ECON 549: Bi-level Programming

Lecture

G Cornelis van Kooten

Thanks to Linda Wong for help with these slides

Bi-level Programming

- Partitions the control over the decision variables between two hierarchical levels

- The decision maker at each level attempts to optimize her individual objective, which usually depends in part on the variables controlled by the decision maker at the other level

Bi-level Programming

- The interaction between the two hierarchical levels reflects a decentralized decision making situation where the higher level (leader/principal) can only influence rather than dictate the choices of the lower level (follower/agent)

- Applicability is constrained by its computational difficulties:
  - nonconvexity,
  - NP-hardness (NP = non-deterministically polynomial),
  - inefficiency of algorithms

Example from Game Theory: Stackelberg

- Suppose firm 1 can fix production levels first and all others will react on this quantity
- Formulated as a BLPP, firm 1 solves profit maximization problem:

$$\max_{x_1} x_1 p\left(\sum_{j=1}^{n} x_j\right) - f_1(x_1)$$

$$s.t.\ x_i \in \operatorname*{argmax}_{x_i} x_i p\left(\sum_{j=1}^{n} x_j\right) - f_i(x_i)$$

Two Modeling Approaches

Optimistic:

- Results from cooperative behaviour of the follower
- If the follower’s reaction set is not a singleton, the leader is allowed to choose the most suitable element from the follower’s feasible set

Pessimistic:

- Results from aggressive non-cooperative follower behaviour
- The leader cannot decide which of the best responses is implemented by follower, and instead chooses a decision that performs best in view of the worst follower response

Optimistic:

$$
\begin{aligned}
&\min_{x\in X, y\in Y} F(x,y) \\
&s.t.\ \ G(x,y) \le 0 \\
&\qquad y = \arg\min_{y'\in Y} f(x,y') \\
&\qquad s.t\ \ g(x,y') \le 0
\end{aligned}
$$

Pessimistic:

$$
\begin{aligned}
&\min_{x\in X} \max_{y\in Z} F(x,y) \\
&s.t.\ \ G(x,y) \le 0 \\
&\qquad Z = \arg\min_{y'\in Y} f(x,y') \\
&\qquad s.t\ \ g(x,y') \le 0
\end{aligned}
$$

- Variables are divided into two classes: upper-level variables *x* and lower-level variables *y*
- *F* and *f* are the upper-level and lower-level objective functions
- *G* and *g* are the upper-level and lower-level constraints

Application Issues

- Even the ‘simplest instance’, the linear BLPP, was shown to be NP-hard – worst case requires exhaustive search

- The optimality conditions proposed generally have few practical uses and the numerical algorithms do not have convenient stopping criteria

Application Issues

- Many times, the solution of a one-level optimization problem is considered superior because solving a BLPP to global optimality is not desirable (due to the time it takes) or not possible due to NP-hardness

Solution Algorithms

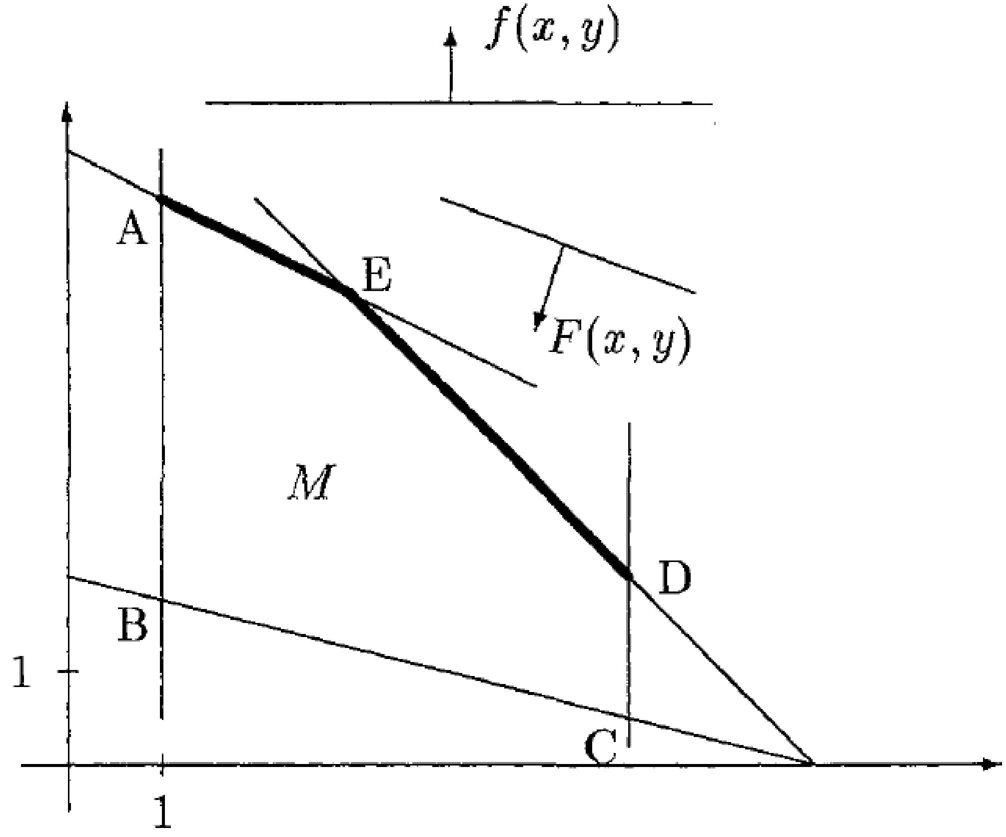
- Reformulate BLPP into ordinary one-level optimization problems
- General approaches:
  1. Apply implicit function theorems to derive a local description of the follower's response function, *y*(*x*), and insert into the original problem:

$$\min_{x}\{F(x, y(x)) : G(x, y(x)) \leq 0, y(x) \in \Psi(x)\}$$

     – Requires uniqueness and continuity of the optimal solution to lower level problem

Numerical Example

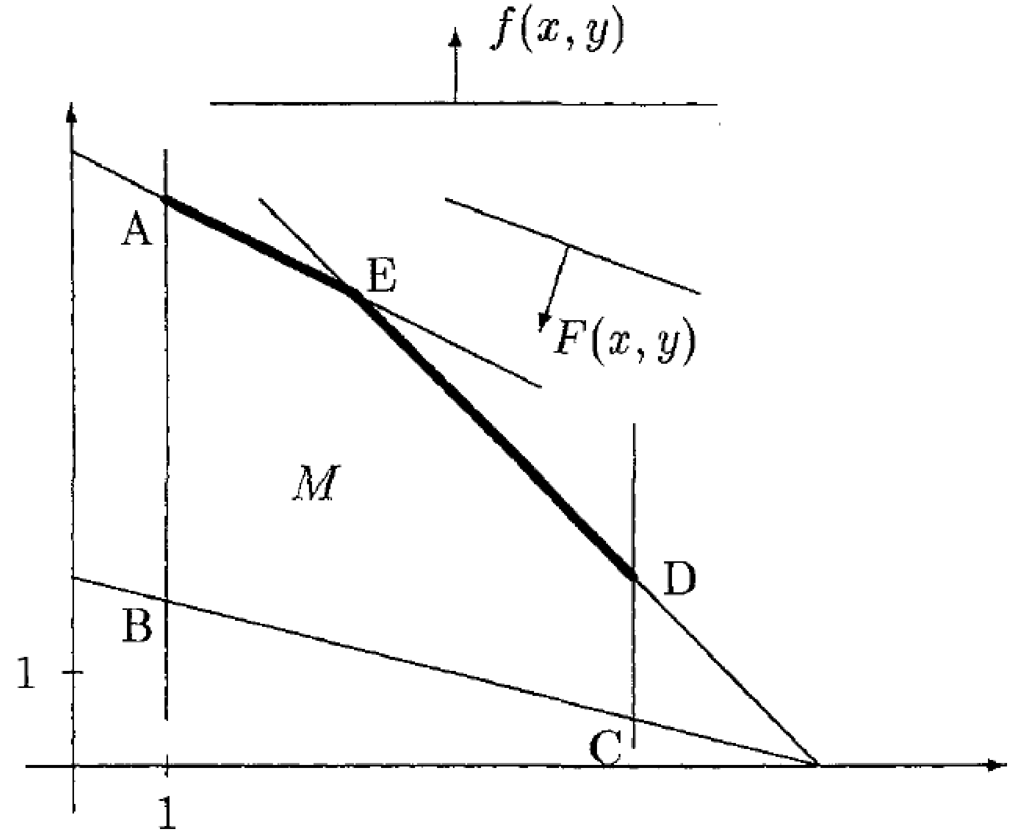
$$\min_{x} x + 3y$$
$$s.t. \, 1 \le x \le 6,$$
$$\min_{y} \; -y$$
$$s.t. \quad x + y \le 8,$$
$$x + 4y \ge 8,$$
$$x + 2y \le 13$$



- M is the feasible set
- For all feasible *x*, the optimal solutions for the lower level problem are indicated by the thick line

Numerical Example (cont)

$$y(x) = \begin{cases} 6.5 - 0.5x & for\ 1 \le x \le 3 \\ 8 - x & for\ 3 \le x \le 6 \end{cases}$$

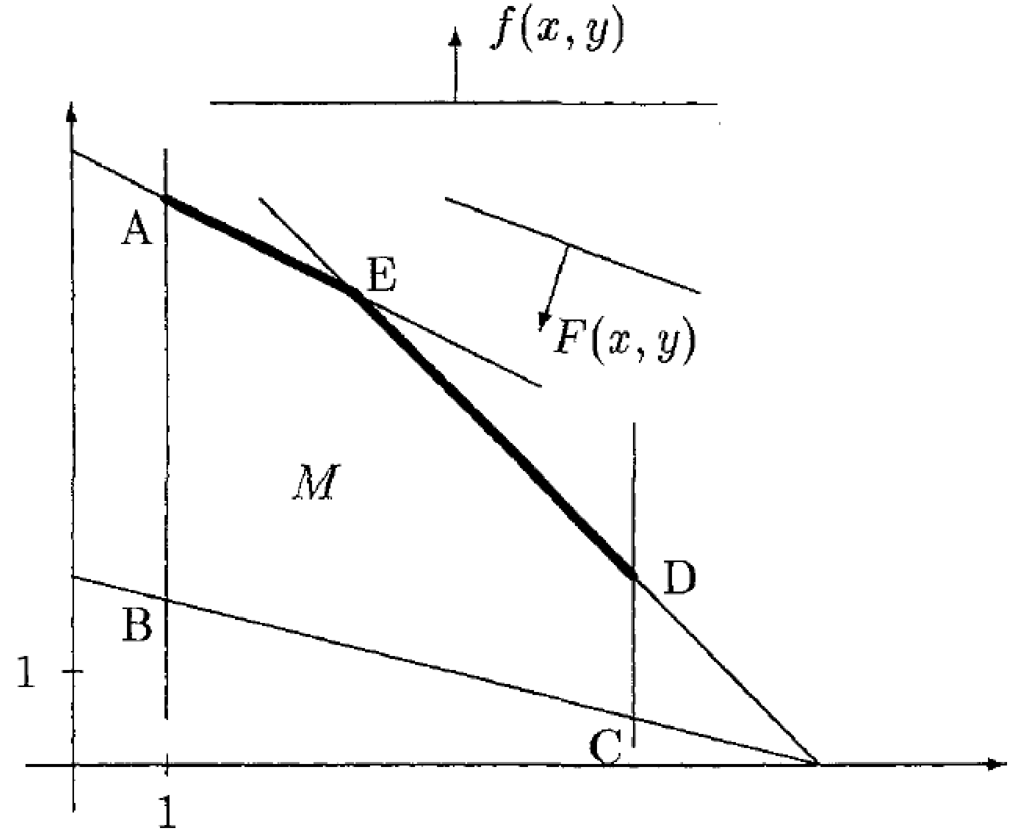


Numerical Example (cont)

$$y(x) = \begin{cases} 6.5 - 0.5x & for\ 1 \le x \le 3 \\ 8 - x & for\ 3 \le x \le 6 \end{cases}$$

$$\Downarrow$$

$$F(x, y(x)) = \begin{cases} 19.5 - 0.5x & for\ 1 \le x \le 3 \\ 24 - 2x & for\ 3 \le x \le 6 \end{cases}$$



Numerical Example (cont)

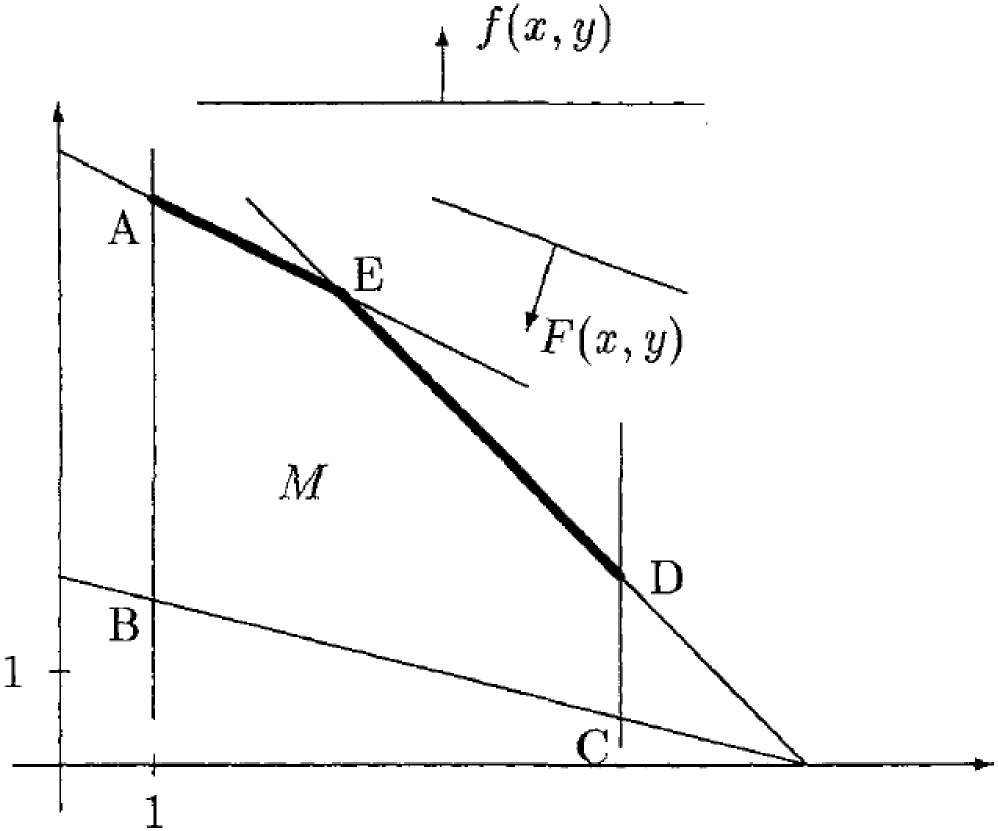
$$y(x) = \begin{cases} 6.5 - 0.5x & for\ 1 \le x \le 3 \\ 8 - x & for\ 3 \le x \le 6 \end{cases}$$

$$\Downarrow$$

$$F(x, y(x)) = \begin{cases} 19.5 - 0.5x & for\ 1 \le x \le 3 \\ 24 - 2x & for\ 3 \le x \le 6 \end{cases}$$

$$\Downarrow$$

$$\min_x \{F(x, y(x)) : 1 \le x \le 6\}$$



Solution Algorithms

2. Replace lower level problem by its Karush-Kuhn-Tucker (KKT) conditions
   – Results in a typical one-level MP which is often called Mathematical Programming with Equilibrium Constraints (MPEC)
   – Only applicable to the optimistic approach; there is no efficient way to use it in the pessimistic one
   – Note: even under certain regularity assumptions, the resulting problem is in general not equivalent to the original bi-level problem – this reformulation introduces new (dual/lagrange) variables, making it difficult to find a solution point of the KKT reformulation that solves the original BLPP

$$
\min_{x \in X, y, \lambda} F(x, y)
$$

$$
s.t. \begin{cases} G(x, y) \le 0, \quad g(x, y) \le 0 \\ \lambda \ge 0, \quad \lambda' g(x, y) = 0 \\ \nabla_y f(x, y) + \lambda' \nabla_y g(x, y) = 0 \end{cases}
$$

- Even under suitable convexity assumptions on the functions *F*, *G* and set *x*, this MP is not easy to solve due to the non-convexities that occur in the complementarity and Lagrangean constraints
- To solve, it is best to use enumeration algorithms, such as branch-and-bound, which involve decomposing the feasible region into a number of ‘branches’, each of which has the format of a standard NLP feasible set

- Cautionary note:

  available algorithms only focus on finding stationary points, which may or may not be global optimal

Numerical Example (again)

$$\min_{x} x + 3y$$

$$s.t. \; 1 \le x \le 6,$$

$$\begin{aligned} \min_{y} \; & -y \\ s.t. \; & x + y \le 8, \\ & x + 4y \ge 8, \\ & x + 2y \le 13 \end{aligned}$$

KKT Conditions:

$$-1 + \lambda_1 - 4\lambda_2 + 2\lambda_3 = 0$$

$$g_1(x, y) = x + y - 8 \le 0$$

$$g_2(x, y) = 8 - x - 4y \le 0$$

$$g_3(x, y) = x + 2y - 13 \le 0$$

$$\lambda_i \ge 0, \; \lambda_i g_i(x, y) = 0, i = 1,2,3.$$

Solution Algorithms

3. Optimal value reformulation: replace the lower level solution set by its description via the optimal value function
   – Equivalent to the initial problem
   – However, the optimal value function is typically non-smooth, even when the functions involved are affine linear ones

$$\min_{x \in X, y} F(x,y)$$

$$s.t. \begin{cases} G(x,y) \leq 0, \\ g(x,y) \leq 0, \\ f(x,y) - \varphi(x) \leq 0 \end{cases}$$

where

$$\varphi(x) := \min_{y} \{ f(x,y) : g(x,y) \leq 0 \}$$

- For the optimal value function to be well-defined, the lower level problem must admit an optimal solution for each possible *x*

General Bilevel Programming Problem (Bard 1998)

$$\min_{x \in X} F(x, y)$$

$$s.t. \quad G(x, y) \leq 0$$

$$\min_{x \in X} \quad f(x, y)$$

$$s.t. \quad g(x, y) \leq 0$$

$$x, y \geq 0$$

Linear Bilevel Programming Problem

$$\begin{aligned}
&\underset{x \in X}{Min} \; F(x, y) = c_1 x + d_1 y \\
&s.t. \quad A_1 x + B_1 y \le b_1 \\
&\qquad \underset{x \in X}{Min} \; f(x, y) = c_2 x + d_2 y \\
&\qquad s.t. \quad A_2 x + B_2 y \le b_2 \\
&\qquad\qquad x, y \ge 0
\end{aligned}$$

Example

$$\min_{x \geq 0} \quad F(x,y) = x - 4y$$

$$\text{s.t.}$$

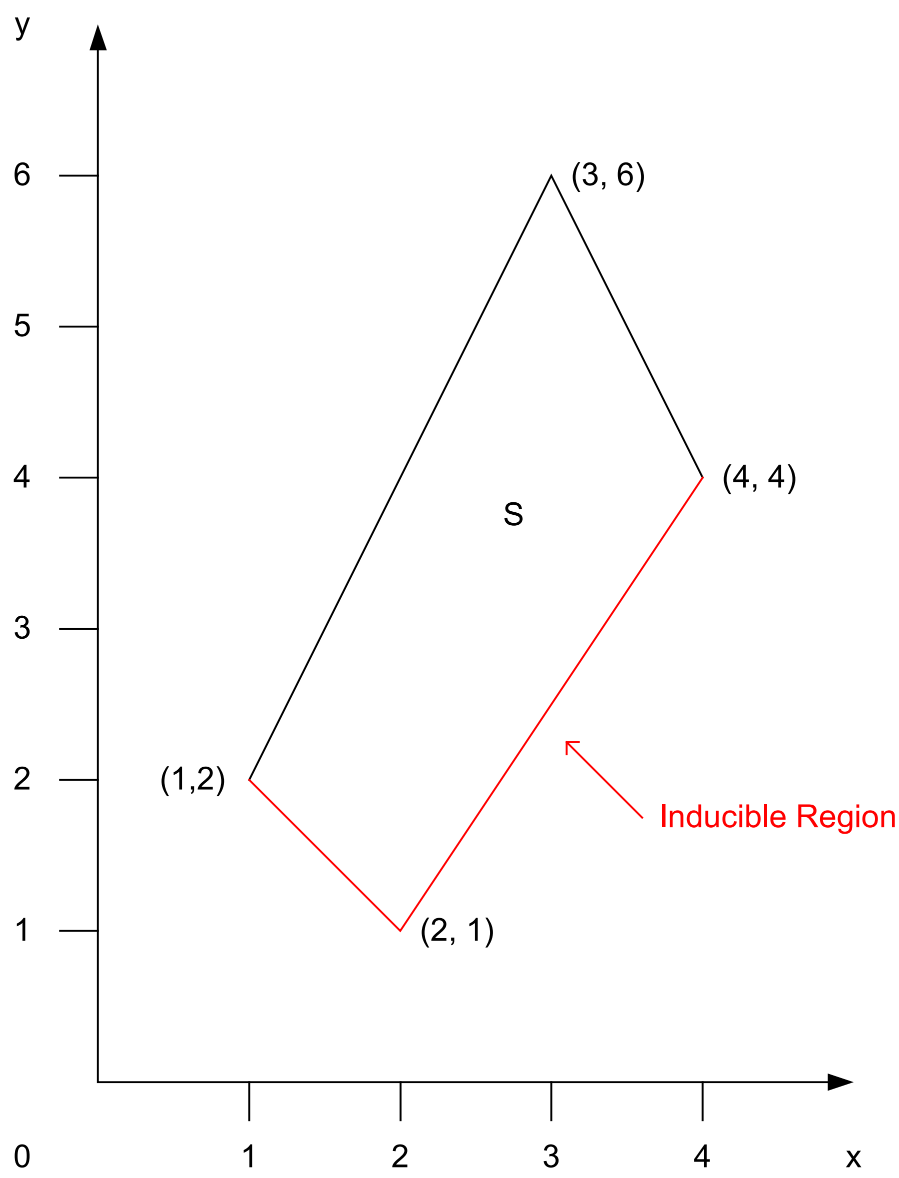
$$\min_{y \geq 0} \qquad f(y) = y$$

$$\text{s.t.}$$

$$\begin{aligned} -x - y &\leq -3 \\ -2x + y &\leq 0 \\ 2x + y &\leq 12 \\ -3x + 2y &\leq -4 \end{aligned}$$

$$\boxed{F(x, y) = x - 4y}$$

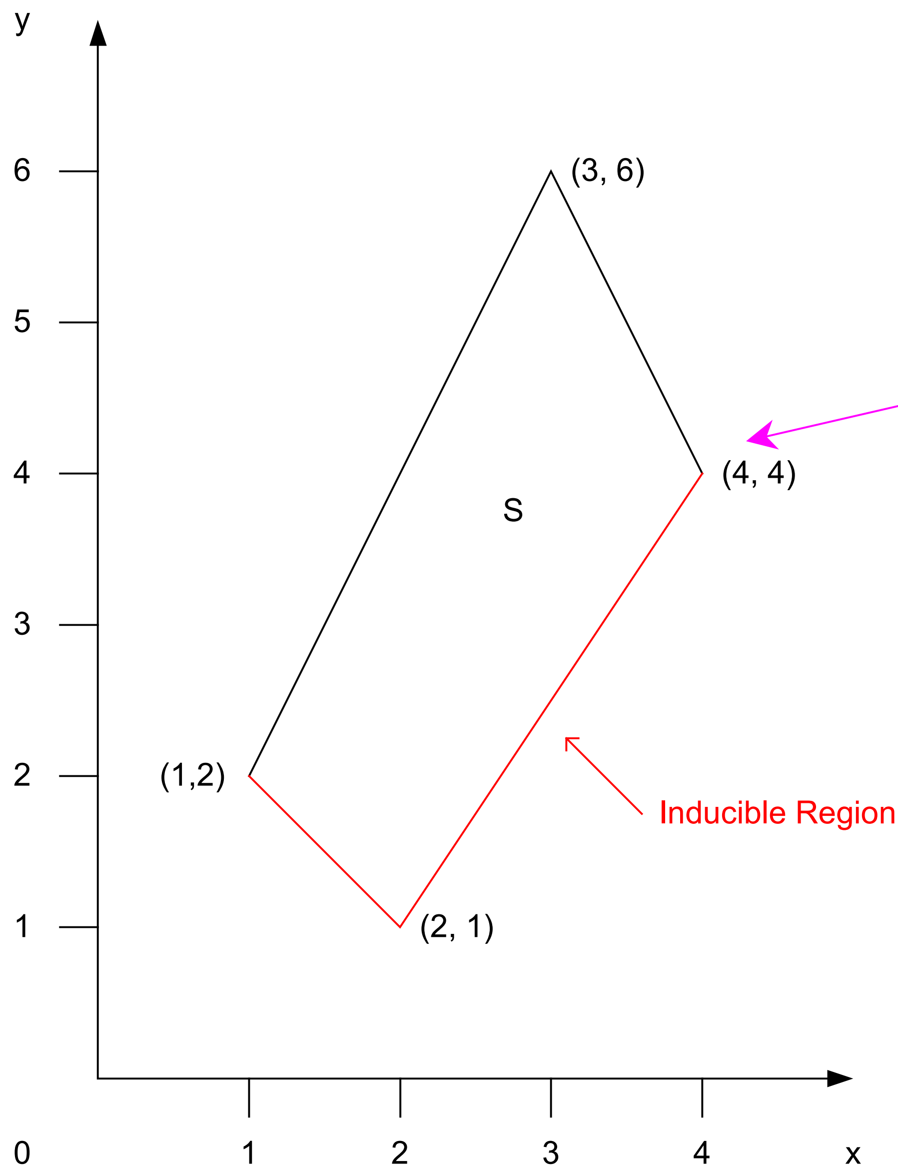
$$\boxed{f(y) = y}$$



Non-convex in general

$$\boxed{F(x, y) = x - 4y}$$

$$\boxed{f(y) = y}$$



Optimal solution

$$x^* = 4$$

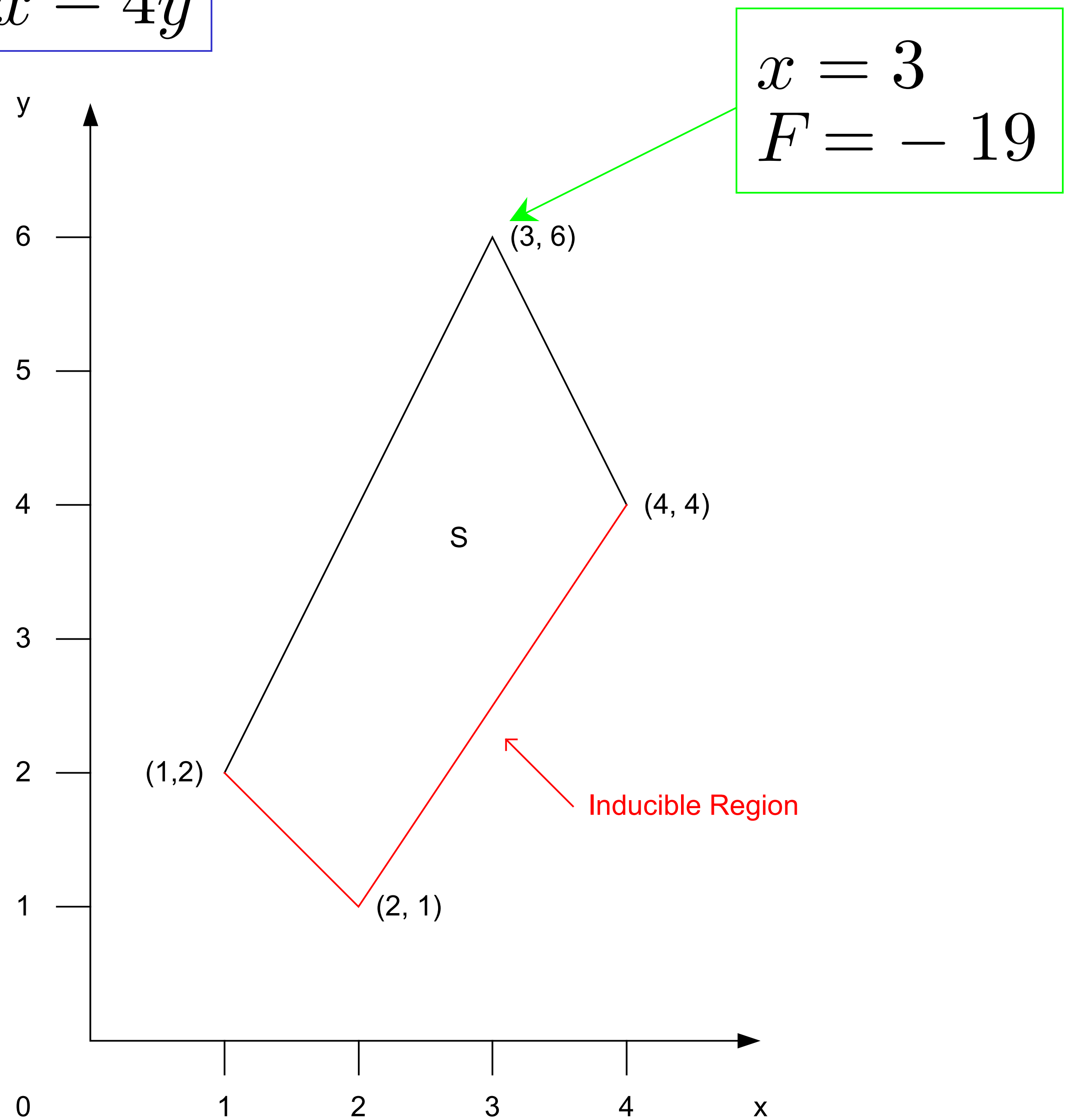
$$y^* = 4$$

$$F^* = -12$$

$$f^* = 4$$

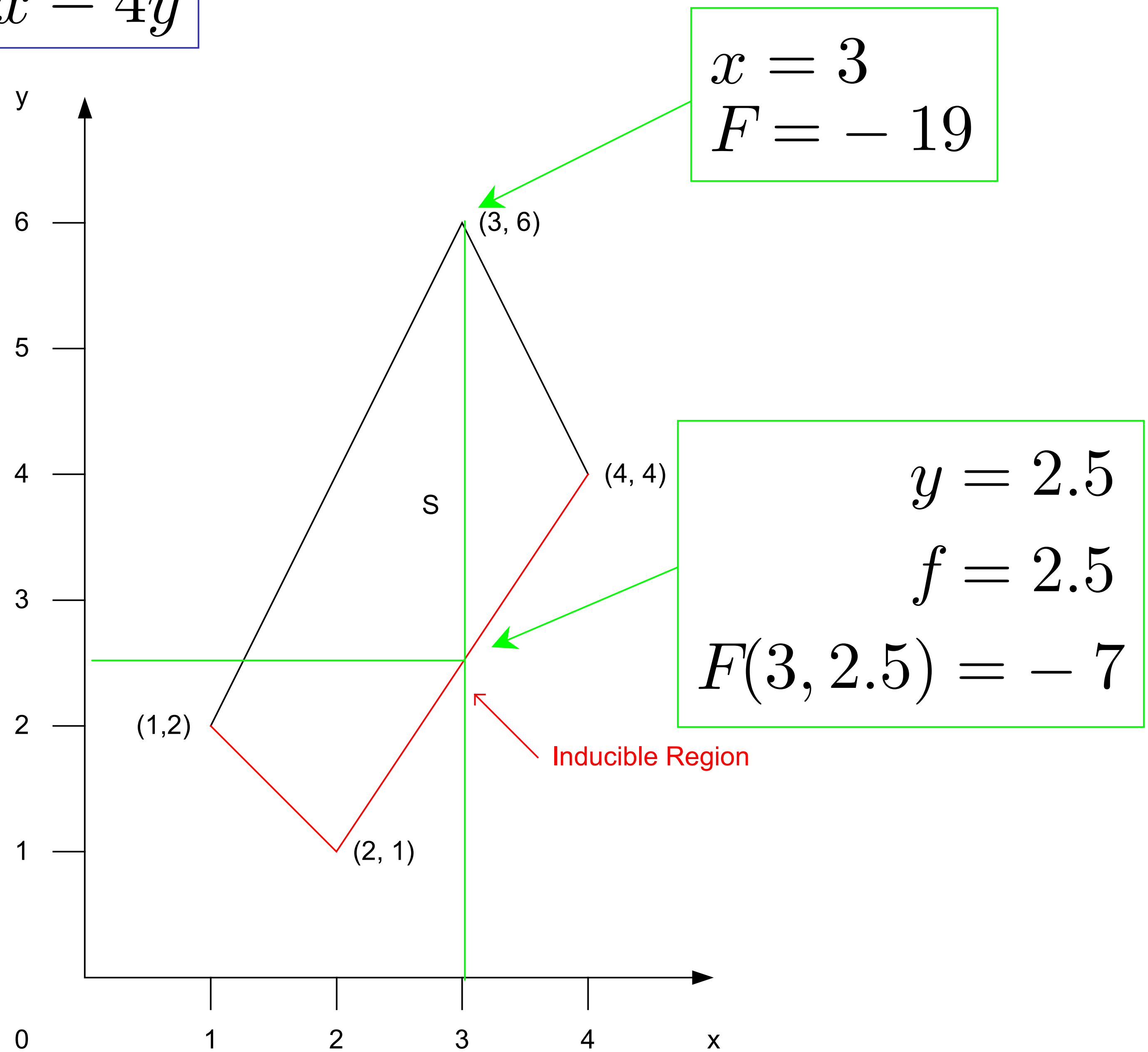
$$F(x, y) = x - 4y$$

$$f(y) = y$$



$$F(x, y) = x - 4y$$

$$f(y) = y$$

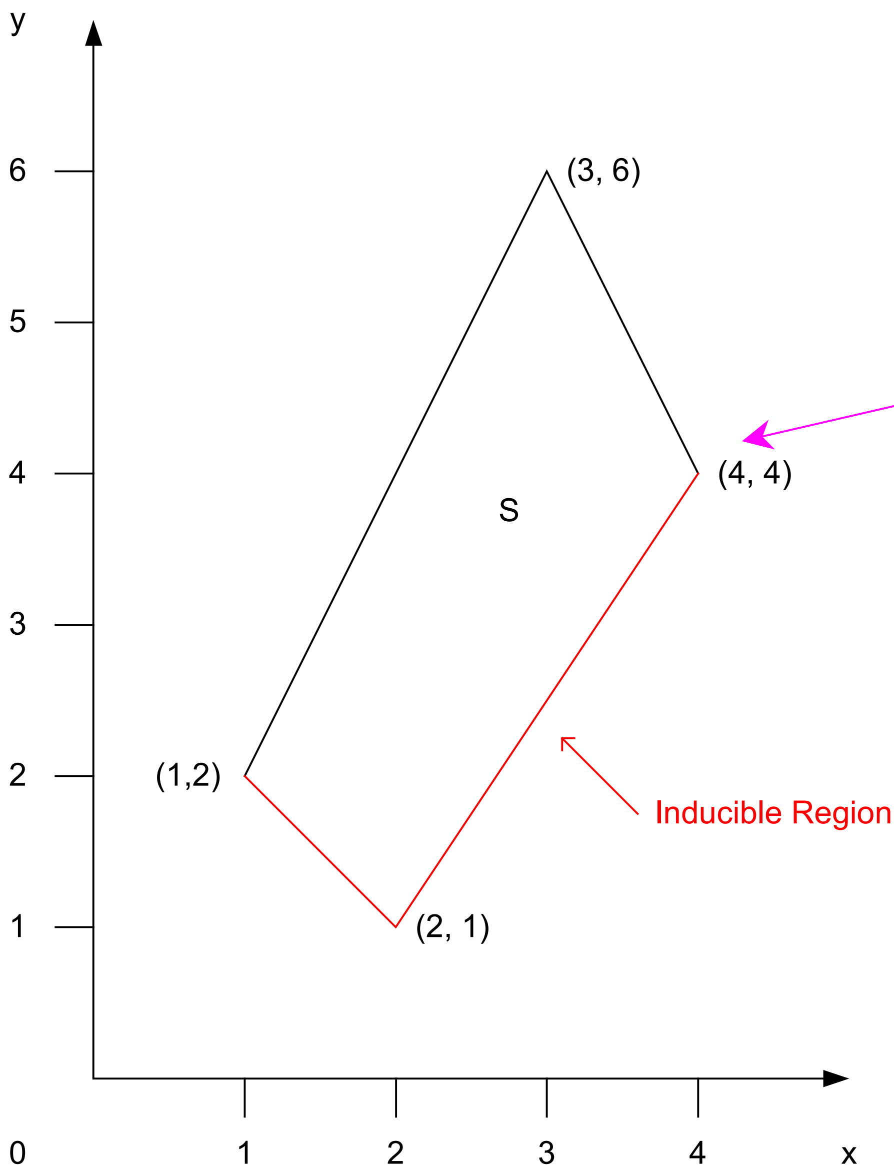

y
6
5
4
3
2
1
0
1
2
3
4
x
(3, 6)
(4, 4)
(1,2)
(2, 1)
S
Inducible Region
x = 3
F = − 19
y = 2.5
f = 2.5
F(3, 2.5) = − 7

Pareto Optimality

- Multiple objective problem.
- Feasible solution B dominates feasible solution A:
  - B at least as good as A w.r.t. every objective,
  - B strictly better than A w.r.t. at least one objective.
- Pareto optimal solutions:
  - Set of all non-dominated feasible solutions.

$$\boxed{F(x, y) = x - 4y}$$

$$\boxed{f(y) = y}$$



Optimal solution

$$x^* = 4$$

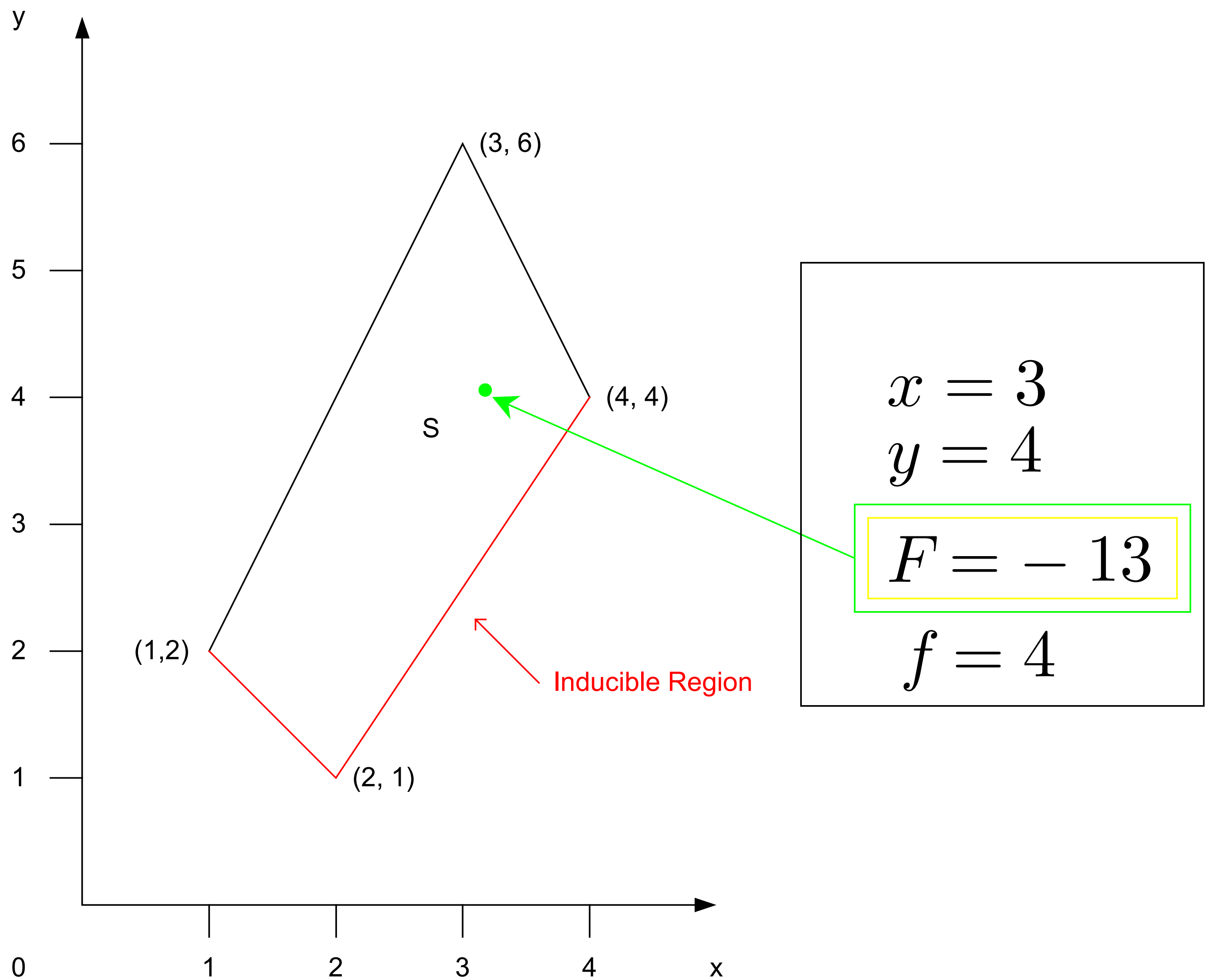
$$y^* = 4$$

$$F^* = -12$$

$$f^* = 4$$

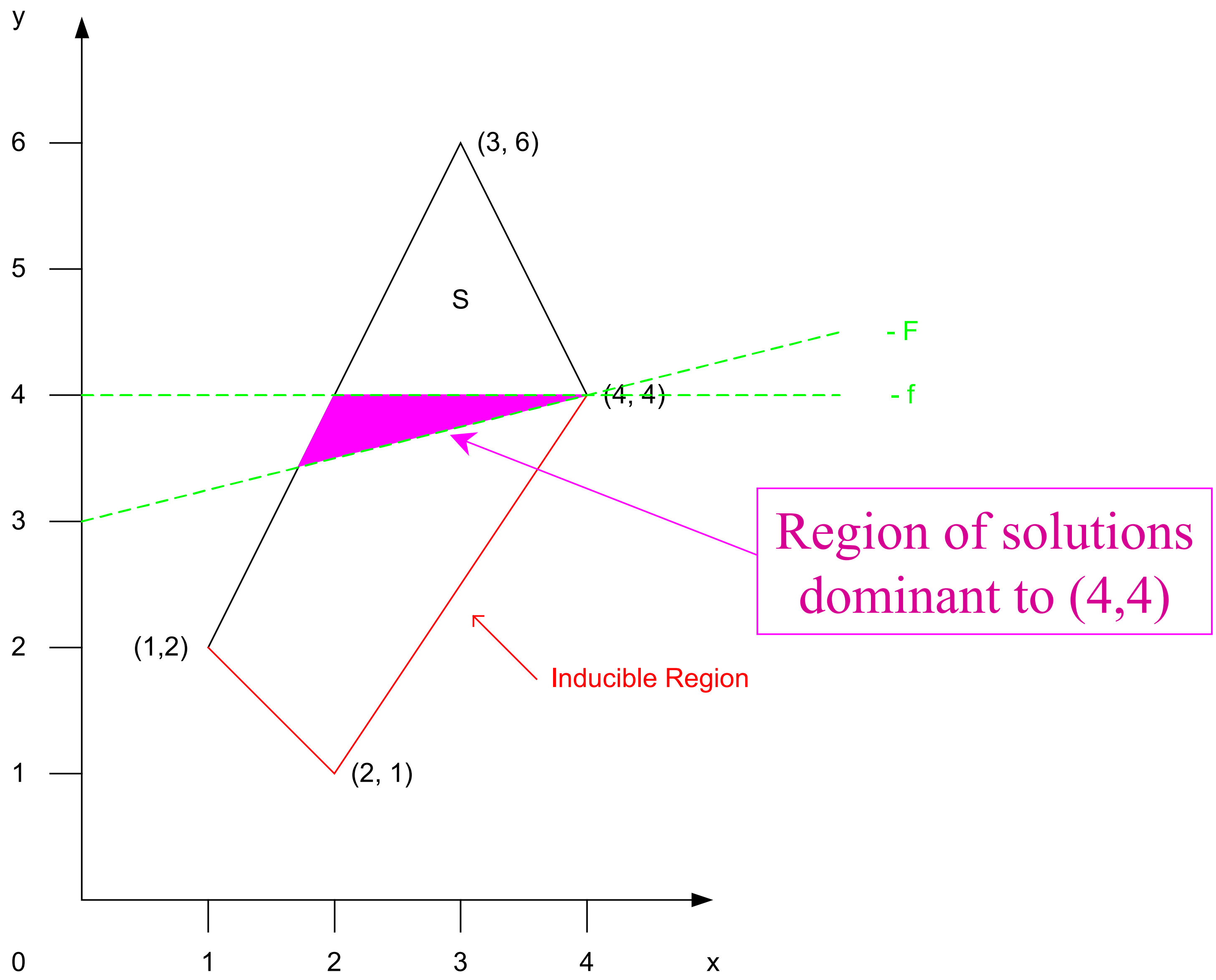
$$\boxed{F(x, y) = x - 4y}$$

$$\boxed{f(y) = y}$$



$$F(x, y) = x - 4y$$

$$f(y) = y$$



y
6
5
4
3
2
1
0
1
2
3
4
x
(3, 6)
S
(4, 4)
(1,2)
(2, 1)
- F
- f
Inducible Region
Region of solutions dominant to (4,4)

Summary of Properties of Bilevel Programs

- No guarantee of solution.
- Order in which decisions are made is important.
- No guarantee of Pareto optimality.
- Non-convex optimization problem.
- All of the above can apply even when all functions are continuous and bounded.

GAMS

Hierarchical Models

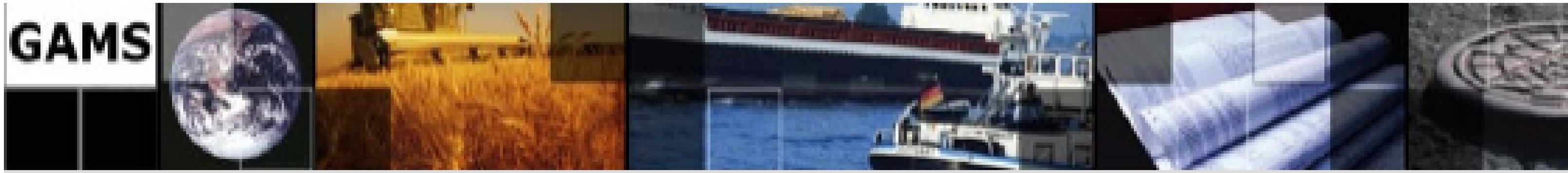
- Bilevel Program:

$$
\begin{array}{ll}
\min\limits_{x,y} & f(x,y) \\
\text{s.t.} & g(x,y) \leq 0, \\
 & y \text{ solves } \min\limits_{s} v(x,s) \text{ s.t. } h(x,s) \leq 0
\end{array}
$$

- Additional Information:

```
$onecho > %emp.info%
Bilevel x min v h
$offecho
```

- EMP Tool automatically creates an MPEC by expressing the lower level optimization problem through its optimality conditions



Bilevel Model

Conejo A J, Castillo E, Minguez R, and Garcia-Bertrand R; Decomposition Techniques in Mathematical Programming, Springer,Berlin, 2006.

```
variables z,x1,x2,x3,x4,h1,h2,u1,u2,u3,u4,v1,v2,v3,v4;
equations defobj,defh1,defh2,a1,e1,e2;

defobj.. z =e= sqr(x1+x2-2) + sqr(x3+x4-2);
a1.. x1-x2 =e= 3;
                                                    Outer Problem

defh1.. h1 =e= sqr(u1-x1) + sqr(u2-x2) + sqr(u3-x3) + sqr(u4-x4);
e1.. 3*u1 +  u2 + 2*u3 + u4 =e= 6;
                                                    Inner Problem 1

defh2.. h2 =e= sqr(v1-x1) + sqr(v2-x2) + sqr(v3-x3) + sqr(v4-x4);
e2.. v1 +  v2+ v3 + 2*v4 =e= 7;
                                                    Inner Problem 2

model bilevel / all /
```

Application to Energy Markets

- BLP can be used to assess the relative benefits of partial or total deregulation

- For example, model can be applied to situations of pure competition, regulated equilibrium, or monopoly equilibrium and outcomes for these situations can be compared

Application to Energy Markets

Model description:

$$
\begin{aligned}
P:\ & \min_{x,p} c_1 x - ps \\
& s.t.\ A_1 x - s = b_1 \\
& \quad x \in P_1 \\
& \quad s \geq 0
\end{aligned}
$$

x: decision variables of the producer (investments, activity levels, etc);
c_1: vector of unit costs;
s: vector of commodity exchanges between supplier and consumers;
p: price vector;
A_1: techno-economic matrix relating the utility's decision variables to production levels;
P_1: polyhedron defined by any remaining constraints.

$$
\begin{aligned}
C:\ & \min_{y,s} c_2 y - ps \\
& s.t.\ A_2 y + s = b_2 \\
& \quad y \in P_2 \\
& \quad s \geq 0
\end{aligned}
$$

y: consumer decision variables (investments, capacities, etc);
c_2: costs associated to the consumer variables;
A_2: techno-economic matrix relating consumer decision variables to consumption;
P_2: polyhedron defined by any remaining constraints.

Application to Energy Markets (cont)

- The shadow prices of the first constraints of each problem define the implicit inverse supply and demand functions and vector s makes the connection between supply and demand
- Equilibrium (p^*, s^*) is achieved when, for a given s^*, the producer sets its activity vector to x^* and the price vector to p^* while, for given p^*, consumers set their activity and exchange vectors to y^* and s^*

Application to Energy Markets (cont)

This framework encompasses a large class of equilibria, but if *p* is constrained to belong to a given set *S*, we derive a so-called *S*-equilibrium, which solves a bi-level program

$$
\begin{aligned}
SEQ:\ & \min_{x,p} c_1 x - ps \\
& s.t.\ A_1 x - s = b_1 \\
& \qquad x \in P_1 \\
& \qquad p \in S \\
& \qquad \min_{y,s} c_2 y - ps \\
& \qquad s.t.\ A_2 y + s = 0 \\
& \qquad\qquad y \in P_2 \\
& \qquad\qquad s \geq 0
\end{aligned}
$$

Principal-Agent Problem in Salvage Harvesting Pine Timber

Principal's problem:

$$\underset{\tau_{s,t},\, H_{a,t},\, \sigma_a}{\text{Maximize}} \; W = \sum_{t=1}^{T} \delta^{t} \left(\sum_{s \in \{P, DP, NP\}} \tau_s \sum_{a \in \{R, N\}} h_{a,s,t} \right) - \delta^{T+1} w \sum_{t=1}^{T} I_{DP,t} \delta^{T+1} + \sum_{s \in \{P, NP\}} \left(v_{s,T+1} - c \right) I_{s,T+1}$$

s.t.

$$I_{P,t+1} = (1 - \beta_t)\, I_{s,t} - \sum_a h_{a,P,t}\;, \forall\, t = 1, \ldots, T;\; a \in \{R, N\} \qquad \text{(Pine inventory)}$$

$$I_{DP,t+1} = (1 - \xi)\, I_{DP,t} + \beta_t I_{P,t} - \sum_a h_{a,DP,t}\;, \forall\, t = 1, \ldots, T;\; a \in \{R, N\} \qquad \text{(Dead pine inventory)}$$

$$I_{NP,t+1} = I_{NP,t} - \sum_a h_{a,NP,t}\;, \forall\, t = 1, \ldots, T;\; a \in \{R, N\} \qquad \text{(Non-pine inventory)}$$

$$I_{s,0} = \overline{I}_s,\; \forall\, s \in \{P, DP, NP\} \qquad \text{(Starting inventory)}$$

Pine harvest constraint that applies to both agents:

$$\sum_{a\in\{R,N\}} (h_{a,P,t} + h_{a,DP,t}) \geq \sigma \sum_{s} h_{a,s,t} \; ; \forall\, t = 1, \ldots, T \quad \text{(Pine harvest constraint)}$$

Replaceable quota-based timber agent

Minimize $\text{Cost} = \sum_{t=1}^{T} \delta^t \sum_{s} (c + \tau_s)\, h_{R,s,t}$

Subject to:

R:

$$\sum_{s} h_{R,s,t} = H_{R,t}, \forall\, t = 1, \ldots, T \qquad \text{(Annual harvest)}$$

$$\sum_{s} (v_{s,t} - c - \tau_s) h_{R,s,t} \geq 0, \forall\, t = 1, \ldots, T \qquad \text{(Positive annual revenue)}$$

Non-replaceable quota-based timber agent

Maximize $\pi = \sum_{t=1}^{T} \delta^t \sum_{s} (v_{s,t} - c - \tau_s)\, h_{N,s,t}$

Subject to

N:

$$\sum_{s} h_{N,s,t} \leq H_{N,t}\,, \forall\, t = 1, \ldots, T \qquad \text{(Annual harvest)}$$

$$h_{N,DP,t} \geq \mu \sum_{s} h_{N,s,t}\,, \forall\, t = 1, \ldots, T \qquad \text{(Dead pine harvest requirement)}$$