Solving equilibrium problems using extended mathematical programming and SELKIE

Youngdae Kim and Michael Ferris

Wisconsin Institute for Discovery
University of Wisconsin-Madison

April 30, 2018

This talk is about

- EMP framework: specifying and solving equilibrium and variational problems in modeling languages such as AMPL, GAMS, Julia, etc.
  - (Generalized) Nash equilibrium problems
  - Multiple optimization problems with equilibrium constraints
  - (Quasi-) Variational inequalities
  - Enables us to expose and exploit some special structures
    - e.g., shared constraints and shared variables
- SELKIE: a decomposition method based on a grouping of agents (a block diagonalization method, including Dantzig-Wolfe)
- Our interfaces and algorithms have been implemented and are available within GAMS/EMP.

Equilibrium = the first-order optimality conditions (KKTs)

An equilibrium of a single optimization (a single agent) under CQs

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x), \\ \text{subject to} \quad & g(x) \le 0, \ (\Rightarrow) \\ & h(x) = 0, \end{aligned} \qquad \begin{aligned} \nabla f(x) - \nabla g(x)^T \lambda - \nabla h(x)^T \mu = 0, \\ 0 \ge g(x) \perp \lambda \le 0, \\ 0 = h(x) \perp \mu, \end{aligned}$$

- Mixed complementarity problem $\text{MCP}([l, u], F) : l \le z \le u \ \perp \ F(z)$

Geometric first-order optimality conditions for a closed convex set K

$$\underset{x \in K}{\text{minimize}} \quad f(x), \ (\Rightarrow) \quad \langle \nabla f(x), y - x \rangle \ge 0, \ \forall y \in K.$$

- Variational inequality $\text{VI}(K, F) : \langle F(x), y - x \rangle \ge 0, \forall y \in K$

Generalizing to N agents: NEP

Nash equilibrium problem: $x = [x_i]_{i=1}^N$

$$\begin{array}{lll} \underset{x_i}{\text{minimize}} & f_i(x_i, x_{-i}), & \\ \text{subject to} & g_i(x_i) \le 0, \ (\Rightarrow) & \\ & h_i(x_i) = 0, & \end{array} \qquad \begin{array}{r} \nabla_{x_i} f_i(x_i, x_{-i}) - \nabla g_i(x_i)\lambda_i - \nabla h_i(x_i)\mu_i = 0, \\ 0 \ge g_i(x_i) \perp \lambda_i \le 0, \\ 0 = h_i(x_i) \perp \mu_i. \end{array}$$

- $x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)^T$.
- Equilibrium: satisfy the KKT conditions of all agents simultaneously.
- Interactions occur only in objective functions.
- Example of an interaction: $f_i(x_i, x_{-i}) = c_i(x_i) - x_i p\left(\sum_{j=1}^N x_j\right)$

NEP + non-disjoint feasible regions: GNEP

Generalized Nash equilibrium problem: $x = [x_i]_{i=1}^N$

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i(x_i, x_{-i}), \\ \text{subject to} \quad & g_i(x_i, x_{-i}) \le 0, \ (\Rightarrow) \\ & h_i(x_i, x_{-i}) = 0, \end{aligned} \qquad \begin{aligned} \nabla_{x_i} f_i(x) - \nabla_{x_i} g_i(x)\lambda_i - \nabla_{x_i} h_i(x)\mu_i = 0, \\ 0 \ge g_i(x) \perp \lambda_i \le 0, \\ 0 = h_i(x) \perp \mu_i. \end{aligned}$$

- Interactions occur in both objective functions and constraints.
- Non-disjoint feasible region:
  $K_i(x_{-i}) = \{x_i \in \mathbb{R}^{n_i} \mid g_i(x_i, x_{-i}) \le 0, h_i(x_i, x_{-i}) = 0\}$.
  - $K_i : \mathbb{R}^{n-n_i} \rightrightarrows \mathbb{R}^{n_i}$ a set-valued mapping
  - e.g., shared resources among agents: $\sum_{i=1}^N x_i \le b$, or strategic interactions

(G)NEP + VI agent: MOPEC

Multiple optimization problems with equilibrium constraints: $x = [x_i]_{i=1}^N, \pi$

$$\begin{array}{lll} \underset{x_i}{\text{minimize}} & f_i(x_i, x_{-i}, \pi), & \nabla_{x_i} f_i(x,\pi) - \nabla_{x_i} g_i(x,\pi)\lambda_i - \nabla_{x_i} h_i(x,\pi)\mu_i = 0, \\ \text{subject to} & g_i(x_i, x_{-i}, \pi) \le 0, & 0 \ge g_i(x,\pi) \perp \lambda_i \le 0, \\ & h_i(x_i, x_{-i}, \pi) = 0, & 0 = h_i(x,\pi) \perp \mu_i, \\ \hline \pi & \in \mathrm{SOL}(K, F), & \pi \in K(x), \langle F(\pi, x), y - \pi \rangle \ge 0, \ \forall y \in K(x). \end{array}$$

- No hierarchy between agents, c.f., MPECs and EPECs
- An example of a VI agent: market clearing conditions

$$0 \le \text{supply} - \text{demand} \quad \perp \quad \text{price} \ge 0$$

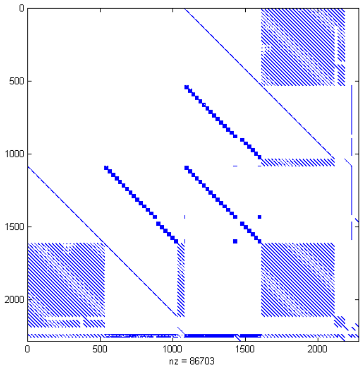
Connections with mixed complementarity problems (MCPs)

- A MOPEC can be computed using an MCP$((x, \lambda, \mu, \pi), F)$:

$$
\begin{aligned}
F_i(x,\pi,\lambda,\mu) \quad &:= \quad \begin{bmatrix} \nabla_{x_i} f_i(x,\pi) - \nabla_{x_i} g_i(x,\pi)\lambda_i - \nabla_{x_i} h_i(x,\pi)\mu_i \\ g_i(x,\pi) \\ h_i(x,\pi) \end{bmatrix}, \\
F_i(x,\pi,\lambda,\mu) \quad &\perp \quad \begin{bmatrix} x_i \\ \lambda_i \leq 0 \\ \mu_i \end{bmatrix}, \quad \text{for} \quad i = 1,\ldots,N, \\
F_\pi(\pi,x,\lambda,\mu) \quad &:= \quad \begin{bmatrix} \tilde{F}(\pi,x) - \nabla_\pi g_\pi(\pi,x)\lambda_\pi - \nabla_\pi h_\pi(\pi,x)\mu_\pi \\ g_\pi(\pi,x) \\ h_\pi(\pi,x) \end{bmatrix}, \\
F_\pi(\pi,x,\lambda,\mu) \quad &\perp \quad \begin{bmatrix} \pi \\ \lambda_\pi \leq 0 \\ \mu_\pi \end{bmatrix}.
\end{aligned}
$$

World Bank Project (Uruguay Round)

- 24 regions, 22 commodities
  - Nonlinear complementarity problem
  - Size: 2200 x 2200
- Short term gains $53 billion p.a.
  - Much smaller than previous literature
- Long term gains $188 billion p.a.
  - Number of less developed countries loose in short term
- Unpopular conclusions - forced concessions by World Bank
- Region/commodity structure not apparent to solver



PATHVI on Nash Equilibria

Name	Elapsed time (secs)		
	PATHVI	PATH	PATHVI/ UMFPACK
vimod1	0.372	4.129	0.437
vimod2	1.098	24.134	0.645
vimod3	3.208	60.553	1.639
vimod4	127.194	66.427	18.319
vimod5	327.970	325.558	40.285
vimod6	2341.193	1841.642	109.960

Specifying (G)NEPs and MOPECs in modeling languages

- Existing method

1. Compute an MCP function F using the KKT conditions.

$$\begin{array}{lll} \underset{x_i}{\text{minimize}} & f_i(x_i, x_{-i}), & \Longrightarrow \quad F_i(x, \lambda_i) = \begin{bmatrix} \nabla_{x_i} f_i - \nabla_{x_i} g_i \lambda_i \\ g_i \end{bmatrix}, \\ \text{subject to} & g_i(x_i, x_{-i}) \leq 0, & \\ & \text{for } i = 1, \ldots, N, & \quad \text{for } i = 1, \ldots, N. \end{array}$$

2. Specify the complementarity relationship.

$$\begin{array}{ll} F \ \texttt{complements} \ (x, \lambda) & \text{in AMPL}, \\ F \ \perp \ (x, \lambda) & \text{in GAMS}. \end{array}$$

3. Solve the resultant MCP$((x, \lambda), F)$ using the PATH solver.

- Cons
  - ★ Prone to errors as we require users to compute derivatives by hand
  - ★ Not easy to modify the problem: a lot of derivative recomputations
  - ★ Agent information is lost in the MCP function F.

The EMP framework

- Automates all the previous steps: no need to derive MCP by hand.
- Annotate equations and variables in an `empinfo` file.
- The framework automatically transforms the problem into another computationally more tractable form.

$$
\begin{aligned}
\underset{x_i}{\text{minimize}} \quad & f_i(x_i, x_{-i}, \pi), \\
\text{subject to} \quad & g_i(x_i, x_{-i}, \pi) \le 0, \\
& h_i(x_i, x_{-i}, \pi) = 0, \\
& \text{for } i = 1, \ldots, N, \\
\hline
\pi \quad & \in \mathrm{SOL}(K, F).
\end{aligned}
$$

```
equilibrium
min f('1') x('1') g('1') h('1')
...
min f('N') x('N') g('N') h('N')
vi F pi K
```

An example of using the EMP framework

- An oligopolistic market equilibrium problem formulated as a NEP:

$$q_i^* \in \underset{q_i \geq 0}{\arg\max} \quad q_i p \left(\sum_{j=1, j \neq i}^{5} q_j^* + q_i \right) - c_i(q_i), \text{ for } i = 1, \ldots, 5.$$

```
variables obj(i); positive variables q(i);
equations defobj(i);
defobj(i)..  obj(i) =E= ...;
model m / defobj /;

file info / '%emp.info%' /;
put info 'equilibrium' /;
loop(i, put 'max', obj(i), q(i), defobj(i) /;);
putclose;
solve m using emp;
```

Special features I: supporting shared constraints

- Shared constraints: agents have shared resources.
- g is a *shared constraint*:

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i(x_i, x_{-i}), \\ \text{subject to} \quad & g(x_i, x_{-i}) \leq 0. \end{aligned}$$

- Examples:
  - Network capacity: $\sum_{i=1}^{N} x_i \leq b$
    Agents send packets through the same network channel.
  - Total pollutants: $\sum_{i=1}^{N} a_i x_i \leq c$
    Agents throw pollutants in the river. The maximum pollutants that can be thrown are set by a policy.

Different types of solutions for shared constraints

- A GNEP equilibrium: replicate g and assign a separate multiplier

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i(x_i, x_{-i}), \\ \text{subject to} \quad & g(x_i, x_{-i}) \leq 0, \quad (\perp \quad \mu_i \leq 0). \end{aligned}$$

- A variational equilibrium: force use of a single g and a single μ

$$\frac{\underset{x_i}{\text{minimize}} \quad f_i(x_i, x_{-i}) - \mu^T g,}{0 \geq g(x) \quad \perp \quad \mu \leq 0.}$$

- Syntactic enhancement

```
equilibrium
visol g
min f('1') x('1') g
...
min f('N') x('N') g
```

Special features II: supporting shared variables

- Shared variables: agents have shared states.
- y is a *shared variable*:

$$\underset{y, x_i}{\text{minimize}} \quad f_i(y, x_i, x_{-i}),$$
$$\text{subject to} \quad h(y, x_i, x_{-i}) = 0.$$

  - For each x, the value of y is implicitly determined by h.
- Syntactic enhancement

```
equilibrium
implicit y h
min f('1') x('1') y
...
min f('N') x('N') y
```

MCP formulation strategies for shared variables

- Replication

$$F_i(x,y,\mu) = \begin{bmatrix} \nabla_{x_i} f_i - \nabla_{x_i} h \mu_i \\ \nabla_{y_i} f_i - \nabla_{y_i} h \mu_i \\ h \end{bmatrix} \quad \perp \quad \begin{bmatrix} x_i \\ y_i \\ \mu_i \end{bmatrix}$$

- Switching

$$\frac{F_i(x,y,\mu) = \begin{bmatrix} \nabla_{x_i} f_i - \nabla_{x_i} h \mu_i \\ \nabla_{y} f_i - \nabla_{y} h \mu_i \end{bmatrix} \quad \perp \quad \begin{bmatrix} x_i \\ \mu_i \end{bmatrix}}{F_h(x,y,\mu) = \begin{bmatrix} h \end{bmatrix} \qquad\qquad \perp \quad \begin{bmatrix} y \end{bmatrix}}$$

- Substitution: eliminate $\mu_i \leftarrow [\nabla_y h]^{-1} \nabla_y f_i$

Strategy	Size of the MCP
replication	$(n + 2mN)$
switching	$(n + mN + m)$
substitution (implicit)	$(n + nm + m)$
substitution (explicit)	$(n + m)$

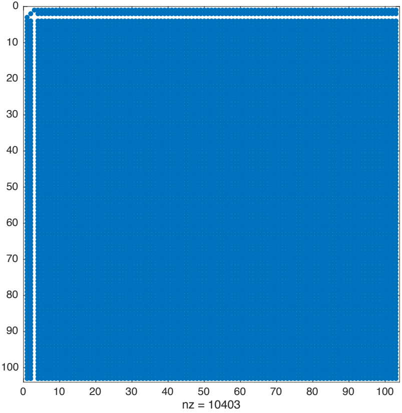
Experimental results: improving sparsity

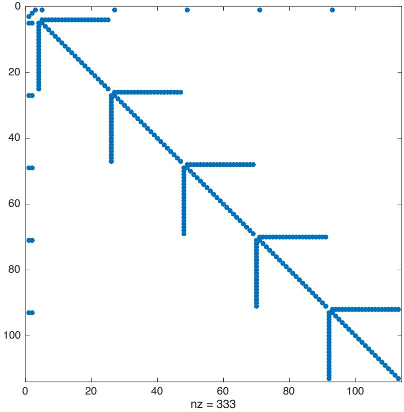
- Replace $p\left(\sum_{i=1}^{N} x_i\right)$ with $p(y)$ in oligopolistic market problem.
  - 1 ISO agent and 5 energy-producing agents
  - Each energy-producing agent has a fixed number of plants: $n/5$.

n	Original		Switching	
	Size	Density (%)	Size	Density (%)
2,500	2,502	99.92	2,508	0.20
5,000	5,002	99.96	5,008	0.10
10,000	10,002	99.98	10,008	0.05
25,000	-	-	25,008	0.02
50,000	-	-	50,008	0.01

n	Original		Switching	
	(Major,Minor)	Time (secs)	(Major,Minor)	Time (secs)
2,500	(2,2639)	57.78	(1,2630)	1.30
5,000	(2,5368)	420.92	(1,5353)	5.83
10,000	-	-	(1,10517)	22.01
25,000	-	-	(1,26408)	148.08
50,000	-	-	(1,52946)	651.14

The sparsity patterns





Jacobian nonzero pattern $n = 100$, $N_a = 20$

Experimental results: general equilibrium conditions

- Agents trade goods to maximize their welfare.
  - z represents an economic state, and t is a policy.
  - h represents partial/general equilibrium conditions.
  - The value of z is implicitly determined by the value of t via h.

$$
\begin{aligned}
\underset{z, t_i \in T_i}{\text{maximize}} \quad & f_i(z, t), \\
\text{subject to} \quad & h(z, t_i, t_{-i}) = 0, \\
& \text{for } i = 1, \ldots, 23.
\end{aligned}
$$

Experimental results: general equilibrium conditions (cont'd)

# Agents	Replication		Switching		Substitution	
	Size	Density (%)	Size	Density (%)	Size	Density (%)
5	570	1.66	350	3.34	1,230	0.77
10	2,290	0.72	1,300	1.70	10,210	0.14
15	5,160	0.50	2,850	1.28	35,190	0.06
20	9,180	0.40	5,000	1.10	84,420	0.03
23	12,144	0.37	6,578	1.03	129,030	0.02

# Agents	Replication		Switching		Substitution	
	(Major,Minor)	Time (secs)	(Major,Minor)	Time (secs)	(Major,Minor)	Time (secs)
5	(18,164)	0.33	(18,173)	0.22	(11,29)	0.38
10	(17,279)	1.52	(17,301)	1.48	(15,436)	8.14
15	(8,22)	1.81	(8,22)	1.68	(129,19806)	814.73
20	(9,28)	4.90	(9,28)	4.73	(13,461)	104.00
23	(9,41)	10.07	(9,41)	8.02	(20,1451)	368.99

Experimental results: modeling mixed behavior

- Revisiting the oligopolistic market equilibrium problem:

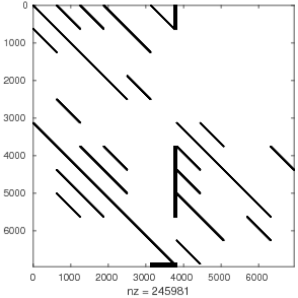
$$\underset{q_i \geq 0}{\text{maximize}} \quad q_i p\left(\sum_{j=1, j\neq i}^{5} q_j + q_i\right) - c_i(q_i), \text{ for } i = 1, \ldots, 5.$$

- Introduce a shared variable $y = p(q)$.
  - If an agent declares y as its decision variable, it is a price-maker.
  - Otherwise, it is a price-taker.

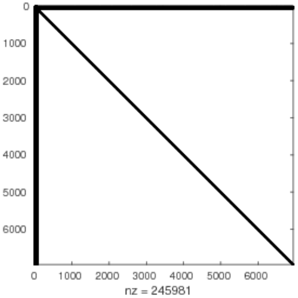
Profit	Competitive	Oligo1	Oligo12	Oligo123	Oligo1234	Oligo12345
Firm 1	123.834	125.513	145.591	167.015	185.958	199.934
Firm 2	195.314	216.446	219.632	243.593	264.469	279.716
Firm 3	257.807	278.984	306.174	309.986	331.189	346.590
Firm 4	302.863	322.512	347.477	373.457	376.697	391.279
Firm 5	327.591	344.819	366.543	388.972	408.308	410.357
Total profit	1207.410	1288.273	1385.417	1483.023	1566.621	1627.875
Social welfare	39063.824	39050.191	39034.577	39022.469	39016.373	39015.125

SELKIE: a motivating example

- A dynamic programming problem in economics
  - 626 agents. Interactions occur only between the first and the others.
  - Decomposable once the first agent's variable value is fixed.



(a) Original Jacobian of MCP

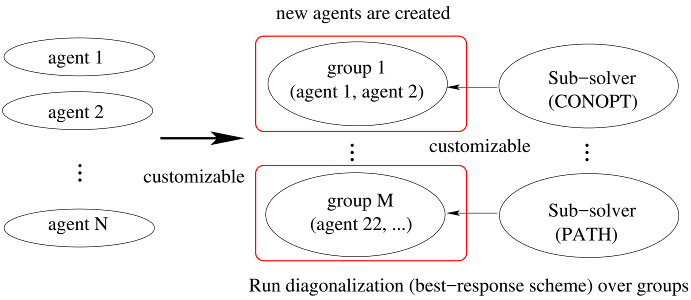


(b) Permuted Jacobian of MCP

- PATH fails to solve the problem, but diagonalization works.

Introduction to SELKIE

- SELKIE is a general-purpose solver for equilibrium problems.
  - A decomposition method based on a grouping of agents
    - ★ Generalize the diagonal dominance theory for convergence
  - Flexible and adaptable in creating and solving sub-problems
    - ★ Works under a single problem specification



An example of using SELKIE

- An oligopolistic market equilibrium problem:

$$\underset{q_i \geq 0}{\text{maximize}} \quad q_i p\left(\sum_{j=1, j\neq i}^{5} q_j + q_i\right) - c_i(q_i), \text{ for } i = 1, \ldots, 5.$$

Group	Iterations			
	Jacobi	GS	GSW	GS(RS)
{{1},{2},{3},{4},{5}}	155	45	28	50
{{1,2},{3,4},{5}}	57	21	22	30
{{1..3},{4,5}}	28	14	14	18
{{1..4},{5}}	22	12	12	16
{{1..5}}	1			

  - GS: Gauss-Seidel
  - GSW: Gauss-Southwell
  - GS(RS): Gauss-Seidel with random sweep

- An automatic detection of independent groups is supported.

Exploiting decomposable structure: revisiting the DP

- Specify the model and the `empinfo` file.

```
equilibrium
min objl A lsqrdef
max obj('1') C('1') ...
...
max obj('625') C('625') ...
```

- Specify groups of agents and their solution methods.

```
agent_group={{1},{2..626}:jacobi}
parallel_jacobi=yes
```

- SELKIE takes about 10 mins to solve it on a 2-core machine.
  - In parallel: 10 mins, in sequential: 20 mins
- PATH fails to solve the problem: `agent_group={{1..626}}`

Conclusion: who knows (and controls) what?

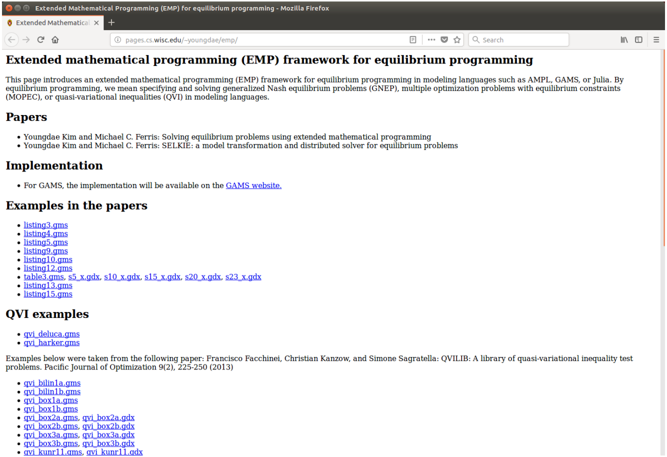
$$\min_{x_i} f_i(x_i, x_{-i}, y(x_i, x_{-i}), \pi) \text{ s.t. } g_i(x_i, x_{-i}, y, \pi) \leq 0, \forall i, \theta(x, y, \pi) = 0$$

$$\pi \text{ solves VI}(h(x, \cdot), C)$$

- Presents an EMP framework to specify and solve (Q)VIs, (G)NEPs, and MOPECs in modeling languages
- Enhance modeling through shared constraints and shared variables
- Can use EMP to write all these problems, and convert to MCP form
- Provides a decomposition-based highly customizable solver SELKIE.
- Enables modelers to convey simple structures to algorithms and allows algorithms to exploit this
- Can evaluate effects of regulations and their implementation in a competitive environment

Examples are available

- See http://pages.cs.wisc.edu/~youngdae/emp.



Extended mathematical programming (EMP) framework for equilibrium programming

This page introduces an extended mathematical programming (EMP) framework for equilibrium programming in modeling languages such as AMPL, GAMS, or Julia. By equilibrium programming, we mean specifying and solving generalized Nash equilibrium problems (GNEP), multiple optimization problems with equilibrium constraints (MOPEC), or quasi-variational inequalities (QVI) in modeling languages.

Papers

- Youngdae Kim and Michael C. Ferris: Solving equilibrium problems using extended mathematical programming
- Youngdae Kim and Michael C. Ferris: SELKIE: a model transformation and distributed solver for equilibrium problems

Implementation

- For GAMS, the implementation will be available on the GAMS website.

Examples in the papers

- listing3.gms
- listing4.gms
- listing5.gms
- listing9.gms
- listing10.gms
- listing12.gms
- table3.gms, s5_x.gdx, s10_x.gdx, s15_x.gdx, s20_x.gdx, s23_x.gdx
- listing13.gms
- listing15.gms

QVI examples

- qvi_deluca.gms
- qvi_harker.gms

Examples below were taken from the following paper: Francisco Facchinei, Christian Kanzow, and Simone Sagratella: QVILIB: A library of quasi-variational inequality test problems. Pacific Journal of Optimization 9(2), 225-250 (2013)

- qvi_bilin1a.gms
- qvi_bilin1b.gms
- qvi_box1a.gms
- qvi_box1b.gms
- qvi_box2a.gms, qvi_box2a.gdx
- qvi_box2b.gms, qvi_box2b.gdx
- qvi_box3a.gms, qvi_box3a.gdx
- qvi_box3b.gms, qvi_box3b.gdx
- qvi_kunr11.gms, qvi_kunr11.gdx