

DM545/DM871

Linear and Integer Programming

Lecture 8

# Integer Linear Programming Modeling

Marco Chiarandini

Department of Mathematics & Computer Science  
University of Southern Denmark

# Midway evaluation

What is working well and what is not:

<https://padlet.com/marco43/dm545>

<https://padlet.com/marco43/dm871>

# Outline

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks

# Outline

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks

# Discrete Optimization

- Often we need to deal with integral inseparable quantities
- Sometimes rounding can go
- Other times rounding not feasible: eg, presence of a bus on a line is 0.3...

# Integer Linear Programming

Linear Objective  
Linear Constraints  
but! integer variables

The world is not linear: "OR is the art and science of obtaining bad answers to questions to which otherwise worse answers would be given"

$$\begin{aligned} \max c^T x \\ Ax \leq b \\ x \geq 0 \end{aligned}$$

Linear Programming  
(LP)

$$\begin{aligned} \max c^T x \\ Ax \leq b \\ x \geq 0 \\ x \text{ integer} \end{aligned}$$

Integer (Linear) Programming  
(ILP)

$$\begin{aligned} \max c^T x \\ Ax \leq b \\ x \in \{0, 1\}^n \end{aligned}$$

Binary Integer Program  
(BIP)  
0/1 Integer Programming

$$\begin{aligned} \max c^T x + h^T y \\ Ax + Gy \leq b \\ x \geq 0 \\ y \geq 0 \\ y \text{ integer} \end{aligned}$$

Mixed Integer (Linear)  
Programming (MILP)

$$\begin{aligned} \max f(x) \\ g(x) \leq b \\ x \geq 0 \end{aligned}$$

Non-linear Programming (NLP)

Recall:

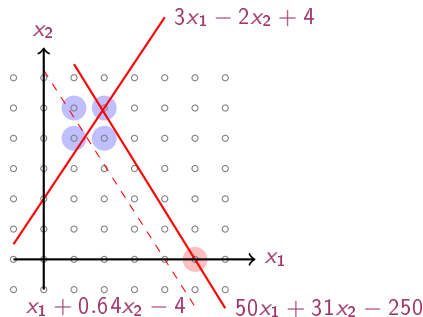
- $\mathbb{Z}$  set of integers
- $\mathbb{Z}^+$  set of positive integer
- $\mathbb{Z}_0^+$  set of nonnegative integers ( $\{0\} \cup \mathbb{Z}^+$ )
- $\mathbb{N}_0$  set of natural numbers, ie, nonnegative integers  $\{0, 1, 2, 3, 4, \dots\}$

# Rounding

$$\begin{aligned} \max \quad & 100x_1 + 64x_2 \\ & 50x_1 + 31x_2 \leq 250 \\ & 3x_1 - 2x_2 \geq -4 \\ & x_1, x_2 \in \mathbb{Z}_0^+ \end{aligned}$$

LP optimum  $(376/193, 950/193)$

IP optimum  $(5, 0)$



Note: rounding does not help in the example above!

↪ feasible region convex but not continuous: Now the optimum can be on the border (vertices) but also [internal](#).

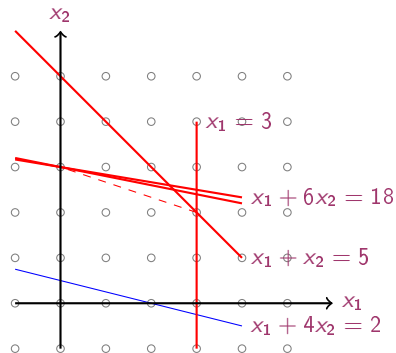
Possible way: solve the [relaxed](#) problem.

- If solution is [integer](#), done.
- If solution is [rational](#) (never irrational) try rounding to the nearest integers (but may exit feasibility region)
  - if in  $\mathbb{R}^2$  then  $2^2$  possible roundings (up or down)
  - if in  $\mathbb{R}^n$  then  $2^n$  possible roundings (up or down)



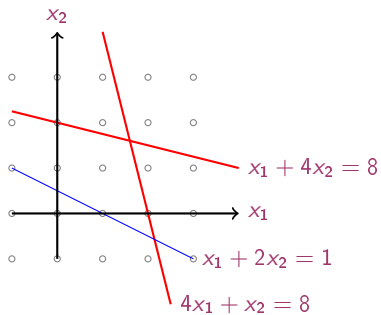
# Cutting Planes

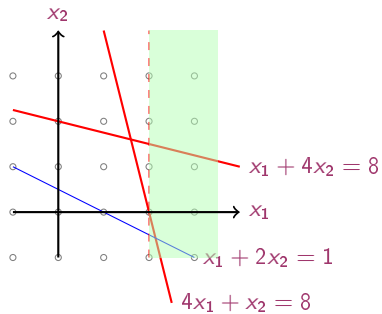
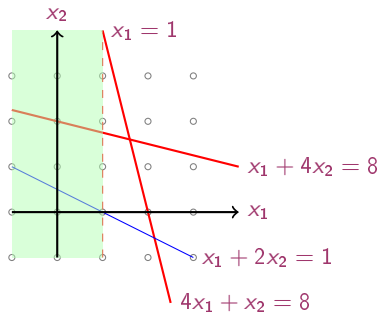
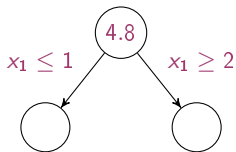
$$\begin{aligned} \max \quad & x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + 6x_2 \leq 18 \\ & x_1 \leq 3 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \text{ integers} \end{aligned}$$

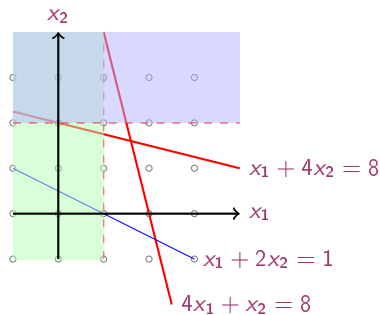
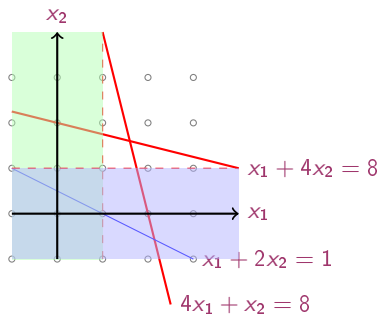
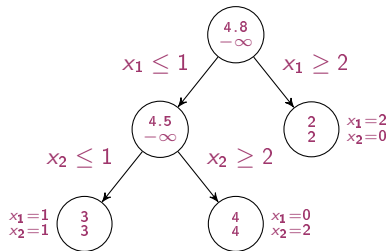


# Branch and Bound

$$\begin{aligned} \max \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 8 \\ & 4x_1 + x_2 \leq 8 \\ & x_1, x_2 \geq 0, \text{ integer} \end{aligned}$$







# Outline

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks

# Mathematical Programming: Modeling

- Find out exactly what the decision maker needs to know:
  - which investment?
  - which product mix?
  - which job  $j$  should a person  $i$  do?
- Define **Decision Variables** of suitable type (continuous, integer valued, binary) corresponding to the needs and **Known Parameters** corresponding to given data.
- Formulate **Objective Function** computing the benefit/cost
- Formulate mathematical **Constraints** indicating the interplay between the different variables.

# How to “build” a constraint

- Formulate relationship between the variables in plain words
- Then formulate your sentences using logical connectives **and**, **or**, **not**, **implies**
- Finally convert the logical statement to a mathematical constraint.

## Example

- “The power plant must not work in both of two neighbouring time periods”
- on/off is modelled using **binary** integer variables
- $x_i = 1$  or  $x_i = 0$
- $x_i = 1$  implies  $\Rightarrow x_{i+1} = 0$
- $x_i + x_{i+1} \leq 1$

# Outline

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks



# The Assignment Problem

## Problem

Common application: **Assignees** are being assigned to perform **tasks**.

Suppose we have  $n$  persons and  $n$  jobs

Each person has a certain proficiency at each job.

Formulate a mathematical model that can be used to find an assignment that maximizes the total proficiency.

# The Assignment Problem

## Model

### Decision Variables:

$$x_{ij} = \begin{cases} 1 & \text{if person } i \text{ is assigned job } j \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } i, j = 1, 2, \dots, n$$

### Objective Function:

$$\max \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} x_{ij}$$

where  $\rho_{ij}$  is person  $i$ 's proficiency at job  $j$

# The Assignment Problem

## Model

### Constraints:

Each person is assigned one job:

$$\sum_{j=1}^n x_{ij} = 1 \text{ for all } i$$

e.g. for person 1 we get  $x_{11} + x_{12} + x_{13} + \cdots + x_{1n} = 1$

Each job is assigned to one person:

$$\sum_{i=1}^n x_{ij} = 1 \text{ for all } j$$

e.g. for job 1 we get  $x_{11} + x_{21} + x_{31} + \cdots + x_{n1} = 1$

# Outline

## 1. Integer Programming

## 2. Modeling

Assignment Problem

**Knapsack Problem**

Set Problems

## 3. More on Modeling

Graph Problems

Modeling Tricks

# The Knapsack Problem

## Problem ..

**Input:** Given a set of  $n$  items, each with a value  $v_i$  and weight  $w_i$  ( $i = 1, \dots, n$ )

**Task:** determine the number of each items to include in a collection so that the total weight is less than a given limit,  $W$ , and the total value is as large as possible.

The “knapsack” name derives from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most useful items.

Assuming we can take at most one of any item and that  $\sum_i w_i > W$ , formulate a mathematical model to determine which items give the largest value.

Model used, eg, in capital budgeting, project selection, etc.

# The Knapsack Problem

## Decision Variables:

$$x_i = \begin{cases} 1 & \text{if item } i \text{ is taken} \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } i = 1, 2, \dots, n$$

## Objective Function:

$$\max \sum_{i=1}^n v_i x_i$$

## Constraints:

Knapsack capacity restriction:

$$\sum_{i=1}^n w_i x_i \leq W$$

# Outline

## 1. Integer Programming

## 2. Modeling

Assignment Problem

Knapsack Problem

**Set Problems**

## 3. More on Modeling

Graph Problems

Modeling Tricks

# Set Covering

## Problem

**Given:** a set of regions, a set of possible construction locations for emergency centers, regions that can be served in less than 8 minutes, cost of installing an emergency center in each location.

**Task:** decide where to install a set of emergency centers such that the total cost is minimized and all regions are safely served

**As a COP:**  $M = \{1, \dots, m\}$  regions,  $N = \{1, \dots, n\}$  centers,  $S_j \subseteq M$  regions serviced by  $j \in N$  in 8 min.

$$\min_{T \subseteq N} \left\{ \sum_{j \in T} c_j \mid \bigcup_{j \in T} S_j = M \right\}$$

regions:  $M = \{1, \dots, 5\}$

centers:  $N = \{1, \dots, 6\}$

cost of centers:  $c_j = 1 \quad \forall j = 1, \dots, 6$

coverages:  $S_1 = (1, 2), S_2 = (1, 3, 5), S_3 = (2, 4, 5), S_4 = (3), S_5 = (1), S_6 = (4, 5)$



## Example

- regions:  $M = \{1, \dots, 5\}$   
centers:  $N = \{1, \dots, 6\}$   
cost of centers:  $c_j = 1 \quad \forall j = 1, \dots, 6$   
coverages:  $S_1 = (1, 2), S_2 = (1, 3, 5), S_3 = (2, 4, 5), S_4 = (3), S_5 = (1), S_6 = (4, 5)$

•

$$A = \begin{array}{c} \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \\ S_1 \quad S_2 \quad S_3 \quad S_4 \quad S_5 \quad S_6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \left[ \begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right] \end{array}$$

**As a BIP:**

**Variables:**

$x \in \mathbb{B}^n$ ,  $x_j = 1$  if center  $j$  is selected,  $0$  otherwise

**Objective:**

$$\min \sum_{j=1}^n c_j x_j$$

**Constraints:**

- incidence matrix:  $a_{ij} = \begin{cases} 1 \\ 0 \end{cases}$
- $\sum_{j=1}^n a_{ij} x_j \geq 1$

### Set covering

cover each of  $M$  at least once

1.  $\min$ ,  $\geq$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \min c^T x \\ Ax \geq 1 \\ x \in \mathbb{B}^n \end{aligned}$$

### Set packing

cover as many of  $M$  without overlap

1.  $\max$ ,  $\leq$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \max c^T x \\ Ax \leq 1 \\ x \in \mathbb{B}^n \end{aligned}$$

### Set partitioning

cover exactly once each element of  $M$

1.  $\max$  or  $\min$ ,  $=$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \max c^T x \\ Ax = 1 \\ x \in \mathbb{B}^n \end{aligned}$$

Generalization:  $RHS \geq 1$

Application examples:

- Aircrew scheduling:  $M$ : legs to cover,  $N$ : rosters
- Vehicle routing:  $M$ : customers,  $N$ : routes

# A good written example of how to present a model:

## 2.1. Notation

Let  $N$  be the set of operational flight legs and  $K$  the set of aircraft types. Denote by  $n^k$  the number of available aircraft of type  $k \in K$ . Define  $\Omega^k$ , indexed by  $p$ , as the set of feasible schedules for aircraft of type  $k \in K$  and let index  $p = 0$  denote the empty schedule for an aircraft. Next associate with each schedule  $p \in \Omega^k$  the value  $c_p^k$  denoting the anticipated profit if this schedule is assigned to an aircraft of type  $k \in K$  and  $a_{ip}^k$  a binary constant equal to 1 if this schedule covers flight leg  $i \in N$  and 0 otherwise. Furthermore, let  $S$  be the set of stations and  $S^k \subseteq S$  the subset having the facilities to serve aircraft of type  $k \in K$ . Then, define  $o_{sp}^k$  and  $d_{sp}^k$  to equal to 1 if schedule  $p$ ,  $p \in \Omega^k$ , starts and ends respectively at station  $s$ ,  $s \in S^k$ , and 0 otherwise.

Denote by  $\theta_p^k$ ,  $p \in \Omega^k \setminus \{0\}$ ,  $k \in K$ , the binary decision variable which takes the value 1 if schedule  $p$  is assigned to an aircraft of type  $k$ , and 0 otherwise. Finally, let  $\theta_0^k$ ,  $k \in K$ , be a nonnegative integer variable which gives the number of unused aircraft of type  $k$ .

## 2.2. Formulation

Using these definitions, the DARSP can be formulated as:

$$\text{Maximize } \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k \theta_p^k \quad (1)$$

subject to:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k \theta_p^k = 1 \quad \forall i \in N, \quad (2)$$

$$\sum_{p \in \Omega^k} (d_{sp}^k - o_{sp}^k) \theta_p^k = 0 \quad \forall k \in K, \forall s \in S^k, \quad (3)$$

$$\sum_{p \in \Omega^k} \theta_p^k = n^k \quad \forall k \in K, \quad (4)$$

$$\theta_p^k \geq 0 \quad \forall k \in K, \forall p \in \Omega^k, \quad (5)$$

$$\theta_p^k \text{ integer} \quad \forall k \in K, \forall p \in \Omega^k. \quad (6)$$

The objective function (1) states that we wish to maximize the total anticipated profit. Constraints (2) require that each operational flight leg be covered exactly once. Constraints (3) correspond to the flow conservation constraints at the beginning and the end of the day at each station and for each aircraft type. Constraints (4) limit the number of aircraft of type  $k \in K$  that can be used to the number available. Finally, constraints (5) and (6) state that the decision variables are nonnegative integers. This model is a Set Partitioning problem with additional constraints.

[from G. Desaulniers, J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis. Daily Aircraft Routing and Scheduling. *Management Science*, 1997, 43(6), 841-855]

# Outline

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks

# Outline

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks

# Matching

## Definition (Matching Theory Terminology)

Matching: set of pairwise non adjacent edges

Covered (vertex): a vertex is covered by a matching  $M$  if it is incident to an edge in  $M$

Perfect (matching): if  $M$  covers each vertex in  $G$

Maximal (matching): if  $M$  cannot be extended any further

Maximum (matching): if  $M$  covers as many vertices as possible

Matchable (graph): if the graph  $G$  has a perfect matching

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & \sum_{e \in E: v \in e} x_e \leq 1 \quad \forall v \in V \\ & x_e \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

Special case: bipartite matching  $\equiv$  assignment problems

# Vertex Cover

Select a subset  $S \subseteq V$  such that each edge has at least one end vertex in  $S$ .

$$\begin{aligned} \min \quad & \sum_{v \in V} x_v \\ & x_v + x_u \geq 1 \quad \forall u, v \in V, uv \in E \\ & x_v \in \{0, 1\} \quad \forall v \in V \end{aligned}$$

**Approximation algorithm:** set  $S$  derived from the LP solution in this way:

$$S_{LP} = \{v \in V : x_v^* \geq 1/2\}$$

(it is a cover since  $x_v^* + x_u^* \geq 1$  implies  $x_v^* \geq 1/2$  or  $x_u^* \geq 1/2$ )

## Proposition

*The LP rounding approximation algorithm gives a 2-approximation:  $|S_{LP}| \leq 2|S_{OPT}|$   
(at most as bad as twice the optimal solution)*

Proof: Let  $\bar{x}$  be opt to IP. Then  $\sum x_v^* \leq \sum \bar{x}_v$ .

$$|S_{LP}| = \sum_{v \in S_{LP}} 1 \leq \sum_{v \in V} 2x_v^* \text{ since } x_v^* \geq 1/2 \text{ for each } v \in S_{LP}$$

$$|S_{LP}| \leq 2 \sum_{v \in V} x_v^* \leq 2 \sum_{v \in V} \bar{x}_v = 2|S_{OPT}|$$



# Maximum Independent Set

Find the largest subset  $S \subseteq V$  such that the induced graph has no edges

$$\begin{aligned} \max \quad & \sum_{v \in V} x_v \\ & x_v + x_u \leq 1 \quad \forall u, v \in V, uv \in E \\ & x_v = \{0, 1\} \quad \forall v \in V \end{aligned}$$

Optimal sol of LP relaxation sets  $x_v = 1/2$  for all variables and has value  $|V|/2$ .

What is the value of an optimal IP solution of a complete graph?

LP relaxation gives an  $O(n)$ -approximation (almost useless)

# Traveling Salesman Problem

- Find the cheapest movement for a drilling, welding, drawing, soldering arm as, for example, in a printed circuit board manufacturing process or car manufacturing process
- $n$  locations,  $c_{ij}$  cost of travel

**Variables:**

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$$

**Objective:**

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

## Constraints:

- 

$$\sum_{j:j \neq i} x_{ij} = 1$$

$$\forall i = 1, \dots, n$$

$$\sum_{i:i \neq j} x_{ij} = 1$$

$$\forall j = 1, \dots, n$$

- cut set constraints

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1$$

$$\forall S \subset N, S \neq \emptyset$$

- subtour elimination constraints

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1$$

$$\forall S \subset N, 2 \leq |S| \leq n - 1$$

# Outline

## 1. Integer Programming

## 2. Modeling

Assignment Problem

Knapsack Problem

Set Problems

## 3. More on Modeling

Graph Problems

Modeling Tricks

# Modeling Tricks

Objective function and/or constraints do not appear to be linear?

- Absolute values
- Minimize the largest function value
- Maximize the smallest function value
- Constraints include variable division
- Constraints are either/or
- A variable must take one of several candidate values

# Modeling: Absolute Values

$$\min \sum_{i=1}^n |f_i(x)|$$

$$\begin{array}{ll} \min & \sum_{i=1}^n z_i \\ \text{s.t.} & z_i \geq f_i(x) \quad i = 1..n \\ & z_i \geq -f_i(x) \quad i = 1..n \\ & z_i \in \mathbb{R} \quad i = 1..n \\ & x \in \mathbb{R}^q \end{array}$$

$n$  additional variables and  $2n$  additional constraints.

$$\begin{array}{ll} \min & \sum_{i=1}^n (z_i^+ + z_i^-) \\ \text{s.t.} & f_i(x) = z_i^+ - z_i^- \quad i = 1..n \\ & z_i^+, z_i^- \geq 0 \quad i = 1..n \\ & x \in \mathbb{R}^q \end{array}$$

$2n$  additional variables and  $n$  additional constraints.

# Modeling: Minimax

Minimize the largest of a number of function values:

$$\min \max\{f_1(x), \dots, f_n(x)\}$$

- Introduce an auxiliary variable  $z$ :

$$\begin{aligned} \min \quad & z \\ \text{s. t. } & f_1(x) \leq z \\ & f_2(x) \leq z \end{aligned}$$

# Modeling: Divisions

Constraints include variable division:

- Constraint of the form

$$\frac{a_1x + a_2y + a_3z}{d_1x + d_2y + d_3z} \leq b$$

- Rearrange:

$$a_1x + a_2y + a_3z \leq b(d_1x + d_2y + d_3z)$$

which gives:

$$(a_1 - bd_1)x + (a_2 - bd_2)y + (a_3 - bd_3)z \leq 0$$



# Modeling: “Either/Or Constraints”

In conventional mathematical models, the solution must satisfy all constraints.  
Suppose that your constraints are “either/or”:

$$\begin{array}{ll} a_1x_1 + a_2x_2 \leq b_1 & \text{or} \\ d_1x_1 + d_2x_2 \leq b_2 \end{array}$$

Introduce new variable  $y \in \{0,1\}$  and a large number  $M$ :

$$\begin{array}{ll} a_1x_1 + a_2x_2 \leq b_1 + My & \text{if } y = 0 \text{ then this is active} \\ d_1x_1 + d_2x_2 \leq b_2 + M(1 - y) & \text{if } y = 1 \text{ then this is active} \end{array}$$

# Modeling: “Either/Or Constraints”

Binary integer programming allows to model alternative choices:

- Eg: 2 feasible regions, ie, disjunctive constraints, not possible in LP.  
introduce  $y$  auxiliary binary variable and  $M$  a big number:

$$Ax \leq b + My$$

if  $y = 0$  then this is active

$$A'x \leq b' + M(1 - y)$$

if  $y = 1$  then this is active

# Modeling: “Either/Or Constraints”

Generally:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1m}x_m &\leq d_1 \\a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2m}x_m &\leq d_2 \\&\vdots \\a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{Nm}x_m &\leq d_N\end{aligned}$$

Exactly  $K$  of the  $N$  constraints must be satisfied.

Introduce binary variables  $y_1, y_2, \dots, y_N$  and a large number  $M$

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1m}x_m &\leq d_1 + My_1 \\a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2m}x_m &\leq d_2 + My_2 \\&\vdots \\a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{Nm}x_m &\leq d_N + My_N\end{aligned}$$

$$y_1 + y_2 + \dots + y_N = N - K$$

$K$  of the  $y$ -variables are 0, so  $K$  constraints must be satisfied

# Modeling: “Either/Or Constraints”

At least  $h \leq k$  of  $\sum_{j=1}^n a_{ij}x_j \leq b_i, i = 1, \dots, k$  must be satisfied

introduce  $y_i, i = 1, \dots, k$  auxiliary binary variables

$$\sum_{j=1}^n a_{ij}x_j \leq b_i + My_i$$

$$\sum_i y_i \leq k - h$$

# Modeling: “Possible Constraints Values”

A constraint must take on one of  $N$  given values:

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_mx_m = d_1 \text{ or}$$

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_mx_m = d_2 \text{ or}$$

$$\vdots$$

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_mx_m = d_N$$

Introduce binary variables  $y_1, y_2, \dots, y_N$ :

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_mx_m = d_1y_1 + d_2y_2 + \dots d_Ny_N$$

$$y_1 + y_2 + \dots y_N = 1$$

# Resume

## 1. Integer Programming

## 2. Modeling

- Assignment Problem

- Knapsack Problem

- Set Problems

## 3. More on Modeling

- Graph Problems

- Modeling Tricks