

5 MPEC

Name

Salvador Pineda Morente

Learning objectives

- Define mathematical programming with equilibrium constraints (MPEC)
- Define bilevel programming and Stackelberg games
- Formulate and solve MPECs using different methods

Bibliography

- Basic reading:
 - Bard, Jonathan F. Practical bilevel optimization: algorithms and applications. Vol. 30. Springer, 1998 (chapter 1).
 - Gabriel, S., Conejo, A., Fuller, D., Hobbs, B., and Ruiz, C. Complementarity modeling in energy markets. Springer, 2012 (chapter 3).
 - Z.Q. Luo, J.S. Pang and D. Ralph: Mathematical Programs with Equilibrium Constraints. Cambridge University Press, 1996.
- Further reading:
 - Hobbs, B. F., Metzler, C. B., Pang, J. S. (2000). Strategic gaming analysis for electric power systems: An MPEC approach. Power Systems, IEEE Transactions on, 15(2), 638-645.
 - Zugno, M., Morales, J. M., Pinson, P., Madsen, H. (2013). A bilevel model for electricity retailers' participation in a demand response market environment. Energy Economics, 36, 182-197.
 - Ruiz, C., Conejo, A. J. (2009). Pool strategy of a producer with endogenous formation of locational marginal prices. Power Systems, IEEE Transactions on, 24(4), 1855-1866.
 - Morales, J. M., Pinson, P., Madsen, H. (2012). A transmission-cost-based model to estimate the amount of market-integrable wind resources. Power Systems, IEEE Transactions on, 27(2), 1060-1069.
 - Scheel, H., Scholtes, S. (2000). Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. Mathem. of Operations Research, 25(1), 1-22.

Hierarchical optimization

Hierarchical optimization (also known as multi-level programming) deals with problems that include multiple players that sequentially make decisions to maximize their objective functions following a hierarchical structure. Decisions of players in higher levels (leaders) have an impact on the objective function or feasible set of players in lower levels (followers).

Stackelberg game

In the simplest hierarchical optimization problem there are only two decision makers. One player, called the leader, makes her decisions first and then the other player, called the follower, determines the optimal reaction to the leader's decisions. This non-cooperative sequential game is known as Stackelberg game. Stackelberg games are usually formulated as bilevel programming problems.

Bilevel programming

Bilevel optimization is a special kind of optimization where one problem is embedded (nested) within another. The outer optimization problem is commonly referred to as the upper-level problem, and the inner optimization problem is commonly referred to as the lower-level problem. These problems involve two kinds of variables, referred to as the upper-level variables and the lower-level variables. A generic bilevel programming problem can be formulated as follows

$$\begin{aligned} \min_x \quad & F_0(x, y) \\ \text{s.t.} \quad & F_i(x, y) \leq 0, \quad i = 1, \dots, M \\ & H_i(x, y) = 0, \quad i = 1, \dots, P \\ \min_y \quad & f_0(x, y) \\ \text{s.t.} \quad & f_i(x, y) \leq 0, \quad i = 1, \dots, m \\ & h_i(x, y) = 0, \quad i = 1, \dots, p \end{aligned}$$

Important definitions in bilevel programming

- Constraint region $\Omega = \{(x, y) : F_i(x, y) \leq 0, H_i(x, y) = 0, f_i(x, y) \leq 0, h_i(x, y) = 0\}$
- Leader feasible region $\Omega^L = \{x : \exists y, (x, y) \in \Omega\}$
- Follower feasible region $\Omega^F(x) = \{y : f_i(x, y) \leq 0, h_i(x, y) = 0\}$
- Follower reaction set $\Pi(x) = \{y : y \in \operatorname{argmin}\{f_0(x, y') : y' \in \Omega^F(x)\}\}$
- Induced region $IR = \{(x, y) : x \in \Omega^L, y \in \Pi(x)\}$

Using these definitions a bilevel problem can be reformulated as a single-level problem as

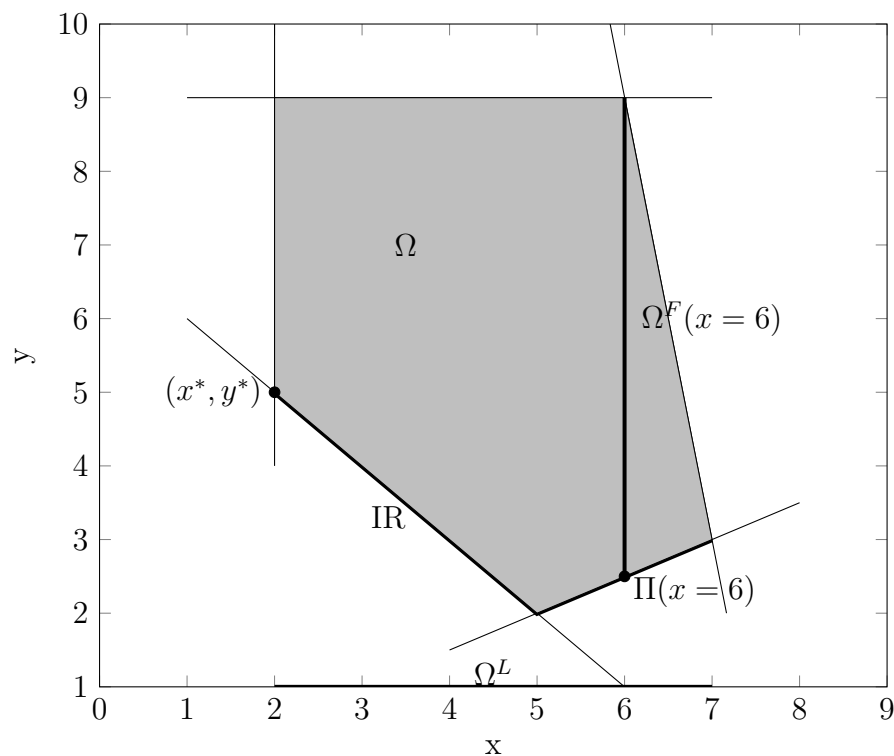
$$\begin{aligned} \min_x \quad & F_0(x, y) \\ \text{s.t.} \quad & (x, y) \in IR \end{aligned}$$

📌 Example 5-1: Bilevel problem

Given the following bilevel optimization problem

$$\begin{aligned}
 \min_x \quad & -y \\
 \text{s.t.} \quad & -x \leq -2 \\
 \min_y \quad & y \\
 \text{s.t.} \quad & -x - y \leq -7 \\
 & y \leq 9 \\
 & x - 2y \leq 1 \\
 & 6x + y \leq 45
 \end{aligned}$$

- Draw in the figure below the constraint region Ω
- Provide the optimal values of x, y for an hypothetical single level optimization problem consisting of the upper-level objective function and all upper- and lower-level constraints
- Provide the optimal values of x, y for an hypothetical single level optimization problem consisting of the lower-level objective function and all upper- and lower-level constraints
- Draw the leader feasible region Ω^L
- Draw the follower feasible region for $x = 6$
- Draw the follower reaction set for $x = 6$
- Draw the induced region IR. Is the IR a convex set?
- Provide the optimal values of x, y for the original bilevel problem



Reformulation of bilevel problem as single-level optimization

If the lower-level is a convex optimization problem and satisfies some constraint qualification, then it can be replaced by its KKT conditions as follows

$$\begin{aligned}
 \min_{x,y,\lambda,\nu} \quad & F_0(x, y) \\
 \text{s.t.} \quad & F_i(x, y) \leq 0, \quad i = 1, \dots, M \\
 & H_i(x, y) = 0, \quad i = 1, \dots, P \\
 & f_i(x, y) \leq 0, \quad i = 1, \dots, m \\
 & h_i(x, y) = 0, \quad i = 1, \dots, p \\
 & \lambda_i \geq 0, \quad \forall i = 1, \dots, m \\
 & \nabla_y f_0(x, y) + \sum_{i=1}^m \lambda_i \nabla_y f_i(x, y) + \sum_{i=1}^p \nu_i \nabla_y h_i(x, y) = 0 \\
 & \lambda_i f_i(x, y) = 0, \quad \forall i = 1, \dots, m
 \end{aligned}$$

This single-level optimization problem is usually referred to as mathematical program with complementarity conditions (MPCC).

Example 5-2: Convexity of MPCC

For affine functions $F_i(x, y)$, $H_i(x, y)$, $f_i(x, y)$, $h_i(x, y)$, is the MPCC formulated above a convex optimization problem? Is there any strictly feasible point?

No, MPCC are not convex optimization problems because they include, by definition, non-affine equality constraints. No, there is none strictly feasible point since either $f_i(x, y)$ or λ_i must be equal to zero by definition.

Mangasarian–Fromovitz constraint qualification (MFCQ)

Let us have the following optimization problem

$$\begin{aligned}
 \min_x \quad & f_0(x) \\
 \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m \\
 & h_i(x) = 0, \quad i = 1, \dots, p
 \end{aligned}$$

We say that MFCQ holds at a point $\tilde{x}$ if the equality constraint gradients are linearly independent at that point and there exists a vector $d \in \mathbb{R}^n$ such that

$$\begin{aligned}
 \nabla h_i(\tilde{x})^T d &= 0, \quad i = 1, \dots, p \\
 \nabla f_i(\tilde{x})^T d &< 0, \quad \forall i \in \mathcal{A}(\tilde{x})
 \end{aligned}$$

where $\mathcal{A}(\tilde{x})$ represents the set of active inequality constraints for $\tilde{x}$.

MFCQ ensures that there exists a feasible path and therefore it is a common requirement in algorithms to solve non-linear optimization problems.

MFQC is violated at any feasible point of a MPCC

Let us have the following generic MPCC

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_i(x) = 0, \quad i = 1, \dots, q \\ & 0 \leq F_i(x) \perp H_i(x) \geq 0, \quad i = 1, \dots, r \end{aligned}$$

where $x \in \mathbb{R}^n$, $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $F_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $H_i : \mathbb{R}^n \rightarrow \mathbb{R}$. To prove that MFQC is violated at any feasible point $\tilde{x}$ we take one complementarity condition and consider the three following cases:

- $F_i(\tilde{x}) > 0 \implies H_i(\tilde{x}) = 0$

Since constraint $H_i(\tilde{x}) \geq 0$ is active, then

$$\nabla H_i(\tilde{x})^T d < 0$$

Additionally, for the equality constraint $F_i(\tilde{x})H_i(\tilde{x}) = 0$ we have

$$\begin{aligned} \nabla (F_i(\tilde{x})H_i(\tilde{x}))^T d = 0 &\implies (\nabla F_i(\tilde{x})H_i(\tilde{x}) + F_i(\tilde{x})\nabla H_i(\tilde{x}))^T d = 0 \implies \\ (F_i(\tilde{x})\nabla H_i(\tilde{x}))^T d = 0 &\implies \nabla H_i(\tilde{x})^T d = 0 \end{aligned}$$

There is no $d \in \mathbb{R}^n$ that satisfies both conditions simultaneously.

- $H_i(\tilde{x}) > 0 \implies F_i(\tilde{x}) = 0$

Since constraint $F_i(\tilde{x}) \geq 0$ is active, then

$$\nabla F_i(\tilde{x})^T d < 0$$

Additionally, for the equality constraint $F_i(\tilde{x})H_i(\tilde{x}) = 0$ we have

$$\begin{aligned} \nabla (F_i(\tilde{x})H_i(\tilde{x}))^T d = 0 &\implies (\nabla F_i(\tilde{x})H_i(\tilde{x}) + F_i(\tilde{x})\nabla H_i(\tilde{x}))^T d = 0 \implies \\ (H_i(\tilde{x})\nabla F_i(\tilde{x}))^T d = 0 &\implies \nabla F_i(\tilde{x})^T d = 0 \end{aligned}$$

There is no $d \in \mathbb{R}^n$ that satisfies both conditions simultaneously.

- $F_i(\tilde{x}) = H_i(\tilde{x}) = 0$

The gradient of this equality constraint is a vector full of zeros and therefore the linear independence of the equality constraint gradients is not satisfied.

Therefore, any feasible point $\tilde{x}$ violates MFQC.

Example 5-3: Single level formulation of bilevel problem

Reformulate the bilevel problem of Example 5-1 as a single-level optimization problem.

$$\begin{aligned} \min_x \quad & -y \\ \text{s.t.} \quad & -x \leq -2 \\ & -x - y \leq -7 \\ & y \leq 9 \\ & x - 2y \leq 1 \\ & 6x + y \leq 45 \\ & \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0 \\ & 1 - \lambda_1 + \lambda_2 - 2\lambda_3 + \lambda_4 = 0 \\ & \lambda_1(-x - y + 7) = 0 \\ & \lambda_2(y - 9) = 0 \\ & \lambda_3(x - 2y - 1) = 0 \\ & \lambda_4(6x + y - 45) = 0 \end{aligned}$$

Solving example 5-3 (Bilevel.gms)

Use GAMS to solve the problem formulated in example 5-3. You can use the non-linear solver CONOPT. Did you obtain the global optimal solution? In case you did not obtain the global optimal solution, set the level of the variables x and y to a value close enough to the optimal one before the SOLVE statement. Did you obtain the optimal solution now?

Without the initial point close enough to the optimal one I do not get the optimal solution. This is because non-linear solvers only provide locally optimal points and since the problem is non-convex, then they do not need to be global optimal points. The performance of non-linear solvers usually depends on choosing appropriate initial solutions.

Product of binary and continuous variable

Let p be a continuous variable with lower and upper bound $\underline{p}$ and $\bar{p}$, respectively. Let x be a binary variable. Then the product

$$z = xp$$

can be equivalently formulated as the following set of linear equalities and inequalities

$$\begin{aligned} z &= p - r \\ \underline{xp} &\leq z \leq x\bar{p} \\ (1 - x)\underline{p} &\leq r \leq (1 - x)\bar{p} \end{aligned}$$

where r is an auxiliary continuous variable.

✚ Example 5-4: Linear bilevel problem (LBP) with binary upper-level variables

Let us have the following Linear bilevel optimization problem with $x \in \mathbb{Z}^m$, $y \in \mathbb{R}^n$ and $x \in \{0, 1\}$

$$\begin{aligned} \min_x \quad & c_1^T x + d_1^T y \\ \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\ & \min_y \quad c_2^T x + d_2^T y \\ & \text{s.t.} \quad A_2 x + B_2 y \leq e_2 \end{aligned}$$

where $c_1, d_1, A_1, B_1, e_1, c_2, d_2, A_2, B_2, e_2$ are vectors and matrices of the appropriate dimensions. Formulate the equivalent single-level problem replacing the lower-level problem by the KKT conditions. Identify the products of variables. Can you solve that problem as a mip? and as a nlp? Then formulate another single-level problem but use the primal-dual formulation discussed at the end of Block 2. Identify the products of variables. Can you solve that problem as a mip? and as a nlp? The single-level problem using KKT condition is

$$\begin{aligned} \min_{x,y,\lambda} \quad & c_1^T x + d_1^T y \\ \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\ & A_2 x + B_2 y \leq e_2 \\ & \lambda \geq 0 \\ & d_2 + B_2^T \lambda = 0 \\ & \lambda^T (A_2 x + B_2 y - e_2) = 0 \end{aligned}$$

The products of variables are λx (which can be linearized because it is the product of binary and continuous variables) and λy (which cannot be linearized because it is the product of two continuous variables). Therefore, this problem must be solved as nlp.

If we use the primal-dual formulation instead we have

$$\begin{aligned} \min_{x,y,\lambda} \quad & c_1^T x + d_1^T y \\ \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\ & A_2 x + B_2 y \leq e_2 \\ & \lambda \geq 0 \\ & d_2 + B_2^T \lambda = 0 \\ & c_2^T x + d_2^T y = c_2^T x + \lambda^T A_2 x - \lambda^T e_2 \end{aligned}$$

This problem only includes products of binary and continuous variables, which can be linearized. By doing so, we obtain a mixed-integer linear programming problem that can be solved to global optimality using branch-and-bound methods.

Mathematical Programming with Equilibrium Constraints (MPEC)

Mathematical Programming with Equilibrium Constraints includes those optimization problems in which some of the constraints are formulated as variational inequalities or complementarity problems. An MPEC can be formulated as follows

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_i(x) = 0, \quad i = 1, \dots, q \\ & 0 \leq F_i(x) \perp H_i(x) \geq 0, \quad i = 1, \dots, r \end{aligned}$$

where $x \in \mathbb{R}^n$, $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $F_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $H_i : \mathbb{R}^n \rightarrow \mathbb{R}$.

Example 5-5: Relation between bilevel programming and MPEC

Can all bilevel problems be equivalently formulated as MPEC? Can all MPECs be equivalently formulated as bilevel problems?

If the lower-level problem of a bilevel problem is convex, then it can be replaced by its KKT optimality conditions and therefore equivalently formulated as an MPEC. If the lower-level problem is not convex, then the KKT are not sufficient for optimality and therefore it cannot be equivalently formulated as an MPEC.

As discussed in Block 3, not all complementarity problems can be equivalently formulated as an optimization problem. In a similar way, not all MPEC can be formulated as bilevel programming problems.

Let us have the following MPEC problem

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_i(x) = 0, \quad i = 1, \dots, q \\ & 0 \leq F_i(x) \perp H_i(x) \geq 0, \quad i = 1, \dots, r \end{aligned}$$

where $x \in \mathbb{R}^n$, $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $F_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $H_i : \mathbb{R}^n \rightarrow \mathbb{R}$.

The different methods to solve this MPEC basically depend on the way the complementarity conditions $F_i(x)H_i(x) = 0$ are handled. We present below four different methods

1. Non-linear formulation. This method consists on formulating the MPEC as a non-linear optimization problem and solve it using a non-linear optimization software.

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_i(x) = 0, \quad i = 1, \dots, q \\ & F_i(x) \geq 0, \quad i = 1, \dots, r \\ & F_i(x)H_i(x) = 0, \quad i = 1, \dots, r \\ & H_i(x) \geq 0, \quad i = 1, \dots, r \end{aligned}$$

2. Regularization. This method consists on relaxing the complementarity conditions and iterating until $\sum_i F_i(x)H_i(x)$ is small enough as follows.

Step 1) Set $t \leftarrow 10^9$

Step 2) Solve the following non-linear optimization problem that satisfies MFCQ

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_i(x) = 0, \quad i = 1, \dots, q \\ & F_i(x) \geq 0, \quad i = 1, \dots, r \\ & \sum_i F_i(x)H_i(x) \leq t \\ & H_i(x) \geq 0, \quad i = 1, \dots, r \end{aligned}$$

Step 3) Set $t \leftarrow t/10$

Step 4) If t is lower than tolerance, stop. Otherwise, go so Step 2).

Methods to solve MPEC (II)

3. Use of SOS1 variables. This method makes use of Special Ordered Sets of type 1 (SOS1 or S1) that are a set of variables for which at most one of them takes a strictly positive value, being all others equal to 0. We use a SOS1 denoted as s_i to reformulate the MPEC as follows

$$\begin{aligned}
 \min_x \quad & f_0(x) \\
 \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\
 & h_i(x) = 0, \quad i = 1, \dots, q \\
 & F_i(x) \geq 0, \quad i = 1, \dots, r \\
 & s_i = \{F_i(x), H_i(x)\}, \quad i = 1, \dots, r \\
 & H_i(x) \geq 0, \quad i = 1, \dots, r
 \end{aligned}$$

4. Fortuny-Amat method. This method uses a vector of binary variables denoted as $u \in \mathbb{B}^m$ and a large constant M to reformulate the MPEC as follows

$$\begin{aligned}
 \min_x \quad & f_0(x) \\
 \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, p \\
 & h_i(x) = 0, \quad i = 1, \dots, q \\
 & F_i(x) \geq 0, \quad i = 1, \dots, r \\
 & F_i(x) \leq u_i M, \quad i = 1, \dots, r \\
 & H_i(x) \leq (1 - u_i) M, \quad i = 1, \dots, r \\
 & H_i(x) \geq 0, \quad i = 1, \dots, r \\
 & u_i \in \{0, 1\}, \quad i = 1, \dots, r
 \end{aligned}$$

✚ Example 5-6: Fill in the following table with the advantages and disadvantages of each method to solve MPECs

Method	Advantages	Disadvantages
Non-linear	- Single optimization problem to solve - No additional sets, variables or large constants	- Only local optimal (even for linear) - Feasible points non-regular - Solution depending on starting point
Regularization	- Satisfy MFQC (regular points) - No additional sets, variables or large constants	- Only local optimal (even for linear) - Solve multiple non-linear problems - Solution depending on starting point
SOS1 variables	- Single optimization problem to solve - Global optimal for linear (mip, b&b) - No additional variables or constants	- Use of special order sets - High computational burden
Fortuny-Amat	- Single optimization problem to solve - Global optimal for linear (mip, b&b)	- Additional binary variables - High computational burden - Proper tuning of large constants

📌 Example 5-7: Linear bilevel programming: single-level non-linear reformulation

Given the following linear bilevel programming problem

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 \min_y \quad & c_2^T x + d_2^T y \\
 \text{s.t.} \quad & A_2 x + B_2 y \leq e_2 : \lambda \\
 & y \geq 0 : \gamma
 \end{aligned}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^q, c_1 \in \mathbb{R}^n, d_1 \in \mathbb{R}^q, A_1 \in \mathbb{R}^p \times \mathbb{R}^n, B_1 \in \mathbb{R}^p \times \mathbb{R}^q, e_1 \in \mathbb{R}^p, c_2 \in \mathbb{R}^n, d_2 \in \mathbb{R}^q, A_2 \in \mathbb{R}^q \times \mathbb{R}^n, B_2 \in \mathbb{R}^q \times \mathbb{R}^q, e_2 \in \mathbb{R}^q$ and λ, γ are the Lagrange multipliers, provide its single-level non-linear reformulation.

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 & A_2 x + B_2 y \leq e_2 \\
 & y \geq 0 \\
 & \lambda \geq 0 \\
 & \gamma \geq 0 \\
 & d_2 + B_2^T \lambda - \gamma = 0 \\
 & \lambda^T (A_2 x + B_2 y - e_2) = 0 \\
 & \gamma^T y = 0
 \end{aligned}$$

✚ Example 5-8: Linear bilevel programming: regularization method

Given the following linear bilevel programming problem

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 \min_y \quad & c_2^T x + d_2^T y \\
 \text{s.t.} \quad & A_2 x + B_2 y \leq e_2 : \lambda \\
 & y \geq 0 : \gamma
 \end{aligned}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^q, c_1 \in \mathbb{R}^n, d_1 \in \mathbb{R}^q, A_1 \in \mathbb{R}^p \times \mathbb{R}^n, B_1 \in \mathbb{R}^p \times \mathbb{R}^q, e_1 \in \mathbb{R}^p, c_2 \in \mathbb{R}^n, d_2 \in \mathbb{R}^q, A_2 \in \mathbb{R}^q \times \mathbb{R}^n, B_2 \in \mathbb{R}^q \times \mathbb{R}^q, e_2 \in \mathbb{R}^q$ and λ, γ are the Lagrange multipliers, explain how to solve it using the regularization method. Provide the complete formulation of any optimization problem to be solved in the regularization method.

Step 1) Set $t \leftarrow 10^9$

Step 2) Solve the following non-linear optimization problem

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 & A_2 x + B_2 y \leq e_2 \\
 & y \geq 0 \\
 & \lambda \geq 0 \\
 & \gamma \geq 0 \\
 & d_2 + B_2^T \lambda - \gamma = 0 \\
 & \lambda^T (A_2 x + B_2 y - e_2) + \gamma^T y \leq t
 \end{aligned}$$

Step 3) Set $t \leftarrow t/10$

Step 4) If t is lower than tolerance, stop. Otherwise, go so Step 2).

📌 Example 5-9: Linear bilevel programming: SOS1 reformulation

Given the following linear bilevel programming problem

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 \min_y \quad & c_2^T x + d_2^T y \\
 \text{s.t.} \quad & A_2 x + B_2 y \leq e_2 : \lambda \\
 & y \geq 0 : \gamma
 \end{aligned}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m, c_1 \in \mathbb{R}^n, d_1 \in \mathbb{R}^m, A_1 \in \mathbb{R}^p \times \mathbb{R}^n, B_1 \in \mathbb{R}^p \times \mathbb{R}^m, e_1 \in \mathbb{R}^p, c_2 \in \mathbb{R}^n, d_2 \in \mathbb{R}^m, A_2 \in \mathbb{R}^q \times \mathbb{R}^n, B_2 \in \mathbb{R}^q \times \mathbb{R}^m, e_2 \in \mathbb{R}^q$ and λ, γ are the Lagrange multipliers, provide its single-level reformulation using special order sets of type 1.

I declare two special order sets of type 1 denoted as s_i^1, s_j^2 .

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 & A_2 x + B_2 y \leq e_2 \\
 & y \geq 0 \\
 & \lambda \geq 0 \\
 & \gamma \geq 0 \\
 & d_2 + B_2^T \lambda - \gamma = 0 \\
 & s_i^1 = \{(\lambda)_i, (A_2 x + B_2 y - e_2)_i\} \quad \forall i = 1, \dots, q \\
 & s_j^2 = \{(\gamma)_j, (y)_j\} \quad \forall j = 1, \dots, m
 \end{aligned}$$

✚ Example 5-10: Linear bilevel programming: Fortuny-Amat reformulation

Given the following linear bilevel programming problem

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 \min_y \quad & c_2^T x + d_2^T y \\
 \text{s.t.} \quad & A_2 x + B_2 y \leq e_2 : \lambda \\
 & y \geq 0 : \gamma
 \end{aligned}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m, c_1 \in \mathbb{R}^n, d_1 \in \mathbb{R}^m, A_1 \in \mathbb{R}^p \times \mathbb{R}^n, B_1 \in \mathbb{R}^p \times \mathbb{R}^m, e_1 \in \mathbb{R}^p, c_2 \in \mathbb{R}^n, d_2 \in \mathbb{R}^m, A_2 \in \mathbb{R}^q \times \mathbb{R}^n, B_2 \in \mathbb{R}^q \times \mathbb{R}^m, e_2 \in \mathbb{R}^q$ and λ, γ are the Lagrange multipliers, provide its single-level mixed-integer linear reformulation using the Fortuny-Amat approach.

I declare two binary variables $u^1 \in \mathbb{B}^q, u^2 \in \mathbb{B}^m$ and a large enough constant M .

$$\begin{aligned}
 \min_x \quad & c_1^T x + d_1^T y \\
 \text{s.t.} \quad & A_1 x + B_1 y \leq e_1 \\
 & x \geq 0 \\
 & e_2 - A_2 x - B_2 y \geq 0 \\
 & e_2 - A_2 x - B_2 y \leq u^1 M \\
 & \lambda \geq 0 \\
 & \lambda \leq (1 - u^1) M \\
 & y \geq 0 \\
 & y \leq u^2 M \\
 & \gamma \geq 0 \\
 & \gamma \leq (1 - u^2) M \\
 & u^1, u^2 \in \{0, 1\}
 \end{aligned}$$

Solving MPECs with GAMS (LinBilevel.gms)

Solve the linear bilevel problem in Example 5-7 using GAMS. You have to randomly generate the elements of all matrices and vectors as follows:

$$\begin{aligned} c_1 &= |\mathcal{N}(1, n)| & d_1 &= |\mathcal{N}(1, m)| & A_1 &= \mathcal{N}(p, n) & B_1 &= \mathcal{N}(p, m) & e_1 &= \mathcal{N}(p, 1) \\ c_2 &= |\mathcal{N}(1, n)| & d_2 &= |\mathcal{N}(1, m)| & A_2 &= \mathcal{N}(q, n) & B_2 &= \mathcal{N}(q, m) & e_2 &= \mathcal{N}(q, 1) \end{aligned}$$

where $\mathcal{N}(i, j)$ denotes a $i \times j$ matrix in which each element is randomly generated according to a standard normal distribution with mean and variance equal to 0 and 1, respectively. You have to solve the problem using the following methods:

1. Using just an MPEC solver (Check the available MPEC solvers for GAMS)
2. Using the non-linear formulation of Example 5-7
3. Using the regularization method of Example 5-8
4. Using SOS1 as described in Example 5-9
5. Using the Fortuny-Amat approach of Example 5-10 with $M = 1$
6. Using the Fortuny-Amat approach of Example 5-10 with $M = 100$
7. Using the Fortuny-Amat approach of Example 5-10 with $M = 100000$

GAMS advices:

- I highly recommend to code all the models in the same GAMS file. Keep in mind that the only different between the formulations in Examples 5-7, 5-8, 5-9, 5-10 is the way complementarity conditions are handled. So the objective function and most constraints are the same in all models.
- Use CPLEX and CONOPT as solvers for mip and nlp models, respectively.
- As we discussed, the performance of non-linear solver depends on the initial point. To evaluate all non-linear solvers in the same conditions make sure you set the initial values of all variables to 0 before each SOLVE statement in GAMS.
- Include the following code line at the beginning of your GAMS file so that we all obtain the same randomly generated matrices and results: `execseed = 504553610;`
- To generate randomly generated numbers that follow a normal distribution you have to use the command `normal(mean,sd)`. For example: `PARAMETER A1(p,n); A1(p,n)=normal(0,1);`
- Remember that you need one SOS1 variable for each complementarity condition
- Remember that for the Fortuny-Amat approach you need one binary for each complementarity condition
- GAMS stops after 1000 seconds running and return `solvstat` equal to 3.
- You can obtain the `modelstat` and time using `ModelName.modelstat` and `ModelName.resusd`, respectively

Solving MPECs with GAMS (LinBilevel.gms)

Fill in the table below for $n = m = 20, p = q = 10$. If a model is infeasible, then write INFES in the optimal value row.

Method	Optimal value	Time (seconds)
MPEC	INFES	0.003
NLP	INFES	0.023
NLP-REG	7.196	0.046
SOS1	6.219	0.023
FA-1	INFES	0.005
FA-100	6.219	0.048
FA-100000	5.872	0.031

Fill in the table below for $n = m = 100, p = q = 50$. If a model is infeasible, then write INFES in the optimal value row.

Method	Optimal value	Time (seconds)
MPEC	INFES	0.063
NLP	INFES	0.294
NLP-REG	5.805	1.241
SOS1	6.129	1000
FA-1	INFES	0.074
FA-100	5.801	1000
FA-100000	4.295	1.274

Comment on the results above.

- Try to avoid MPEC, NLP and FA with low values of M since they can be infeasible
- For small problems, SOS1 and FA with proper values of M provide the optimal solution in a reasonable time
- For large problems, the computational burden of SOS1 and FA with proper values of M may increase exponentially
- NLP-REG provides local optimal solutions in a reasonable time for small and large problems
- Setting values extremely large of M may create significant numerical issues such as binary variables different from 0 and 1.

📌 Example 5-11: Generation expansion of power producers (II)

In Example 4-13 we formulated a single-level optimization model to determine the optimal capacity and production level of a power producer. In this example we will formulate this problem in a different way as following:

1) For known values of the generation capacities, write down the equilibrium problem corresponding to a set of price-taker power producers (similar to Example 4-9). Assume that minimum production levels are equal to zero ($\underline{x}_g = 0$), a linear production cost function ($a_g = c_g = 0$) and a linear inverse demand function ($p = \alpha_t - \beta_t \sum_g x_{gt}$).

$$\begin{aligned} 0 &\leq (b_g - p_t + \bar{\eta}_{gt}) \perp x_{gt} \geq 0, \quad \forall g, t \\ 0 &\leq \bar{x}_g - x_{gt} \perp \bar{\eta}_{gt} \geq 0, \quad \forall g, t \\ p_t &= \alpha_t - \beta_t \sum_g x_{gt}, \quad \forall t \end{aligned}$$

2) Now formulate an optimization problem that determines the optimal capacity of a given generator $\bar{x}_g$ subject to the market equilibrium of price-takers power producer. Assume also a linear investment cost function such as $\mathcal{C}_g^I = i_g \bar{x}_g$. Which type of problem did you obtain?

$$\begin{aligned} \min_{\bar{x}_g, x_{gt}, \bar{\eta}_{gt}, p_t} \quad & i_g \bar{x}_g + \sum_t (b_g - p_t) x_{gt} \\ \text{s.t.} \quad & 0 \leq (b_g - p_t + \bar{\eta}_{gt}) \perp x_{gt} \geq 0, \quad \forall g, t \\ & 0 \leq \bar{x}_g - x_{gt} \perp \bar{\eta}_{gt} \geq 0, \quad \forall g, t \\ & p_t = \alpha_t - \beta_t \sum_g x_{gt}, \quad \forall t \end{aligned}$$

The problem we obtained is an MPEC.

Solving Example 5-11 with GAMS (CapExpMPEC.gms)

Given the following data

	Unit	$\underline{x}_g$	$\bar{x}_g$	b_g	i_g
	g_1	0	-	10	100
	g_2	0	50	20	-

t	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
α_t	10	20	30	40	50	60	70	80	90	100

$$\beta_t = 1, \quad \forall t$$

- First, use your code CournotEqui.gms to fill in the table below with the profit of g_1 for the perfect competition and the Cournot equilibrium cases. To do so, fix $\bar{x}_{g_2}$ to 50 and $\bar{x}_{g_1}$ to the different values of the table. To compute the profit, you have to account both for production and investment costs of generating unit g_1 . Note that you will have to modify the code to consider several time periods. What is the optimal investment decision for each case?

$\bar{x}_{g_1}$	2	4	6	8	10	12	14	16	18	20
PerfectCom	84	136	156	144	100	48	-28	-128	-252	-400
CournotEqui	320	600	841	1049	1225	1361	1465	1535	1571	1581

The optimal investment under perfect competition is 6 MW and the optimal investment under Cournot assumptions is 20MW

- Second, determine the optimal capacity of generating unit g_1 given the capacity of g_2 according to the single-level problem formulated in Example 4-13. Note that you can do that by solving the code GenExpEqu.gms with the variable $\bar{x}_{g_2}$ fixed to 50. Fill in the table below assuming that both players behave either as price-takers or a la Cournot. Are the optimal investment equal to the ones obtained in 1)? And the profits? Why?

Model	Competition	$\bar{x}_{g_1}^*$	profit of g_1
Single-level	Perfect competition	13.3	0
Single-level	Cournot competition	13.9	1460.2

Solving Example 5-11 with GAMS (CapExpMPEC.gms)

Neither the investment decisions nor the profits are the same in 1) and 2). The reason is that the single level optimization problem in Example 4-13 assumes that investment decisions and production decisions are made simultaneously. Remember, that this is in fact one of the conditions for strategic games. However, for the results of 1) we are assuming that these decisions are made sequentially. First, we determine the investment decisions (which is fixed to a given value) and then we compute the equilibrium at the production level. By doing so, investment decisions can be made anticipating their effects on production decisions and, in turn, on prices and profits. This is called backward induction and we will talk more about it in the next chapter.

We can also explain it looking only at the perfect competitive results. As shown in class, under perfect competition, power producers invest until their profit is equal to 0. Note that using the GEnExpEqu.gms code, we obtain a profit of g_1 equal to 0 as expected. However, in the results in 1) we can see that if we reduce the capacity of g_1 to 6 MW, we create some scarcity in the system, prices increase, and also does the profit of g_1 . Producer g_1 can do so under a sequential game in which she plays the role of the leader.

- 3) Finally, code the formulation of Example 5-11 in GAMS (CapExpMPEC.gms). Then solve this MPEC for the same data above. You can use any of the different methods to solve the MPEC although I recommend using the Fortuny-Amat approach with a proper value of the large constants M . Fill in the table below assuming that both players behave either as price-takers or a la Cournot. Are the optimal investment equal to the ones obtained in 1)? And the profits? Why?

Model	Competition	$\bar{x}_{g_1}^*$	profit of g_1
MPEC	Perfect competition	6.3	156.3
MPEC	Cournot competition	20.0	1580.6

In the MPEC formulation the investment decision is made anticipating the outcome of the equilibrium of the market and then the investment decisions and the profit are equal to those in 1). The MPEC formulation allows us to take the sequence of decisions into account.

- 4) Which solution do you think is closer to reality, the one provided in 2) or the one provided in 3)? Justify your answer.

The solution provided in 3) is closer to reality because investment decisions are made first, and then production decisions are made knowing the decided capacities. Therefore, there is an actual sequence on these decisions that was not considered in the results of 2).