MBA 8023: Optimization
Introduction to Linear Programming

Ling-Chieh Kung

Department of Information Management
National Taiwan University

December 4, 2013

Introduction

- In the following weeks, we will study Linear Programming (LP).
  - It is used a lot in practice.
  - It also provides important theoretical properties.
  - It is good starting point for all OR subjects.
- We will study:
  - The basic properties.
  - LP formulation.
  - The simplex method for solving LPs.
  - Conditions for feasibility, unboundedness, and optimality.
  - Integer Programming.

Road map

- **Basic ideas.**
- LP formulation examples.
- Linearization techniques.

Basic elements of an LP

- A **linear program** (LP) is a mathematical program whose objective function and constraints are all **linear** and variables are all **fractional**.
- In general, any LP can be expressed as

$$
\begin{aligned}
\min \quad & f(x) = \sum_{j=1}^{n} c_j x_j \quad (\textbf{objective function}) \\
\text{s.t.} \quad & g_i(x) = \sum_{j=1}^{n} A_{ij} x_j \leq b_i \quad \forall i = 1, ..., m \quad (\textbf{constraints}) \\
& x_j \in \mathbb{R} \quad \forall j = 1, ..., n. \quad (\textbf{decision variable})
\end{aligned}
$$

  - A_{ij}s: the **constraint coefficients**.
  - b_is: the right-hand-side values (**RHSs**).
  - c_js: the **objective coefficients**.
  - As a convention, we will ignore $x_j \in \mathbb{R}$ in the sequel.

Transformation

- How about a maximization objective function?
  - $\max f(x) \Leftrightarrow \min -f(x)$.
- How about equality or greater-than-or-equal-to constraint?
  - $g_i(x) \geq b_i \Leftrightarrow -g_i(x) \leq -b_i$.
  - $g_i(x) = b_i \Leftrightarrow g_i(x) \leq b_i$ and $g_i(x) \geq b_i$ (which is $-g_i(x) \leq -b_i$).
- For example,

$$\begin{array}{rl} \max & x_1 - x_2 \\ \text{s.t.} & -2x_1 + x_2 \geq -3 \\ & x_1 + 4x_2 = 5. \end{array} \quad \Leftrightarrow \quad \begin{array}{rl} \min & -x_1 + x_2 \\ \text{s.t.} & 2x_1 - x_2 \leq 3 \\ & x_1 + 4x_2 \leq 5 \\ & -x_1 - 4x_2 \leq -5. \end{array}$$

Matrix representation of an LP

- An LP can also be expressed in the **matrix representation**:

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax \leq b. \end{aligned}$$

  - $A \in \mathbb{R}^{m \times n}$: the **constraint matrix**.
  - $b \in \mathbb{R}^{m}$: the **RHS vector** (a column vector).
  - $c \in \mathbb{R}^{n}$: the **objective vector** (a row vector).
- For example,

$$\begin{array}{rl} \max & x_1 - x_2 \\ \text{s.t.} & -2x_1 + x_2 \geq 3 \\ & x_1 + 4x_2 = 5. \end{array} \Rightarrow A = \begin{bmatrix} 2 & -1 \\ 1 & -4 \\ -1 & 4 \end{bmatrix}, b = \begin{bmatrix} -3 \\ 5 \\ -5 \end{bmatrix}, c = \begin{bmatrix} -1 & 1 \end{bmatrix}.$$

- The matrix representation will be used a lot in this course.

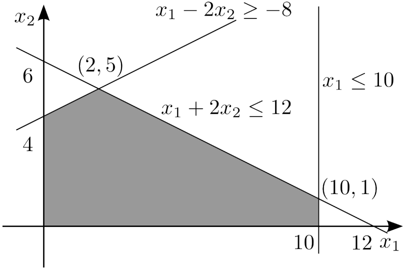
Sign constraints

- For some reasons that will be clear in a couple weeks, we distinguish between two kinds of constraints:
  - **Sign constraints**: $x_i \geq 0$ or $x_i \leq 0$.
  - **Functional constraints**: all others.
- For a variable x_i:
  - It is **nonnegative** if $x_i \geq 0$.
  - It is **nonpositive** if $x_i \leq 0$.
  - It is **unrestricted in sign** (urs.) or **free** if there is no sign constraint for it.

Example

- Here is an example of LP:

$$
\begin{array}{rrcrcr}
\min & 2x_1 & + & x_2 & & \\
\text{s.t.} & x_1 & & & \leq & 10 \\
 & x_1 & + & 2x_2 & \leq & 12 \\
 & x_1 & - & 2x_2 & \geq & -8 \\
 & x_1 & & & \geq & 0 \\
 & & & x_2 & \geq & 0.
\end{array}
$$



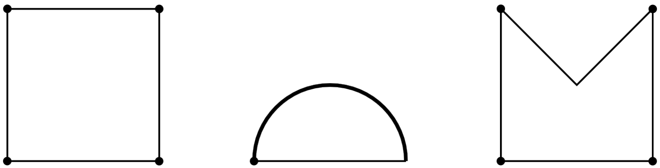
Extreme points

- We need to first define **extreme points** for a set:

Definition 1 (Extreme points)

For a set S, a point x is an extreme point if there does not exist a three-tuple (x^1, x^2, λ) such that $x^1 \in S \setminus \{x\}$, $x^2 \in S \setminus \{x\}$, $\lambda \in (0,1)$, and

$$x = \lambda x^1 + (1-\lambda)x^2.$$



Local v.s. global optima

- Recall the following result from Nonlinear Programming:

Proposition 1

For a convex function over a convex feasible region, a local minimum is a global minimum.

  - For a concave function over a convex feasible region, a local maximum is a global maximum.

Proposition 2

For any concave function that has a global minimum, there exists a global minimum that is an extreme point.

  - It is **not** "a global minimum must be an extreme point."

Solving a linear program

- Now we know when we minimize $f(\cdot)$ over a convex feasible region F:
  - If $f(\cdot)$ is **convex**, search for a **local min**.
  - If $f(\cdot)$ is **concave**, search among the **extreme points** of F.
- How are these related to Linear Programming?
- We will show that, for any linear program:
  - The feasible region is convex.
  - The objective function is both convex and concave.
- Then the results will mean a lot to Linear Programming!

Solving a linear program

Proposition 3

The feasible region of a linear program is convex.

Proof. First, note that the feasible region of a linear program is the intersection of several half spaces (each one is determined by an inequality constraint) and hyperplanes (each one is determined by an equality constraint). It is trivial to show that half spaces and hyperplanes are always convex. It then remains to show that the intersection of convex sets is convex, which is also trivial. □

Solving a linear program

Proposition 4

A linear function is both convex and concave.

Proof. To show that a function $f(\cdot)$ is convex and concave, we need to show that $f(\lambda x^1 + (1-\lambda)x^2) = \lambda f(x^1) + (1-\lambda) f(x^2)$. Let $f(x) = c \cdot x + b$ be a linear function, $c \in \mathbb{R}^n, b \in \mathbb{R}$, then

$$\begin{aligned} f\Big(\lambda x^1 + (1-\lambda)x^2\Big) &= c \cdot \Big(\lambda x^1 + (1-\lambda)x^2\Big) + b \\ &= \lambda(c \cdot x^1 + b) + (1-\lambda)(c \cdot x^2 + b) \\ &= \lambda f(x^1) + (1-\lambda) f(x^2). \end{aligned}$$

Therefore, a linear function is both convex and concave. □

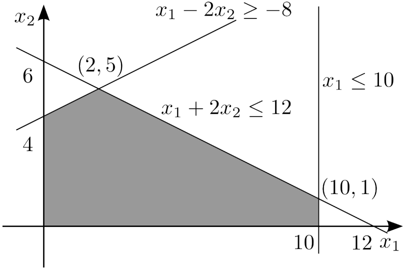
Solving a linear program

- To solve a linear program, we only need to search for a local minimum.
  - As long as we find a **feasible improving direction**, just move along that direction.
- We only need to search among extreme points of the feasible region.
  - We may **keep moving** until we reach the **end** of the feasible region.
- These two properties form the foundation of the **graphical approach** for solving two-dimensional linear programs.
- They also allow us to use the **simplex method** for solving n-dimensional linear programs.

Graphical approach

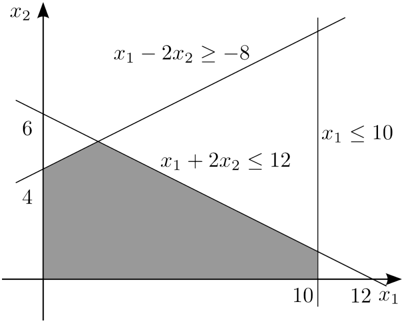
- For linear programs with only two decision variables, we may solve them with the graphical approach.
- Consider the following example:

$$\begin{array}{rrcrcr} \max & 2x_1 & + & x_2 & & \\ \text{s.t.} & x_1 & & & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 12 \\ & x_1 & - & 2x_2 & \geq & -8 \\ & x_1 & & & \geq & 0 \\ & & & x_2 & \geq & 0. \end{array}$$

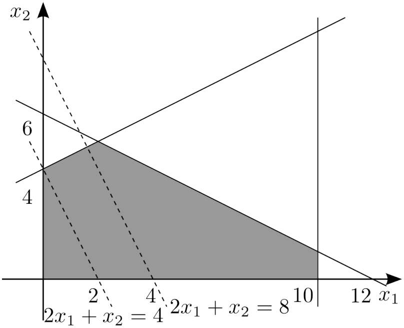


Graphical approach

- Step 1: Draw the feasible region.

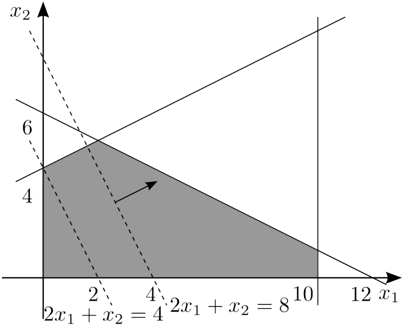


- Step 2: Draw an **isocost line**.
  - All points on it have the same objective value.
  - isoprofit/isoquant line sometimes.

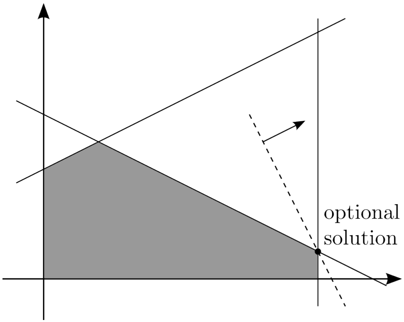


Graphical approach

- Step 3: Indicate the direction to push the isocost line.
  - The direction that increases the objective value for a maximization problem.

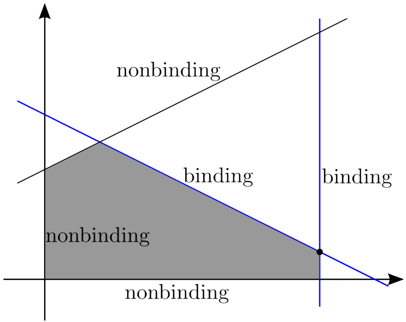


- Step 4: Push the isocost line to the "end" of the feasible region.
  - Stop when any further step makes all points on the isocost line infeasible.



Graphical approach

- Step 5: Identify the **binding constraints** at the optimal solution.



- Step 6: Set the binding constraints to equalities and solve the linear system for an optimal solution.
  - In the example, the binding constraints are $x_1 \leq 10$ and $x_1 + 2x_2 \leq 12$. Therefore, we solve

$$\left[\begin{array}{cc|c} 1 & 0 & 10 \\ 1 & 2 & 12 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 0 & 10 \\ 0 & 2 & 2 \end{array}\right]$$
$$\to \left[\begin{array}{cc|c} 1 & 0 & 10 \\ 0 & 1 & 1 \end{array}\right]$$

    and obtain an optimal solution $(x_1^*, x_2^*) = (10, 1)$.
- Step 7: Plug in and find z^*, the associated objective value.

Road map

- Basic ideas.
- **LP formulation examples.**
  - Resource allocation.
  - Material blending.
  - Production and inventory.
- Linearization techniques.

Introduction

- It is important to learn how to model a practical situation as a linear program.
- This process is typically called **LP formulation** or **modeling**.
- We will introduce three types of LP problems, demonstrate how to formulate them, and discuss some important issues.
  - Resource allocation, material blending, production and inventory.
  - There are certainly many other types of LP problems.
- For large-scale problems, **compact formulations** are used.

Resource allocation (1/3)

- We produce products to sell.
- Each product requires some resources. **Resources are limited**.
- We want to maximize the total sales revenue while ensuring resources are enough.

Resource allocation: the problem (2/3)

- We may produce desks and tables.[1]
  - Producing a desk requires four units of wood, one hour of labor, and 30 minutes of machine time.
  - Producing a table requires five units of wood, two hours of labor, and 20 minutes of machine time.
- We may sell everything we produce.
- For each day, we have
  - Two workers, each works for eight hours.
  - One machine that can run for eight hours.
  - A supply of 36 units of wood.
- Desks and tables are sold at $800 and $600 per unit, respectively.

[1] *Operations Research: Applications and Algorithms* by W. Winston, 4th ed.

Resource allocation: formulation (3/3)

- Let

$$x_1 = \text{number of desks produced in a day and}$$
$$x_2 = \text{number of tables produced in a day.}$$

- The complete formulation is

$$\begin{array}{rrcrcrl} \max & 800x_1 & + & 600x_2 & & & \\ \text{s.t.} & 4x_1 & + & 5x_2 & \leq & 36 & \text{(wood)} \\ & x_1 & + & 2x_2 & \leq & 16 & \text{(labor)} \\ & 30x_1 & + & 20x_2 & \leq & 480 & \text{(machine)} \\ & \multicolumn{6}{l}{x_i \geq 0 \quad \forall i = 1, 2} \end{array}$$

  - **Clearly** define decision variables **in front of** your formulation.
  - Write **comments** after the objective function and constraints.
  - Do not forget nonnegativity constraints.

Material blending (1/5)

- In some situations, we need to determine not only products to produce but also **materials** to input.
- This is because we have some **flexibility** in making the products.
- For example, in making orange juice, we may use orange, sugar, water, etc. Different ways of **blending** these materials results in different qualities of juice.
- The goal is to save money (lower the proportion of expensive materials) while maintaining **quality**.

Material blending: the problem (2/5)

- We blend materials 1, 2, and 3 to make products 1 and 2.[2]
- The quality of a product, which depends on the proportions of these three materials, must meet the standard:
  - Product 1: at least 40% of material 1; at least 20% of material 2.
  - Product 2: at least 50% of material 1; at most 30% of material 3.
- At most 100 kg of product 1 and 150 kg of product 2 can be sold.
- Prices for products 1 and 2 are \$10 and \$15 per kg, respectively.
- Costs for materials 1 to 3 are \$8, \$4, and \$3 per kg, respectively.
- Amount of a product made equals the amount of materials input.
- We want to maximize the total profit.

[2] *Operations Research: Applications and Algorithms* by W. Winston, 4th ed.

Material blending: decision variables (3/5)

- Let

$$x_{11} = \text{kg of material 1 used for product 1,}$$
$$x_{12} = \text{kg of material 1 used for product 2,} \ldots$$
$$x_{32} = \text{kg of material 3 used for product 2.}$$

- How to find the production quantities of products and the purchasing quantities of materials?

	Product 1	Product 2	Purchasing quantity
Material 1	x_{11}	x_{12}	$x_{11} + x_{12}$
Material 2	x_{21}	x_{22}	$x_{21} + x_{22}$
Material 3	x_{31}	x_{32}	$x_{31} + x_{32}$
Production quantity	$x_{11} + x_{21} + x_{31}$	$x_{12} + x_{22} + x_{32}$	

Material blending: quality constraints (4/5)

- The objective function is

$$\begin{aligned} &\max\ 10(x_{11}+x_{21}+x_{31})+15(x_{12}+x_{22}+x_{32}) \\ &\qquad -8(x_{11}+x_{12})-4(x_{21}+x_{22})-3(x_{31}+x_{32}) \\ =&\max\ 2x_{11}+7x_{12}+6x_{21}+11x_{22}+7x_{31}+x_{32}. \end{aligned}$$

- In product 1, how to guarantee at least 40% are material 1?

$$\frac{x_{11}}{x_{11}+x_{21}+x_{31}} \geq 0.4.$$

- It is conceptually correct. However, it is **nonlinear**!
- Let's fix the nonlinearity by taking the denominator to the RHS:

$$x_{11} \geq 0.4(x_{11}+x_{21}+x_{31}).$$

  Though equivalent, they are just different.
  - In total we have four quality constraints.

Material blending: formulation (5/5)

- The complete formulation is

$$
\begin{aligned}
\max \quad & 10(x_{11} + x_{21} + x_{31}) + 15(x_{12} + x_{22} + x_{32}) \\
& - 8(x_{11} + x_{12}) - 4(x_{21} + x_{22}) - 3(x_{31} + x_{32}) \\
\text{s.t.} \quad & x_{11} \geq 0.4(x_{11} + x_{21} + x_{31}), \quad x_{21} \geq 0.2(x_{11} + x_{21} + x_{31}) \\
& x_{12} \geq 0.5(x_{12} + x_{22} + x_{32}), \quad x_{13} \leq 0.3(x_{12} + x_{22} + x_{32}) \\
& x_{11} + x_{21} + x_{31} \leq 100, \quad x_{12} + x_{22} + x_{32} \leq 150 \\
& x_{ij} \geq 0 \quad \forall i = 1, ..., 3, j = 1, 2.
\end{aligned}
$$

  - We may need to **redefine** decision variables when necessary.
  - We may use **multi-dimensional variables**.
  - We need to **linearize** nonlinear constraints or objective functions.

Production and inventory (1/6)

- When we are making decisions, we may need to consider what will happen in the **future**.
- This creates **multi-period** problems.
- In particular, in many cases products produced today may be **stored** and then sold in the future.
  - Maybe production is cheaper today.
  - Maybe the price is higher in the future.
- So the production decision must be jointly considered with the **inventory** decision.

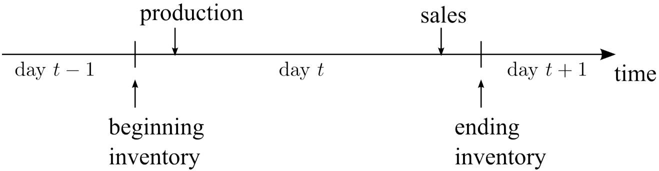
Production and inventory: the problem (2/6)

- Suppose we are going to produce and sell a product in four days.[3]
- For each day, there are different amounts of demands to fulfill.
  - Days 1, 2, 3, and 4: 100, 150, 200, and 170 units, respectively.
- The unit production costs are different for different days:
  - Days 1, 2, 3, and 4: \$9, \$12, \$10, and \$12 per unit, respectively.
- The prices are all **fixed**. So maximizing profits is the same as minimizing costs.
- We may store a product and sell it later.
  - The **inventory cost** is \$1 per unit per day.
  - E.g., producing 620 units on day 1 to fulfill all demands costs $9 \times 620 + 1 \times 150 + 2 \times 200 + 3 \times 170 = 6640$ dollars.

[3] *Operations Research: Applications and Algorithms* by W. Winston, 4th ed.

Production and inventory: the problem (3/6)

- Timing:



  - Beginning inventory + production − sales = ending inventory.
  - Inventory costs are assessed according to **ending inventory**.

Production and inventory: variables (4/6)

- We need to determine the production quantities. Let

$$x_t = \text{production quantity of day } t, t = 1, ..., 4.$$

- Is that information enough?
- So we also need to determine the inventory quantities. Let

$$y_t = \text{ending inventory of day } t, t = 1, ..., 4.$$

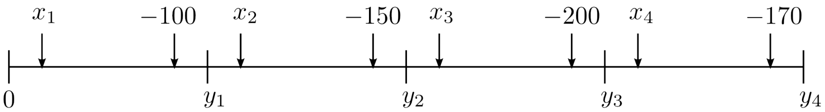
  - It is important to specify "ending"!

Production and inventory: formulation (5/6)

- The objective function is

$$\min\ 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4.$$

- We need to keep an eye on our inventory:
  - Day 1: $x_1 - 100 = y_1$.
  - Day 2: $y_1 + x_2 - 150 = y_2$.
  - Day 3: $y_2 + x_3 - 200 = y_3$.
  - Day 4: $y_3 + x_4 - 170 = y_4$.



- This is typically called the **inventory balancing** constraint.

Production and inventory: formulation (6/6)

- The complete formulation is

$$
\begin{aligned}
\min \quad & 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4 \\
\text{s.t.} \quad & x_1 - 100 = y_1 \\
& y_1 + x_2 - 150 = y_2 \\
& y_3 + x_3 - 200 = y_3 \\
& y_3 + x_4 - 170 = y_4 \\
& x_t, y_t \geq 0 \quad \forall t = 1, ..., 4.
\end{aligned}
$$

  - Is it guaranteed to satisfy all the demands?
- The main idea is to use inventory variables to **connect multiple periods**. Otherwise periods will be unconnected.
- In general, some constraints may be **redundant**.

Road map

- Basic ideas.
- LP formulation examples.
- **Advanced formulation techniques**
  - Compact formulation.
  - Linearization.

Compact formulations (1/5)

- Most problems in practice are of **large scales**.
  - The number of variables and constraints are huge.
- Many variables can be grouped together:
  - E.g., x_t = production quantity of day $t, t = 1, ..., 4$.
- Many constraints can be grouped together:
  - E.g., $x_t \geq 0$ for all $t = 1, ..., 4$.
- In modeling large-scale problems, we must use **compact formulations** to enhance readability and efficiency.
- In general, we may use the following three instruments:
  - Indices $(i, j, k, ...)$.
  - Summation $(\sum)$.
  - For all $(\forall)$.

Production and inventory (2/5)

- The problem:
  - We have four periods.
  - In each period, we first produce and then sell.
  - Unsold products become ending inventories.
  - Want to minimize the total cost.
- Indices:
  - Because things will **repeat in each period**, it is natural to use an index for periods. Let $t \in \{1, ..., 4\}$ be the index of periods.
- The objective function:
  - $\min\ 9x_1 + 12x_2 + 10x_3 + 12x_4 + y_1 + y_2 + y_3 + y_4$.
  - $\min\ 9x_1 + 12x_2 + 10x_3 + 12x_4 + \sum_{t=1}^{4} y_t$.
  - Denote the unit cost on day t as C_t, $t = 1, ..., 4$:

$$\min \sum_{t=1}^{4} (C_t x_t + y_t).$$

Compacting the constraints (3/5)

- The original constraints:
  - $x_1 - 100 = y_1$, $y_1 + x_2 - 150 = y_2$, $y_2 + x_3 - 200 = y_3$, $y_3 + x_4 - 170 = y_4$.
- Denote the demand on day t as D_t, $t = 1, ..., 4$.
- The compact constraint:
  - For $t = 2, ..., 4 : y_{t-1} + x_t - D_t = y_t$.
  - We cannot apply this to day 1 as y_0 is undefined!
  - For $t = 1$, $x_1 - D_t = y_1$.
- To group the four constraints into one compact constraint, we add y_0 as a decision variable:

$$y_t = \text{ending inventory of day } t, t = 0, ..., 4.$$

- Then the set of inventory balancing constraints are written as

$$y_{t-1} + x_t - D_t = y_t \quad \forall t = 1, ..., 4$$

  - Certainly we need to set up the initial inventory: $y_0 = 0$.

The complete compact formulation (4/5)

- The compact formulation is

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{4} (C_t x_t + y_t) \\
\text{s.t.} \quad & y_{t-1} + x_t - D_t = y_t \quad \forall t = 1, ..., 4 \\
& y_0 = 0 \\
& x_t, y_t \geq 0 \quad \forall t = 1, ..., 4.
\end{aligned}
$$

  - **Do not forget** "$\forall t = 1, ..., 4$"! Without that, the formulation is wrong.
  - Nonnegativity constraints for multiple sets of variables can be combined to save some "≥ 0".
- One convention is to:
  - Use **lowercase** letters for variables (e.g., x_t).
  - Use **uppercase** letters for parameters (e.g., C_t).

Parameter declaration (5/5)

- When creating parameter sets, it is fine to

  denote C_t as the unit production cost on day $t, t = 1, ..., 4$.

  - Do not need to specify values.
  - Need to specify **range** through **indices**.
- It is also fine to

  Denote $C = [9\ 12\ 10\ 12]$ as the production cost vector.

  - C_t is naturally its tth element and has no ambiguity.
  - The **values** should be indicated when defining the name.
- In either case, we should indicate the **physical meaning**.

Maximum and minimum functions (1/6)

- Maximum and minimum functions are nonlinear.
- If we are lucky enough, they can be **linearized** for us to construct an equivalent linear formulation.
- As the first example, how would you linearize

$$\max \ \min\{x_1, x_2\}?$$

Maximum and minimum functions (2/6)

- First attempt:

$$\begin{array}{rl} \max & y \\ \text{s.t.} & y = \min\{x_1, x_2\}. \end{array}$$

- Some observations:

$$\begin{array}{rcl} y = \min\{x_1, x_2\} & \Rightarrow & y \leq x_1, y \leq x_2 \\ y \leq x_1, y \leq x_2 & \Rightarrow & y \leq \min\{x_1, x_2\} \end{array}$$

- Second attempt:

$$\begin{array}{rl} \max & y \\ \text{s.t.} & y \leq x_1, y \leq x_2. \end{array}$$

  - The feasible region becomes larger. The two programs are not identical.
  - But **at any optimum**, either $y = x_1$ or $y = x_2$. Why?
  - So the two programs are equivalent.

Maximum and minimum functions (3/6)

- This technique can be applied on more general LPs.
- The following two problems are equivalent:

$$\begin{array}{rl} \max & \min\{x_1, 0\} \\ \text{s.t.} & x_1 + x_2 = 1 \\ & x_i \geq 0 \quad \forall i = 1, 2. \end{array} \quad \leftrightarrow \quad \begin{array}{rl} \max & y \\ \text{s.t.} & y \leq x_1 \\ & y \leq 0 \\ & x_1 + x_2 = 1 \\ & x_i \geq 0 \quad \forall i = 1, 2. \end{array}$$

- This technique works only for "max min" and "min max". For "min min" and "max max, it does not work!

Absolute value functions (4/6)

- An absolute value function can be viewed as a special maximum function: $|x| = \max\{x, -x\}$. So the above technique applies.
- The following three problems are equivalent:

$$\begin{array}{rl} \min & |x_1| \\ \text{s.t.} & x_1 + x_2 = 1 \\ & x_i \geq 0 \quad \forall i = 1,2 \end{array} \quad \leftrightarrow \quad \begin{array}{rl} \min & \max\{x_1, -x_1\} \\ \text{s.t.} & x_1 + x_2 = 1 \\ & x_i \geq 0 \quad \forall i = 1,2 \end{array}$$

$$\leftrightarrow \quad \begin{array}{rl} \min & y \\ \text{s.t.} & y \geq x_1 \\ & y \geq -x_1 \\ & x_1 + x_2 = 1 \\ & x_i \geq 0 \quad \forall i = 1,2. \end{array}$$

- This technique works only for "max min" and "min max". For "min min" and "max max, it does not work!

Locating fire stations (5/6)

- Consider the following problem of locating the fire station.[4]

  Monroe County is trying to determine where to place one county fire station. The locations of the county's four major towns are given in the following coordinates measured in miles. Town 1 is at $(10, 20)$*; town 2 is at* $(60, 20)$*; town 3 is at* $(40, 30)$*; town 4 is at* $(80, 60)$*. Town 1 averages 20 fires per year; town 2, 30 fires; town 3, 40 fires; and town 4, 25 fires. The county wants to build the fire station in a location that minimizes the average "distance to travel" of a fire engine. Since most roads run in either an east-west or a north-south direction, we assume that the fire engine can only do the same. Thus, if the fire station were located at* $(30, 40)$ *and a fire occurred at town 4, the "distance to travel" is* $|80 - 30| + |60 - 40| = 70$ *miles to the fire. Formulate a linear program that determines where the fire station should be located.*

[4] *Operations Research: Applications and Algorithms* by W. Winston, 4th ed.

Locating fire stations (6/6)

- Let (x, y) be the location of the fire station.
- Let (X_i, Y_i) be the location of the city i and F_i be the frequency of having fire in city i, $i = 1, ..., 4$.
- We solve
$$\min \sum_{i=1}^{4} F_i\Big(|x - X_i| + |y - Y_i|\Big).$$
- This can be linearized by introducing new variables w_i and z_i such that $w_i = |x - X_i|$ and $z_i = y - Y_i$:

$$\begin{array}{lll}\min & \sum_i F_i(w_i + z_i) & \\ \text{s.t.} & w_i \geq x - X_i & \forall i = 1, ..., 4 \\ & w_i \geq X_i - x & \forall i = 1, ..., 4 \\ & z_i \geq y - Y_i & \forall i = 1, ..., 4 \\ & z_i \geq Y_i - y & \forall i = 1, ..., 4.\end{array}$$