Chapter 3
Integer Linear Programming

Discrete Mathematics II

Le Hong Trang, Nguyen An Khuong, Huynh Tuong Nguyen
Faculty of Computer Science and Engineering
HCMC University of Technology

Contents

Transportation Problem — Linear Programming Model

Statement of an instance

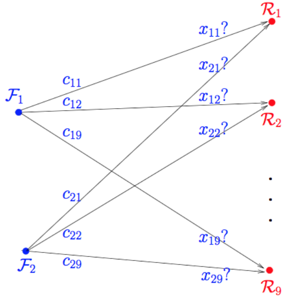
Suppose a company has 2 factories $\mathcal{F}_1, \mathcal{F}_2$ and 9 retail outlets $\mathcal{R}_1, \ldots, \mathcal{R}_9$:

- the total supply of the product from each factory $\mathcal{F}_1$ is a_i;
- the total demand for the product at each outlet $\mathcal{R}_j$ is b_j;
- The cost of sending one unit of the product from factory $\mathcal{F}_i$ to outlet $\mathcal{R}_j$ is equal to c_{ij},

where $i = 1, 2$ and $j = 1, 2, \ldots, 9$.
The problem is to determine a transportation scheme between the factories and the outlets so as to minimize the total transportation cost, subject to the specified supply and demand constraints..

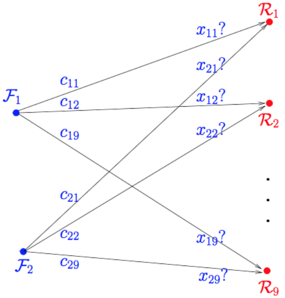
- Objective: minimum cost of transporting.
- Constraint: total supply of the factories and, total demand for the product of the outlets.
- Variable: the size of the shipment from $\mathcal{F}_i$ to $\mathcal{R}_j$, where $i = 1, 2$ and $j = 1, 2, \ldots, 9$.

Transportation Problem: Mathematical Formulation



- Variable: $x_{ij} \geq 0$.
- Objective
  - transportation cost from $\mathcal{F}_i$ to $\mathcal{R}_j$: $c_{ij}x_{ij}$,
  - objective function: $\sum_{ij} c_{ij}x_{ij}$.
- Constraints
  - total supply of $\mathcal{F}_i$: $\sum_{j=1}^{9} x_{ij} \leq a_i$,
  - total demand of $\mathcal{R}_j$: $\sum_{i=1}^{2} x_{ij} \geq b_j$.

Transportation Problem: Mathematical Formulation (cont.)



$$\min_x \quad \sum_{ij} c_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{9} x_{ij} \leq a_i, \quad i = 1, 2, \qquad (1)$$

$$\sum_{i=1}^{2} x_{ij} \geq b_j, \quad j = 1, \ldots, 9, \qquad (2)$$

$$x_{ij} \geq 0, i = 1, 2, \quad j = 1, \ldots, 9. \qquad (3)$$

Mathematical Formulation

Steps

1. Define decision variables.
2. Determine the objective function.
3. Establish constraints.

Knapsack Problems — Integer Linear Programming Model

There are many different knapsack problems. The first and classical one is the binary knapsack problem.

Binary (0-1) knapsack problem

A tourist is planning a tour in the mountains. He has a lot of objects which may be useful during the tour. For example ice pick and can opener can be among the objects. We suppose that

- Each object has a positive value and a positive weight, the value is the degree of contribution of the object to the success of the tour;
- The objects are independent from each other (e.g. can and can opener are not independent as any of them without the other one has limited value);
- The knapsack of the tourist is strong and large enough to contain all possible objects;
- The strength of the tourist makes possible to bring only a limited total weight;
- But within this weight limit the tourist want to achieve the maximal total value.

Mathematical Formulation

- The following notations are used to the mathematical formulation of the problem:

n	the number of objects;
j	the index of the objects;
w_j	the weight of object j;
v_j	the value of object j;
b	the maximal weight what the tourist can bring.

- For each object j a so-called *binary* or *zero-one* decision variable, say x_j, is introduced:

$$x_j = \begin{cases} 1 & \text{if object } j \text{ is present on the tour} \\ 0 & \text{if object } j \text{ isn't present on the tour.} \end{cases}$$

- Notice that

$$w_j x_j = \begin{cases} w_j & \text{if object } j \text{ is present on the tour,} \\ 0 & \text{if object } j \text{ isn't present on the tour} \end{cases}$$

is the weight of the object in the knapsack.

Mathematical Formulation (cont.)

- Similarly $v_j x_j$ is the value of the object on the tour. The total weight in the knapsack is

$$\sum_{j=1}^{n} w_j x_j$$

which may not exceed the weight limit.

- Hence the mathematical form of the problem is as follows.

$$\max_{x} \quad \sum_{j=1}^{n} v_j x_j \tag{4}$$

subject to

$$\sum_{j=1}^{n} w_j x_j \leq b, \tag{5}$$

$$x_j = 0 \text{ or } 1, \quad j = 1, \dots, n. \tag{6}$$

Solving the problem in this case

- The difficulty of the problem is caused by the integrality requirement. If constraint (6) is substituted by the relaxed constraint, i.e. by

$$0 \leq x_j \leq 1, \quad j = 1, \dots, n\,, \tag{7}$$

  then the Problem (4), (5), and (7) is a linear programming problem. (7) means that not only a complete object can be in the knapsack but any part of it.
- Moreover in this special case, it is not necessary to apply the simplex method or any other LP algorithm to solve it, as its optimal solution is described by

Theorem

Suppose: v_j, w_j $(j = 1, \dots, n)$-all positive, and satisfies

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \dots \geq \frac{v_n}{w_n}. \tag{8}$$

Then there is an index p $(1 \leq p \leq n)$ and an optimal sol. $\mathbf{x}^*$ s.t.

$$x_1^* = x_2^* = \dots = x_{p-1}^* = 1,\ x_{p+1}^* = x_{p+2}^* = \dots = x_{p+1}^* = 0.$$

Remarks

- Notice that there is only at most one non-integer component in $\mathbf{x}^*$. This property will be used at the numerical calculations.
- From the point of view of B&B the relation of the Problems (4), (5), and (6) and (4), (5), and (7) is very important. Any feasible solution of the first one is also feasible in the second one. But the opposite statement is not true.
- In other words the set of feasible solutions of the first problem is a proper subset of the feasible solutions of the second one. This fact has two important consequences:
  - The optimal value of the Problem (4), (5), and (7) is an upper bound of the optimal value of the Problem (4), (5), and (6).
  - If the optimal solution of the Problem (4), (5), and (7) is feasible in the Problem (4), (5), and (6) then it is the optimal solution of the latter problem as well.
- These properties are used in the course of the branch and bound method intensively.

Linear Program (LP)

- Recall the standard form of LP:

$$\min_{x} \quad c^T x$$

s.t.

$$Ax = b, \tag{9}$$

$$x \geq 0, \tag{10}$$

where $c \in \mathbb{R}^n$, A is an $m \times n$ matrix with full row rank, and $b \in \mathbb{R}^m$.

- A polyhedron is a set of the form $\{x \in \mathbb{R}^n | Bx \geq d\}$ for some matrix B.
- Let $P \in \mathbb{R}^n$ be a given polyhedron. A vector $x \in P$ is an extreme point of P if there does not exist $y, z \in P$, and $\lambda \in (0, 1)$ such that $x = \lambda y + (1 - \lambda)z$.

Linear Program in General form

- General form of LP:

$$\min_{x} \quad c^T x$$

s.t.

$$Ax = b, \tag{11}$$

$$Cx \leq d, \tag{12}$$

$$x \geq 0. \tag{13}$$

Question

Is there any way to transform a general form to standard one?

Integer Linear Program (ILP)

- An integer linear program is a linear program where variables are constrained to be integers.

$$\min_x \quad c^T x$$

s.t.

$$Ax = b, \tag{14}$$

$$x \geq 0 \text{ and } x \in \mathbb{Z}^n. \tag{15}$$

Question

Why do we consider the problem with only equality constraints but not inequality ones?

Remark

Often a mix is desired of integer and non-integer variables, called Mixed Integer Linear Programs (MILP).

Basic Solutions and Extreme Points

- Let $S = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$,the feasible set of LP. Since A is full row rank, if the feasible set is not empty, then we must $m \leq n$. Without loss of generality, we assume that $m < n$.
- Let $A = (B, N)$, where B is an $m \times m$ matrix with full rank, i.e., $det(B) \neq 0$. Then, B is called a *basis*.
- Let $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$. We have $Bx_B + Nx_N = b$. Setting $x_N = 0$ gives $x_B = B^{-1}b$. $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ is called a *basic solution*. x_B is called *basic variables*, x_N is called *nonbasic variables*.
- If the basic solution is also feasible, this is, $B^{-1}b \geq 0$, then $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ is said to be a *basic feasible solution*.

Basic Solutions and Extreme Points

- $\hat{x} \in S$ is an extreme point of S *if and only if* $\hat{x}$ is a basic feasible solution.
- Two extreme points are *adjacent* if they differ in only one basic variable.

Theorem (Basic theorem of LP)

Consider the linear program $\min\{c^T x | Ax = b, x \geq 0\}$. *If* S *has at least one extreme point and there exists an optimal solution, then there exists an optimal solution that is an extreme point.*

- The feasible set of standard form linear program has at least one extreme point.
- Therefore, we claim that the optimal value of a linear program is either $-\infty$, or is attained an extreme point (basic feasible solution) of the feasible set.

A Naive Algorithm for Solving Linear Program

- Let $\min\{c^T x | Ax = b, x \geq 0\}$ be a bounded linear program.
- Enumerate all bases $\mathcal{B} \in \{1, 2, \ldots, n\}$, $\binom{m}{n} = O(n^m)$, too many.
- Compute associated basic solution $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$.
- Return the one which has largest objective function value among the feasible basic solutions.
- Running time is $O(n^m \cdot m^3)$.

Question

Are there more efficient algorithms?

Simplex Method

- Simplex method was invented by George Dantzig (1914–2005) (father of linear programming).
- Suppose we have a basic feasible solution $\hat{x} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$, $A = (B, N)$.
- Let $x \in S$ be any feasible solution of the LP. Let $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ and $c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$. Then, $Bx_B + Nx_N = b$ and $x_B = B^{-1}b - B^{-1}Nx_N$. We have

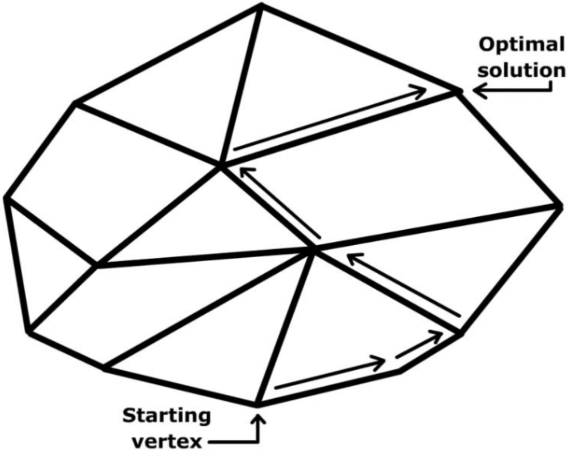
$$\begin{aligned} c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T B^{-1} b - c_B^T B^{-1} N x_N + c_N^T x_N \\ &= c^T \hat{x}^T + (c_N^T - c_B^T B^{-1} N) x_N. \end{aligned}$$

- Let $r_N = (c_N^T - c_B^T B^{-1} N)$, called *reduced cost*. If $r_N \geq 0$, then $c^T x \geq c^T \hat{x}^T$ and the current extreme point $\hat{x}$ is *optimal*.

Simplex Method

- Otherwise, there must exist an $r_i < 0$, we can let current nonbasic variable x_i become a basic one $x_i > 0$ (*entering variable*).
- Suitably choosing basic variable to become a nonbasic one (*leaving variable*), we can get a new basic feasible solution whose objective value is less than that of the current basic feasible solution $\hat{x}$.
- Geometrically, the simplex method moves from one extreme point to one of its *adjacent* extreme point.
- Since there are only a finite number of extreme points, the method terminates finitely at an *optimal solution* or detects that the problem is *infeasible* or it is *unbounded*.

Geometry of Simplex Method



Simplex Method

- Step 0: Compute an initial basis B and the basic feasible $B^{-1}b$ solution $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$.
- Step 2: if $r_N = (c_N^T - c_B^T B^{-1} N)$, **stop**, x is an optimal solution. Otherwise **goto** Step 2.
- Choose j satisfying $c_i^T - c_B^T B^{-1} a_j < 0$, if $\bar{a}_j = B^{-1} a_j \leq 0$, **stop**, the LP is infeasible. Otherwise, **goto** Step 3.
- Step 3: compute the step size

$$\lambda = \min\{\frac{\bar{b}_i}{\bar{a}_{ij}} | \bar{a}_{ij} > 0\} = \frac{\bar{b}_r}{\bar{a}_{rj}}$$

Let $x := x + \lambda d_j$, where $d_j = \begin{pmatrix} B^{-1} a_j \\ e_j \end{pmatrix}$. **goto** Step 1.

Simplex Tableau

x_B	x_N	rhs
B	N	b
c_B^T	c_N^T	0

It implies that

x_B	x_N	rhs
I	$B^{-1}N$	$B^{-1}b$
0	$c_N^T - c_B^T B^{-1} N$	$-c_B^T B^{-1} b$

Simplex Method: An example

- Consider the following LP

$$
\begin{aligned}
\min_{x} \quad & -7x_1 - 2x_2 \\
\text{s.t.} \quad & \\
& -x_1 + 2x_2 + x_3 = 4, \\
& 5x_1 + x_2 + x_4 = 20, \\
& 2x_1 + 2x_2 - x_5 = 7, \\
& x \geq 0.
\end{aligned}
$$

- The initial tableau should be

x_1	x_2	x_3	x_4	x_5	rhs
-1	2	1	0	0	4
5	1	0	1	0	20
2	2	0	0	-1	7
-7	-2	0	0	0	0

Simplex Method: example (cont.)

- Choose the initial basis to be $B = (a_1, a_3, a_4)$, we have basic $x_B = (x_1, x_3, x_4)^T$. The simplex tableau is then

x_1	x_2	x_3	x_4	x_5	rhs
0	3	1	0	$-\frac{1}{2}$	$7\frac{1}{2}$
0	-4	0	1	$2\frac{1}{2}$	$2\frac{1}{2}$
1	1	0	0	$-\frac{1}{2}$	$3\frac{1}{2}$
0	5	0	0	$-7\frac{1}{2}$	$24\frac{1}{2}$

- The basic feasible solution is $x_B = (x_1, x_3, x_4)^T = (3\frac{1}{2}, 7\frac{1}{2}, 2\frac{1}{2})^T$.
- Since $r_5 = -\frac{7}{2} < 0$, x_5 is chosen entering variable. $\lambda = \frac{2\frac{1}{2}}{2\frac{1}{2}} = 1$, then x_4 is leaving variable. The new basic variable should be $x_B = (x_1, x_3, x_5)^T$. The new tableau is obtained as below.

x_1	x_2	x_3	x_4	x_5	rhs
0	$\frac{11}{5}$	1	$\frac{1}{5}$	0	8
0	$-\frac{8}{5}$	0	$\frac{2}{5}$	1	1
1	$\frac{1}{5}$	0	$\frac{1}{5}$	0	4
0	$-\frac{5}{3}$	0	$\frac{7}{5}$	0	28

Simplex Method: example (cont.)

- Similarly, choose x_2 as the entering variable, then $\lambda = \min\{\frac{8}{11/5}, \frac{4}{1/5}\} = \frac{40}{11}$. Hence x_3 is leaving variable. Therefore, the tableau is

x_1	x_2	x_3	x_4	x_5	rhs
0	1	$\frac{5}{11}$	$\frac{1}{11}$	0	$\frac{40}{11}$
0	0	$\frac{8}{11}$	$\frac{6}{11}$	1	$\frac{75}{11}$
1	0	$-\frac{1}{11}$	$\frac{2}{11}$	0	$\frac{36}{11}$
0	0	$\frac{3}{11}$	$\frac{16}{11}$	0	$30\frac{2}{11}$

- Since $r_N = \left(\frac{3}{11}, \frac{16}{11}\right) \geq 0$, the current basic feasible solution $x = \left(\frac{36}{11}, \frac{40}{11}, 0, 0, \frac{75}{11}\right)^T$ is optimal with the optimal value is $-30\frac{2}{11}$.

Branch & Bound Method – Introduction

- This is the *divide and conquer* method. We divide a large problem into a few smaller ones. This is the *"branch"* part.
- The *conquering* part is done by estimate how good a solution we can get for each smaller problems.
    - To do so, we may have to divide the problem further, until we get a problem that we can handle, that is the *"bound"* part.
- We will use the *linear programming relaxation* to estimate the optimal solution of an integer programming.
- For an integer programming model $\mathcal{P}$, the linear programming model we get by dropping the requirement that all variables must be integers is called the linear programming relaxation of $\mathcal{P}$.

Numerical Example 1: Integer Programming

$$\max(Z = -x_1 + 4x_2) \quad (16)$$

Subject to

$$-10x_1 + 20x_2 \leq 22 \quad (17)$$

$$5x_1 + 10x_2 \leq 49 \quad (18)$$

$$x_1 \leq 5 \quad (19)$$

$$x_i \geq 0, x_i \in \mathbb{Z} \quad \forall i \in \{1, 2\} \quad (20)$$

Numerical Example 1 (cont') : Linear programming relaxation

With linear programming relaxation, we drop $x_i \in \mathbb{Z}$

$$\max(Z = -x_1 + 4x_2) \tag{21}$$

Subject to

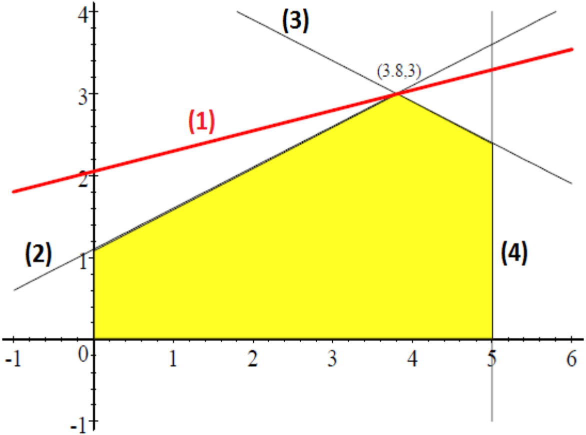
$$-10x_1 + 20x_2 \leq 22 \tag{22}$$

$$5x_1 + 10x_2 \leq 49 \tag{23}$$

$$x_1 \leq 5 \tag{24}$$

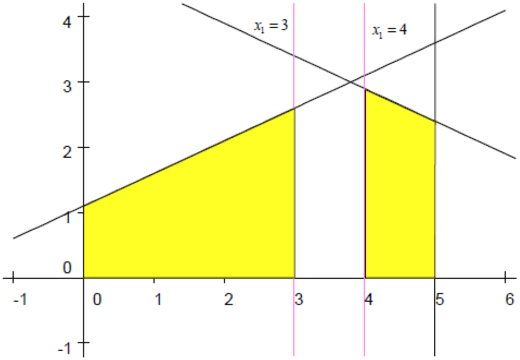
$$x_i \geq 0 \ \ \forall i \in \{1, 2\} \tag{25}$$

Numerical Example 1 (cont')



Optimal solution of relaxation is (3.8, 3) with Z = 8.2.

Numerical Example 1 (cont'): Linear programming relaxation



Since optimal solution of relaxation is (3.8, 3), we consider two cases: $x_1 \geq 4$ and $x_1 \leq 3$.

Example 1 (cont') – LP relaxation: $x_1 \geq 4$

Case $x_1 \geq 4$:

$$\max(Z = -x_1 + 4x_2) \tag{26}$$

Subject to

$$-10x_1 + 20x_2 \leq 22 \tag{27}$$

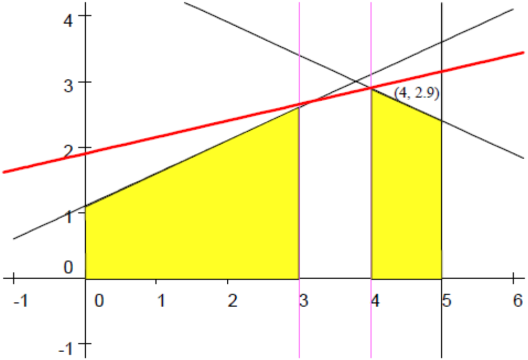
$$5x_1 + 10x_2 \leq 49 \tag{28}$$

$$x_1 \leq 5 \tag{29}$$

$$x_1 \geq 4 \tag{30}$$

$$x_2 \geq 0 \tag{31}$$

Numerical Example 1 (cont') – LP relaxation: $x_1 \geq 4$



has optimal solution at (4, 2.9) with Z = 7.6.
Then we consider two cases: $x_2 \geq 3$ and $x_2 \leq 2$.

Numerical Example 1 (cont') – LP relaxation: $x_1 \geq 4, x_2 \geq 3$

Case $x_1 \geq 4$, $x_2 \geq 3$:

$$\max(Z = -x_1 + 4x_2) \tag{32}$$

Subject to

$$-10x_1 + 20x_2 \leq 22 \tag{33}$$

$$5x_1 + 10x_2 \leq 49 \tag{34}$$

$$4 \leq x_1 \leq 5 \tag{35}$$

$$x_2 \geq 3 \tag{36}$$

has no feasible solution ($5x_1 + 10x_2 \geq 50$) so the IP has no feasible solution either.

Numerical Example 1 (cont') – LP relaxation: $x_1 \geq 4, x_2 \leq 2$

Case $x_1 \geq 4$, $x_2 \leq 2$:

$$\max(Z = -x_1 + 4x_2) \tag{37}$$

Subject to

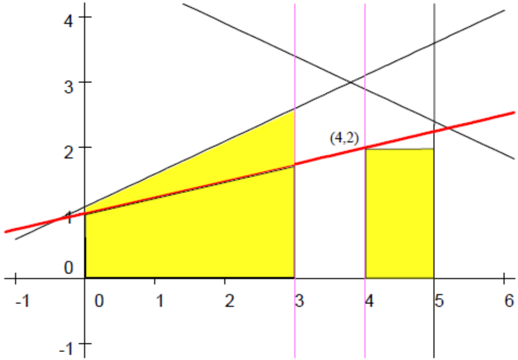
$$-10x_1 + 20x_2 \leq 22 \tag{38}$$

$$5x_1 + 10x_2 \leq 49 \tag{39}$$

$$4 \leq x_1 \leq 5 \tag{40}$$

$$0 \leq x_2 \leq 2 \tag{41}$$

Numerical Example 1 (cont') – LP relaxation: - $x_1 \geq 4, x_2 \leq 2$



has an optimal solution at (4, 2) with Z = 4.
This is the optimal solution of the IP as well. Currently, the best value of Z for the original IP is Z = 4.

Numerical Example 1 (cont') – LP relaxation: $x_1 \leq 3$

Come back case $x_1 \leq 3$:

$$\max(Z = -x_1 + 4x_2) \tag{42}$$

Subject to

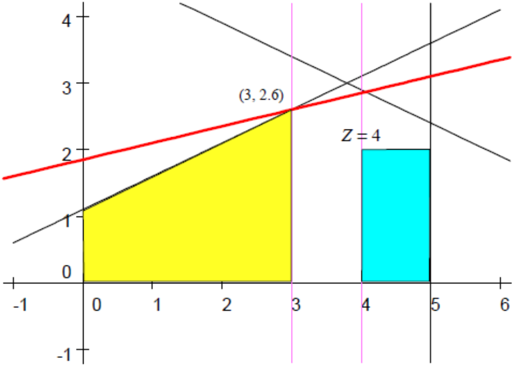
$$-10x_1 + 20x_2 \leq 22 \tag{43}$$

$$5x_1 + 10x_2 \leq 49 \tag{44}$$

$$0 \leq x_1 \leq 3 \tag{45}$$

$$x_2 \geq 0 \tag{46}$$

Numerical Example 1 (cont'): LP relaxation



has an optimal solution at (3, 2.6) with Z = 7.4.
We branch out further to two cases: $x_2 \geq 3$ and $x_2 \leq 2$.

Example 1 (cont') – LP relaxation: $x_1 \leq 3, x_2 \geq 3$

Case $x_1 \leq 3$, $x_2 \geq 3$:

$$\max(Z = -x_1 + 4x_2) \tag{47}$$

Subject to

$$-10x_1 + 20x_2 \leq 22 \tag{48}$$

$$5x_1 + 10x_2 \leq 49 \tag{49}$$

$$0 \leq x_1 \leq 3 \tag{50}$$

$$x_2 \geq 3 \tag{51}$$

has no feasible solution ($-10x_1$ $+20x_2 \geq$ 30).
The IP has no solution either.

Numerical Example 1 (cont') – LP relaxation: - $x_1 \leq 3, x_2 \leq 2$

Case $x_1 \leq 3$, $x_2 \leq 2$:

$$\max(Z = -x_1 + 4x_2) \tag{52}$$

Subject to

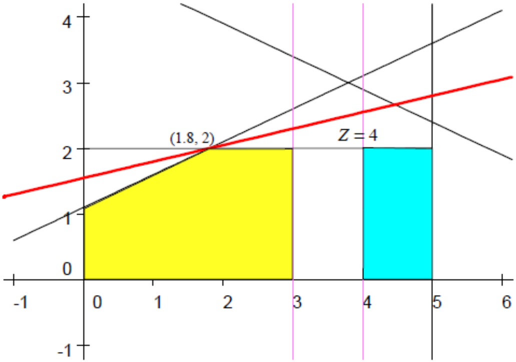
$$-10x_1 + 20x_2 \leq 22 \tag{53}$$

$$5x_1 + 10x_2 \leq 49 \tag{54}$$

$$0 \leq x_1 \leq 3 \tag{55}$$

$$0 \leq x_2 \leq 2 \tag{56}$$

Numerical Example 1 (cont') – LP relaxation: $x_1 \leq 3, x_2 \leq 2$



has an optimal at (1.8, 2) with Z = 6.2.
We branch further with two cases: $x_1 \geq 2$ or $x_1 \leq 1$

Numerical Example 1 (cont') – LP relaxation: $x_1 \geq 2, x_2 \leq 2$

Case $x_1 \geq 2$, $x_2 \leq 2$:

$$\max(Z = -x_1 + 4x_2) \tag{57}$$

Subject to

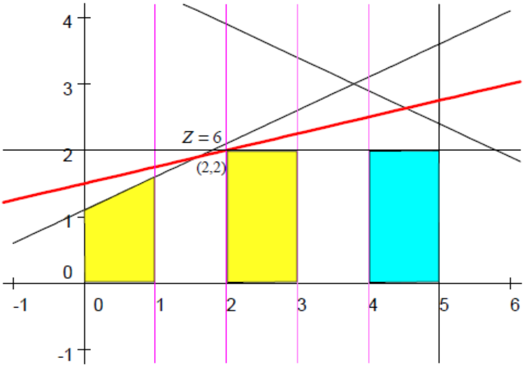
$$-10x_1 + 20x_2 \leq 22 \tag{58}$$

$$5x_1 + 10x_2 \leq 49 \tag{59}$$

$$2 \leq x_1 \leq 3 \tag{60}$$

$$0 \leq x_2 \leq 2 \tag{61}$$

Numerical Example 1 (cont') – LP relaxation: $x_1 \geq 2, x_2 \leq 2$



has an optimal at (2, 2), with Z = 6.
Since this is better than the incumbent Z = 4 at (4, 2), this new integer solution is our current best solution.

Numerical Example 1 (cont') – LP relaxation: $x_1 \leq 1, x_2 \leq 2$

Case $x_1 \leq 1$, $x_2 \leq 2$:

$$\max(Z = -x_1 + 4x_2) \tag{62}$$

Subject to

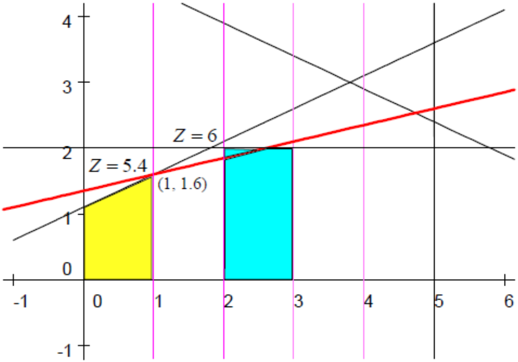
$$-10x_1 + 20x_2 \leq 22 \tag{63}$$

$$5x_1 + 10x_2 \leq 49 \tag{64}$$

$$0 \leq x_1 \leq 1 \tag{65}$$

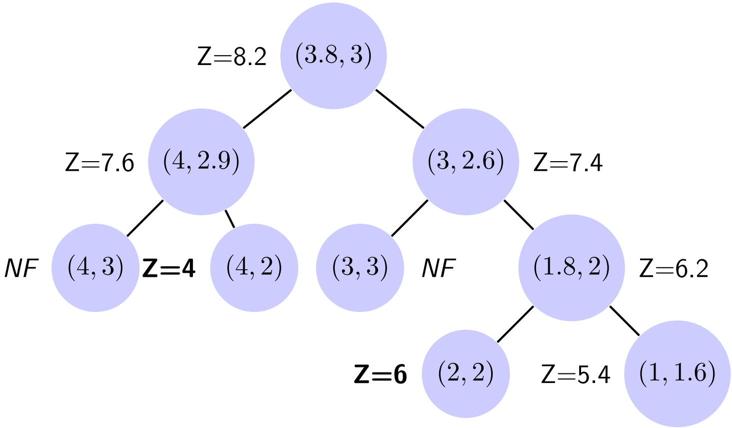
$$0 \leq x_2 \leq 2 \tag{66}$$

Numerical Example 1 (cont') – LP relaxation: $x_1 \leq 3, x_2 \leq 2$



has an optimal at (1, 1.6) with Z = 5.4.
Then any integer solution in this region can not give us a solution with the value of Z greater than 5.4. This branch is fathomed.

Numerical Example 1 (cont') – LP relaxation: conclusion



Numerical Example 2 – Binary IP Problem

$$\max(Z = 9x_1 + 5x_2 + 6x_3 + 4x_4) \quad (67)$$

Subject to

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \quad (68)$$

$$x_3 + x_4 \leq 1 \quad (69)$$

$$-x_1 + x_3 \leq 0 \quad (70)$$

$$-x_2 + x_4 \leq 0 \quad (71)$$

$$x_i \in \{0, 1\}, (x_i \in [0, 1], x_i \in \mathbb{Z}) \quad (72)$$

Numerical Example 2: LP Relaxation

$$\max(Z = 9x_1 + 5x_2 + 6x_3 + 4x_4) \tag{73}$$

Subject to

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \tag{74}$$

$$x_3 + x_4 \leq 1 \tag{75}$$

$$-x_1 + x_3 \leq 0 \tag{76}$$

$$-x_2 + x_4 \leq 0 \tag{77}$$

$$x_i \leq 1, for\ 1 \leq i \leq 4 \tag{78}$$

$$0 \leq x_i \tag{79}$$

has optimal solution at ($\frac{5}{6}$, 1, 0, 1) (why?-HW on Simplex Method!) with Z = 16.5

Has two branch, $x_1 = 0$ or $x_1 = 1$

Numerical Example 2 – LP Relaxation: $x_1 = 0$

With $x_1 = 0$, problem becomes

$$\max(Z = 5x_2 + 4x_4) \tag{80}$$

Subject to

$$\begin{aligned} 3x_2 + 2x_4 &\leq 10 \\ x_3 + x_4 &\leq 1 \\ -x_2 + x_4 &\leq 0 \\ x_i &\in \{0, 1\} \end{aligned}$$

has the optimal solution at (0, 1, 0, 1) with Z = 9. (Current best solution)

Numerical Example 2 – LP Relaxation: $x_1 = 1$

With $x_1 = 1$, the LP relaxation

$$\max(Z = 9 + 5x_2 + 6x_3 + 4x_4) \tag{81}$$

Subject to

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 4 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ -x_2 + x_4 &\leq 0 \\ x_i &\leq 1 \; for \; 2 \leq i \leq 4 \\ 0 &\leq x_i \end{aligned}$$

has the optimal solution at (1, 0.8, 0, 0.8) with Z = 16.2
Branch: $x_2 = 0$ or $x_2 = 1$

Numerical Example 2 – LP Relaxation: $x_1 = 1$, $x_2 = 0$

In this case, we have $x_4 = 0$ as well

$$\max(Z = 9 + 6x_3) \tag{82}$$

Subject to

$$\begin{aligned} 5x_3 &\leq 4 \\ x_3 &\leq 1 \\ 0 &\leq x_3 \end{aligned}$$

has the optimal solution at (1, 0, 0.8, 0) with Z = 13.8
Branch: $x_3 = 0$ or $x_3 = 1$

- With $x_3 = 0$ The optimal solution is (1, 0, 0, 0) with Z = 9 (not better than current solution)
- With $x_3 = 1$. Have no feasible solution

Numerical Example 2 – LP Relaxation: $x_1 = 1$, $x_2 = 1$

With $x_1 = 0$, the LP relaxation

$$\max(Z = 14 + 6x_3 + 4x_4) \tag{83}$$

Subject to

$$\begin{aligned} 5x_3 + 2x_4 &\le 1 \\ x_3 + x_4 &\le 1 \\ 0 \le x_3 &\le 1 \\ 0 \le x_4 &\le 1 \end{aligned}$$

has the optimal solution at (1, 1, 0, 0.5) with Z = 16

Numerical Example 2 – LP Relaxation: $x_1 = 1$, $x_2 = 1$

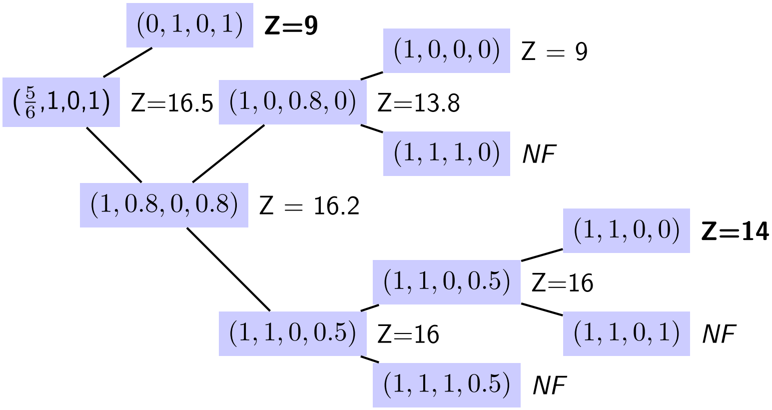
$$\max(Z = 14 + 6x_3 + 4x_4) \tag{84}$$

$$\begin{aligned} S.t. \qquad 5x_3 + 2x_4 &\leq 1 \\ x_3 + x_4 &\leq 1 \\ 0 \leq x_3 &\leq 1 \\ 0 \leq x_4 &\leq 1 \end{aligned}$$

- With $x_3 = 0$, Z = 16 is still feasible solution at (1, 1, 0, 0.5)
  - With $x_4 = 0$, (1, 1, 0, 0), Z = 14 (new optimal solution)
  - With $x_4 = 1$, (1, 1, 0, 1) is not feasible
- With $x_3 = 1$, No feasible solution

$\Rightarrow$ The current best solution (1, 1, 0, 0) with $Z = 14$ is the optimal solution.

Numerical Example 2 (cont') – LP relaxation: conclusion



Branch & Bound: General Strategy

- Solve the linear relaxation of the problem. If the solution is integer, then we are done. Otherwise create two new subproblems by branching on a fractional variable.
- A node (subproblem) is not active when any of the following occurs:
  1. The node is being branched on;
  2. The solution is integral;
  3. The subproblem is infeasible;
  4. You can fathom the subproblem by a bounding argument.
- Choose an active node and branch on a fractional variable. Repeat until there are no active subproblems.

Branch & Bound: Steps

- Omitted the integrality constraints from the original problem in order to obtain a relaxation. Thus a LP problem is obtained and solve this LP problem.
- Divide a problem into subproblems
- Calculate the LP relaxation of a subproblem
  - The LP problem has no feasible solution, done;
  - The LP problem has an integer optimal solution; done.
  - Compare the optimal solution with the best solution we know (the incumbent).
  - The LP problem has an optimal solution that is worse than the incumbent, done.

    In all the cases above, we know all we need to know about that subproblem. We say that subproblem is fathomed.
  - The LP problem has an optimal solution that are not all integer, better than the incumbent. In this case we would have to divide this subproblem further and repeat.

Exercise: An 0-1 Knapsack Problem Instance

- Consider the problem

$$\max_x \quad 8x_1 + 11x_2 + 6x_3 + 4x_4$$

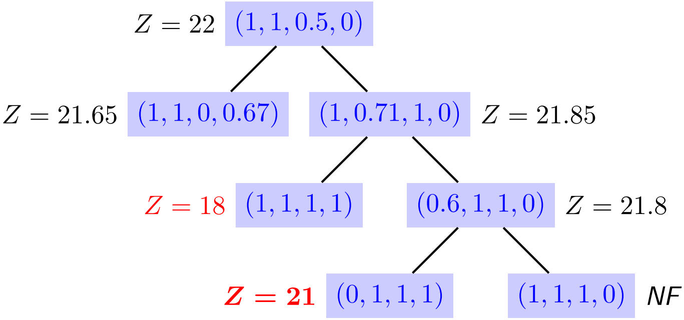
s.t.

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14,$$
$$x \in \{0,1\}^4.$$

- The linear relaxation solution is $x = (1, 1, 0.5, 0)$ with a value of 22. The solution is not integral.
- Choose x_3 to branch. The next two subproblems will have $x_3 = 0$ and $x_3 = 1$, respectively.
- ...

Exercise: An 0-1 Knapsack Problem Instance

- The tree search:



- Theo ptimal solution is $\boldsymbol{x = (0, 1, 1, 1)}$.

How to Branch?

- We want to divide the current problem into two or more subproblems that are easier than the original. A commonly used branching method:

$$x_i \leq \lfloor x_i^* \rfloor, x_i \geq \lceil x_i^* \rceil,$$

  where x_i^* is a fractional variable.
- Which variable to branch? A commonly used branching rule: Branch the most fractional variable.
- We would like to choose the branching that minimizes the sum of the solution times of all the created subproblems.
- How do we know how long it will take to solve each subproblem?
  - Answer: We don't.
  - Idea: Try to predict the difficulty of a subproblem.
- A good branching rule: The value of the linear programming relaxation changes a lot.

Which Node to Select?

- An important choice in branch and bound is the strategy for selecting the next subproblem to be processed.
- Goals: (i) Minimizing overall solution time; (ii) Finding a good feasible solution quickly.
- Some commonly used search strategies:
  - Best First,
  - Depth-First.

The Best First Approach

- One way to minimize overall solution time is to try to minimize the size of the search tree. We can achieve this by choosing the subproblem with the *best bound* (lowest lower bound if we are minimizing).
- *Drawbacks* of Best First
  - Doesn't necessarily find feasible solutions quickly since feasible solutions are "more likely" to be found deep in the tree.
  - Node setup costs are high. The linear program being solved may change quite a bit from one node evaluation to the next.
  - Memory usage is high. It can require a lot of memory to store the candidate list, since the tree can grow "broad".

The Depth-First Approach

- The depth first approach is to always choose the deepest node to process next. Just dive until you prune, then back up and go the other way.
- This avoids most of the problems with best first: The number of candidate nodes is minimized (saving memory). The node set-up costs are minimized.
- LPs change very little from one iteration to the next. Feasible solutions are usually found quickly.
- *Drawback*: If the initial lower bound is not very good, then we may end up processing lots of non-critical nodes.
- Hybrid Strategies: Go depth-first until you find a feasible solution, then do best-first search.

Rule of Fathoming

A subproblem is fathomed when

1. The relaxation of the subproblem has an optimal solution with $z < z^*$ where z^* is the current best solution;
2. The relaxation of the subproblem has no feasible solution;
3. The relaxation of the subproblem has an optimal solution that has all integer values (or all binary if it is an BIP).

Software for ILP

- Matlab optimization toolbox: `bintprog` – a built-in function for mixed integer linear programming.
- Some of softwares which are proprietary but *free for academic use*:
  - `CPLEX`: `www-03.ibm.com/.../ibmilogcpleoptistud`
  - `Gurobi`: `gurobi.com`
  - `MOSEK`: `www.mosek.com`
  - `SCIP`: `scip.zib.de`

  These softwares can also be used as solvers in some modeling tools.
- Many open-source solvers were developed in the literature.

Software for ILP: Modeling Tools

- CVX: `cvxr.com/cvx`
- YALMIP: `yalmip.github.io`
- TOMLAB: `tomopt.com/tomlab`

Questions

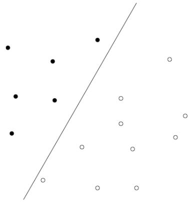
- What is a modeling tool?
- How does it work?
- How do we use it?

→ *See the next slide for an example of the use of CVX.*

Example: Solving the Linear Classification Problem using CVX

Statement

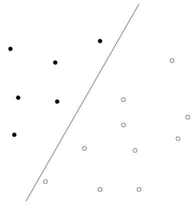
Given two sets of points in $\mathbb{R}^n$, denoted by $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_M\}$, we need to classify the points.



Its applications in Computer Science

- Machine learning,
- Pattern recognition,
- Data mining,
- ...

Solving the Linear Classification Problem using CVX



- Use a hyperplane $a^T x + b = 0$, hence the followings must be satisfied:

$$a^T x_i + b > 0, i = 1, \ldots, N \text{ and } \quad a^T y_i + b < 0, i = 1, \ldots, M.$$

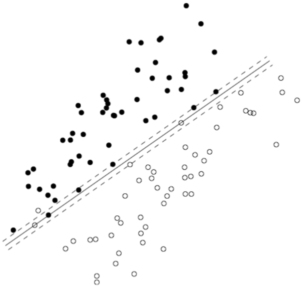
- Since the strict inequalities are homogeneous in a and b, they are feasible iff the set of nonstrict linear inequalities:

$$a^T x_i + b > 1, i = 1, \ldots, N \text{ and } \quad a^T y_i + b < -1, i = 1, \ldots, M.$$

Solving the Linear Classification Problem using CVX

Classification for inseparable sets

$$
\begin{aligned}
\text{min} \quad & \mathbf{1}^T u + \mathbf{1}^T u \\
\text{s.t.} \quad & a^T x_i + b \geq 1 - u_i, i = 1, \ldots, N, \\
& a^T y_i + b \leq -1 + v_i, i = 1, \ldots, M, \\
& u, v \succeq 0.
\end{aligned}
$$

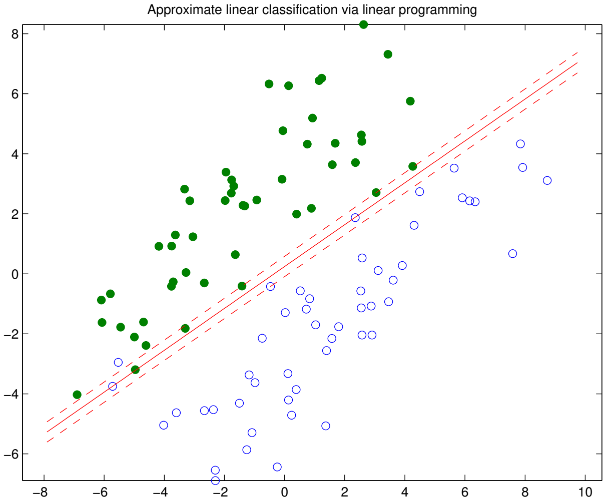


Solving the Linear Classification Problem using CVX

```
n = 2;
randn('state',2);
N = 50; M = 50;
Y = [1.5+0.9*randn(1,0.6*N), 1.5+0.7*randn(1,0.4*N);
     2*(randn(1,0.6*N)+1), 2*(randn(1,0.4*N)-1)];
X = [-1.5+0.9*randn(1,0.6*M),  -1.5+0.7*randn(1,0.4*M);
      2*(randn(1,0.6*M)-1), 2*(randn(1,0.4*M)+1)];
T = [-1 1; 1 1];
Y = T*Y;  X = T*X;

% Solution via CVX
cvx_begin
    variables a(n) b(1) u(N) v(M)
    minimize (ones(1,N)*u + ones(1,M)*v)
    X'*a - b >= 1 - u;
    Y'*a - b <= -(1 - v);
    u >= 0;
    v >= 0;
cvx_end
```

Solving the Linear Classification Problem using CVX



Sudoku Game

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Problem statement

Given a grid of 9×9, the grid is divided into 9 subgrids of 3×3. Some cells of subgrids are filled by digits in $\{1, 2, \ldots, 9\}$. The objective is to fill remaining cells such that

- each column, each row, and each subgrid that compose the grid, contain all of the digits from 1 to 9.

Sudoku Game: Model

$$\text{Variable } x_{ijk} = \begin{cases} 1 & \text{if cell } (i,j) \text{ contains } k; \\ 0 & \text{otherwise.} \end{cases}$$

$$\min_{x} \quad \mathbf{0}^T \boldsymbol{x}$$

subject to

$$\sum_{i=1}^{9} x_{ijk} = 1, \quad j,k = 1,\dots,9,$$

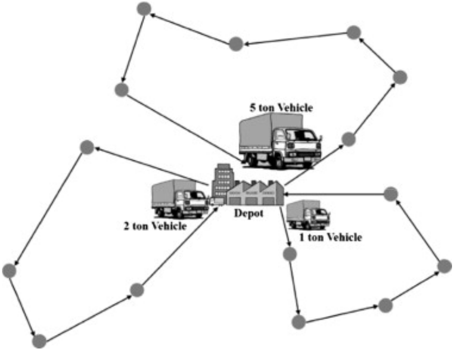
$$\sum_{j=1}^{9} x_{ijk} = 1, \quad i,k = 1,\dots,9,$$

$$\sum_{k=1}^{9} x_{ijk} = 1, \quad i,j = 1,\dots,9,$$

$$\sum_{j=3q+2}^{3q} \sum_{i=3p+2}^{3p} x_{ijk} = 1, \quad k = 1,\dots,9; p,q = 1,2,3,$$

$$x_{ijk} \in \{0,1\}, \quad x_{ijk} = 1, \quad \forall (i,j,k) \in G.$$

Vehicle Routing



Vehicle Routing

Problem statement

Let $G = (V, E)$ be a complete graph, where $V = 0, \ldots, n$ is the set of vertices and E is the set of edges.

- We assume that vertex 0 denotes the depot, whereas vertex $i = 1, \ldots, n$ is customer.
- A nonnegative value, denoted by c_{ij}, is the *cost* associated with each edge of G that connects vertex i and j.
- Let $S \subseteq V$, we denote by $r(S)$ the minimum number of vehicles required for all customers in S.

Given a positive integer number k, the capacitated vehicle routing problem is asked for seeking k simple routes that *minimizing a cost function of the sum of cost of edges of the routes, and satisfying simultaneously:*

(i) each route starts at the depot vertex,

(ii) each customer belongs exactly one route,

(iii) the sum of demands of the vertices visited by a route is always lower than the capacity C of the vehicle.

Vehicle Routing: Model

- We use $x_{ij} \in \{0,1\}$ to indicates if the edge connecting vertex i and j, is an edge of the solution.

$$\min_x \sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij}$$

subject to

$$\sum_{i\in V} x_{ij} = 1 \quad \text{for all } j \in V \setminus \{0\},$$

$$\sum_{j\in V} x_{ij} = 1 \quad \text{for all } i \in V \setminus \{0\},$$

$$\sum_{i\in V} x_{i0} = k,$$

$$\sum_{j\in V} x_{0j} = k,$$

$$\sum_{i\notin S}\sum_{j\in S} x_{ij} \geq r(S) \quad \text{for all } S \subseteq V \setminus \{0\}, S \neq \emptyset,$$

$$x_{ij} \in \{0,1\} \quad \text{for all } i,j \in V.$$

Facility Location — Home?

Problem statement

We need to build warehouses for a set of supermarkets.

- The locations of the supermarkets have been decided, but the locations of the warehouses are yet to be chosen.
- Several good candidate locations for the warehouses have been determined, but it remains to decide how many warehouses to open and at which candidate locations to build them.
- Opening many warehouses would be advantageous as this would reduce the average distance a truck has to drive from warehouse to supermarket, and hence reduce the delivery cost. However, opening a warehouse is costly.

How can we the build warehouses so that the tradeoff between delivery cost and the cost of building new facilities is optimal?

Homeworks

- Simplex Method: details.
- Read carefully and do all Examples, Exercises in Chapter 7 of [4]: F.R. Giordano, W.P. Fox and S.B. Horton, A First Course in Mathematical Modeling, 5th ed., Cengage, 2014.
- Do all Exercises in the file named "BT Chương 3.pdf".