Stochastic Programming, Equilibria and Extended Mathematical Programming

Michael C. Ferris

Joint work with: Michael Bussieck, Jan Jagla, Lutz Westermann and Roger Wets

Supported partly by AFOSR, DOE and NSF

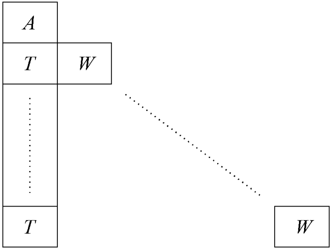
University of Wisconsin, Madison

Workshop on Role of Uncertainty and Risk, Center for Energy Policy and Economics, ETH, Zurich
November 2, 2011

Stochastic recourse

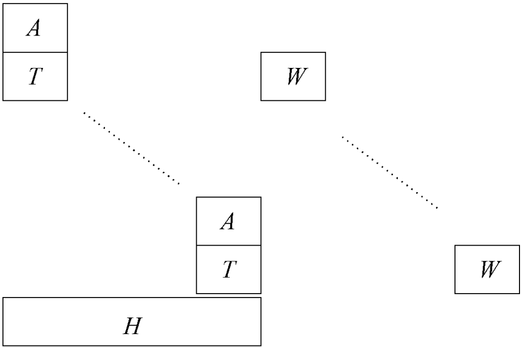
- Two stage stochastic programming, x is here-and-now decision, recourse decisions y depend on realization of a random variable
- $\mathbb{R}$ is a risk measure (e.g. expectation, CVaR)

$$\begin{aligned} \text{SP:} \ \min \quad & c^\top x + \mathbb{R}[d^\top y] \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0, \\ \forall \omega \in \Omega : \quad & T(\omega)x + W(\omega)y(\omega) \geq h(\omega), \\ & y(\omega) \geq 0. \end{aligned}$$



Key-idea: Non-anticipativity constraints

- Replace x with $x_1, x_2, \ldots, x_K$
- Non-anticipativity: $(x_1, x_2, \ldots, x_K) \in L$ (a subspace) - the H constraints



Computational methods exploit the separability of these constraints, essentially by dualization of the non-anticipativity constraints.

- Primal and dual decompositions (Lagrangian relaxation, progressive hedging , etc)
- L shaped method (Benders decomposition applied to det. equiv.)
- Trust region methods and/or regularized decomposition

Models with explicit random variables

- Model transformation:
  - Write a core model as if the random variables are constants
  - Identify the random variables and decision variables and their staging
  - Specify the distributions of the random variables
- Solver configuration:
  - Specify the manner of sampling from the distributions
  - Determine which algorithm (and parameter settings) to use
- Output handling:
  - Optionally, list the variables for which we want a scenario-by-scenario report

Example: Farm Model (core model)

- Allocate land (L) for planting crops $x(c)$ to max (p/wise lin) profit
- Yield rate per crop c is $F{*}Y(c)$
- Can purchase extra crops b and sell s, but must have enough crops d to feed cattle

$$\begin{aligned} \max_{x,b,s\geq 0} \quad & profit = p(x,b,s) \\ \text{s.t.} \quad & \sum_c x(c) \leq L, \\ & F{*}Y(c) * x(c) + b(c) - s(c) \geq d(c) \end{aligned}$$

- Random variables are F, realized at stage 2: structured $T(\omega)$
- Variables x stage 1, b and s stage 2.
- landuse constraints in stage 1, requirements in stage 2.

Can now generate the *extensive form* problem or pass on directly to specialized solver

Stochastic Programming as an EMP

Three separate pieces of information (extended mathematical program) needed

1. emp.info: model transformation

```
randvar F discrete 0.25 0.8 // below
                   0.50 1.0 // avg
                   0.25 1.2 // above

stage 2 F b s req
```

2. solver.opt: solver configuration (benders, sampling strategy, etc)

```
4 "ISTRAT"  * solve universe problem (DECIS/Benders)
```

3. dictionary: output handling (where to put all the "scenario solutions")

How does this help?

- Clarity/simplicity of model
- Separates solution process from model description
- Models can be solved by the extensive form equivalent, exisiting codes such as LINDO and DECIS, or decomposition approaches such as Benders, ATR, etc
- Allows description of compositional (nonlinear) random effects in generating ω

$$\text{i.e. } \omega = \omega_1 \times \omega_2, \;\; T(\omega) = f(X(\omega_1), Y(\omega_2))$$

- Easy to write down multi-stage problems
- Automatically generates "COR", "TIM" and "STO" files for Stochastic MPS (SMPS) input

Sampling methods

But what if the number of scenarios is too big (or the probability distribution is not discrete)? use sample average approximation (SAA)

- Take sample $\xi_1, \ldots, \xi_N$ of N realizations of random vector ξ
  - viewed as historical data of N observations of ξ, or
  - generated via Monte Carlo sampling
- for any $x \in X$ estimate $f(x)$ by averaging values $F(x, \xi_j)$

$$\text{(SAA):} \min_{x \in X} \left\{ \hat{f}_N(x) := \frac{1}{N} \sum_{j=1}^{N} F(x, \xi_j) \right\}$$

- Nice theoretical asymptotic properties
- Can use standard optimization tools to solve the SAA problem
- EMP = SLP $\implies$ SAA $\implies$ (large scale) LP

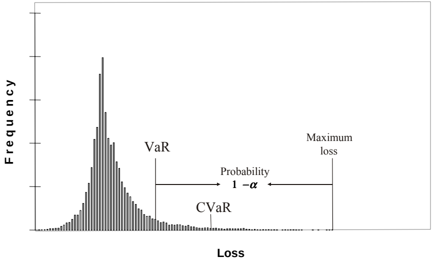
Risk Measures

- Classical: utility/disutility $u(\cdot)$:

$$\min_{x \in X} f(x) = \mathbb{E}[u(F(x,\xi))]$$

- Modern approach to modeling risk aversion uses concept of risk measures

$\overline{CVaR}_\alpha$: mean of upper tail beyond α-quantile (e.g. $\alpha = 0.95$)



- mean-risk, semi-deviations, mean deviations from quantiles, VaR, CVaR
- Römisch, Schultz, Rockafellar, Urasyev (in Math Prog literature)
- Much more in mathematical economics and finance literature
- Optimization approaches still valid, different objectives

Example: Portfolio Model (core model)

- Determine portfolio weights w_j for each of a collection of assets
- Asset returns v are random, but jointly distributed
- Portfolio return $r(w, v)$
- Minimize a "risk" measure

$$\begin{array}{ll} \max & 0.2 * \mathbb{E}(r) + 0.8 * \underline{CVaR}_\alpha(r) \\ \text{s.t.} & r = \sum_j v_j * w_j \\ & \sum_j w_j = 1, \; w \geq 0 \end{array}$$

- Jointly distributed random variables v, realized at stage 2
- Variables: portfolio weights w in stage 1, returns r in stage 2
- Coherent risk measures $\mathbb{E}$ and $\underline{CVaR}$

Other EMP information

- emp.info: model transformation

```
expected_value EV_r r
cvarlo         CVaR_r r alpha
stage          2 v r defr
jrandvar       v("att") v("gmc") v("usx") discrete
   table of probabilities and outcomes
```

- Variables are assigned to $\mathbb{E}(r)$ and $\underline{CVaR}_\alpha(r)$; can be used in model (appropriately) for objective, constraints, or be bounded
- Problem transformation: theory states this expression can be written as convex optimization using:

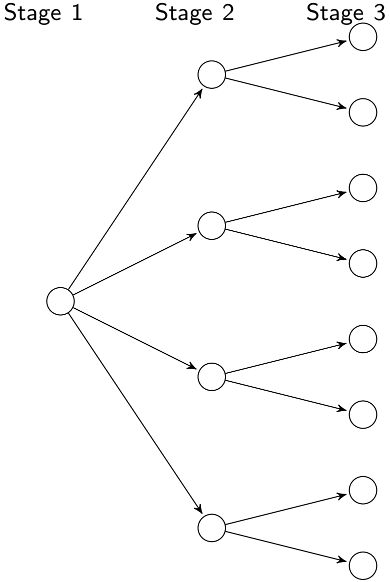
$$\underline{CVaR}_\alpha(r) = \max_{a \in \mathbb{R}} \left\{ a - \frac{1}{\alpha} \sum_{j=1}^{N} Prob_j * (a - r_j)_+ \right\}$$

Solution options

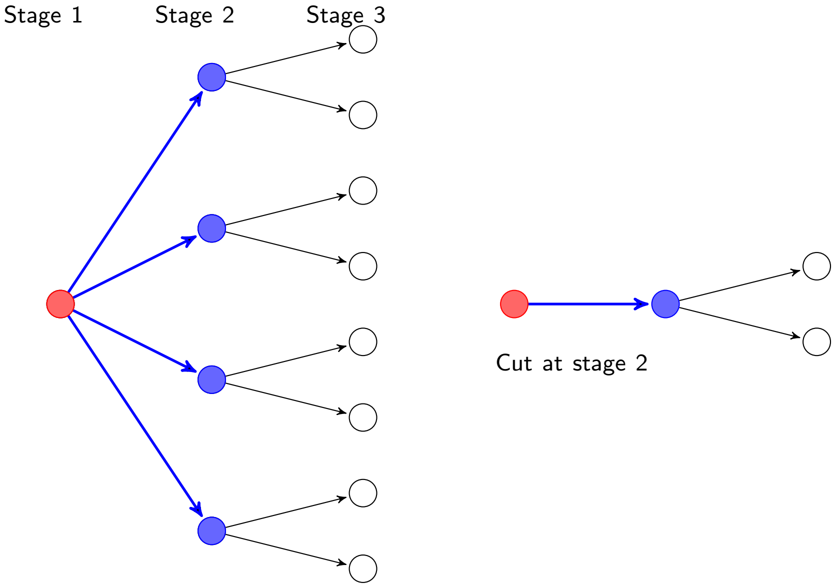
- Form the extensive form equivalent
- Solve using LINDO api (stochastic solver)
- Convert to two stage problem and solve using DECIS or any number of competing methods
- Problem with $3^{40} \approx 1.2 * 10^{19}$ realizations in stage 2
  - DECIS using Benders and Importance Sampling: < 1 second (and provides confidence bounds)
  - CPLEX on a presampled approximation:

sample	samp. time(s)	CPLEX time(s) for solution			cols (mil)
500	0.0	5	(4.5 barrier,	0.5 xover)	0.25
1000	0.2	18	(16 barrier,	2 xover)	0.5
10000	28	195	(44 barrier,	151 xover)	5
20000	110	1063	(98 barrier,	965 xover)	10

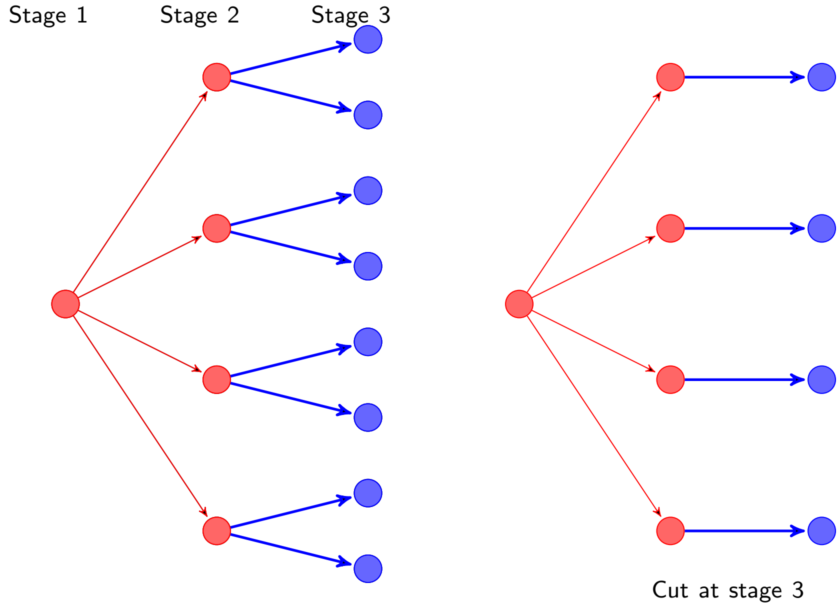
Multi to 2 stage reformulation



Multi to 2 stage reformulation



Multi to 2 stage reformulation



Additional techniques requiring extensive computation

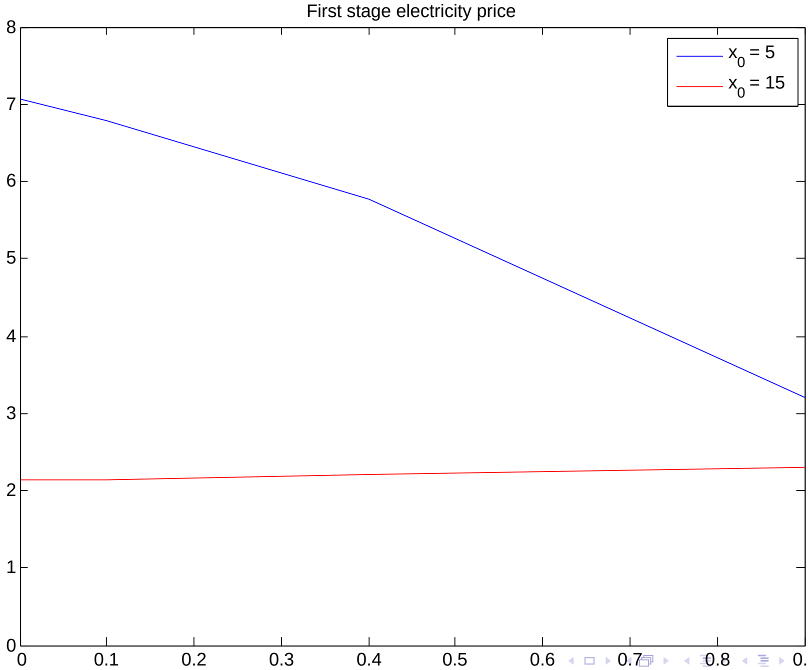
- Continuous distributions, sampling functions, density estimation
- Chance constraints: $Prob(T_i x + W_i y_i \geq h_i) \geq 1 - \alpha$ - can reformulate as MIP and adapt cuts (Luedtke) empinfo: chance E1 E2 0.95
- Use of discrete variables (in submodels) to capture logical or discrete choices (logmip - Grossmann et al)
- Robust or stochastic programming
- Decomposition approaches to exploit underlying structure identified by EMP
- Nonsmooth penalties and reformulation approaches to recast problems for existing or new solution methods (ENLP)
- Conic or semidefinite programs - alternative reformulations that capture features in a manner amenable to global computation

Risk averse hydro (Philpott, MCF, Wets)

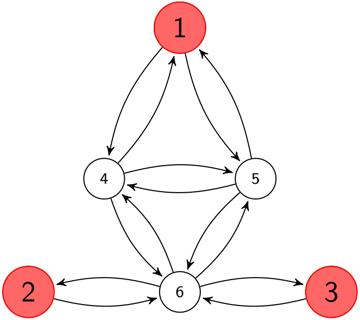
- Hydro agents solve two stage stochastic program with increasing risk aversion
- Thermal agents solve two stage stochastic program (in effect risk neutral)
- Prices cleared in both periods by Walras
- Modeled as a MOPEC:

$$\min_{x_i \in X_i} c(x_i, x_{-i}, p)$$

$$0 \leq S(x, p) - D(x, p) \perp p \geq 0$$


First stage electricity price
x_0 = 5
x_0 = 15
0
1
2
3
4
5
6
7
8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

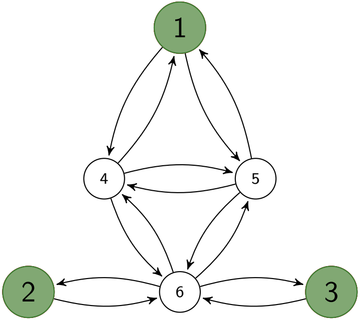
Spatial Price Equilibrium (Dirkse)



$n \in \{1, 2, 3, 4, 5, 6\}$
$L \in \{1, 2, 3\}$

Supply quantity: S_L
Production cost: $\Psi(S_L) = ..$

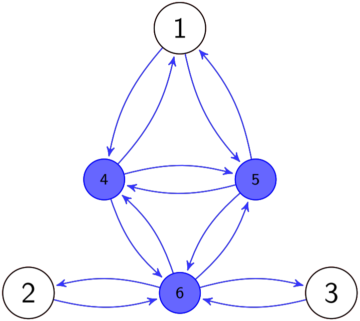
Spatial Price Equilibrium (Dirkse)



$n \in \{1, 2, 3, 4, 5, 6\}$
$L \in \{1, 2, 3\}$

Supply quantity: S_L
Production cost: $\Psi(S_L) = ..$
Demand: D_L
Unit demand price: $\theta(D_L) = ..$

Spatial Price Equilibrium (Dirkse)



$n \in \{1, 2, 3, 4, 5, 6\}$
$L \in \{1, 2, 3\}$

Supply quantity: S_L
Production cost: $\Psi(S_L) = ..$
Demand: D_L
Unit demand price: $\theta(D_L) = ..$
Transport: T_{ij}
Unit transport cost: $c_{ij}(T_{ij}) = ..$

One large system of equations and inequalities to describe this (GAMS).

Nonlinear Program Model (Monopolist)

- One producer controlling all regions
- Full knowledge of demand system
- Full knowledge of transportation system

$$
\begin{aligned}
\max_{D,S,T} \quad & \sum_{l \in L} \theta_l(D_l) D_l - \sum_{l \in L} \Psi_l(S_l) - \sum_{i,j} c_{ij}(T_{ij}) T_{ij} \\
\text{s.t.} \quad & S_l - D_l + \sum_{i,l} T_{il} - \sum_{l,j} T_{lj} = 0 \ \forall l \in L \\
& D, S, T \in F
\end{aligned}
$$

EMP = NLP

2 agents: NLP + VI Model (Monopolist)

- One producer controlling all regions
- Full knowledge of demand system
- Price-taker in transportation system

$$\max_{D,S,T} \quad \sum_{l \in L} \theta_l(D_l) D_l - \sum_{l \in L} \Psi_l(S_l) - \sum_{i,j} \overset{p_{ij}}{\cancel{c_{ij}(T_{ij})}} T_{ij}$$

$$\text{s.t.} \quad S_l - D_l + \sum_{i,l} T_{il} - \sum_{l,j} T_{lj} = 0 \ \forall l \in L$$

$$D, S, T \in F$$

$$p_{ij} = \quad c_{ij}(T_{ij})$$

empinfo: vi tcDef tc

2 agents: NLP + VI Model (Monopolist)

- One producer controlling all regions
- Full knowledge of demand system
- Price-taker in transportation system

$$\max_{D,S,T} \quad \sum_{l \in L} \theta_l(D_l) D_l - \sum_{l \in L} \Psi_l(S_l) - \sum_{i,j} \overset{p_{ij}}{\cancel{c_{ij}(T_{ij})}} T_{ij}$$

$$\text{s.t.} \quad S_l - D_l + \sum_{i,l} T_{il} - \sum_{l,j} T_{lj} = 0 \ \forall l \in L$$

$$D, S, T \in F$$

$$p_{ij} = \quad c_{ij}(T_{ij})$$

$$\text{EMP} = \text{MOPEC} \implies \text{MCP}$$

Classic SPE Model (NLP + VI agents)

- One producer controlling all regions
- Price-taker in demand system
- Price-taker in transportation system

$$\max_{D,S,T} \quad \sum_{l \in L} \overset{\pi_l}{\cancel{\theta_l(D_l)}} D_l - \sum_{l \in L} \Psi_l(S_l) - \sum_{i,j} \overset{p_{ij}}{\cancel{c_{ij}(T_{ij})}} T_{ij}$$

$$\text{s.t.} \quad S_l - D_l + \sum_{i,l} T_{il} - \sum_{l,j} T_{lj} = 0 \;\forall l \in L$$

$$D, S, T \in F$$

$$p_{ij} = \quad c_{ij}(T_{ij})$$

$$\pi_l = \quad \theta_l(D_l)$$

empinfo: vi tcDef tc
pricedef price

Classic SPE Model (NLP + VI agents)

- One producer controlling all regions
- Price-taker in demand system
- Price-taker in transportation system

$$\begin{aligned}
\max_{D,S,T} \quad & \sum_{l \in L} \overset{\pi_l}{\cancel{\theta_l(D_l)}} D_l - \sum_{l \in L} \Psi_l(S_l) - \sum_{i,j} \overset{p_{ij}}{\cancel{c_{ij}(T_{ij})}} T_{ij} \\
\text{s.t.} \quad & S_l - D_l + \sum_{i,l} T_{il} - \sum_{l,j} T_{lj} = 0 \ \forall l \in L \\
& D, S, T \in F \\
p_{ij} = \quad & c_{ij}(T_{ij}) \\
\pi_l = \quad & \theta_l(D_l)
\end{aligned}$$

$$\text{EMP} = \text{MOPEC} \implies \text{MCP}$$

Cournot-Nash equilibrium (multiple agents)

Assumes that each agent:

- Treats other agent decisions as fixed
- Is a price-taker in transport and demand

EMP info file

```
equilibrium
max obj('one') vars('one') eqns('one')
max obj('two') vars('two') eqns('two')
max obj('three') vars('three') eqns('three')
vi tcDef tc pricedef price
```

$\text{EMP} = \text{MOPEC} \implies \text{MCP}$

Bilevel Program (Stackelberg)

- Assumes one leader firm, the rest follow
- Leader firm optimizes subject to expected follower behavior
- Follower firms act in a Nash manner
- All firms are price-takers in transport and demand

EMP info file

```
bilevel obj('one') vars('one') eqns('one')
max obj('two') vars('two') eqns('two')
max obj('three') vars('three') eqns('three')
vi tcDef tc pricedef price
```

EMP = bilevel $\Longrightarrow$ MPEC $\Longrightarrow$ (via NLPEC) NLP(μ)

Design: Stochastic competing agent models (with Wets)

- Competing agents (consumers, or generators in energy market)
- Each agent minimizes objective independently (cost)
- Market prices are function of all agents activities
- Additional twist: model must "hedge" against uncertainty
- Facilitated by allowing contracts bought now, for goods delivered later
- Conceptually allows to transfer goods from one period to another (provides wealth retention or pricing of ancilliary services in energy market)
- Can investigate new instruments to mitigate risk, or move to system optimal solutions from equilibrium (or market) solutions

Example as MOPEC: agents solve a Stochastic Program

Each agent minimizes:

$$u_a = \sum_s \pi_s \left(\kappa - f(q_{a,s,*})\right)^2$$

Budget time 0: $\sum_i p_{0,i} q_{a,0,i} + \sum_j v_j y_{a,j} \leq \sum_i p_{0,i} e_{a,0,i}$

Budget time 1: $\sum_i p_{s,i} q_{a,s,i} \leq \sum_i p_{s,i} \sum_j D_{s,i,j} y_{a,j} + \sum_i p_{s,i} e_{a,s,i}$

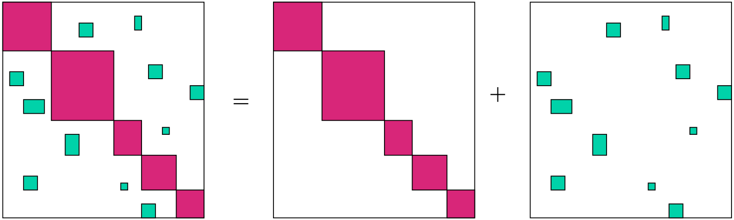
Additional constraints (complementarity) outside of control of agents:

$$\text{(contract) } 0 \leq -\sum_a y_{a,j} \perp v_j \geq 0$$

$$\text{(walras) } 0 \leq -\sum_a d_{a,s,i} \perp p_{s,i} \geq 0$$

Model and solve

- Can model financial instruments such as “financial transmission rights”, “spot markets”, “reactive power markets”
- Reduce effects of uncertainty, not simply quantify
- Use structure in preconditioners
  - Use nonsmooth Newton methods to formulate complementarity problem
  - Solve each “Newton” system using GMRES
  - Precondition using “individual optimization” with fixed externalities



What is EMP?

Annotates existing equations/variables/models for modeler to provide/define additional structure

- equilibrium
- vi (agents can solve min/max/vi)
- bilevel (reformulate as MPEC, or as SOCP)
- disjunction (or other constraint logic primitives)
- randvar
- dualvar (use multipliers from one agent as variables for another)
- extended nonlinear programs (library of plq functions)

Currently available within GAMS

Conclusions

- Modern optimization within applications requires multiple model formats, computational tools and sophisticated solvers
- EMP model type is clear and extensible, additional structure available to solver
- Extended Mathematical Programming available within the GAMS modeling system
- Able to pass additional (structure) information to solvers
- Embedded optimization models automatically reformulated for appropriate solution engine
- Exploit structure in solvers
- Extend application usage further