KNITRO 5.1
for Nonlinear Optimization

Todd Plantenga, Richard Waltz
Ziena Optimization, Inc.

May, 2007

Outline

- KNITRO overview
  - Product, where to use it
  - Three solvers
- Release history
  - KNITRO 5.0 enhancements
  - KNITRO 5.1 enhancements
- Examples
  - Strategic bidding with MPCCs
  - Sample AMPL code

KNITRO Overview

KNITRO Product

- Solve mathematical optimization problems
- **Robust** (3 different solvers)
- **Large-scale** (sparse linear algebra)
- **Embeddable** (multiple platforms, thread-safe)
- **Commercially supported** (Ziena since 2001)

Mathematical Problem Form

- KNITRO solves
  problems that can be written as

$$\begin{array}{lll} \min_{x \in \mathbf{R}^n} & f(x) & \\ \text{subject to} & g_i(x) \geq 0 & i \in I \\ & h_j(x) = 0 & j \in E \\ & l_i \leq x_i \leq u_i & \end{array}$$

- *x* must be continuous (no integer variables)
- *f(x)*, *g(x)*, *h(x)* must be smooth
- No convexity requirement

Optimization Problem Types

- General nonlinear problems with constraints
- Unconstrained
- Bound constrained
- Equality constrained
- Heuristics for better performance on
  - Complementarity constraints (MPCCs)
  - Nonlinear systems of equations
  - Least squares (linear and nonlinear)
  - Linear programs (LPs)
  - Quadratic programs (convex and nonconvex)

Where is KNITRO Used?

Optimal power flow

Strategic bidding, option pricing

Portfolio optimization, financial investments

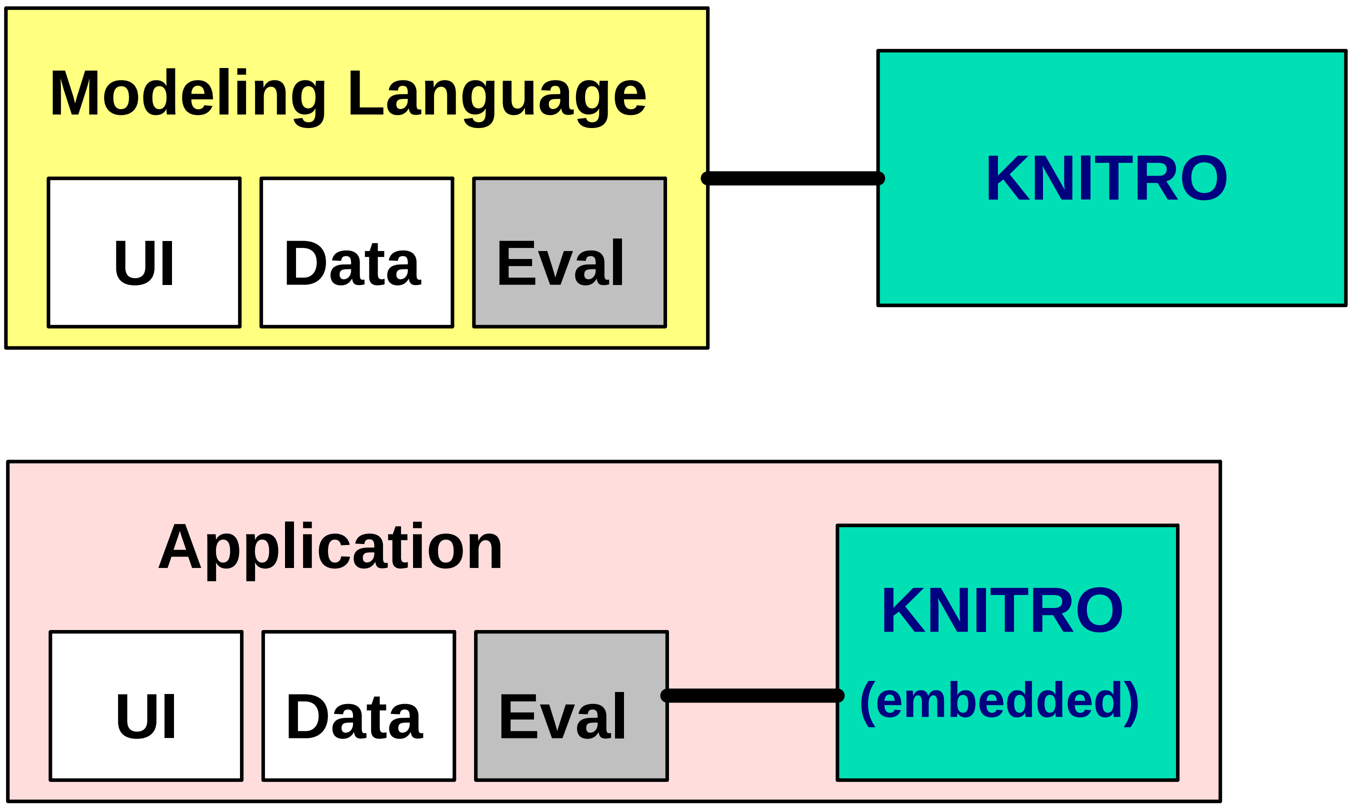
Semiconductor circuit layout

Chemical process control

Engineering design

Academic research (many applications)

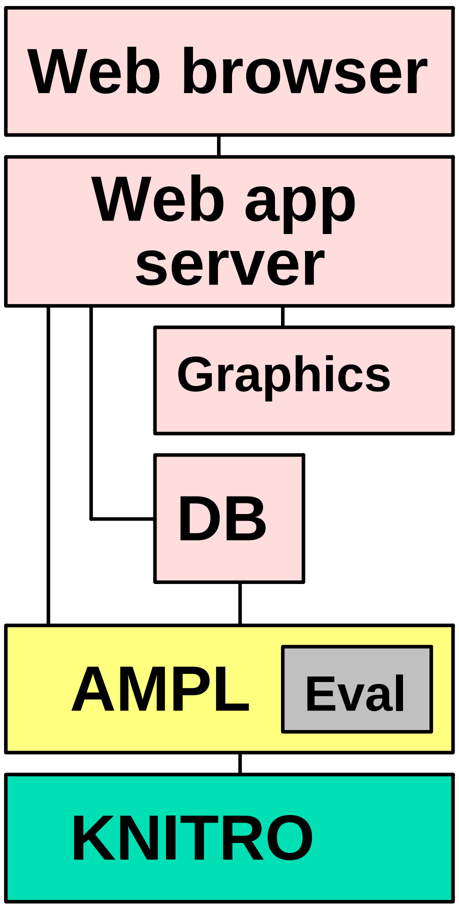
How is KNITRO Used?



"Eval" computes partial derivative values

Dynamic Application Example

Users modify problem, KNITRO solves in real time



Example tech stack:

- Browser with Javascript
- Apache/JBoss or Apache/Tomcat Java-based application server
- Graphics tool to generate plot images
- Data from RDBMS
- AMPL for modeling
- KNITRO for solving

Why Buy KNITRO?

- Power of 3 different solvers
- Large-scale performance
- Commercial support
- Ongoing state-of-the-art development

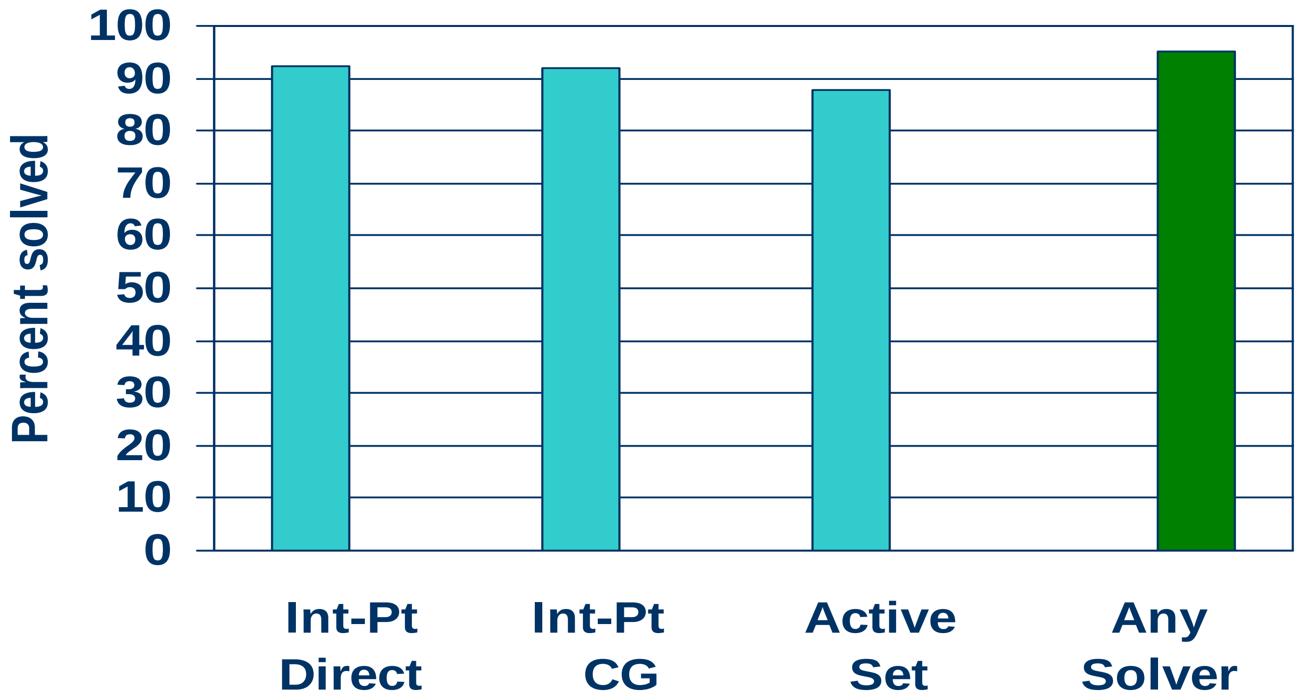
3 Different Solvers/Algorithms

- Interior-point / barrier
  - KNITRO **Interior/Direct** (alg=1, since release 2.0) (handles ill-conditioned problems)
  - KNITRO **Interior/CG** (alg=2, since release 1.0) (handles large/dense Hessians)
- Active Set SLQP
  - KNITRO **Active Set** (alg=3, since release 4.0)

    (able to warm start, crossover from Interior-point solvers)
- All state-of-the-art. KNITRO's underlying theory, published in academic papers by Ziena's founders, is cited in over 200 other publications (Google Scholar).

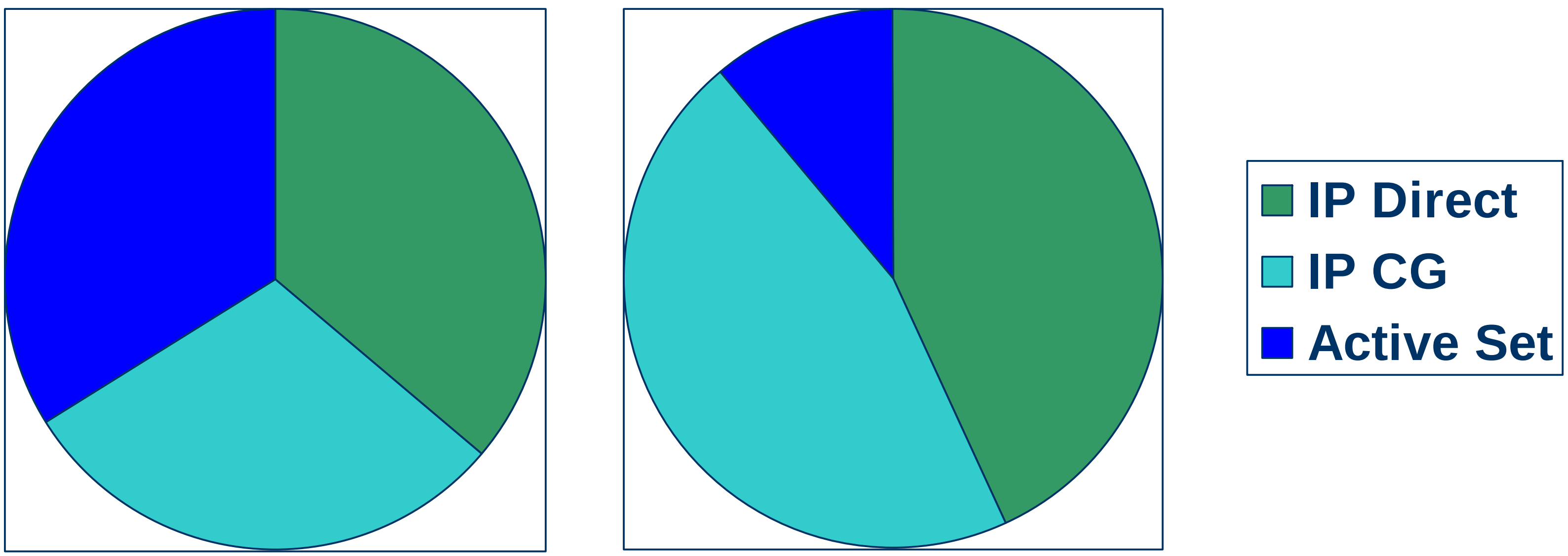
3 Solvers are more Robust

- 920 test problems
- 95.2% can be solved by at least one solver



3 Solvers are more Efficient

- 920 test problems
- "Best" solver when both find a solution:



Function Evaluations **CPU Time**

When to use Interior/Direct

- Second derivatives (Hessian) available
- Often the fastest
- Better than Interior/CG if ill-conditioned:

Problem BQPGauss (2003 variables)

Solver	iters	time	time/iter
Interior/Direct	19	3	0.2
Interior/CG	27	1310	48.5

When to use Interior/CG

- Second derivatives (Hessian) not available
- Hessian available but large (Interior/Direct spends too much time factorizing):

Problem CVXQP2 (10,000 variables)

Solver	iters	time	time/iter
Interior/Direct	14	2638	188.4
Interior/CG	11	401	36.5

When to use Active Set

- Warm start: a good start point is known, or a series of similar problems are solved

Example: solve for different values of a constraint parameter.

Active Set takes fewer iterations (problem HS66).

Parameter	Interior/Direct	Active Set
0.0 (cold start)	115	105
0.4 (warm)	13	3
0.8 (warm)	11	3
0.0 (warm)	10	1

KNITRO is Flexible Software

- Full platform support for
  - Windows: 32-bit and 64-bit
  - Linux: 32-bit and 64-bit
  - Mac OS X: x86 and PowerPC
  - Solaris: SPARC
- Simple API for applications coded in
  - C, C++, Fortran, Java
- Available as a solver for
  - AMPL
  - AIMMS
  - Frontline (Excel spreadsheets)
  - GAMS
  - Mathematica
  - TOMLAB (MATLAB and LabView)

KNITRO is Commercial Software

- Production libraries
  - Thread-safe for embedding
  - Debugging capabilities
  - Examples provided in C, C++, Fortran, Java
- Ziena incorporated in 2001
- Ziena license manager
- Ziena support (Annual Maintenance Service)
  - Software assistance, bug fixes
  - Optimization problem formulation
  - Free upgrades for new features

KNITRO Availability

- **http://www.ziena.com**
- **Free** “student” download
  - 300 variable limit
- **Free** 30-day trial of full strength version
- Academic pricing for University research

KNITRO Release History

ziena OPTIMIZATION INC.

KNITRO

KNITRO Release History

5.1 – Currently Available

5.1.0 (Nov 2006) – minor release

5.0 – No longer available (but still supported)

5.0.3 (Jun 2006) – bug fix

5.0.0 (Feb 2006) – major release

4.0 – No longer available (but still supported)

4.0.0 (Oct 2004) – major release

3.1 – No longer available

3.1.0 (Sep 2003) – minor release

3.0.0 (Apr 2003) – major release

(see *http://www.ziena.com/release_hist.htm*)

KNITRO 5.0 Enhancements

New in KNITRO 5.0

- Capability to solve **MPCC** problems
- **Crossover** from an interior-point solution to the Active Set solver for highly accurate solutions
- **Multi-start** option for trying to find the global solution
- Additional barrier update rule
- Streamlined programming interface
- Built-in derivative checking
- Improvements in algorithm efficiency and robustness

MPCC Problems

Math **P**rogramming with **C**omplementarity **C**onstraints
aka: Equilibrium Constraints / MPEC / NCP

$$\min f(x)$$
$$\text{s.t. } g(x) \geq 0$$
$$h(x) = 0$$
$$0 \leq x_1 \perp x_2 \geq 0$$

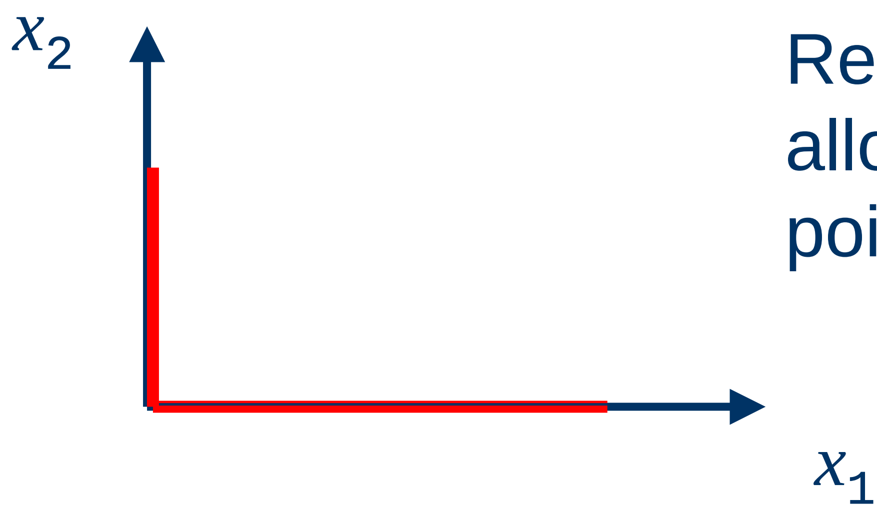
Applications:
- Strategic bidding
- Economic models
- Contact problems
- Traffic equilibrium
- Disjunctive conditions

The complementarity condition means

$x_1 \geq 0$, $x_2 \geq 0$ and
either $x_1 = 0$ or $x_2 = 0$ (equivalently, $x_1 x_2 = 0$)

Why is MPCC Hard?

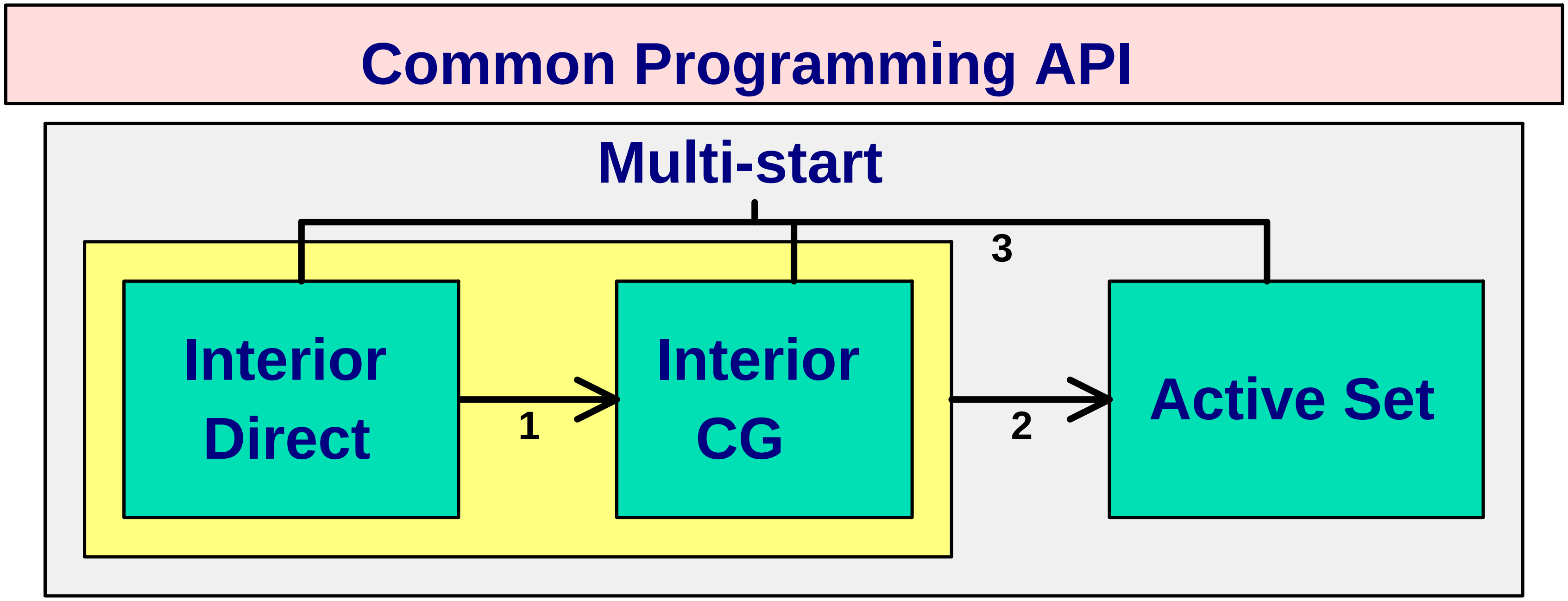
$$\min \; f(x)$$
$$\text{s.t.} \;\; g(x) \geq 0$$
$$h(x) = 0$$
$$0 \leq x_1 \perp x_2 \geq 0$$



- Constraint qualification fails at feasible points
- Multipliers undefined, no interior path

Recent research breakthroughs allow reformulation for interior-point solvers

Solver Interactions



1. Interior/Direct falls back on Interior/CG for improved robustness
2. Interior solvers can cross over to Active set
3. Multi-start searches for best local minima

Crossover

Crossover cleans up the Interior-point solution by switching to the Active Set solver at the end

- Cleanly identifies active set of constraints
- More accurate solution and sensitivities
- Standard for LP but very difficult for NLP because it requires **2 solvers**

- Can be switched off as there may be overhead costs

Crossover Example (HS35)

Interior-point solution (6 interior-point iterations):

```
Constraint Vector                    Lagrange Multipliers
c[   0] =  2.99999960166e+000,   lambda[   0] =  2.22222343515e-001
Solution Vector
x[   0] =  1.33333347124e+000,   lambda[   1] = -3.43504445156e-008
x[   1] =  7.77777685286e-001,   lambda[   2] = -1.71771265657e-008
x[   2] =  4.44444222565e-001,   lambda[   3] = -3.83389961963e-008
```

Crossover solution (1 additional crossover iteration):

```
Constraint Vector                    Lagrange Multipliers
c[   0] =  3.00000000000e+000,   lambda[   0] =  2.22222306741e-001
Solution Vector
x[   0] =  1.33333333333e+000,   lambda[   1] =  0.00000000000e+000
x[   1] =  7.77777777778e-001,   lambda[   2] =  0.00000000000e+000
x[   2] =  4.44444444444e-001,   lambda[   3] =  0.00000000000e+000
```

Multi-start

- KNITRO solvers converge to a local minimum
- Different start points may go to different minima
- Efficient restarting (especially Active Set solver)
- Framework in place for sophisticated global search

KNITRO 5.1 Enhancements

New in KNITRO 5.1

- New options to improve Interior-point performance
- **Intel MKL** (Math Kernel Library) included for 20-30% speedup on BLAS/LAPACK operations
- **Newpoint** capability saves iterates, especially useful when progress is slow
- **Multi-start** generation of new start points improved with several new options
- **Mathematica** interface provides compiled object alternative for large-scale problems

Examples

Strategic Bidding for Electricity

- Your company sells electric power
- You and other producers submit competitive bids to generate power
- An ISO purchases at a single “spot price”

- Your strategic guidance:
  - submit low bids ≤ spot price
  - submit high bids to drive up the spot price

Strategic Bidding for Electricity

$b_{i,j}$	bid price by company i, plant j
$c_{i,j}$	generating cost per kilowatt
$G_{i,j}$	generating capacity
$g_{i,j}$	kilowatts to generate for ISO

ISO problem:

$$\begin{array}{ll} \min\limits_{g_{i,j}} & \sum\limits_{i,j} b_{i,j} g_{i,j} \\ \text{subject to} & 0 \leq g_{i,j} \leq G_{i,j} \\ & \sum\limits_{i,j} g_{i,j} \geq \text{Demand} \end{array}$$

Spot price π_D is multiplier

Strategic Bidding for Electricity

Your problem ($i = 1$):

$$\max_{b_j} \quad \sum_j (\pi_D - c_j) g_j$$

$$\text{subject to} \begin{cases} c_j \le b_j \\ \text{ISO solution} \end{cases}$$

ISO problem: $b_{i,j} \to g_{i,j}, \pi_D$

- Bilevel linear optimization problem
- Can solve iteratively, but converges slowly

Strategic Bidding for Electricity

- Replace ISO problem with KKT conditions
- Solve as a nonlinear MPEC with KNITRO 5.0

$$
\begin{aligned}
b_{i,j} - \pi_D + \pi^U_{i,j} - \pi^L_{i,j} = 0 \quad & \forall i,j \\
0 \le g_{i,j} \le G_{i,j} \quad & \forall i,j \\
\sum_{i,j} g_{i,j} \ge Demand &
\end{aligned}
$$

$$
\begin{aligned}
0 \le \pi_D &\perp \sum_{i,j} g_{i,j} - Demand \ge 0 \\
0 \le \pi^U &\perp G - g \ge 0 \\
0 \le \pi^L &\perp g \ge 0
\end{aligned}
$$

Strategic Bidding for Electricity

- What about bids from competitors?
- Use stochastic optimization
  - Same nonlinear MPEC formulation
  - Different scenarios for bids
  - Larger, but KNITRO handles large-scale problems

AMPL Interface

- Algebraic Modeling and Programming Language
- Intuitive syntax
- Interactive user interface
- Extremely efficient evaluation of partial derivatives for optimization solvers
- File I/O, database connectivity
- Lightweight software component
- Sold by Ziena, same license manager as KNITRO

AMPL Interface Sample Code

```
# Example nonlinear AMPL model.

# Define variables and make them non-negative.
var x{j in 1..3} >= 0;

# Define the objective function to be minimized.
minimize obj:
      1000 - x[1]^2 - 2*x[2]^2 - x[3]^2 - x[1]*x[2] – x[1]*x[3];

# Equality constraint.
s.t.  c1: 8*x[1] + 14*x[2] + 7*x[3] - 56 = 0;

# Inequality constraint.
s.t.  c2: x[1]^2 + x[2]^2 + x[3]^2 -25 >= 0;

# Define initial point.
data;
let x[1] := 2;
let x[2] := 2;
let x[3] := 2;
```

AMPL Interface User Session

```
ampl: option solver knitroampl;
ampl: option knitro_options "outlev=0 alg=2";
ampl: model testproblem.mod;
ampl: solve;

KNITRO 5.1.0: LOCALLY OPTIMAL SOLUTION FOUND.
objective 9.360004e+02; feasibility error 0.000000e+00
6 major iterations; 7 function evaluations

ampl: display x[3];
x[3] = 7.99997
ampl: option knitro_options “outlev=0 alg=2 maxcrossit=1”;
ampl: solve;

KNITRO 5.1.0: LOCALLY OPTIMAL SOLUTION FOUND.
objective 9.360000e+02; feasibility error 0.000000e+00
5 major iterations; 6 function evaluations

ampl: display x[3];
x[3] = 8
```