf{x}\|_p$ is convex, ≥ 0 everywhere
- x^α is convex for $\alpha \leq 0$, $\alpha \geq 1$; $-x^\alpha$ is convex for $0 \leq \alpha \leq 1$
- x^p for even $p > 0$ is convex everywhere,
  decreasing on $x \leq 0$, increasing on $x \geq 0$, *etc.*
- $-\log x$ and $x \log x$ are convex and increasing on $x > 0$
- $\sin x$ is concave on $0 \leq x \leq \pi$, convex on $\pi \leq x \leq 2\pi$,
  increasing on $0 \leq x \leq \pi/2$ and $3\pi/2 \leq x \leq 2\pi$, decreasing . . .
  ≥ -1 and ≤ 1 everywhere
- $\mathbf{x}^T\mathbf{M}\mathbf{x}$ is convex if $\mathbf{M}$ is positive semidefinite
- $e^{\alpha x}$ is convex, increasing everywhere for $\alpha > 0$, *etc.*
- $-(\Pi_i\, x_i)^{1/n}$ is convex where all $x_i > 0$ . . . ***etc., etc.***

Problem analysis

Proof of Convexity *(cont'd)*

Apply properties of convexity

- Certain expressions are convex:
  - $-f(\mathbf{x})$ for any concave f
  - $\alpha f(\mathbf{x})$ for any convex f and $\alpha > 0$
  - $f(\mathbf{x}) + g(\mathbf{x})$ for any convex f and g
  - $f(\mathbf{Ax} + \mathbf{b})$ for any convex f
  - $f(g(\mathbf{x}))$ for any convex nondecreasing f and convex g
  - $f(g(\mathbf{x}))$ for any convex nonincreasing f and concave g
- Use these with preceding to assess whether node expressions are convex on their domains

Apply properties of concavity, similarly

Deduce status of each nonlinear expression

- Convex, concave, or indeterminate
- Lower and upper bounds

Problem analysis

Testing Convexity Analyzers

Principles

- Disprovers can establish nonconvexity, suggest convexity
- Provers can establish convexity, suggest nonconvexity

Test problems

- Established test sets:
  COPS (17), CUTE (734), Hock & Schittkowski (119),
  Netlib (40), Schittkowski (195), Vanderbei (29 groups)
- Submissions to NEOS Server

Design of experiments

- Run a prover and a disprover on each test problem
- Check results for consistency
- Collect and characterize problems found to be convex
- Inspect functions not proved or disproved convex, to suggest possible enhancements to analyzers

Example 2

Analysis of a Nonlinear Problem

Torsion model (parameters and variables)

```
param nx > 0, integer;     # grid points in 1st direction
param ny > 0, integer;     # grid points in 2nd direction

param c;                   # constant

param hx := 1/(nx+1);      # grid spacing
param hy := 1/(ny+1);      # grid spacing

param area := 0.5*hx*hy;   # area of triangle

param D {i in 0..nx+1,j in 0..ny+1} =

  min( min(i,nx-i+1)*hx, min(j,ny-j+1)*hy );

                           # distance to the boundary

var v {i in 0..nx+1, j in 0..ny+1};

                           # definition of the
                           # finite element approximation
```


Example 2

Problem Analysis *(cont'd)*

Torsion model (objective and constraints)

```
var linLower = sum {i in 0..nx, j in 0..ny}
   (v[i+1,j] + v[i,j] + v[i,j+1]);

var linUpper = sum {i in 1..nx+1, j in 1..ny+1}
   (v[i,j] + v[i-1,j] + v[i,j-1]);

var quadLower = sum {i in 0..nx,j in 0..ny} (
   ((v[i+1,j] - v[i,j])/hx)**2 + ((v[i,j+1] - v[i,j])/hy)**2 );

var quadUpper = sum {i in 1..nx+1, j in 1..ny+1} (
   ((v[i,j] - v[i-1,j])/hx)**2 + ((v[i,j] - v[i,j-1])/hy)**2 );

minimize Stress:
  area * ((quadLower+quadUpper)/2 - c*(linLower+linUpper)/3);

subject to distanceBound {i in 0..nx+1, j in 0..ny+1}:
  -D[i,j] <= v[i,j] <= D[i,j];
```

Example 2

Problem Analysis *(cont'd)*

Output from AMPL's presolver

```
Presolve eliminates 2704 constraints and 204 variables.
Substitution eliminates 4 variables.

Adjusted problem:
2500 variables, all nonlinear
0 constraints
1 nonlinear objective; 2500 nonzeros.
```


Choice of a Solver

Relational database

- Table of identifiable *problem* characteristics
- Table of *solvers* and general information about them
- Table of all valid problem-solver pairs

Database queries

- Most specialized solvers
- Moderately specialized solvers:
  "hard" criteria such as convexity not used
- General-purpose solvers

Room for enhancement

- Add data from NEOS Server runs
- Automatically apply "best" solver (or solvers)

Example 2 (continued)

Choice of a Solver

Output from Dr. AMPL prototype (analysis)

```
Problem type
-------------
 -Problem has bounded variables
 -Problem has no constraints

Analyzing problem using only objective
--------------------------------------
 -This objective is quadratic
 -Problem is a QP with bounds

 -0.833013 <= objective <= 0.8359

Problem convexity
------------------
 Nonlinear objective looks convex on its domain.

 Detected 0/0 nonlinear convex  constraints,
          0/0 nonlinear concave constraints.
```


Example 2

Solver Choice

Output from Dr. AMPL (solver recommendations)

```
### Specialized solvers, based on all properties ###

        MOSEK
        OOQP

### Specialized solvers, excluding "hard" properties ###

        BLMVM
        FortMP
        L-BFGS-B
        MINLP
        MOSEK
        OOQP
        PathNLP
        SBB
        TRON

 ### General-purpose solvers ###

        KNITRO
        LANCELOT
        LOQO
```

Example 2

Solver Choice *(cont'd)*

Output from MOSEK solver run

```
ampl: model torsion.mod;
ampl: data torsion.dat;

ampl: option solver kestrel;
ampl: option kestrel_options 'solver=mosek';

ampl: solve;

Job has been submitted to Kestrel
Kestrel/NEOS Job number    : 280313
Kestrel/NEOS Job password  : ExPXrRcP

MOSEK finished.
(interior-point iterations - 11, simplex iterations - 0)

Problem status    : PRIMAL_AND_DUAL_FEASIBLE
Solution status   : OPTIMAL

Primal objective  : -0.4180876313
Dual objective    : -0.4180876333
```


Example 2

Solver Choice *(cont'd)*

Output from OOQP solver run

```
ampl: model torsion.mod;
ampl: data torsion.dat;

ampl: option solver kestrel;
ampl: option kestrel_options 'solver=ooqp';

ampl: solve;

Job has been submitted to Kestrel
Kestrel/NEOS Job number    : 280305
Kestrel/NEOS Job password  : VwLyfaVl

Check the following URL for progress report :
     http://www-neos.mcs.anl.gov/neos/neos-cgi/
          check-status.cgi?job=280305&pass=VwLyfaVl

Executing algorithm...
Finished call

OOQP completed successfully.

ampl: display Stress;
Stress = -0.333296
```

Example 2

Solver Choice *(cont'd)*

Output from TRON solver run

```
ampl: option solver kestrel;
ampl: option kestrel_options 'solver=tron';
ampl: solve;

Job has been submitted to Kestrel
Kestrel/NEOS Job number    : 280036
Kestrel/NEOS Job password  : xXbXViVa
Executing algorithm...

TRON: ------- SOLUTION ------- Finished call

Number of function evaluations                          9
Number of gradient evaluations                          9
Number of Hessian evaluations                           9
Number of conjugate gradient iterations                18

Projected gradient at final iterate              6.21e-07
Function value at final iterate               -0.41808763

Total execution time                             0.87 sec
Percentage in function evaluations                    24%
Percentage in gradient evaluations                    15%
Percentage in Hessian evaluations                     33%
```


Context . . .

Stand-alone

- A solver-like tool for AMPL
- An independent analysis tool like (or within) Mprobe
  - Invokes AMPL to get **.nl** file

Centralized optimization server

- A solver-like service at the NEOS Server
  - Compare the current "benchmark solver"

Decentralized optimization services

- An independent service
  - Listed on a central "registry"
  - Contacted directly by modeling systems