BDC5101

Deterministic Operations Research Models

Zhenyu Hu

Semester II, 2018/2019

Software Implementation

Wyndor Glass Co.

TABLE 3.1 Data for the Wyndor Glass Co. problem

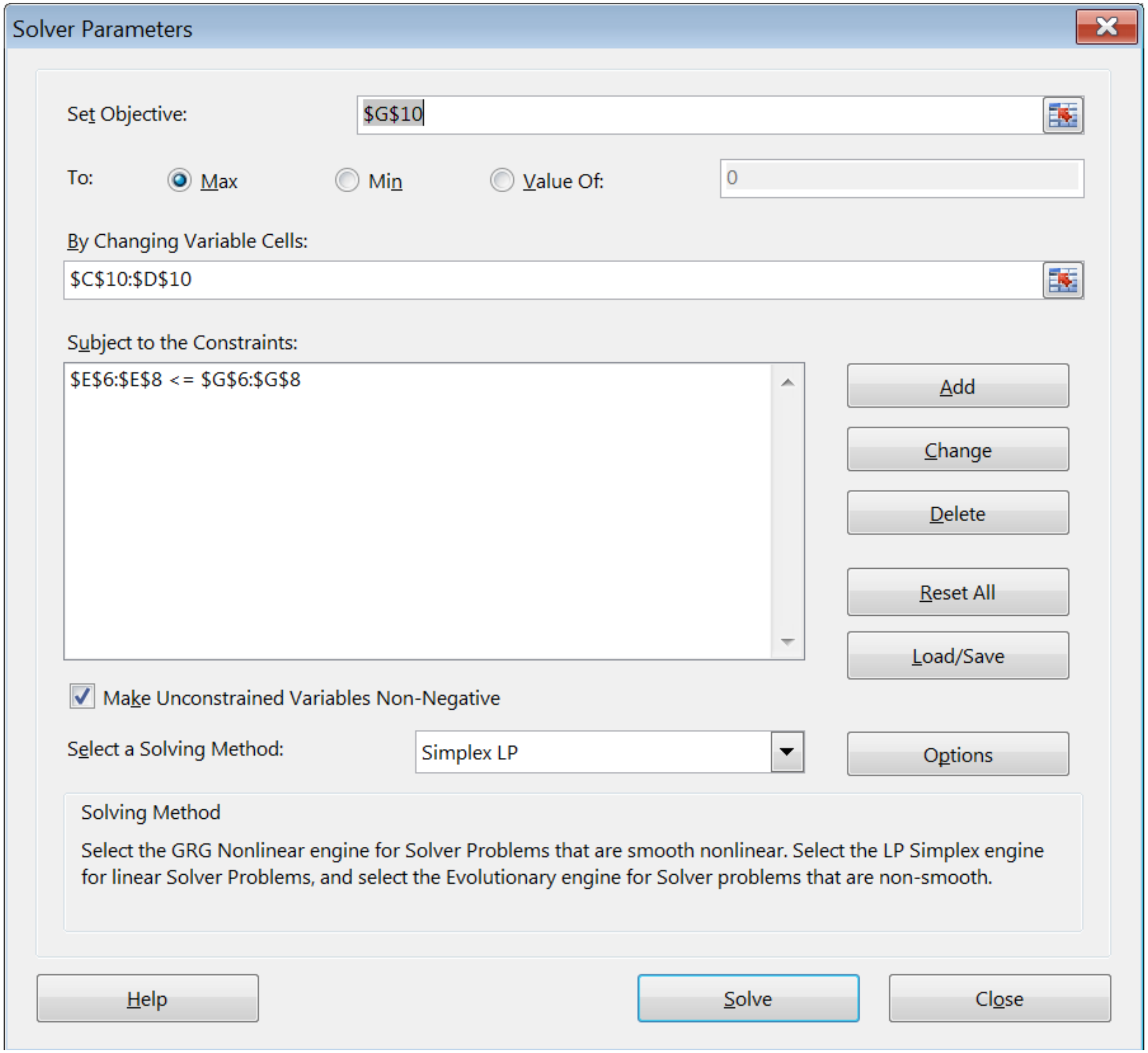
	Production Time per Batch, Hours		
	Product		
Plant	**1**	**2**	**Production Time Available per Week, Hours**
1	1	0	4
2	0	2	12
3	3	2	18
Profit per batch	\$3,000	\$5,000	

$$max\ 3x_1 + 5x_2$$
$$x_1 \leq 4,$$
$$2x_2 \leq 12,$$
$$3x_1 + 2x_2 \leq 18,$$
$$x_1 \geq 0, x_2 \geq 0$$

Model in Solver

	A	B	C	D	E	F	G	H
1			Doors	Windows				
2		Profit per batch ($1,000)	3	5				
3								
4								
5			Hours Used per Batch Produced		Hours Used		Hours Available	
6		Plant 1	1	0	0	<=	4	
7		Plant 2	0	2	0	<=	12	
8		Plant 3	3	2	0	<=	18	
9								
10		Batches Produced	0	0		Total Profit	0	
11								

Model in Solver



Model in Gurobi

```python
from gurobipy import *

m = Model("Wyndor")

# Creat variables
# addVar(lb=0.0, ub=GRB.INFINITY, obj=0.0, vtype=GRB.CONTINUOUS, name="", column=None)
x1 = m.addVar(name = "x1")
x2 = m.addVar(name = "x2")

# Set objective
m.setObjective(3*x1 + 5*x2, GRB.MAXIMIZE)

# Add constraint:
m.addConstr(x1 <= 4, "Plant1")

m.addConstr(2*x2 <= 12, "Plant2")

m.addConstr(3*x1 + 2*x2 <= 18, "Plant3")
```

Solving the Model and Display Output

- **Solving the model**

```
# Solving the model
m.optimize()
```

- **Display solution and optimal value**

```
#  Print optimal solutions and optimal value
for v in m.getVars():
    print(v.VarName, v.x)

print('Obj:', m.objVal)
```

Separating Model and Data: Model

```
########Model Set-up##############

m = Model("Wyndor")

# Creat variables
# addVars ( *indices, lb=0.0, ub=GRB.INFINITY, obj=0.0, vtype=GRB.CONTINUOUS, name="" )

x = m.addVars(N, name = "x")

# Set objective
m.setObjective( quicksum(profit[i]*x[i] for i in range(N)), GRB.MAXIMIZE)

# Add constraints:
m.addConstrs(( quicksum(rate[i,j]*x[j] for j in range(N)) <= capacity[i] for i in range(M) ), "Plant")
```

Separating Model and Data: Data

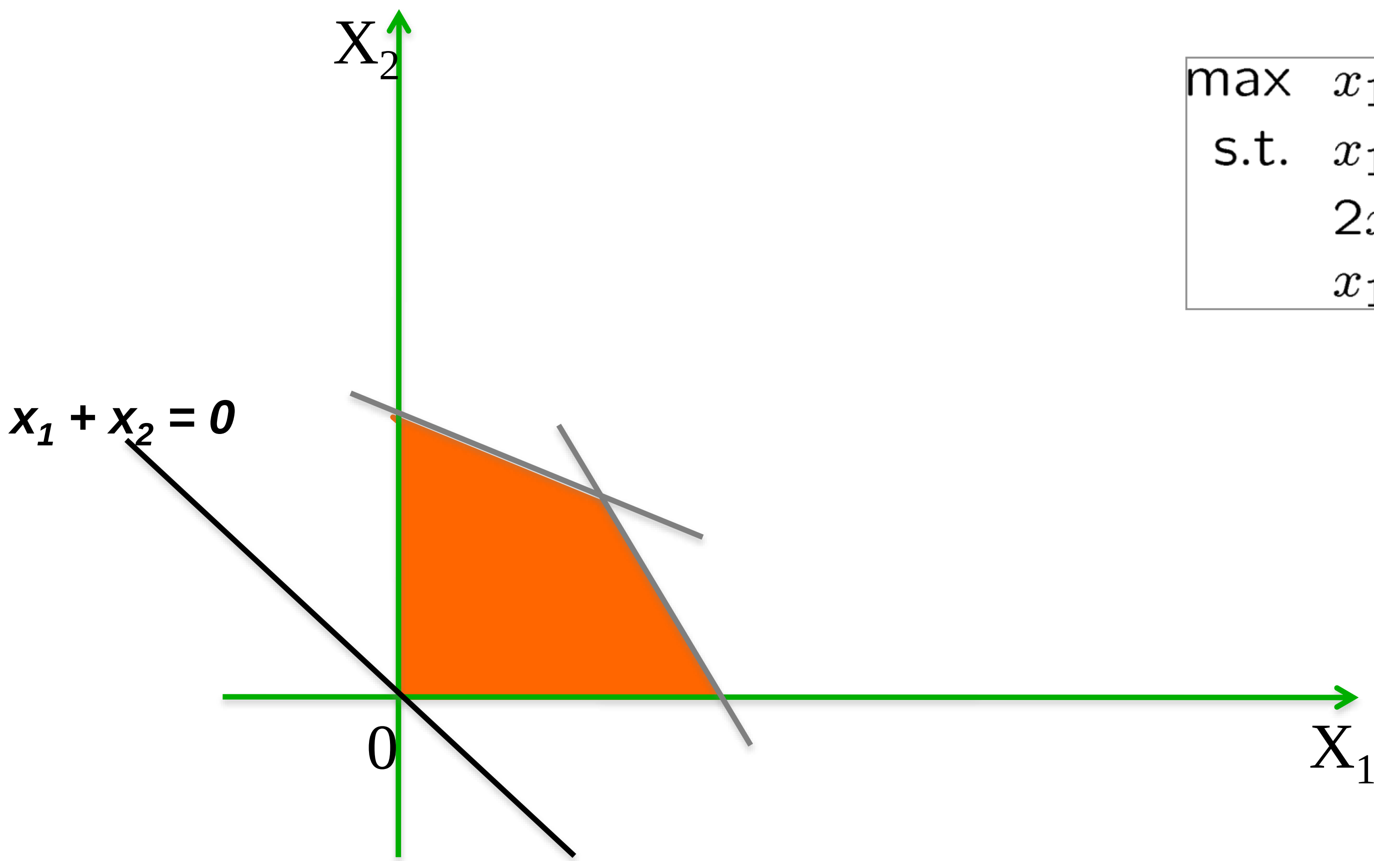
```python
# Objective coefficient: profit for each product
profit = np.array([3, 5])
# Constraint right-hand-side: capacity for each plant
capacity = np.array([4, 12, 18])

# A matrix: the consumption of capacity at plant i of product j
rate = np.array([[1, 0],
                 [0, 2],
                 [3, 2]])
# From A matrix, extract the number of products: N and the number of plants: M
M, N = rate.shape
```

Geometry of Linear Programming

Geometry of LP in 2D

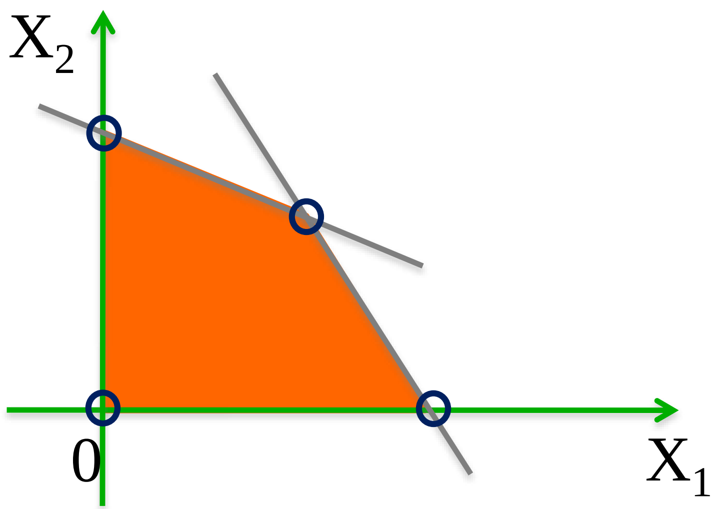
- **Objective:** ***maximize*** $x_1 + x_2$



$$
\begin{aligned}
\text{max} \quad & x_1 + x_2 \\
\text{s.t.} \quad & x_1 + 2x_2 \leq 3 \\
& 2x_1 + x_2 \leq 3 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

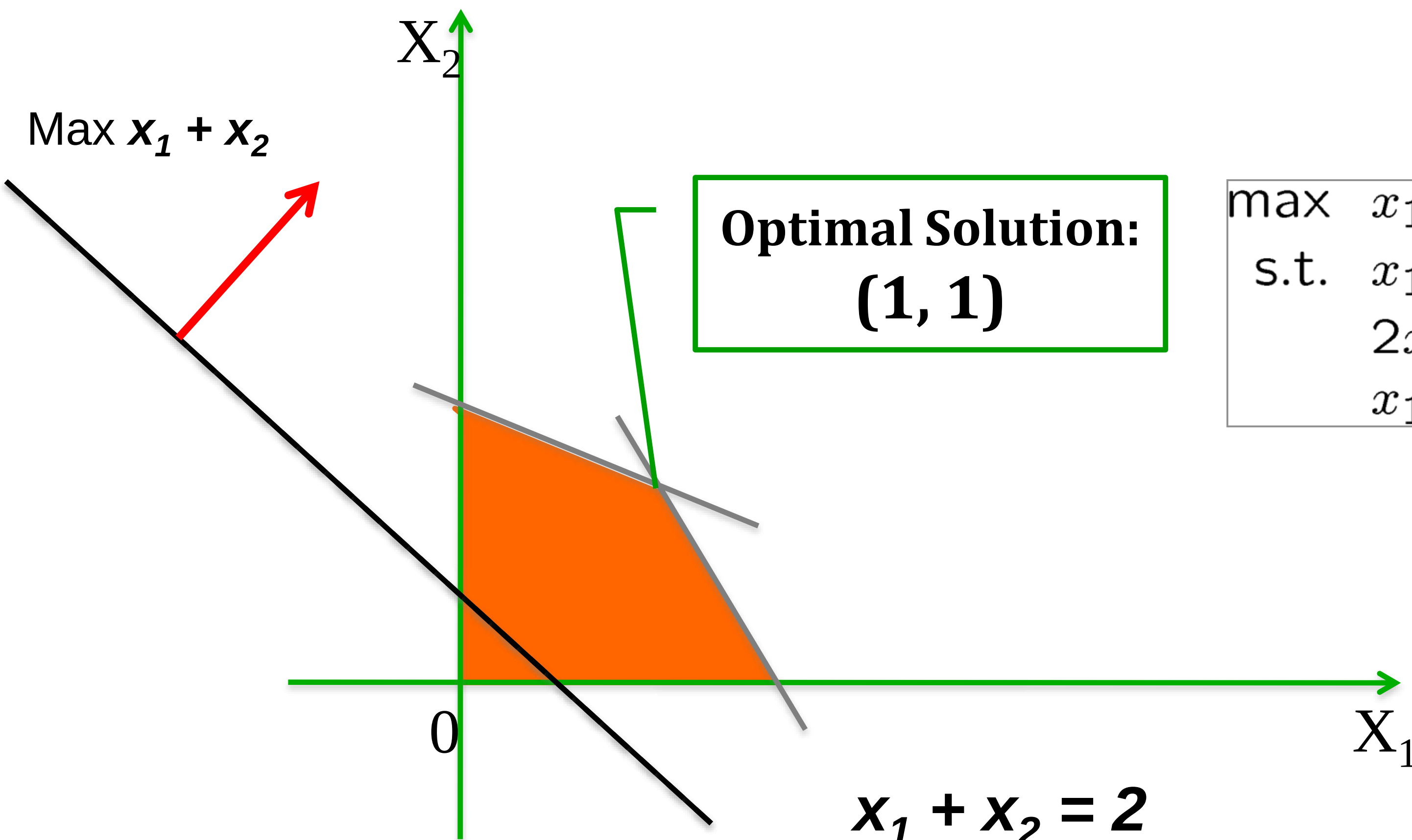
Geometry of LP in 2D

- **<u>Feasible Region</u>: The set of all the allowed solutions; a region (polygon) bounded by the constraints**
  - **Each equality constraint defines a line**
  - **Each inequality constraint defines a half-space**
- **<u>Extreme Points (Corner-Point Feasible Solution)</u>: Corner points on the boundary of the feasible region. E.g., (0, 0), (1.5, 0), (0, 1.5), and (1, 1)**
- **<u>Infeasible problem</u>: A problem with an empty feasible region**



Geometry of LP in 2D

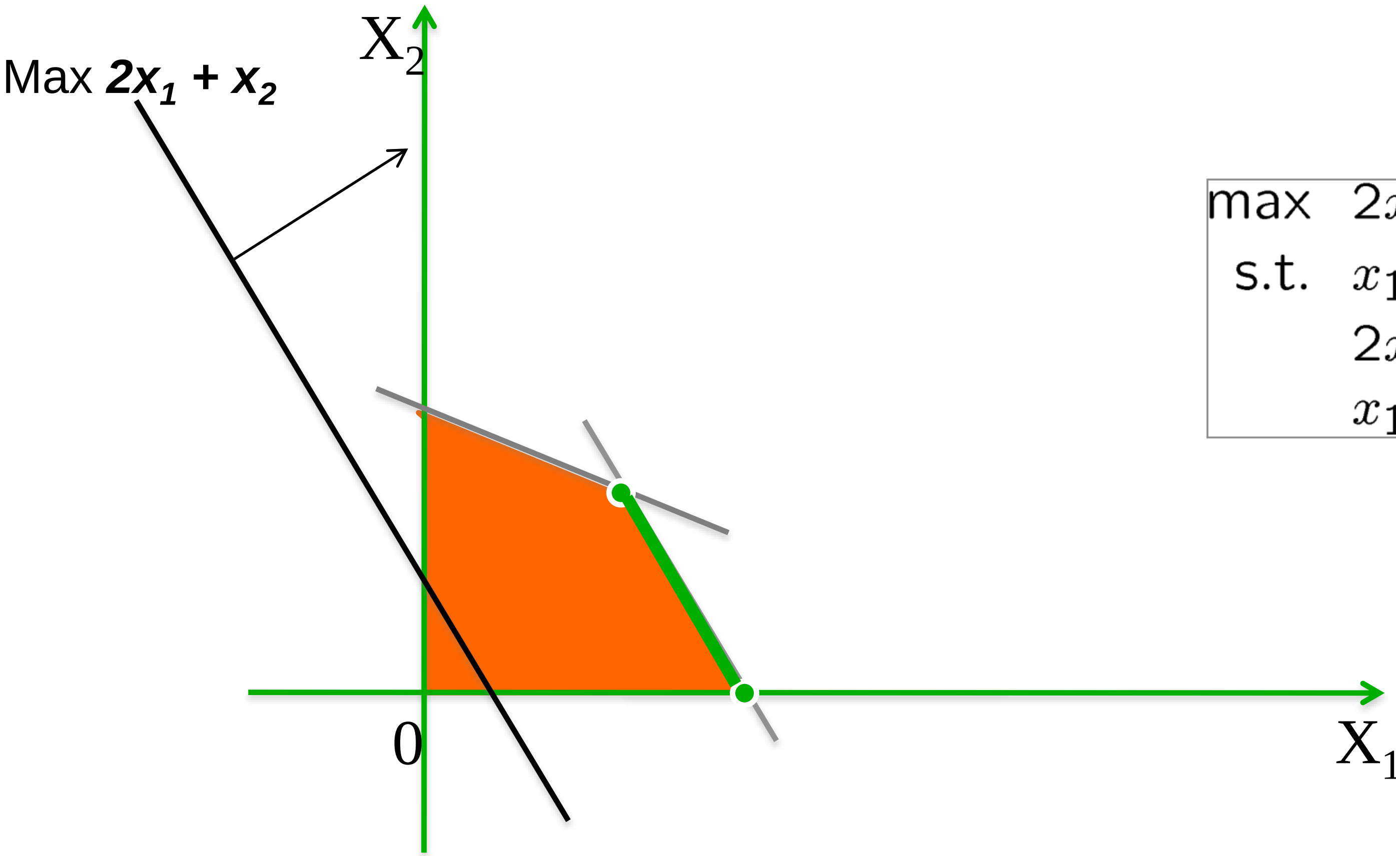
- **<u>Optimal Solution:</u> The *best feasible* solution**
- **For any feasible LP with a <u>finite</u> optimal solution, there exists an optimal solution that is an extreme point**



$$\begin{array}{ll} \max & x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{array}$$

Geometry of LP in 2D

- **Optimal solutions may NOT be <u>unique</u>**

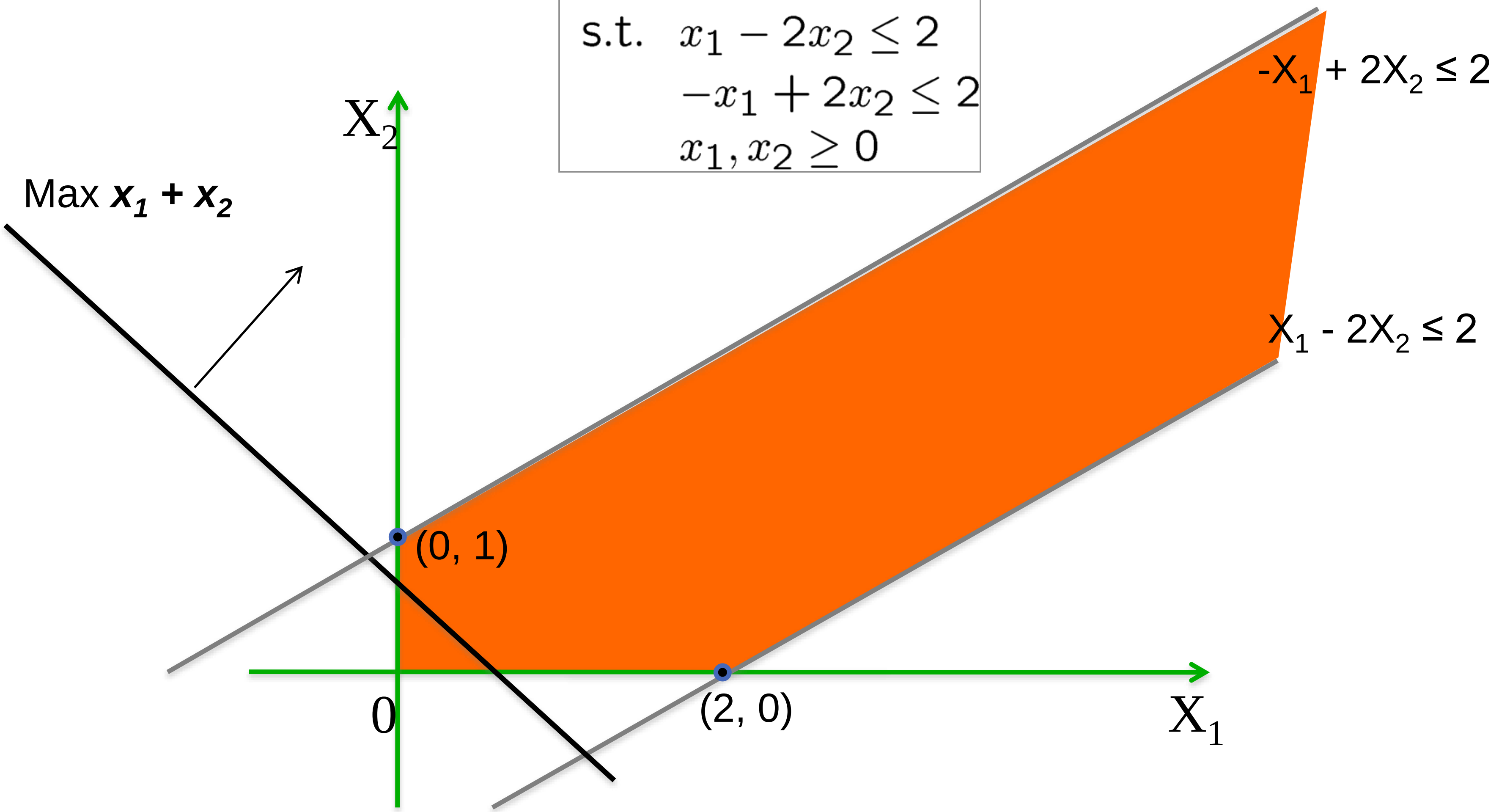


$$\begin{array}{ll} \max & 2x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{array}$$

Geometry of LP in 2D

- **Optimal solution may NOT be <u>finite</u>**

$$\begin{array}{ll} \max & x_1 + x_2 \\ \text{s.t.} & x_1 - 2x_2 \leq 2 \\ & -x_1 + 2x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{array}$$



Summary

- **Solutions of LP**
  - **Unique optimal solution**
  - **Multiple optimal solutions**
  - **The optimal objective is not bounded**
  - **Infeasible**

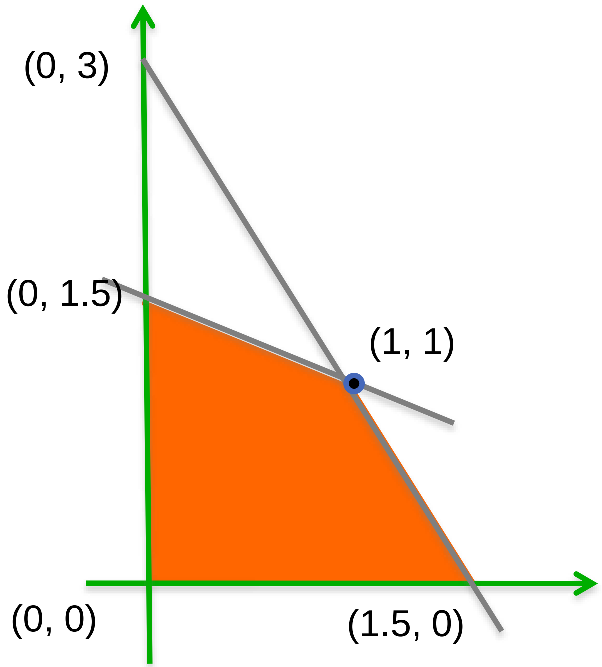
Active constraints

- <u>Binding (or active) constraints:</u> The constraints that are satisfied at *equality* for a given solution
  - All equality constraints (if satisfied) are binding

- <u>Non-binding (or inactive) constraints</u> are satisfied at *strict inequality* for a given solution

- The inequality level (= RHS – LHS) is known as the <u>slack</u>
  - Binding constraints have zero slack by definition

Back to Extreme Point



$$\begin{array}{ll} \max & x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \le 3 \\ & 2x_1 + x_2 \le 3 \\ & x_1, x_2 \ge 0 \end{array}$$

Constraints	Binding?	Slack
$X_1 + 2X_2 \le 3$		
$2X_1 + X_2 \le 3$		
$X_1 \ge 0$		
$X_2 \ge 0$		

How many binding constraints will an extreme point have?

What if we add constraints $x_2 \le 1.5$ and $2x_1 \ge 0$?

Extreme Point: An Algebraic Perspective

Consider a feasible region in n-dimensional space:

Polyhedron:

$$a_{i1}x_1 + \cdots + a_{in}x_n \geq b_i, i \in M_1$$

$$a_{j1}x_1 + \cdots + a_{jn}x_n \leq b_j, j \in M_2$$

$$a_{k1}x_1 + \cdots + a_{kn}x_n = b_k, k \in M_3$$

- **A vector $x^* = (x_1^*, \dots, x_n^*)$ is called basic solution if**
  - **All equality constraints are active;**
  - **Out of all constraints that are active at x^*, there are n of them that are linearly independent.**
- **x^* is called a basic feasible solution (BFS) if it is a basic solution and it is feasible.**

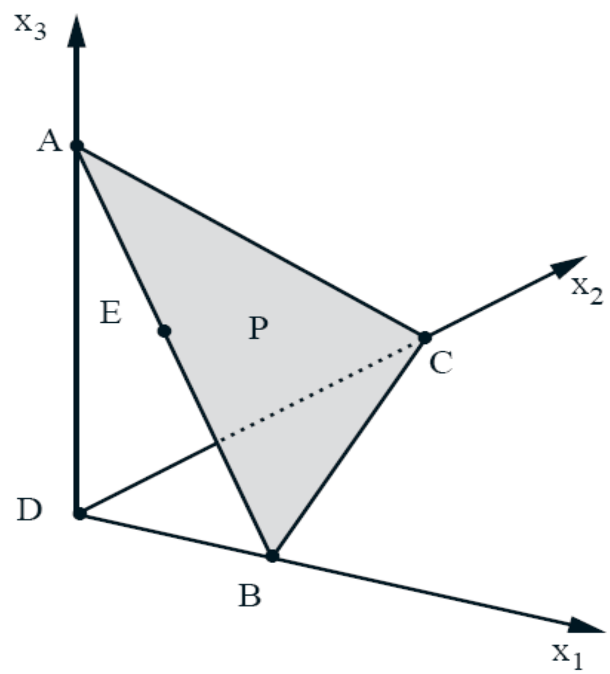
3-D Example

Consider a feasible region in 3-dimensional space:

$$x_1 + x_2 + x_3 = 1,$$

$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0$$

- **Which points are basic solutions, basic feasible solutions?**



3-D Example

Consider a feasible region in 3-dimensional space

$$x_1 \leq 1, x_2 \leq 1,\ x_3 \leq 1,$$

$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0$$

- **What are the extreme points/BFS?**

Insights: Portfolio Management Problem

- There are 100 stocks: stock i has return r_i per dollar invested.
- Budget constraint: b dollars available.
- How many stocks to invest and how to budget?

$$
\begin{aligned}
max \quad & r_1 x_1 + \cdots + r_{100} x_{100} \\
s.t. \quad & x_1 + \cdots + x_{100} \leq b, \\
& x_1 \geq 0, \ldots, x_{100} \geq 0
\end{aligned}
$$

Insights: Portfolio Management Problem

- **Suppose the stocks 1-10 are the top 10% riskiest stocks and at most half of the budget can be allocated to these stocks.**
- **How many stocks to invest and how to budget?**

$$
\begin{aligned}
\max \quad & r_1 x_1 + \cdots + r_{100} x_{100} \\
s.t. \quad & x_1 + \cdots + x_{100} \leq b, \\
& x_1 + \cdots + x_{10} \leq b/2, \\
& x_1 \geq 0, \ldots, x_{100} \geq 0
\end{aligned}
$$

Extreme Point in Standard Form

- For general form, it is not easy to check whether a basic solution is feasible or not; so consider
$$\boldsymbol{Ax = b}$$
$$\boldsymbol{x \geq 0}$$
- $\boldsymbol{A}$ is an $\boldsymbol{m \times n}$ matrix. Assume $\boldsymbol{m \leq n}$ and the $\mathbf{m}$ rows of $\boldsymbol{A}$ are linearly independent.
- A vector $\boldsymbol{x}$ is a basic solution if $\boldsymbol{Ax = b}$ and there exist indices $B(1), \dots, B(m)$ such that
  - The columns $\boldsymbol{A}_{B(1)}, \dots, \boldsymbol{A}_{B(m)}$ are linearly independent;
  - If $i \neq B(1), \dots, B(m)$, then $\boldsymbol{x_i = 0}$.

Some Terminology

- The columns $A_{B(1)}, \ldots, A_{B(m)}$ are called **basic columns**.
- The $m \times m$ matrix

$$\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}],$$

is called the **basis matrix**.
- The corresponding components of the vector x: $\boldsymbol{x}_B = (x_{B(1)}, \ldots, x_{B(m)})$ are called **basic variables**; the remaining components are called **nonbasic**.
  - $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$
  - If x_i is a nonbasic variable, then $x_i = 0$.

An algorithm to find extreme point

1. Convert an LP to standard form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq \mathbf{0}$$

2. Choose m linearly independent columns $A_{B(1)}, \ldots, A_{B(m)}$ from the matrix A.
3. Let $x_i = 0$ if $i \neq B(1), \ldots, B(m)$.
4. Solve the system of m equations $Ax = b$ for the unknowns $x_{B(1)}, \ldots, x_{B(m)}$.
5. If x_i are all non-negative ($x \geq 0$), then we have found an extreme point (BFS); otherwise go back to step 2.

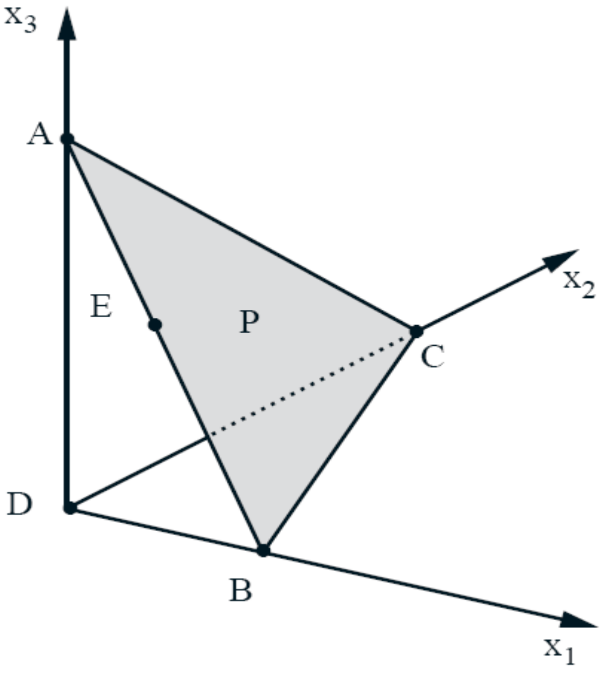
Example

Consider following standard form:

$$x_1 + x_2 + x_3 = 1,$$

$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0$$

- A is a 1×3 matrix (m=1):
  - Choose column 1;
  - Let $x_2 = 0, x_3 = 0$;
  - Solve $x_1 = 1$;
  - Check non-negativity.



Example

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}, \boldsymbol{x} \geq \mathbf{0}.$$

- **How to find a BFS?**

- **Can the constraints be expressed in inequalities?**

Summary

- Each basic solution has *m* basic variables
  - Rest of variables are nonbasic variables and are set to zero
- Values of basic variables given by:
  - Solution to a system of *m* equations
- A basic feasible solution (BFS)
  - Basic solution where all *m* basic variables are nonnegative

Simplex Method

Naïve Approach

- **Consider an LP in standard form**

$$\begin{aligned} \min \quad & c'x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

1. **Find all extreme points.**
2. **Compute $c'x$ at these extreme points.**
3. **Pick the smallest one.**

- **Reduces from infinite to finite; how many possible basic solutions are there?**
  - **n = 20, m = 10: about 0.2 million;**
  - **n = 30, m = 15: about 155 million.**

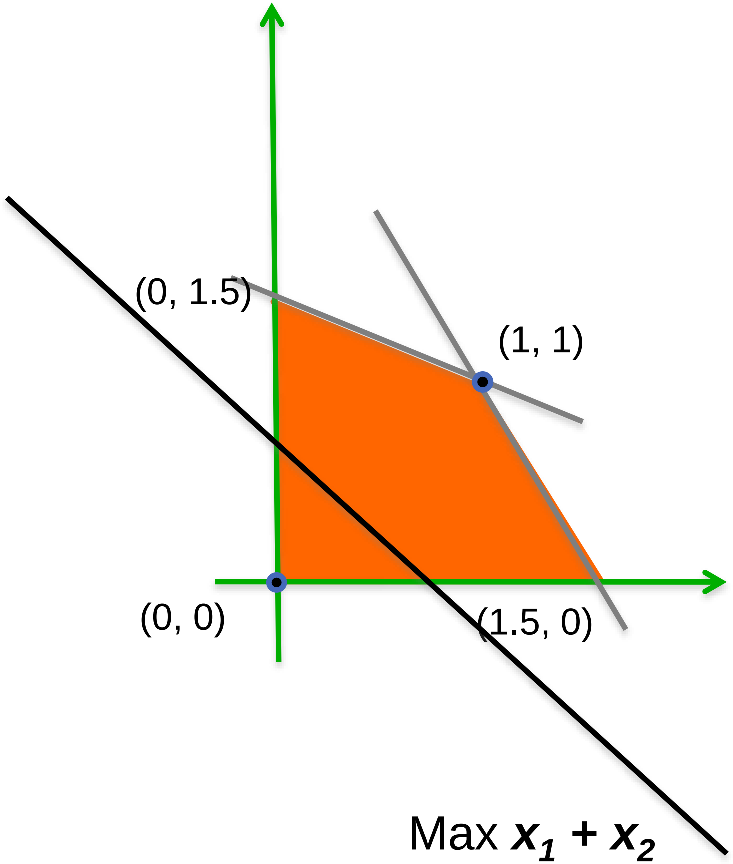
A General Idea

1. **Find a feasible solution;**
2. **Determine whether the current solution is optimal or not: optimality condition.**
   - **If yes, terminate; else proceed to step 3.**
3. **Search the neighborhood of the current solution to find another feasible solution with lower cost, and go back to step 2.**

Optimality Condition

- **Why is (0, 0) not optimal?**

- **Why is (1, 1) optimal?**



A General Optimization Problem

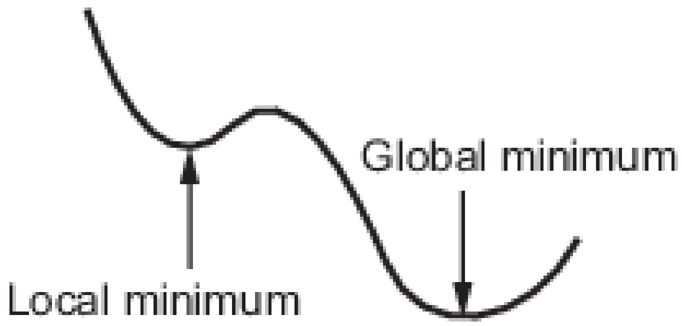
- **A general problem**

$$\begin{aligned} &min \quad && f(x) \\ &s.t. && x \in \mathcal{F} \end{aligned}$$

- **For LP**
  - $f(x) = c'x$
  - $\mathcal{F} = \{x \mid Ax = b, x \geq 0\}$

Local Optimality vs Global Optimality

- For a feasible solution $x \in \mathcal{F}$, if no other feasible solution in the neighborhood of x leads to lower cost, x is called a local optimal solution.
- For a feasible solution $x \in \mathcal{F}$, if no other feasible solution in $\mathcal{F}$ leads to lower cost, x is called a global optimal solution.

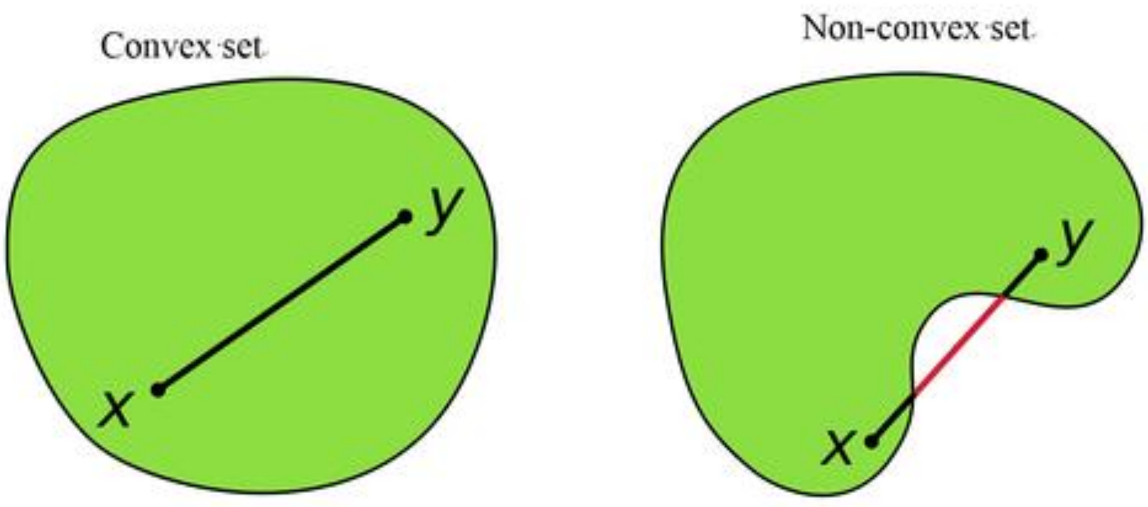


Convex Set and Convex Function

- **A set $\mathcal{F}$ is called convex set if for every $\lambda \in [\mathbf{0}, \mathbf{1}]$ and $x, y \in \mathcal{F}$, we have**

$$\lambda \boldsymbol{x} + (\mathbf{1} - \lambda)\boldsymbol{y} \in \mathcal{F}$$

  - **Polyhedron $\{x \mid Ax = b, x \geq 0\}$ is a convex set.**

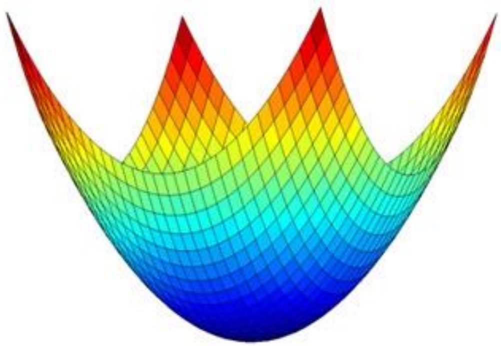


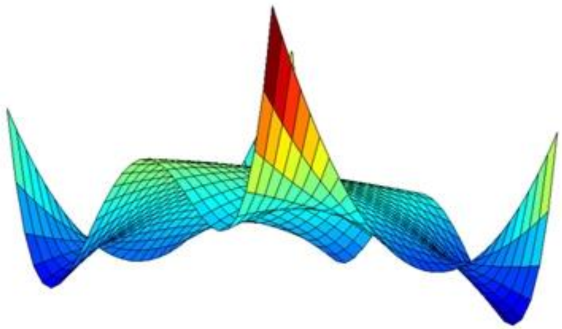
Convex Set and Convex Function

- **A function $f(x)$ is called convex on a convex set $\mathcal{F}$ if for every $\lambda \in [0,1]$ and $x, y \in \mathcal{F}$, we have**

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

  - **A linear function $f(x) = c'x$ is a convex function.**





A General Principle

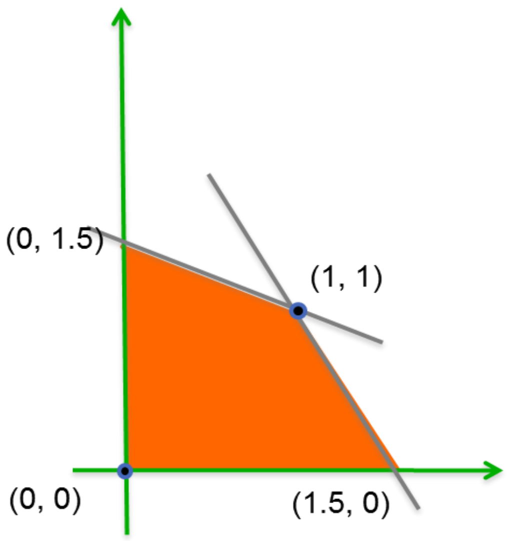
- If $\mathcal{F}$ is a convex set and $f(x)$ is a convex function, then

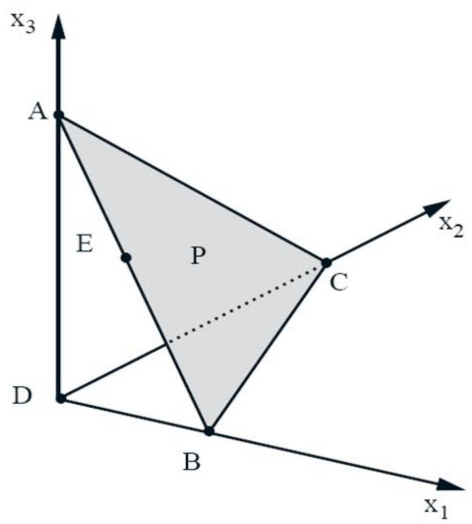
$$\begin{array}{ll} min & f(x) \\ s.t. & x \in \mathcal{F} \end{array}$$

  - is called a convex programming problem;
  - a local optimal solution is a global optimal solution.
- LP is a special case of convex programming.
  - It is sufficient to check feasible solution nearby.
  - It is sufficient to check BFS nearby.

BFS nearby: Adjacent BFS

- **Two distinct BFSs in n-dimensional space are said to be adjacent if they share n-1 linearly independent active constraints.**
  - **For standard form, this implies that all but one of their nonbasic variables are the same.**



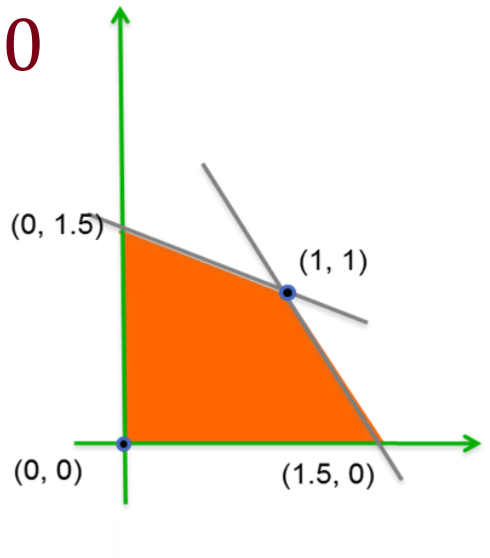


Example

$$\begin{aligned} x_1 + 2x_2 + x_3 \qquad\quad &= 3, \\ 2x_1 + x_2 \qquad\quad + x_4 &= 3, \end{aligned}$$

$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0$$

- **Which one is adjacent BFS**
  - $(0, 0, 3, 3), (0, 1.5, 0, 1.5)$
  - $(0, 0, 3, 3), (1, 1, 0, 0)$
  - $(1,1, 0, 0), (0, 1.5, 0, 1.5)$
  - $(1,1, 0, 0), (1.5, 0, 1.5, 0)$



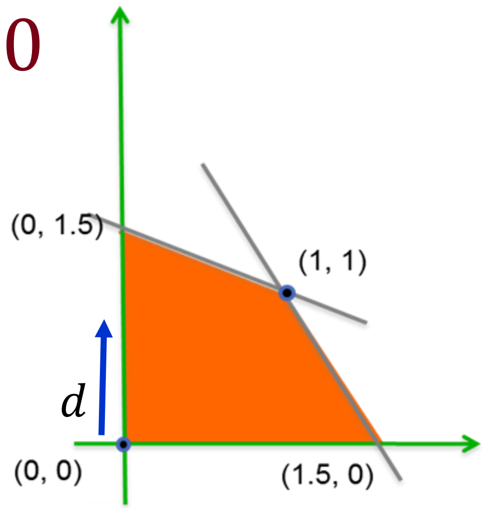
Example

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}, \boldsymbol{x} \geq \mathbf{0}.$$

- **Which one is adjacent BFS**
  - $(0, 0, 0, 8, 12, 4, 6), (4, 0, 0, 4, 12, 0, 6)$
  - $(0, 0, 0, 8, 12, 4, 6), (0, 6, 0, 2, 6, 4, 0)$
  - $(0, 6, 0, 2, 6, 4, 0),\ (4, 0, 0, 4, 12, 0, 6)$

Example: Iteration 0

$$x_1 + 2x_2 + x_3 \qquad = 3,$$
$$2x_1 + x_2 \qquad + x_4 = 3,$$
$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0$$



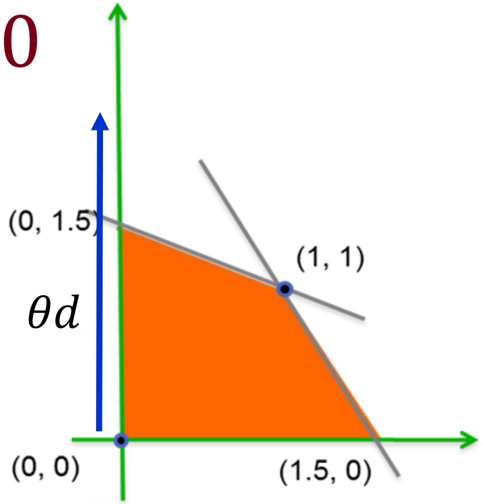
- **Adjacent BFS**
  - $(0, \overset{\Downarrow}{0}, 3, 3) \rightarrow (0, ?\,, ?\,, ?\,)$
- **How to move?**
  - Stay on edge: $d = (0, 1, d_3, d_4)$
  - Stay feasible: $\begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 3 \\ 3 \end{bmatrix} + \theta \begin{bmatrix} 0 \\ 1 \\ d_3 \\ d_4 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \theta \geq 0$
  - $\Rightarrow d = (0, 1, -2, -1)$

Example: Iteration 0

$$
\begin{aligned}
x_1 + 2x_2 + x_3 \qquad\quad &= 3, \\
2x_1 + x_2 \qquad\quad + x_4 &= 3,
\end{aligned}
$$

$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0$$

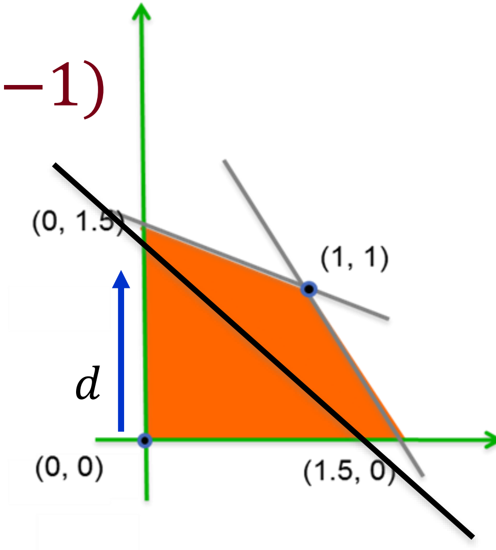
- **Adjacent BFS**
  - $(0, \overset{\Downarrow}{0}, 3, 3) \rightarrow (0, ?\,, ?\,, ?\,)$
- **How far to move?**
  - Stay feasible: $\begin{bmatrix} 0 \\ 0 \\ 3 \\ 3 \end{bmatrix} + \theta \begin{bmatrix} 0 \\ 1 \\ -2 \\ -1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
  - $\Rightarrow \theta \leq 1.5$
  - $(0, 0, 3, 3) + 1.5(0, 1, -2, -1) = (0, 1.5, 0, 1.5)$



Example: Iteration 0

$$-\min -x_1 - x_2$$
$$x_1 + 2x_2 + x_3 \qquad = 3,$$
$$2x_1 + x_2 \qquad + x_4 = 3,$$
$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0$$

- **Change in cost along direction** $d = (0, 1, -2, -1)$



- $(-1, -1, 0, 0)\begin{bmatrix} 0 \\ 0 \\ 3 \\ 3 \end{bmatrix}$

- $(-1, -1, 0, 0)\begin{bmatrix} 0 \\ 0 \\ 3 \\ 3 \end{bmatrix} + \theta(-1, -1, 0, 0)\begin{bmatrix} 0 \\ 1 \\ -2 \\ -1 \end{bmatrix}$

Reduced Cost

- Now we consider the objective: $\boldsymbol{c}'\boldsymbol{x}$
  - $\boldsymbol{c}'\boldsymbol{x}$;
  - $\boldsymbol{c}'(\boldsymbol{x}+\theta\boldsymbol{d})$;
  - $\boldsymbol{d}$ is the direction changing the nonbasic variable with index $\boldsymbol{j}$.
- Reduced cost in direction $\boldsymbol{j}$ is defined as

$$\overline{c_j} = \frac{\boldsymbol{c}'(\boldsymbol{x}+\theta\boldsymbol{d}) - \boldsymbol{c}'\boldsymbol{x}}{\theta} = \boldsymbol{c}'\boldsymbol{d}$$

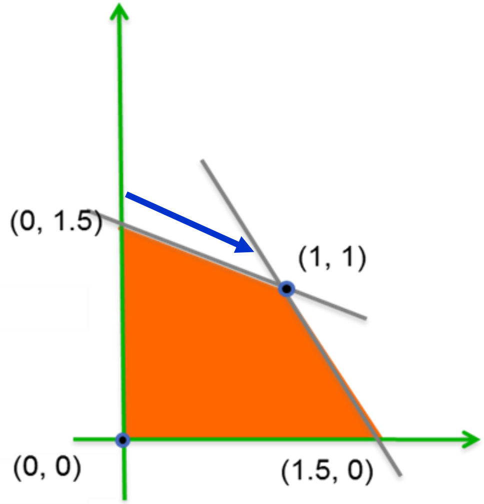
  - If $\overline{c_j} < 0$
  - else

Example: Iteration 1

$$\begin{aligned} x_1 + 2x_2 + x_3 \qquad\quad &= 3, \\ 2x_1 + x_2 \qquad\quad + x_4 &= 3, \end{aligned}$$

$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0$$

- **Adjacent BFS**
  - $(0, 1.5, 0, 1.5) \rightarrow (?\,, ?\,, 0, ?\,)$
- **Moving direction**
  - $d = (1, d_2, 0, d_4)$
  - $\begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ d_2 \\ 0 \\ d_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ d_2 \\ 0 \\ d_4 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ 0 \\ -1.5 \end{bmatrix}$
  - $(0, 1.5, 0, 1.5) + \theta(1, -0.5, 0, -1.5) \geq \mathbf{0}, \Rightarrow \theta \leq 1$
  - $(0, 1.5, 0, 1.5) + 1(1, -0.5, 0, -1.5) = (1,1,0,0)$

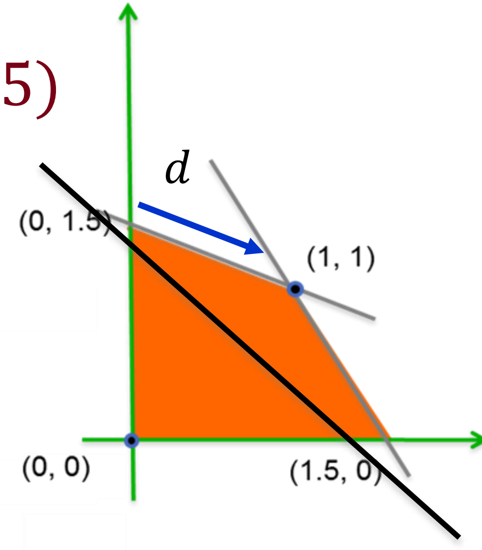


Example: Iteration 1

$$-\min -x_1 - x_2$$
$$x_1 + 2x_2 + x_3 \qquad = 3,$$
$$2x_1 + x_2 \qquad + x_4 = 3,$$
$$x_1 \geq 0, x_2 \geq 0,\ x_3 \geq 0, x_4 \geq 0$$

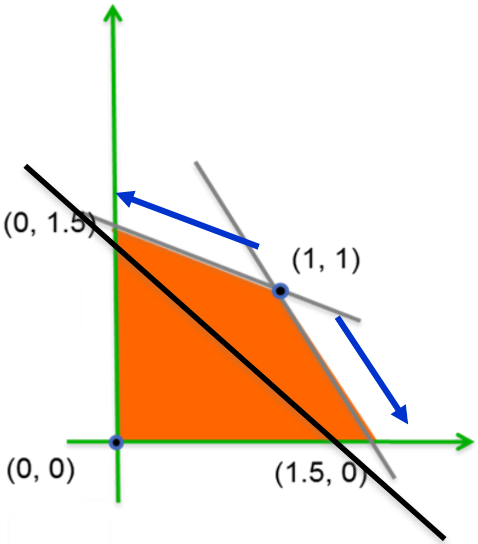
- **Reduced cost in direction:** $d = (1, -0.5, 0, -1.5)$

  - $\bar{c_1} = (-1, -1, 0, 0) \begin{bmatrix} 1 \\ -0.5 \\ 0 \\ -1.5 \end{bmatrix} = -0.5$



Example: Iteration 2

$$- \, min \, -x_1 \, -x_2$$
$$x_1 + 2x_2 + x_3 \qquad = 3,$$
$$2x_1 + x_2 \qquad + x_4 = 3,$$
$$x_1 \geq 0, x_2 \geq 0, \, x_3 \geq 0, x_4 \geq 0$$



- **BFS:** $(1, 1, 0, 0)$
  - $(1, 1, 0, \mathbf{0}) + \theta(d_1, d_2, 1, \mathbf{0})$
  - $(1, 1, \mathbf{0}, 0) + \theta(d_1, d_2, \mathbf{0}, 1)$
- **Reduced cost**
  - $\bar{c}_3 = \boldsymbol{c'd} = \frac{1}{3}$
  - $\bar{c}_4 = \boldsymbol{c'd} = \frac{1}{3}$

Optimality Condition and Simplex Method

1. **Find a BFS x;**
2. **Determine whether the current solution is optimal or not: is the reduced cost $\bar{c}_j$ for all the nonbasic variable x_j positive?**
   - **If yes, terminate; else proceed to step 3.**
3. **Pick a nonbasic variable x_j for which the reduced cost is negative and move along this direction, i.e., $\boldsymbol{x} + \theta^* \boldsymbol{d}$**
   - $d_j = 1$;
   - $d_i = 0$, **for** $i \neq j, i \neq B(1), \ldots, B(m)$;
   - $\theta^* = max\{\theta \mid \boldsymbol{x} + \theta \boldsymbol{d} \geq \boldsymbol{0}\}$