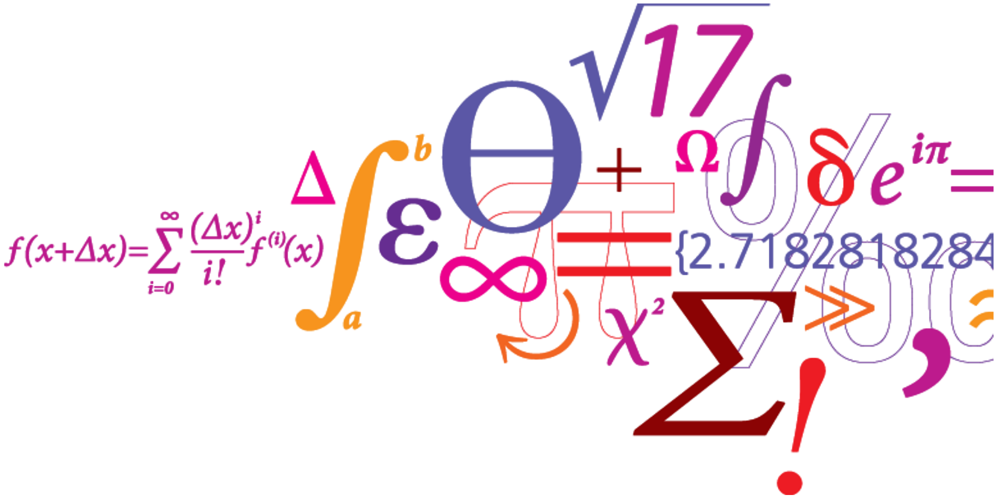
Large-Scale Optimization Problem in Energy Systems: Applications of Decomposition Techniques

Lecture: Introduction to Bilevel Programming

Lesia Mitridati

November 5th, 2018



DTU Electrical Engineering
Department of Electrical Engineering

Learning Objectives

At the end of these 2 sessions you should be able to…

- Define Stackelberg games
- Formulate Stackelberg games as bilevel programs and Mathematical Problems with Equilibrium Constraints (MPECs)
- Solve MPECs using 3 solution methods
- Recognize Stackelberg games in real life problems related to power systems

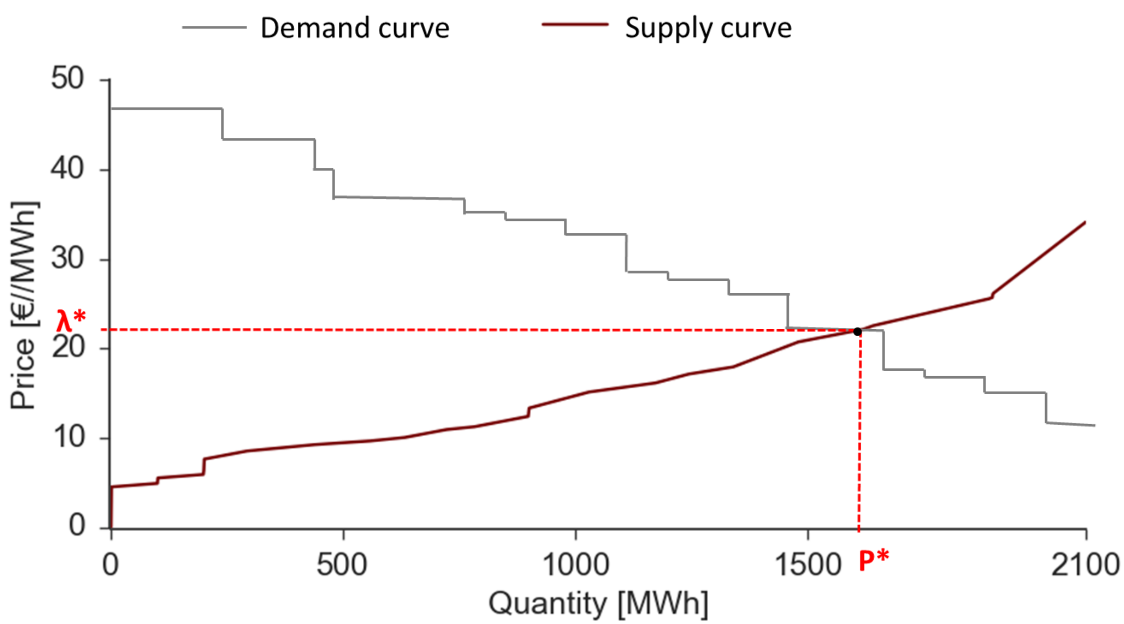
Reminder

In the previous lectures you talked about…

- Market clearing as an equilibrium problem
- Difference between perfect / imperfect competition
- <u>Strategic market players</u> can exercise “market power”

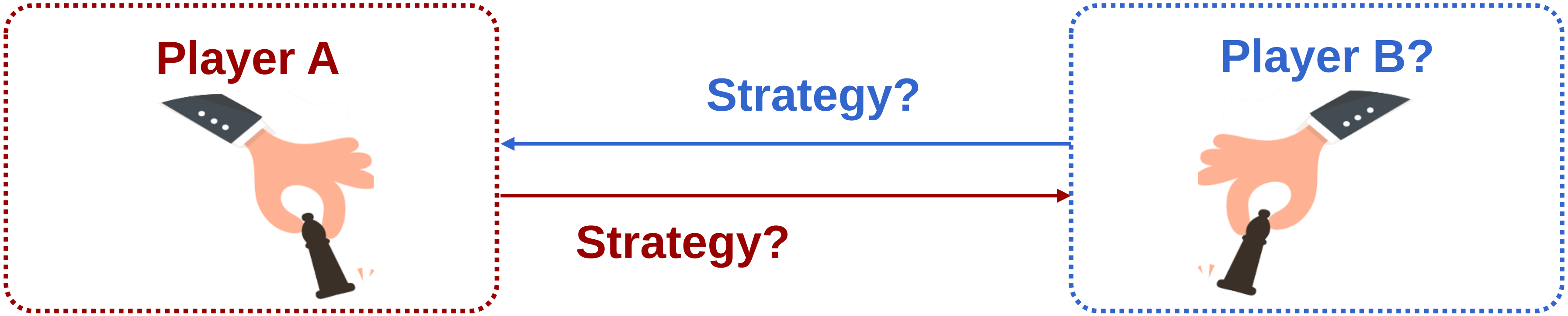
Market as an Equilibrium

- Perfect competition: no producer can exercise market power (i.e. their **actions do not influence** spot price)
- Each producer maximizes its profit



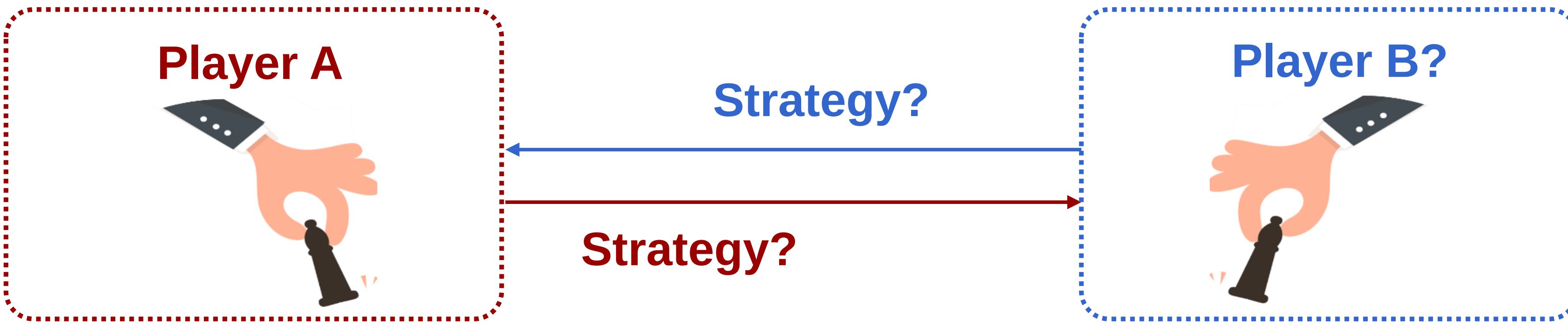
- Market clearing as an equilibrium
- All producers are satisfied with market outcome

Equilibrium Problem is a Static game



- Players simultaneously choose their strategies
- Players know each others' payoffs – But not each others' strategies
- When game is over: players receive payoffs based on combination of strategies

Equilibrium Problem is a Static Game

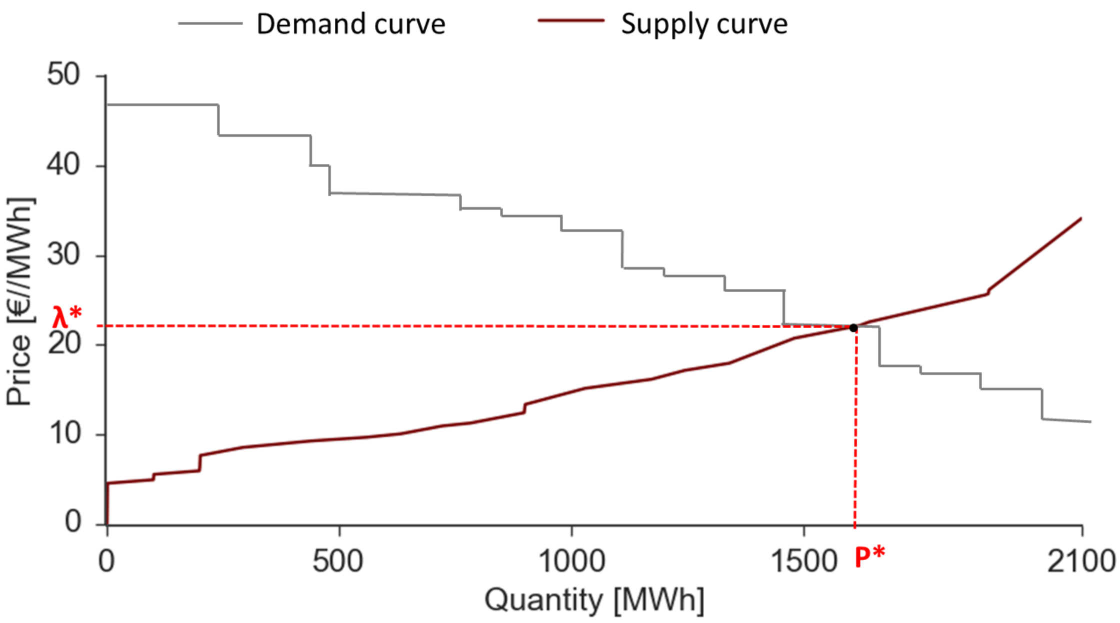


- Players <u>simultaneously</u> choose their strategies
- Players know each others' <u>payoffs</u> – But not each others' <u>strategies</u>
- When game is over: players receive payoffs based on <u>combination of strategies</u>

Question: How does this problem change if one player has a strategic advantage?

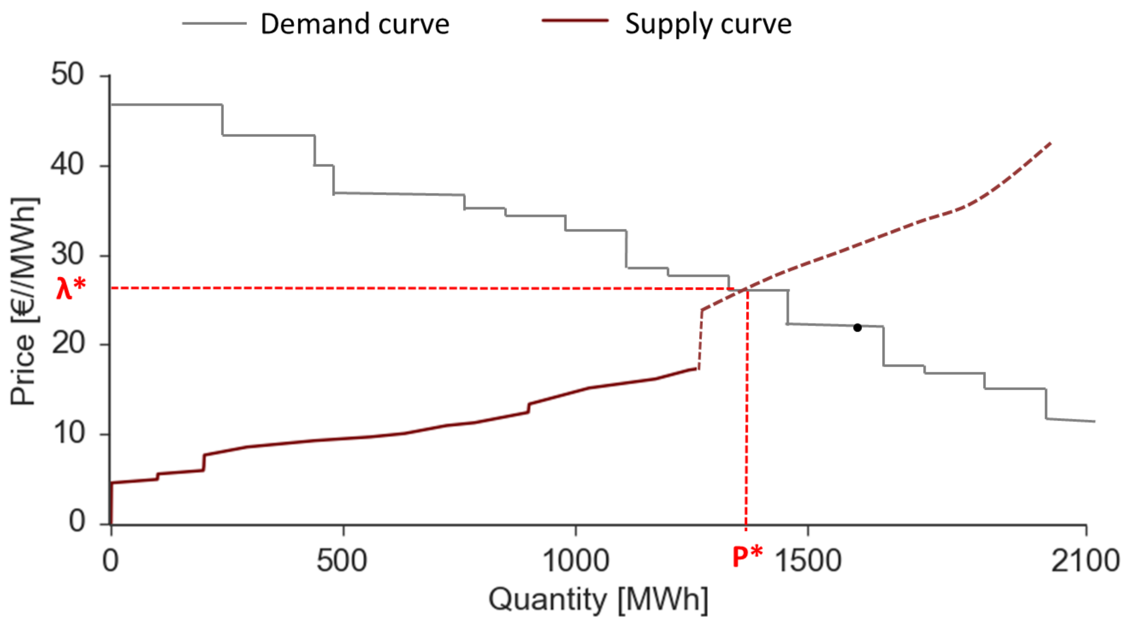
Strategic Producer

- A large producer participating in the day-ahead market
- Can exercise ”market-power” to increase its profit: i.e. its **action influences** market equilibrium (spot price)



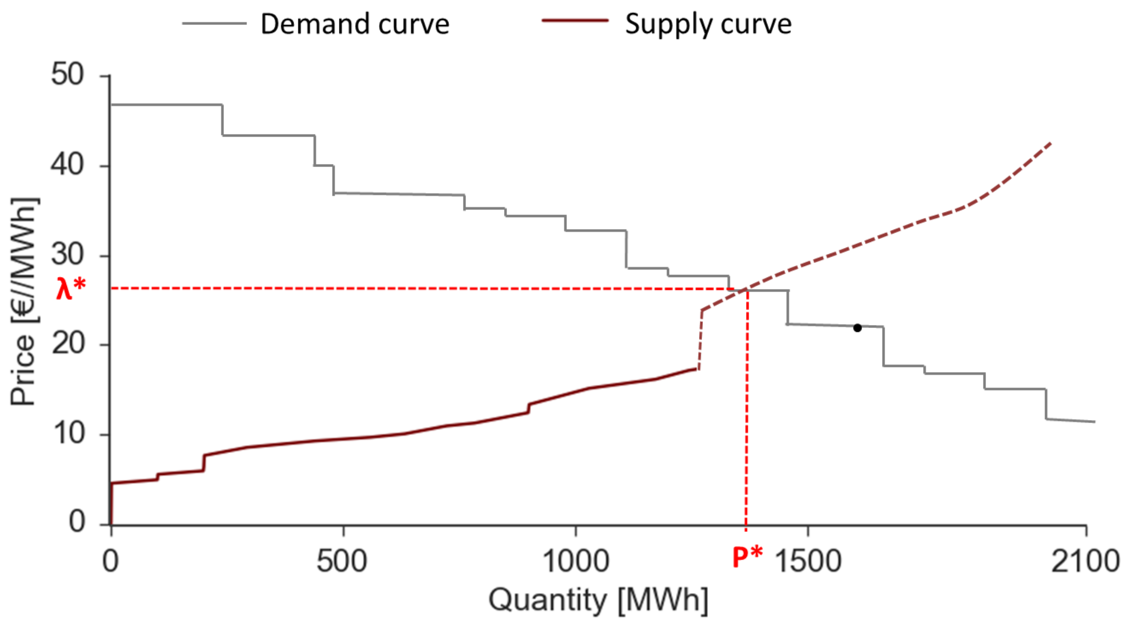
Strategic Producer

- A large producer participating in the day-ahead market
- Can exercise ”market-power” to increase its profit: i.e. its **action influences** market equilibrium (spot price)



Strategic Producer

- A large producer participating in the day-ahead market
- Can exercise ”market-power” to increase its profit: i.e. its **action influences** market equilibrium (spot price)



Question: How can the strategic producer design an optimal strategy, knowing that it can influence market outcomes?

Example 1

- 2 firms producing computers
- Can chose between high-end / low-end product range

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 / 10	30 / 50
Firm A: High-end	50 / 30	20 / 20

Example 1

- Static game: is there a dominant strategy?

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 / 10	30 / 50
Firm A: High-end	50 / 30	20 / 20

Example 1

- Static game: is there a dominant strategy?

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 \ 10	30 \ 50
Firm A: High-end	50 \ 30	20 \ 20

Example 1

- Static game: is there a dominant strategy? NO!

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 / 10	30 / 50
Firm A: High-end	50 / 30	20 / 20

Example 1

- Dynamic game: what if firm A can choose its strategy first? And firm B enters the market 2nd?

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 \ 10	30 \ 50
Firm A: High-end	50 \ 30	20 \ 20

Example 1

- Dynamic game: what if firm A can choose its strategy first? And firm B enters the market 2nd?

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 / 10	30 / 50
Firm A: High-end	50 / 30	20 / 20

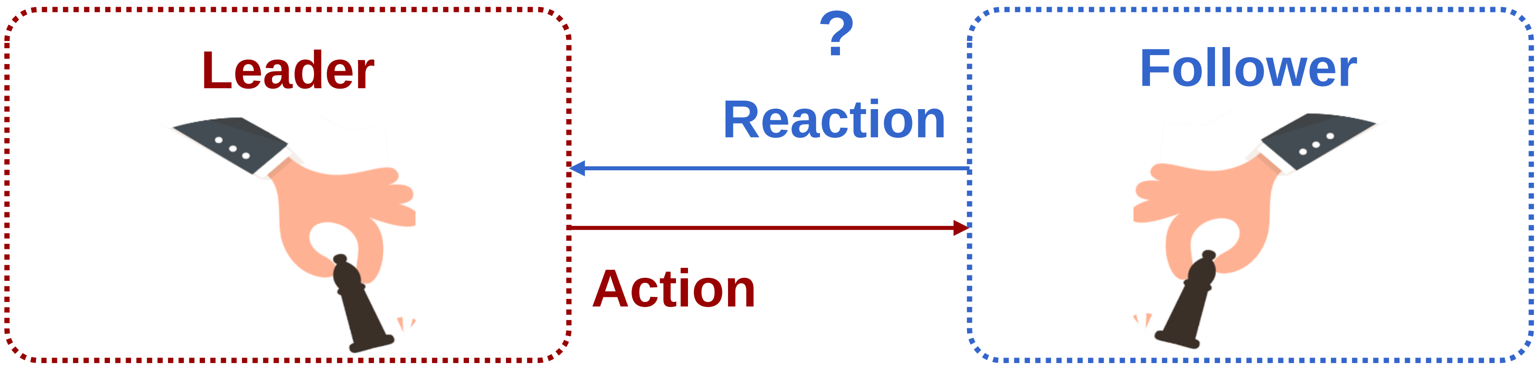
Example 1

- Dynamic game: what if firm B can choose its strategy first? And firm A enters the market 2nd?

	Firm B: Low-end	Firm B: High-end
Firm A: Low-end	10 (A), 10 (B)	30 (A), 50 (B)
Firm A: High-end	50 (A), 30 (B)	20 (A), 20 (B)

Dynamic Game: Stackelberg Game

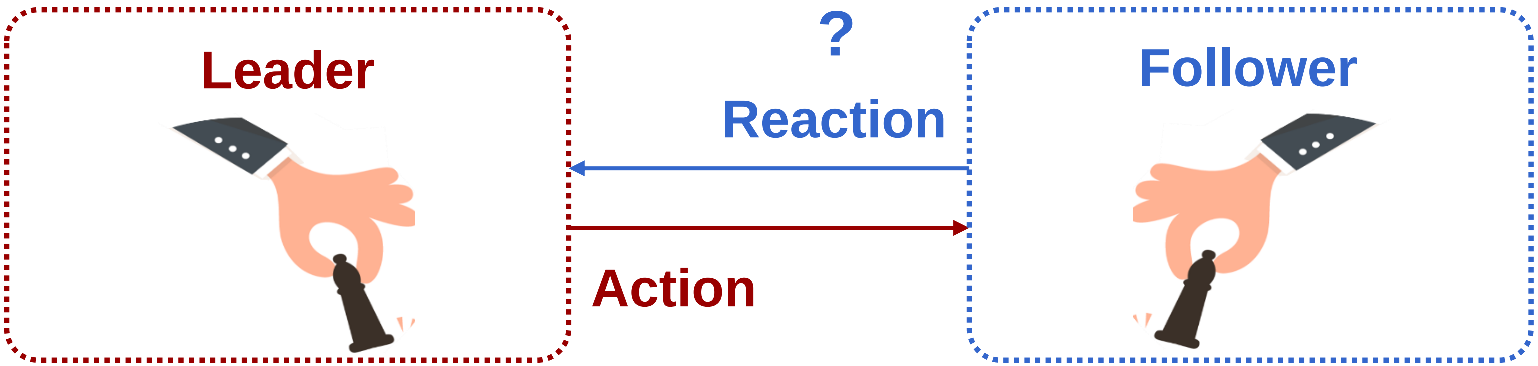
- 2-stage dynamic game



- Has a strategic advantage (plays first)
- Action **influences** optimal reaction of follower

Dynamic Game: Stackelberg Game

- 2-stage dynamic game

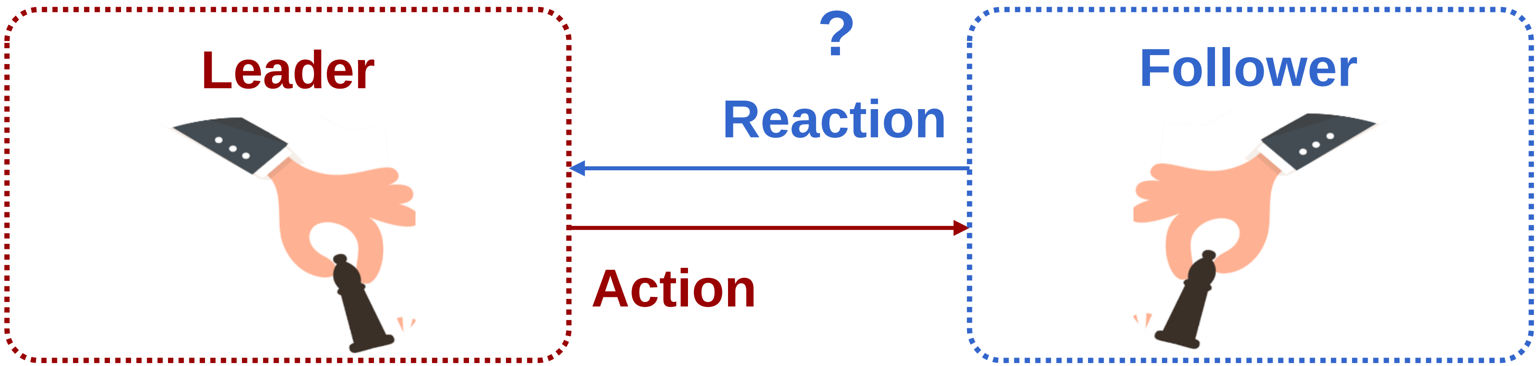


- Has a strategic advantage (plays first)
- Action **influences** optimal reaction of follower

- Finds optimal reaction to leader’s action
- Reaction **influences** leader’s profit

Dynamic Game: Stackelberg Game

- 2-stage dynamic game

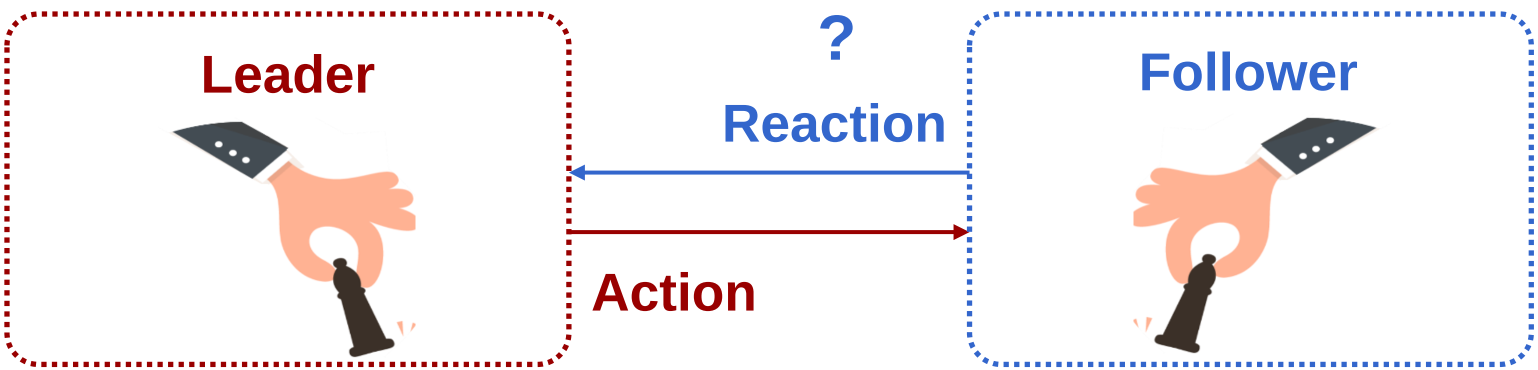


- Has a strategic advantage (plays first)
- Action **influences** optimal reaction of follower
- Tries to **anticipate** the follower's reaction

- Finds optimal reaction to leader's action
- Reaction **influences** leader's profit

Dynamic Game: Stackelberg Game

- 2-stage dynamic game



- Has a strategic advantage (plays first)
- Action **influences** optimal reaction of follower
- Tries to **anticipate** the follower's reaction

- Finds optimal reaction to leader's action
- Reaction **influences** leader's profit

Question: How can the leader anticipate the follower's reaction?

Example 2

- 2-player game

Leader

$$\min_{x} \; -y$$
$$\text{s.t.} \; -x \leq -2$$

Y (x*)

X*

Follower

$$\min_{y} \; y$$
$$\text{s.t.} \; y \leq 9$$

Example 2

- 2-player game

Leader

$$\min_{x} \; -y$$

$$\text{s.t.} \; -x \leq -2$$

Y (x*)

X*

Follower

$$\min_{y} \; y$$

$$\text{s.t.} \; y \leq 9$$

$$-x - y \leq -7$$

$$x - 2y \leq 1$$

$$6x + y \leq 45$$

Constraints linking the action of leader and reaction of follower

Example 2

- 2-player game

Leader

$$\min_{x,y} \; -y$$
$$\text{s.t.} \; -x \leq -2$$

$\xleftarrow{Y^*(x^*)}$

$\xrightarrow{X^*}$

Follower

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

- Reaction of follower constrained by the action of leader **(x is fixed in the follower's problem)**
- Leader tries to anticipate on the optimal reaction of the follower **(y(x) is a variable in the leader's problem)**

Example 2 – Group work!

Given the following problem:

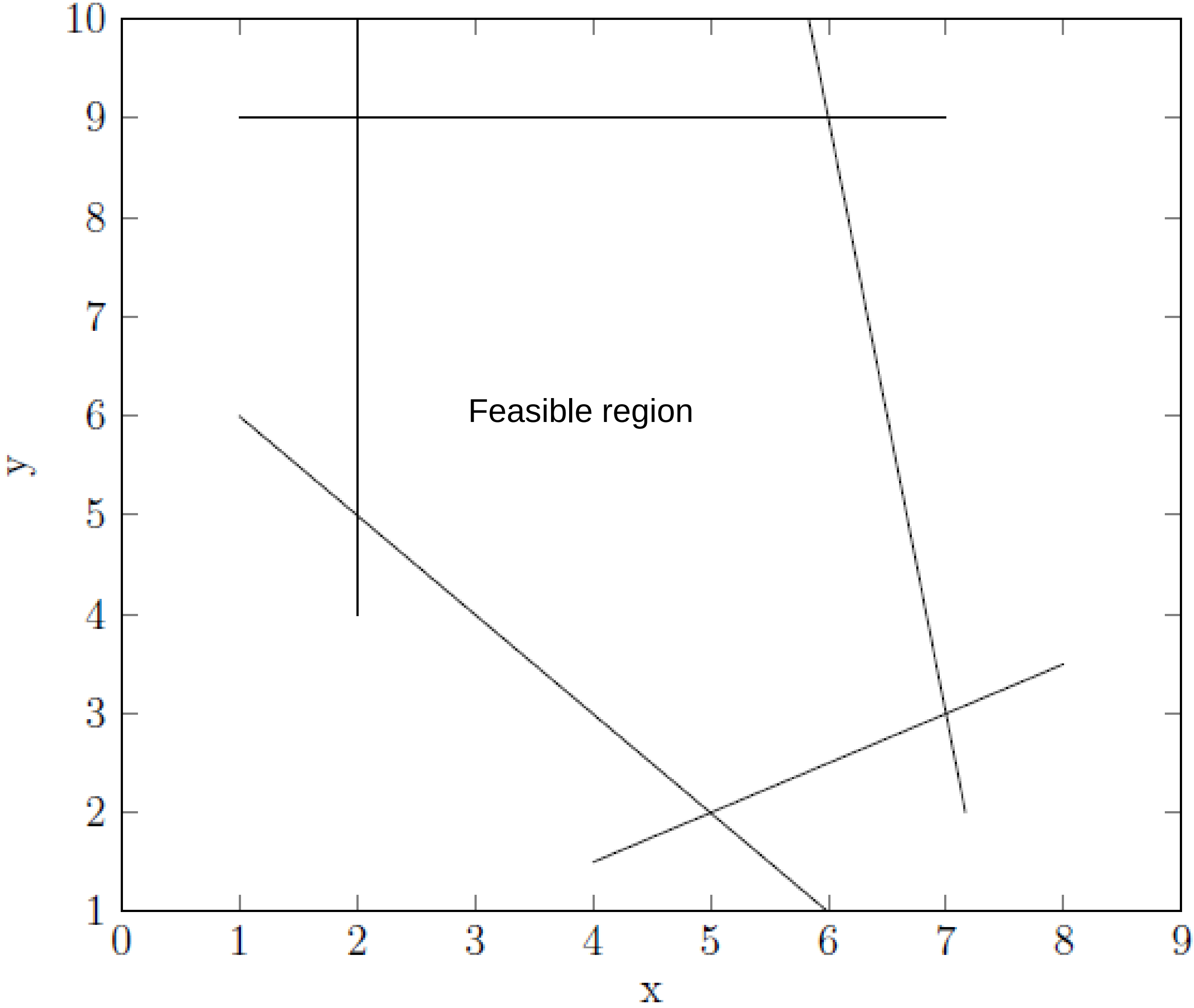
- 2 teams: leader and follower
- 10 minutes 

Questions:

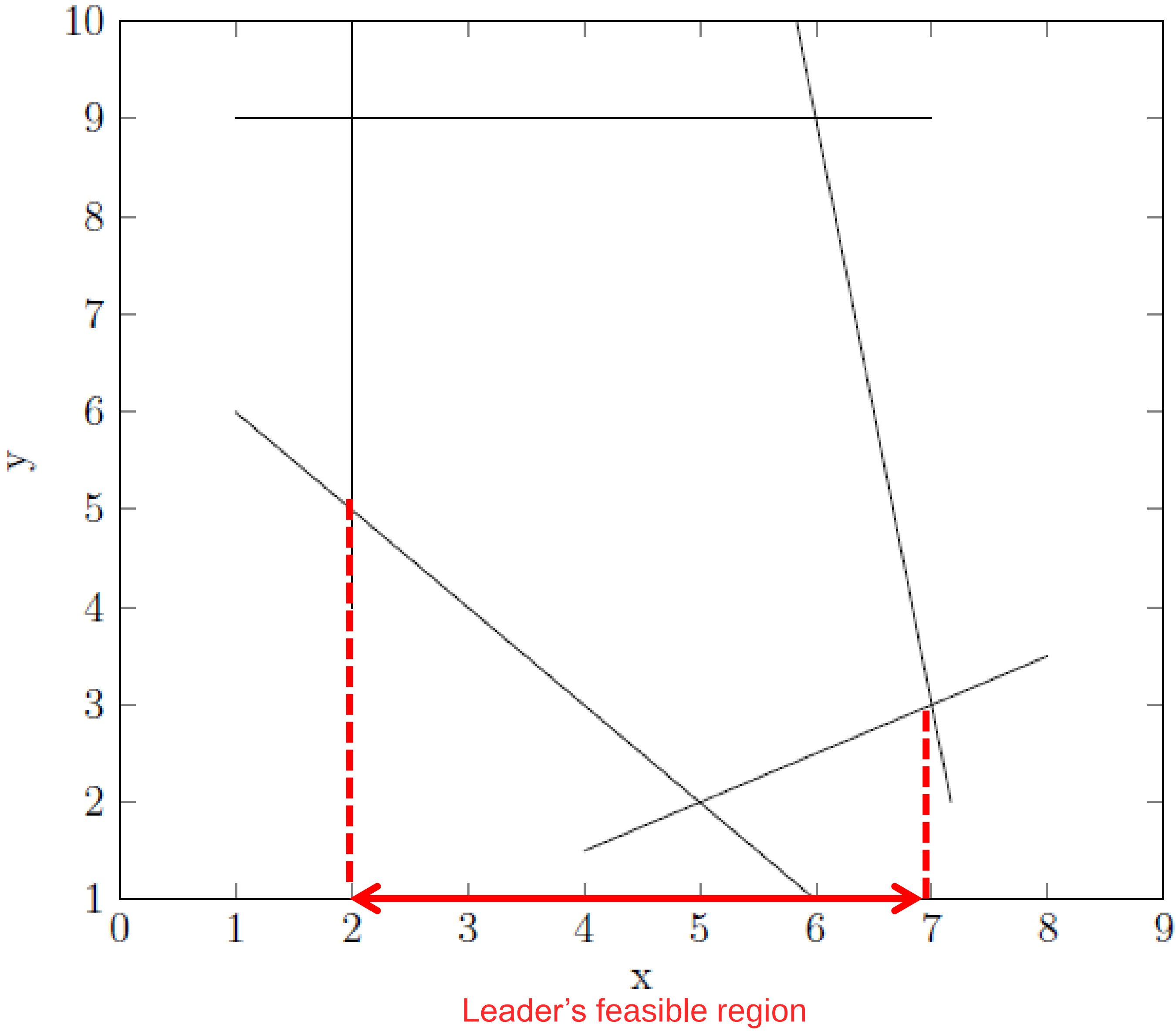
1. Both players: what is the joint feasible region of the leader – follower game? (draw)
2.
   a. Follower: what is your optimal reaction (as a function of the leader's action)?

   b. Leader: what is your optimal action?

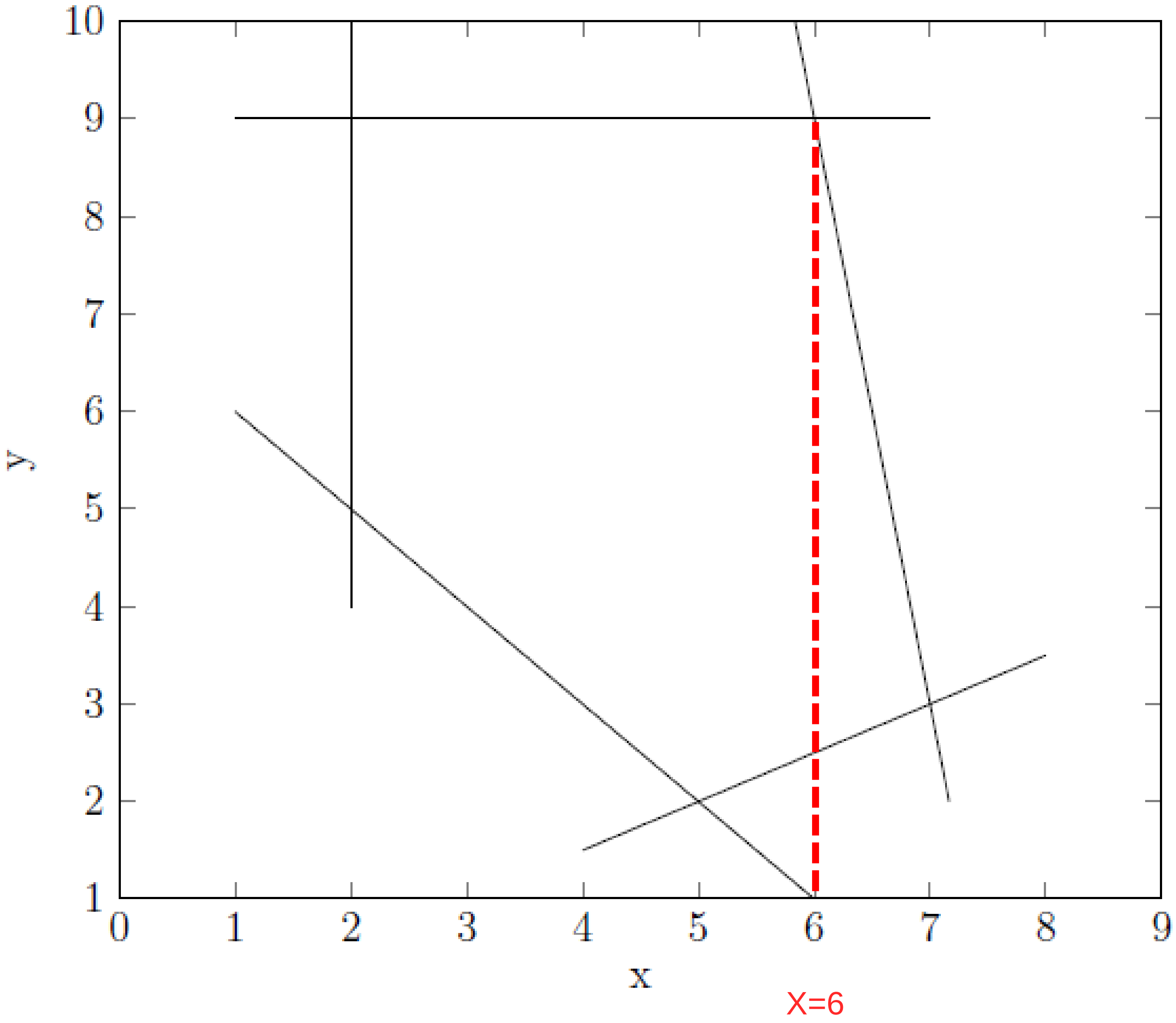
Example 2 - Solution



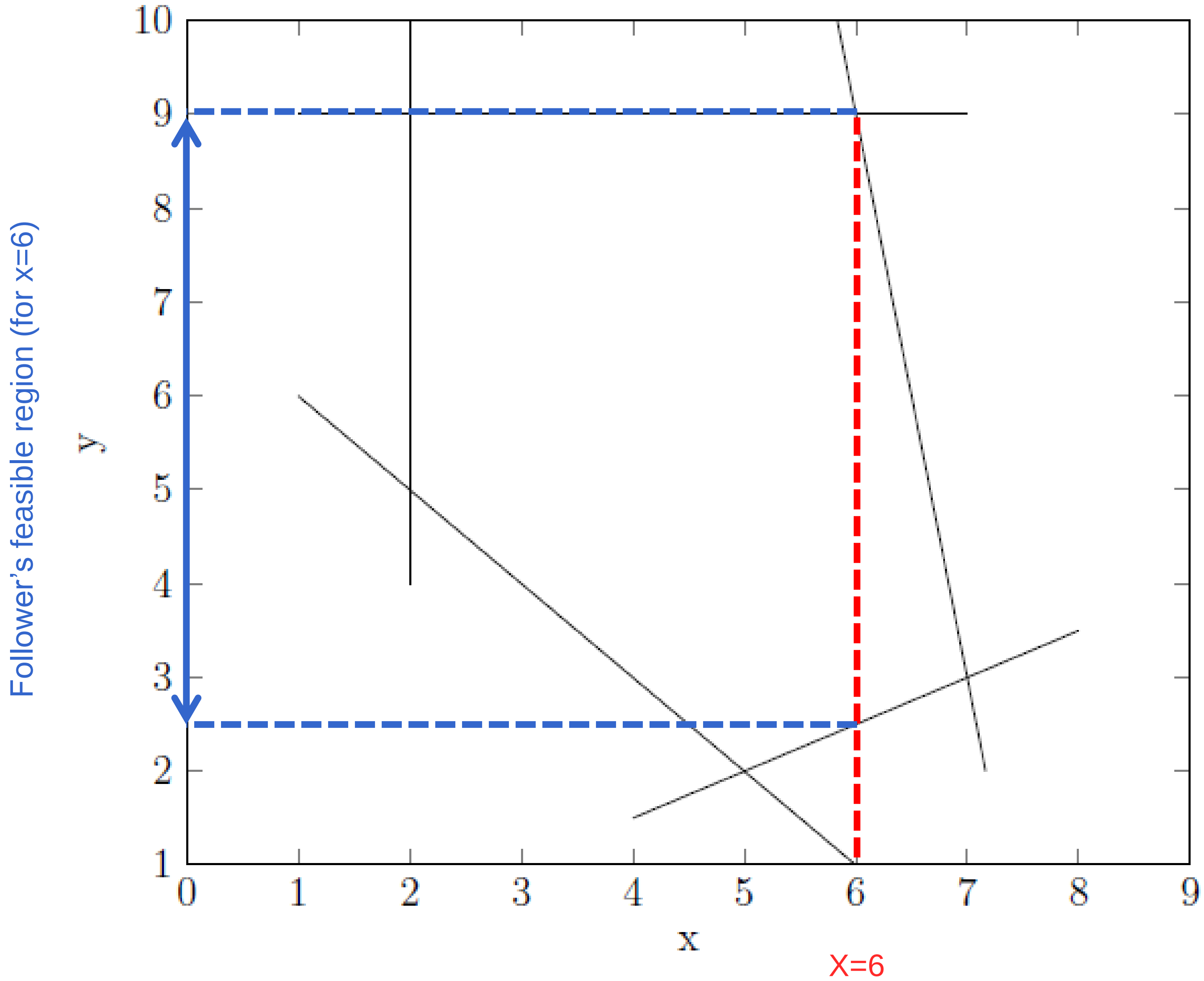
Example 2 - Solution



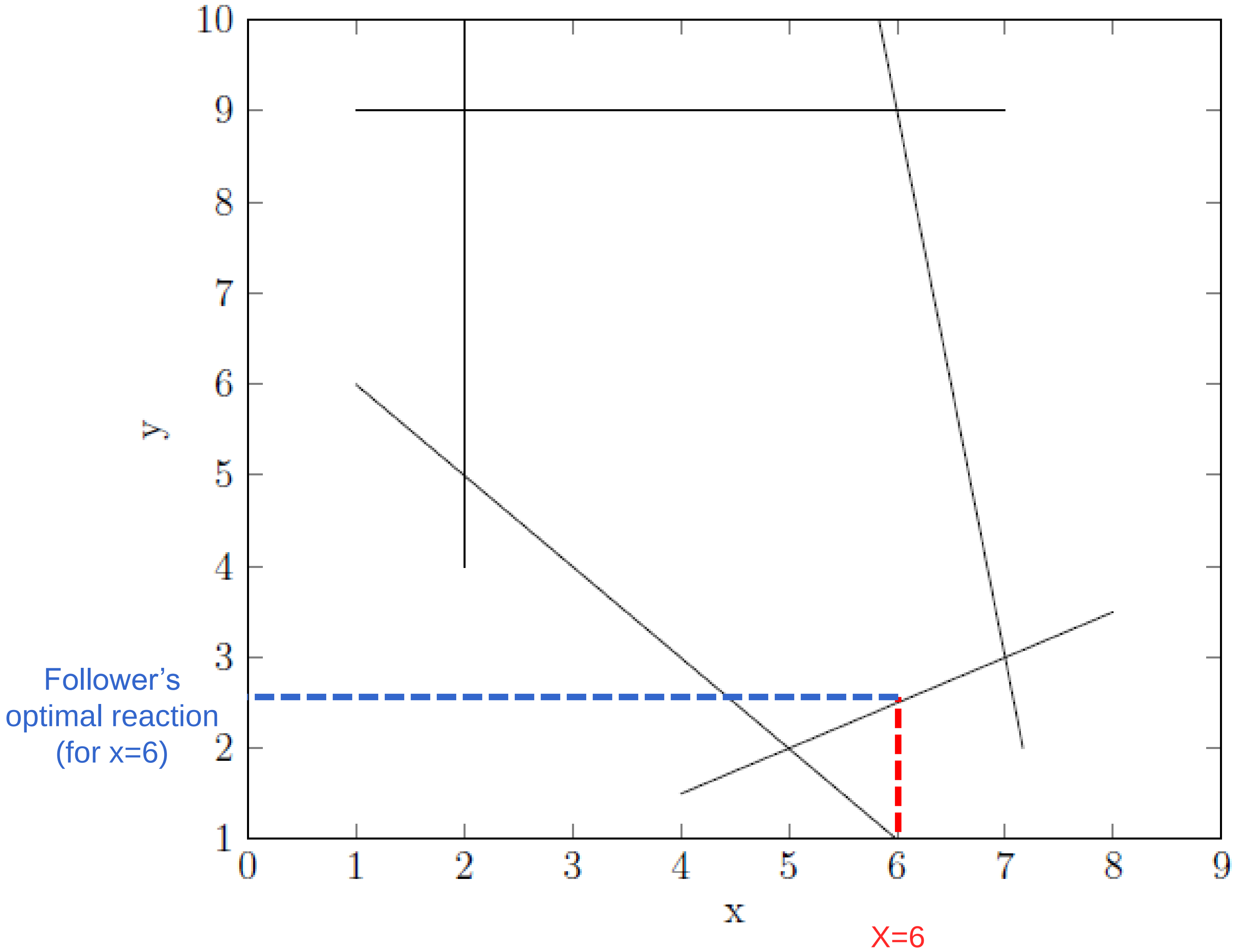
Example 2 - Solution



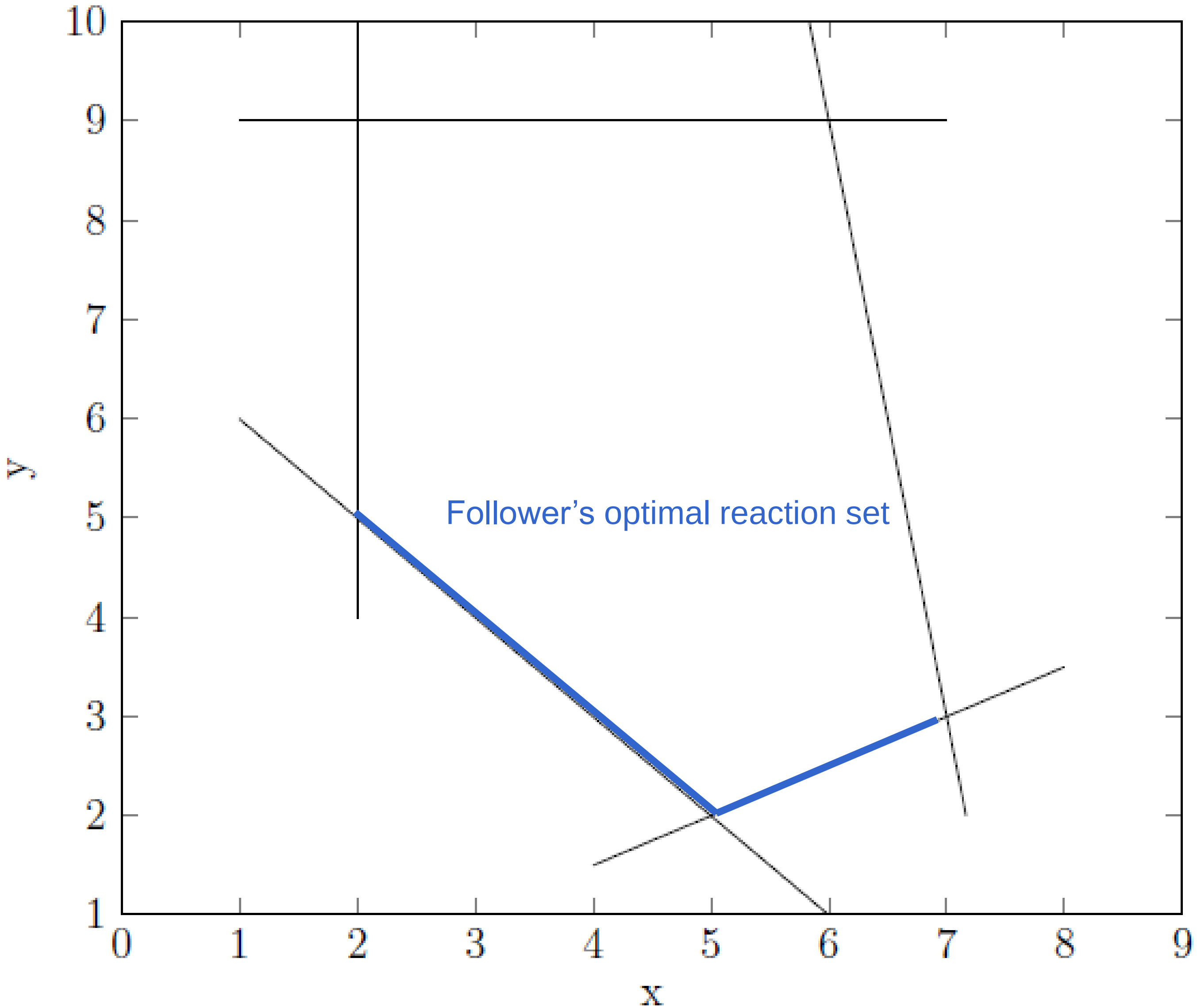
Example 2 - Solution



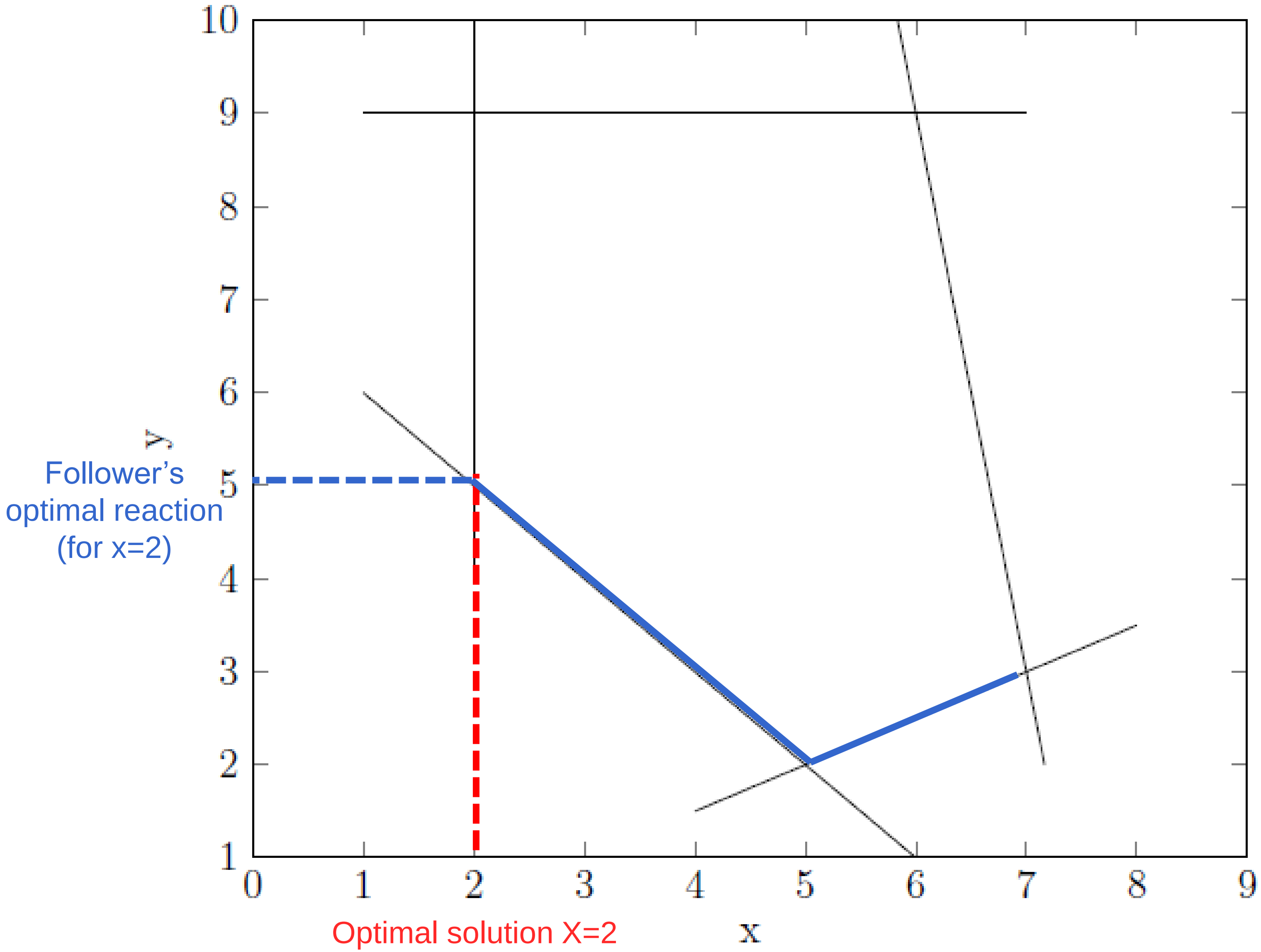
Example 2 - Solution



Example 2 - Solution



Example 2 - Solution



Example 2 as a Stackelberg Game

- Given the following bilevel optimization problem:

Leader

$$\min_{x,y} \quad -y$$

$$\text{s.t.} \quad -x \leq -2$$

Y*(x*)

X*

Follower

$$\min_{y} \quad y$$

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

Question: How can the leader integrate the follower's optimal reaction in its own startegy?

Example 2 as a Bilevel Optimization Problem

- The follower's problem becomes a constraint of the leader's problem: Bilevel optimization problem

$$\min_{x,y} \; -y$$

$$\text{s.t.} \; -x \leq -2$$

Leader

(x,y variable)

X* ↓ ↑ Y*(x*)

$$\text{s.t.} \; \min_{y} \; y$$

$$\text{s.t.} \; y \leq 9$$

$$-x - y \leq -7$$

$$x - 2y \leq 1$$

$$6x + y \leq 45$$

Follower

(x fixed)

From a Stackelberg Game to a Bilevel Program

$$
\begin{aligned}
\min_{x,y}\ & F_0(x,y) \\
\text{s.t. } & H_i(x,y) = 0, && i = 1, \dots, M \\
& G_i(x,y) \leq 0, && i = 1, \dots, P
\end{aligned}
$$

Leader's optimization problem

x*

y*(x*)

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

Follower's optimization problem

From a Stackelberg Game to a Bilevel Program

$$
\begin{aligned}
\min_{x,y} \quad & F_0(x,y) \\
\text{s.t.} \quad & H_i(x,y) = 0, && i = 1, \ldots, M \\
& G_i(x,y) \leq 0, && i = 1, \ldots, P \\
& \min_{y} \quad f_0(x,y) \\
& \quad \text{s.t.} \quad h_i(x,y) = 0, \quad : \lambda_i, && i = 1, \ldots, m \\
& \quad \qquad g_i(x,y) \leq 0, \quad : \mu_i, && i = 1, \ldots, p
\end{aligned}
$$

Leader's optimization problem

Follower's optimization problem

From a Stackelberg Game to a Bilevel Program

$$
\begin{aligned}
\min_{x,y} \quad & F_0(x,y) \\
\text{s.t.} \quad & H_i(x,y) = 0, && i = 1, \dots, M \\
& G_i(x,y) \le 0, && i = 1, \dots, P \\
& \min_{y} \quad f_0(x,y) \\
& \quad \text{s.t.} \quad h_i(x,y) = 0, \quad : \lambda_i, \; i = 1, \dots, m \\
& \quad \qquad\;\; g_i(x,y) \le 0, \quad : \mu_i, \; i = 1, \dots, p
\end{aligned}
$$

Upper-level optimization problem

Lower-level optimization problem

From a Stackelberg Game to a Bilevel Program

$$
\begin{aligned}
\min_{x,y} \quad & F_0(x,y) \\
\text{s.t.} \quad & H_i(x,y) = 0, && i = 1, \ldots, M \\
& G_i(x,y) \leq 0, && i = 1, \ldots, P \\
& \min_{y} \quad f_0(x,y) \\
& \quad \text{s.t.} \quad h_i(x,y) = 0, && : \lambda_i,\ i = 1, \ldots, m \\
& \quad\quad\quad g_i(x,y) \leq 0, && : \mu_i,\ i = 1, \ldots, p
\end{aligned}
$$

Upper-level optimization problem

Lower-level optimization problem

Question: How can we solve the leader's optimization problem with commercial solvers?

Mathematical Problem with Equilibrium Constraints (MPEC)

- Assumptions: Lower-level problem is convex and satisfies constraint qualifications (ex: Linear Program)

?

- Objective: recast bilevel program as a single-level program

Mathematical Problem with Equilibrium Constraints (MPEC)

- Assumptions: Lower-level problem is convex and satisfies constraint qualifications (ex: Linear Program)
- KKT conditions are sufficient and necessary

?

- Objective: recast bilevel program as a single-level program

Mathematical Problem with Equilibrium Constraints (MPEC)

- Assumptions: Lower-level problem is convex and satisfies constraint qualifications (ex: Linear Program)
- KKT conditions are sufficient and necessary
- ➢ Replace the lower-level optimization problem by its KKT conditions

$$
\begin{aligned}
\min_{x,y} \quad & F_0(x,y) \\
\text{s.t.} \quad & H_i(x,y) = 0, && i = 1, \ldots, M \\
& G_i(x,y) \leq 0, && i = 1, \ldots, P \\
& \min_{y} \quad f_0(x,y) \\
& \quad \text{s.t.} \quad h_i(x,y) = 0, \quad && : \lambda_i,\ i = 1, \ldots, m \\
& \quad \qquad g_i(x,y) \leq 0, \quad && : \mu_i,\ i = 1, \ldots, p
\end{aligned}
$$

- ➢ Objective: recast bilevel program as a single-level program

Mathematical Problem with Equilibrium Constraints (MPEC)

- Assumptions: Lower-level problem is convex and satisfies constraint qualifications (ex: Linear Program)
- KKT conditions are sufficient and necessary
- ➢ Replace the lower-level optimization problem by its KKT conditions

$$
\begin{aligned}
\min_{x,y,\lambda,\mu} \quad & F_0(x,y) \\
\text{s.t.} \quad & H_i(x,y) = 0, \quad i = 1, ..., M \\
& G_i(x,y) \le 0, \quad i = 1, ..., P \\
& \boxed{\begin{aligned}
& \nabla_y f_0(x,y) + \sum_{i=1}^{m} \lambda_i \nabla_y h_i(x,y) + \sum_{i=1}^{p} \mu_i \nabla_y g_i(x,y) = 0 \\
& h_i(x,y) = 0, \qquad\qquad i = 1, ..., m \\
& 0 \le \mu_i \perp g_i(x,y) \le 0, \quad i = 1, ..., p
\end{aligned}}
\end{aligned}
$$

- ➢ Bilevel program recast as a single-level program

Mathematical Problem with Equilibrium Constraints (MPEC)

- Assumptions: Lower-level problem is convex and satisfies constraint qualifications (ex: Linear Program)
- KKT conditions are sufficient and necessary

➢ Replace the lower-level optimization problem by its KKT conditions

Only y is variables of the lower-level problem! (not x)

$$
\begin{aligned}
\min_{x,y,\lambda,\mu} \quad & F_0(x,y) \\
\text{s.t.} \quad & H_i(x,y) = 0, \quad i = 1,...,M \\
& G_i(x,y) \leq 0, \quad i = 1,...,P \\
& \boxed{\nabla_y} f_0(x,y) + \sum_{i=1}^{m} \lambda_i \nabla_y h_i(x,y) + \sum_{i=1}^{p} \mu_i \nabla_y g_i(x,y) = 0 \\
& h_i(x,y) = 0, \quad i = 1,...,m \\
& 0 \leq \mu_i \perp g_i(x,y) \leq 0, \quad i = 1,...,p
\end{aligned}
$$

➢ Bilevel program recast as a single-level program

Mathematical Problem with Equilibrium Constraints (MPEC)

Solution Methods:

1. MPEC solver in Gams (KNITRO)
   - Non-linear complementarity constraints
2. Fortuny-Amat linearization (MILP)
   - Linearize complementarity conditions
   - Binary variables
   - Fine-tuning constant M
3. SOS1 variables (Python)
   - Set of variables, among which only one is non-zero
   - Treated by the solver as binaries

Mathematical Problem with Equilibrium Constraints (MPEC) – Solution Methods

Fortuny-Amat linearization (MILP)

Replace the complementarity conditions:

$$\boxed{0 \le \mu_i \perp g_i(x,y) \le 0, \quad i = 1, \dots, p}$$

By the constraints:

$$\boxed{\begin{aligned} &0 \le \mu_i \le M u_i, && i = 1, \dots, p \\ &0 \le -g_i(x,y) \le M(1-u_i), && i = 1, \dots, p \\ &u_i \in \{0,1\} && i = 1, \dots, p \end{aligned}}$$

Prepare for this afternoon…

1. Write the KKT conditions of the lower-level optimization problem in example 2 and reformulate the bilevel optimization problem as a single-level optimization problem

2. Solve the MPEC in example 2 with GAMS (or another programming language) using at least one of the following methods:
   - Complementarity solver (GAMS)
   - Using the Fortuny-Amat approach with M=1
   - Using the Fortuny-Amat approach with M=100
   - Using the Fortuny-Amat approach with $M=10^{16}$
   - Using SOS1 variables (Python)

Example 2 – Bilevel Formulation

$$\begin{aligned}
\min_{x,y,\mu} \quad & -y \\
\text{s.t.} \quad & -x \leq -2 \\
& \min_{y} \quad y \\
& \quad \text{s.t.} \quad -x - y \leq -7 \quad : \mu_1 \\
& \qquad\qquad y \leq 9 \quad : \mu_2 \\
& \qquad\qquad x - 2y \leq 1 \quad : \mu_3 \\
& \qquad\qquad 6x + y \leq 45 \quad : \mu_4
\end{aligned}$$

Example 2 – MPEC Formulation

$$\min_{x,y,\mu} \quad -y$$

$$\text{s.t.} \quad -x \leq -2$$

$$1 - \mu_1 + \mu_2 - 2\mu_3 + \mu_4 = 0$$

$$0 \leq \mu_1 \perp (-x - y + 7) \leq 0$$

$$0 \leq \mu_2 \perp (y - 9) \leq 0$$

$$0 \leq \mu_3 \perp (x - 2y - 1) \leq 0$$

$$0 \leq \mu_4 \perp (6x + y - 45) \leq 0$$

Example 2 – GAMS Code (MPEC)

```
OPTIONS mpec = KNITRO;

FREE VARIABLES
x
y
Obj_leader
;

POSITIVE VARIABLES
mu_1
mu_2
mu_3
mu_4
;

EQUATIONS
OF,constr_UL,constr_LL1,constr_LL2,constr_LL3,constr_LL4,constr_lagrange;

*primal constraints
OF.. Obj_leader =e= -y;
constr_UL.. x =g= 2;
constr_LL1..  x+y =g= 7;
constr_LL2..  -y =g= -9;
constr_LL3.. -x+2*y =g= -1;
constr_LL4..  -6*x - y =g= -45;

*KKT conditions
constr_lagrange.. 1 - mu_1 + mu_2 - 2*mu_3 + mu_4 =e= 0;
```

Example 2 – GAMS Code (MPEC)

```
MODEL MPEC
/OF
constr_UL
constr_LL1.mu_1
constr_LL2.mu_2
constr_LL3.mu_3
constr_LL4.mu_4
constr_lagrange/
;

SOLVE MPEC using mpec min Obj_leader;
Display x.l,y.l;
```

Complementarity conditions

Example 2 – GAMS Code (MILP)

```
PARAMETERS
M /100/
;

FREE VARIABLES
x
y
Obj_leader
;

POSITIVE VARIABLES
mu_1
mu_2
mu_3
mu_4
;

BINARY VARIABLES
u1
u2
u3
u4
;
```

Example 2 – GAMS Code (MILP)

```
EQUATIONS
OF,constr_UL,constr_LL1,constr_LL2,constr_LL3,constr_LL4
constr_lagrange,KKT_11,KKT_12,KKT_21,KKT_22,KKT_31,KKT_32,KKT_41,KKT_42;

*primal constraints
OF.. Obj_leader =e= -y;
constr_UL.. x =g= 2;
constr_LL1..  x+y =g= 7;
constr_LL2..  -y =g= -9;
constr_LL3.. -x+2*y =g= -1;
constr_LL4..  -6*x - y =g= -45;

*KKT conditions
constr_lagrange.. 1 - mu_1 + mu_2 - 2*mu_3 + mu_4 =e= 0;

KKT_11..  x+y-7 =l= M*(1-u1);
KKT_12..  mu_1 =l= M*u1;

KKT_21..  -y+9 =l= m*(1-u2);
KKT_22..  mu_2 =l= M*u2;

KKT_31..  -x+2*y+1 =l= M*(1-u3);
KKT_32..  mu_3 =l= M*u3;

KKT_41..  -6*x - y+45 =l= M*(1-u4);
KKT_42..  mu_4 =l= M*u4;
```

Example 2 – GAMS Code (MILP)

```
MODEL MPEC
/OF,constr_UL,constr_LL1,constr_LL2,constr_LL3,constr_LL4
constr_lagrange,KKT_11,KKT_12,KKT_21,KKT_22,KKT_31,KKT_32,KKT_41,KKT_42/
;

SOLVE MPEC using MIP Minimizing Obj_leader;
Display x.l,y.l;
```

Example 2 – Python Code (SOS 1 Variables)

```
import os
import pandas as pd
import scipy.stats as sp
import gurobipy as gb
import numpy as np

class expando(object):
    '''
    A small class which can have attributes set
    '''
    pass


class example_2:
    def __init__(self):
        self.data = expando()
        self.variables = expando()
        self.constraints = expando()
        self._build_model()


    def optimize(self):
        self.model.optimize()

    def _build_model(self):

        self.model = gb.Model()
        self._build_variables()
        self._build_objective()
        self._build_constraints()

    def _build_variables(self):

        #indexes shortcuts
        m = self.model


        self.variables.x =  m.addVar(lb=2,name='x')

        self.variables.y =  m.addVar(lb=-gb.GRB.INFINITY,name='y')
```

Example 2 – Python Code (SOS 1 Variables)

```
    self.variables.mu_1 =  m.addVar(name='mu 1') # positive dual variable
    self.variables.mu_2 =  m.addVar(name='mu 2') # positive dual variable
    self.variables.mu_3 =  m.addVar(name='mu 3') # positive dual variable
    self.variables.mu_4 =  m.addVar(name='mu 4') # positive dual variable

    self.variables.l_1 =  m.addVar(name='l 1') # positive auxiliary variable
    self.variables.l_2 =  m.addVar(name='l 2') # positive auxiliary variable
    self.variables.l_3 =  m.addVar(name='l 3') # positive auxiliary variable
    self.variables.l_4 =  m.addVar(name='l 4') # positive auxiliary variable

    m.update()

def _build_objective(self): # building the objective function of the leader!

    #indexes shortcuts
    m = self.model

    m.setObjective(-self.variables.y,
        gb.GRB.MINIMIZE)


def _build_constraints(self):

    #indexes shortcuts
    m = self.model

    # lower level inequality constraints

    self.constraints.c_1=m.addConstr(
                - 7 + self.variables.y + self.variables.x,
                gb.GRB.EQUAL,
                self.variables.l_1)

    self.constraints.c_2=m.addConstr(
                9 - self.variables.y,
                gb.GRB.EQUAL,
                self.variables.l_2)
```

Example 2 – Python Code (SOS 1 Variables)

```python
        self.constraints.c_3=m.addConstr(
                    1 + 2*self.variables.y - self.variables.x,
                    gb.GRB.EQUAL,
                    self.variables.l_3)


        self.constraints.c_4=m.addConstr(
                    45 - self.variables.y - 6*self.variables.x,
                    gb.GRB.EQUAL,
                    self.variables.l_4)

        # Stationarity condition

        self.constraints.L=m.addConstr(1-self.variables.mu_1+self.variables.mu_2-2*self.variables.mu_3+self.variables.mu_4,
                                       gb.GRB.EQUAL,
                                       0)

        # complementarity constraints

        self.constraints.SOS_1=m.addSOS(gb.GRB.SOS_TYPE1,[self.variables.l_1,self.variables.mu_1])
        self.constraints.SOS_2=m.addSOS(gb.GRB.SOS_TYPE1,[self.variables.l_2,self.variables.mu_2])
        self.constraints.SOS_3=m.addSOS(gb.GRB.SOS_TYPE1,[self.variables.l_3,self.variables.mu_3])
        self.constraints.SOS_4=m.addSOS(gb.GRB.SOS_TYPE1,[self.variables.l_4,self.variables.mu_4])


solution = example_2()
solution.optimize()

print('x=',solution.variables.x.x)
print('y=',solution.variables.y.x)
```

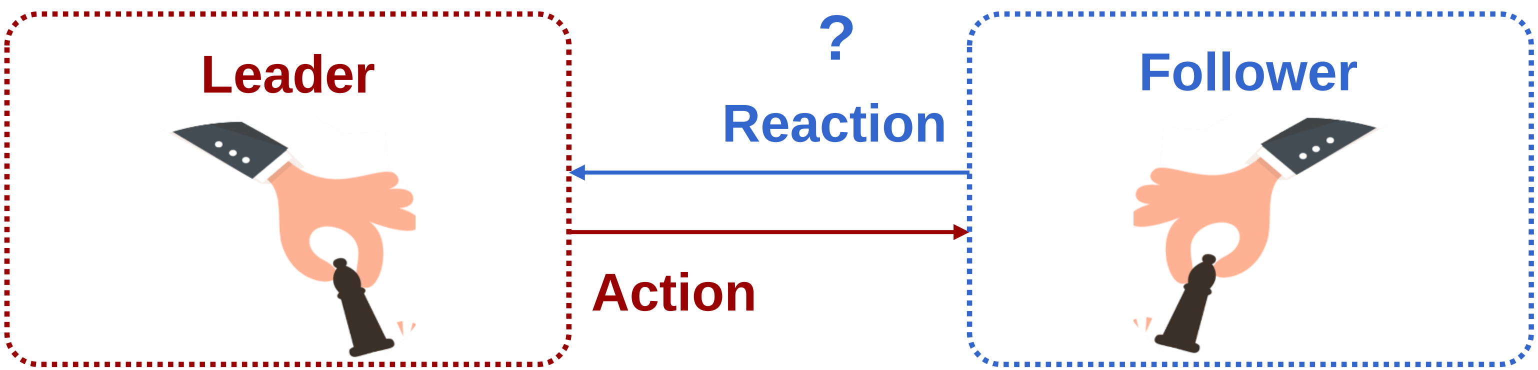
To go further…

- What happens if the lower-level is not convex, or does not satisfy constraint qualification?
- What happens if the optimal reaction of the follower is not unique?
- How would you generalize a Stackelberg game with one leader and multiple followers?
- Is it realistic to assume that the leader perfectly knows the reaction of the follower(s)?
- **Prepare for Wednesday: Can you think of other examples of Stackelberg games in power systems?**

Discussion: Examples of Stackelberg Games in Power Systems and Markets?

Stackelberg Game Model

- 2-stage dynamic game



- Has a strategic advantage (plays first)
- Action **influences** optimal reaction of follower
- Tries to **anticipate** the follower's reaction

- Finds optimal reaction to leader's action
- Reaction **influences** leader's profit

Question: How can the leader anticipate the follower's reaction?

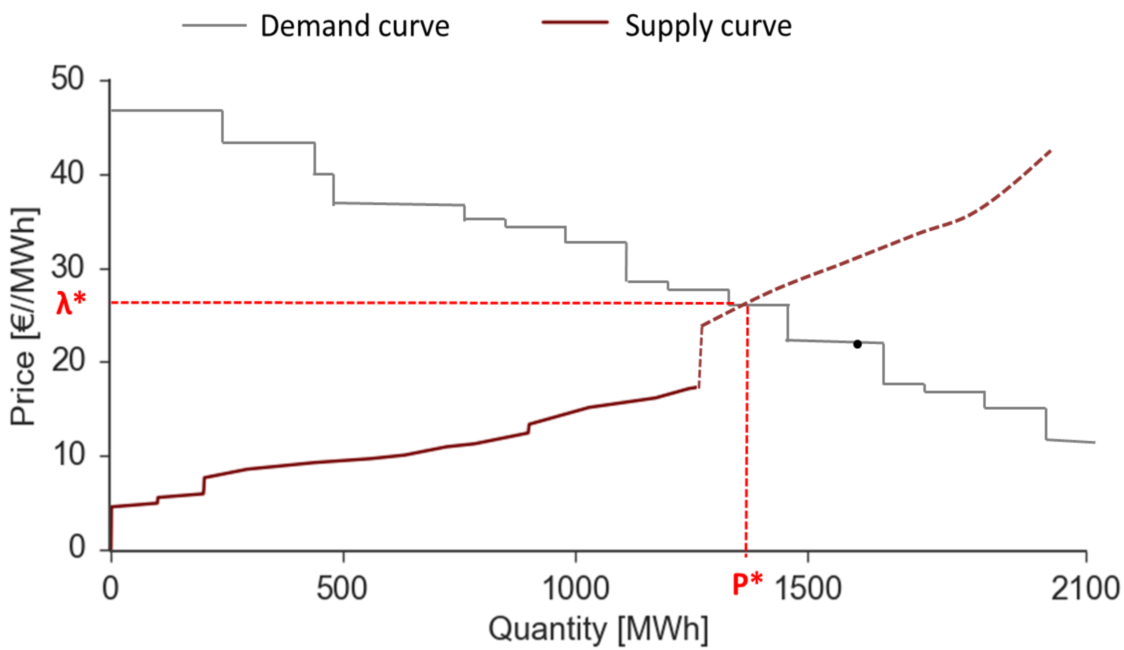
Clue: remember the strategic bidder problem?

How to recognize and formulate a Stackelberg game:

1. Who is the leader / the follower?
2. How does the action of the leader influence the reaction of the follower?
3. What is the optimal reaction of the follower to the leader's action?
4. How does the leader anticipate the reaction of the follower?

Clue: remember the strategic bidder problem?

- A large producer participating in the day-ahead market
- Can exercise ”market-power” to increase its profit: i.e. its **action influences** market equilibrium (spot price)



- **Question: Is this an example of a Stackelberg game?**

Learning Objectives

At the end of these 2 sessions you should be able to…

- Define Stackelberg games
- Formulate Stackelberg games as bilevel programs and Mathematical Problems with Equilibrium Constraints (MPECs)
- Solve MPECs using 3 solution methods
- Recognize Stackelberg games in real life problems related to power systems