EMP support in JuMP

Olivier Huber & Michael C. Ferris

WID — University of Wisconsin–Madison

June 29, 2018

Outline

1. What is Extended Mathematical Programming (EMP)
2. EMP solver
3. Modeling of EMP problems in JuMP

What is EMP?

EMP (Extended Mathematical Programming) is a framework to model optimization problem that do not fit into "classical" optimization problem via simple annotations

EMP in GAMS

- Multi-players problems (MOPEC): (Generalized) Nash Equilibrium
- Complementarity Problems and Variational Inequalities
- Bilevel Problems (MPEC)
- Multistage stochastic programming, Disjunctive Programs

Motivation for EMP support

Motivations & Goals for designing a tool

- Ease of use for non-experts: the modeler uses simple annotations
- The model is augmented/transformed programmatically
- Provide different schemes for solving the problem
- Pass/Exploit the structure of the problem (feasible set)
- No errors/typos during the reformulation
- Allow for easy comparisons between schemes

JAMSDWriter.jl: MPB Solver

- Loosely based on `AmplNLWriter.jl`
- Standard LQP/NLP solver
- Translate the EMP data structure

EMP.jl: modeling package

- Equilibrium support
- Optimal Value Function (OVF) support

VI background

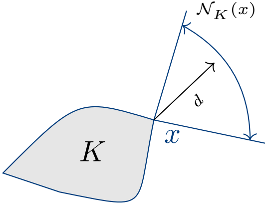
VI of the first kind

$F\colon \mathbb{R}^n \to \mathbb{R}^n$, Z convex subset of $\mathbb{R}^n$. Find $z \in Z$ such that

$$\langle F(z), z - y \rangle \geq 0 \qquad \forall y \in Z$$

If $F = \nabla f$ or $F = (\nabla_x L, -\nabla_\lambda L)$, VI stems from an optimization problem

Normal cone operator: $N_K(x) := \{d \mid \langle y - x, d \rangle \leq 0 \quad \forall y \in K\}$.



$$d \in \mathcal{N}_K(x) \qquad VI(F, K) \equiv 0 \in F(z) + N_K(z)$$

Complementary Problem $0 \leq x \perp F(x) \geq 0 \quad \equiv \quad 0 \in F(x) + N_{\mathbb{R}^n_+}(x)$

where $x \perp F(x) \quad \equiv \quad x^T F(x) = 0$

(Generalized) Nash Equilibrium Problem [(G)NEP]

For each agent i: x_i is the decision variable, x_{-i} is the other variables

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i(x_i, x_{-i}) \\ \text{subject to} \quad & g_i(x_i, x_{-i}) \in K_i \end{aligned}$$

K_i convex cone ($\mathbb{R}^n_+$, $\mathbb{R}^n_-$, $\{0\}$, SOC, ...)

example of coupling constraints

- shared resources among agents: $\sum_{i=1}^N x_i \leq u$
- strategic decisions

For each agent i: x_i is the decision variable, x_{-i} is the other variables

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i(x_i, x_{-i}) \\ \text{subject to} \quad & g_i(x_i, x_{-i}) \in K_i \end{aligned}$$

K_i convex cone ($\mathbb{R}^n_+$, $\mathbb{R}^n_-$, $\{0\}$, SOC, $\ldots$)

example of coupling constraints

- shared resources among agents: $\sum_{i=1}^N x_i \leq u$
- strategic decisions

- No hierarchy between players, (otherwise MPECs or EPECs)
- An example of a VI agent: market clearing conditions

$$0 \leq \text{supply} - \text{demand} \quad \perp \quad \text{price} \geq 0$$

MOPEC is solved by simultaneously solving all the KKT systems

First Order Optimality Conditions under CQ

$$\begin{array}{ll} \underset{x_i}{\text{minimize}} & f_i(x_i, x_{-i}) \\ \text{subject to} & g_i(x_i, x_{-i}) \in K_i \end{array} \Longrightarrow \begin{array}{c} \nabla_{x_i} f_i(x_i, x_{-i}) + \langle \nabla_{x_i} g_i(x_i, x_{-i}), \lambda \rangle = 0 \\ K_i^* \ni \lambda_i \perp g_i(x_i, x_{-i}) \in K_i \end{array}$$

Solution methodology for GNEP

MOPEC is solved by simultaneously solving all the KKT systems

First Order Optimality Conditions under CQ

$$\begin{array}{lll} \underset{x_i}{\text{minimize}} & f_i(x_i, x_{-i}) \\ \text{subject to} & g_i(x_i, x_{-i}) \in K_i \end{array} \Longrightarrow \begin{array}{c} \nabla_{x_i} f_i(x_i, x_{-i}) + \langle \nabla_{x_i} g_i(x_i, x_{-i}), \lambda \rangle = 0 \\ K_i^* \ni \lambda_i \perp g_i(x_i, x_{-i}) \in K_i \end{array}$$

- Resulting MCP: $X \ni z \perp F(z) \in X^*$
- X is a simply bounded set
- F is non-monotone in general
- PATH or PATHAVI can solve many NEP instances (GNEP more challenging)

Problem type

Objective function		or	Constraint	
$\underset{x}{\text{minimize}}$	$f(x) + \boldsymbol{\rho(F(x))}$		$\underset{x}{\text{minimize}}$	$f(x)$
			subject to	$g(x) \leq 0$
subject to	$g(x) \leq 0$			$\boldsymbol{\rho(F(x)) \leq \alpha}$

$$\rho(v) := \sup_{u \in U} \langle G(v), u \rangle - k(u)$$

$G(v) := Bv + b$, k is convex, U polyhedral

Aravkin et al. [2013]

Quadratic Support (QS) examples

Fitting Examples: apply ρ to $F(x)$ — Aravkin et al. [2013]

	ℓ_1	ℓ_2	Huber	Hinge Loss
U	$[-1,1]^m$	0	$[-\lambda,\lambda]^m$	$[0,1]^m$
Q	0	$\frac{1}{2}I$	$\frac{1}{2}I$	0
$G(y)$	y	y	y	$y-\varepsilon$

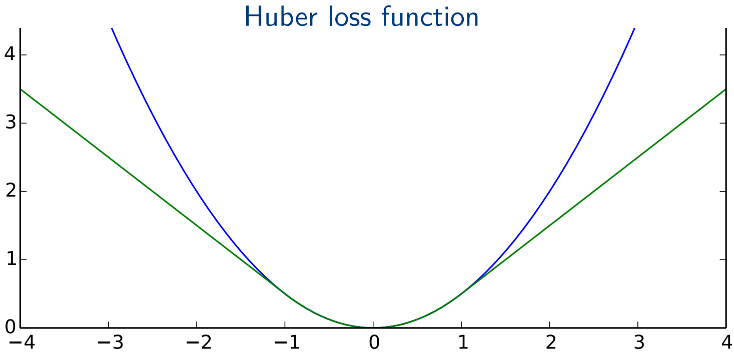
and more: soft Hinge, Vapnik, Hubnik, elastic net

Quadratic Support (QS) examples

Fitting Examples: apply ρ to $F(x)$ — Aravkin et al. [2013]

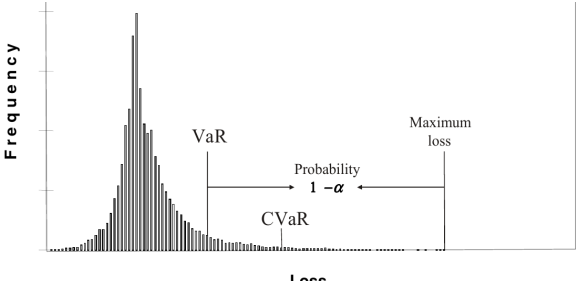
	ℓ_1	ℓ_2	Huber	Hinge Loss
U	$[-1,1]^m$	0	$[-\lambda,\lambda]^m$	$[0,1]^m$
Q	0	$\frac{1}{2}I$	$\frac{1}{2}I$	0
$G(y)$	y	y	y	$y-\varepsilon$

and more: soft Hinge, Vapnik, Hubnik, elastic net



Coherent Risk Measure Examples — Shapiro et al. [2009]

- Expected value: $\mathbb{E}[Z] = \sigma_{\{p_k\}}(Z)$
- Upper CVaR (AV@R, expected shortfall): $\mathrm{CVaR}_{1-\alpha}(z) = \sigma_{D(\alpha)}(Z)$
  with $\mathcal{D}(\alpha) := \left\{ y \in \prod_k [0, \alpha^{-1}] \mid \mathbb{E}[y] = 1 \right\}$
- Algebra of support function: $\lambda\mathbb{E}[z] + (1-\lambda)\mathrm{CVaR}_{1-\alpha}(z)$ is the support function of $\lambda\{p_k\} + (1-\lambda)\mathcal{D}(\alpha)$



Saddle-point reformulation

Game with payoff function: $\bar{L}(x,y) = f(x) + y^T G(F(x)) - k(y)$

$$\min_{x \in X} \bar{L}(x,y) \qquad \max_{y \in \bar{Y}} \bar{L}(x,y)$$

Dual problem (QP)

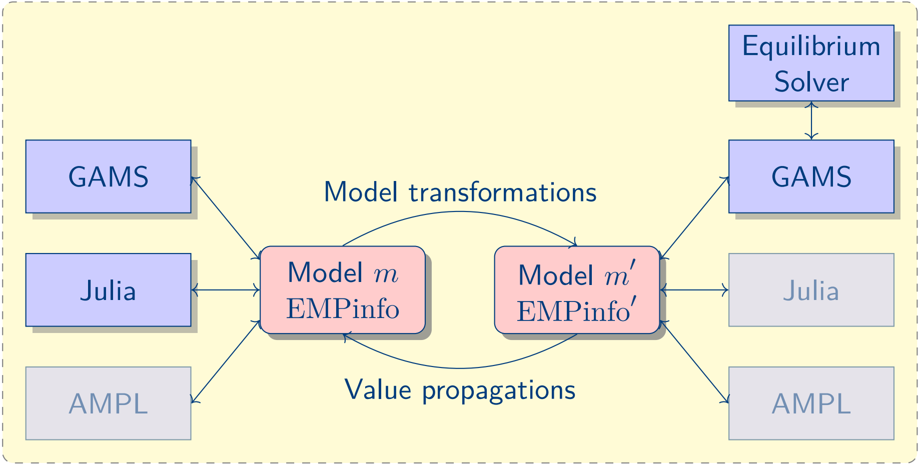
Find a minimization problem with the same objective function value

Conjugate reformulation

- ρ (Fenchel) conjugate of $k + \delta_U$: $\rho(u) = \inf_{u=u_1+u_2} k^*(u_1) + \sigma_U(u_2)$
- $k(u) = u^T Q u = \|L^T u\|_2$ (Q psd)

$$\rho(F(x)) = \inf_s \ \tfrac{1}{2}\|s\|_2^2 + \sigma_U(F(x) - Ls)$$

EMP solver



The model representation inside the EMP solver is independent of any model language

EMP info storage: mathematical programm

Optimization Problem definition

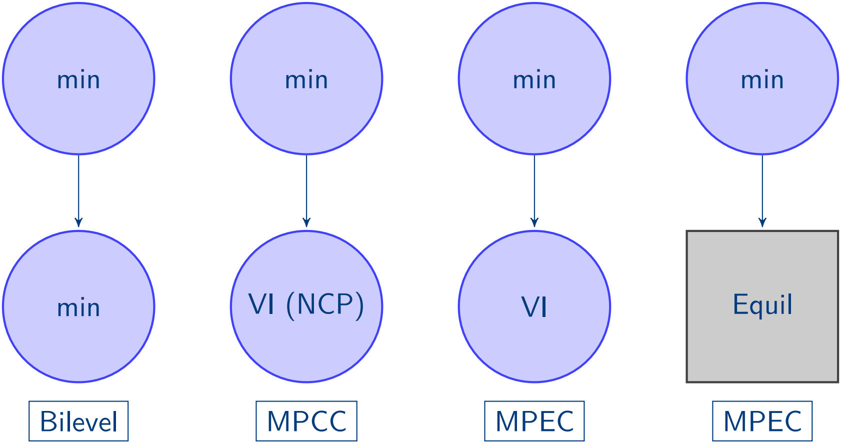
- Mathematical Programm (MP) is the basic building block
- The optimization problem is described by a graph
- Graph with 2 types of nodes: MP or Equilibrium (collection of MPs)
- Examples: bilevel, MPEC, EPEC, MOPEC

EMP info storage: mathematical programm

Optimization Problem definition

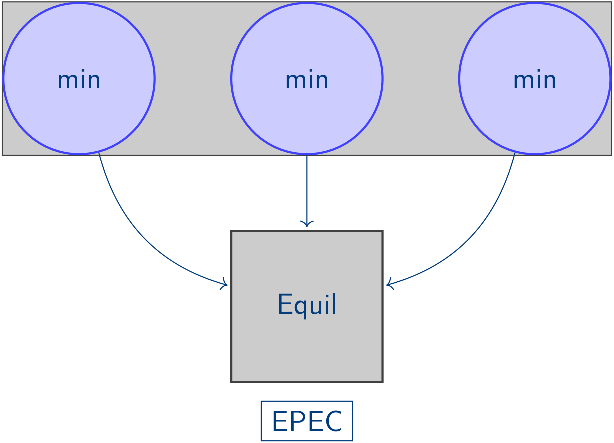
- Mathematical Programm (MP) is the basic building block
- The optimization problem is described by a graph
- Graph with 2 types of nodes: MP or Equilibrium (collection of MPs)
- Examples: bilevel, MPEC, EPEC, MOPEC

- Easy extraction of subproblems (splitting methods (diag, Gauss-Seidel))
- Fix variable values, simply expressions.



min
min
min
min
min
VI (NCP)
VI
Equil
Bilevel
MPCC
MPEC
MPEC

Example II



Model object

- JuMP container for variables and equations
- list of Mathematic Programms
- list of Equilibriums
- list of equations (for global indices)
- list of special variables

Mathematical Programm (MP)

- variables and equations indices
- complementary information $(0 \leq F(x) \perp x \geq 0)$
- objective variable / objective equation / sense
- MP in MP
- equilibriums in MP

EMP.jl API example I


```
mopec = EMP.Model()

ag = MathPrgm(mopec)
mkt = MathPrgm(mopec)

EquilibriumProblem(mopec, [ag, mkt])

n = 3

@variableMP(mkt, y >= 0)
@variableMP(ag, x[1:n] >= 0)
@variableMP(mkt, p[1:n] >= 0)

JuMP.fix(p[2], 1.)
JuMP.setvalue(x[1:n], ones(n))

vipair(mkt, [b[i] + ATmat[i]*y - x[i] for i=1:n], p)
vipair(mkt, sum(-ATmat[i]*p[i] for i=1:n), y)

@constraintMP(ag, sum(p[i]*x[i] for i=1:n) <= sum(p[i]*b[i] for i=1:n) )
@NLobjectiveMP(ag, :Max, sum(s[i] * log(x[i]) for i=1:n))

solveEMP(mopec)
```

Monolitic approach

- A unique container for variables and equations
- the variables and equations are assigned to a MP

⊕ Easy to defined shared variables or equations

⊖ challenges in modeling: what is a natural way to create the partition?

MP-based approach

- Each MP has its own JuMP Model
- The collection of MP in the graph defines the optimition problem

⊕ ownership is trivial

⊖ shared variables/equations are difficult to define

⊖ consistency of the whole model needs to be checked

⊖ Global indices for variables and equations

Future work

EMP solver

- Add automatic/symbolic differentiation support
- Add ability to use Julia to solve the optimization problem https://github.com/mlubin/cmpb

EMP.jl

- User-defined OVF functions
- Define UI/API for hierarchical optimization problems

A. Y. Aravkin, J. V. Burke, and G. Pillonetto. Sparse/robust estimation and kalman smoothing with nonsmooth log-concave densities: modeling, computation, and theory. *Journal of Machine Learning Research*, 14(1): 2689–2728, 2013.

R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35(2): 183–238, 1993.

R. T. Rockafellar. Extended nonlinear programming. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications 2*, pages 381–399. Springer, 2000.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming. Modeling and Theory*. SIAM Publishing, Philadelphia, Pennsylvania, 2009.

AFW '16 Electricity Generation example

- Electricity generation example with $\mathrm{CVaR}\left(\cdot\right)$ in the objective function
- Equilibrium problem with 3 agents and 10 scenarios
- VI (MCP) and conjugate better than dual QP (primal)