Modelling and Optimization

INF170

#10:
Set Covering/Set Packing/Set Partitioning Problems

AHMAD HEMMATI

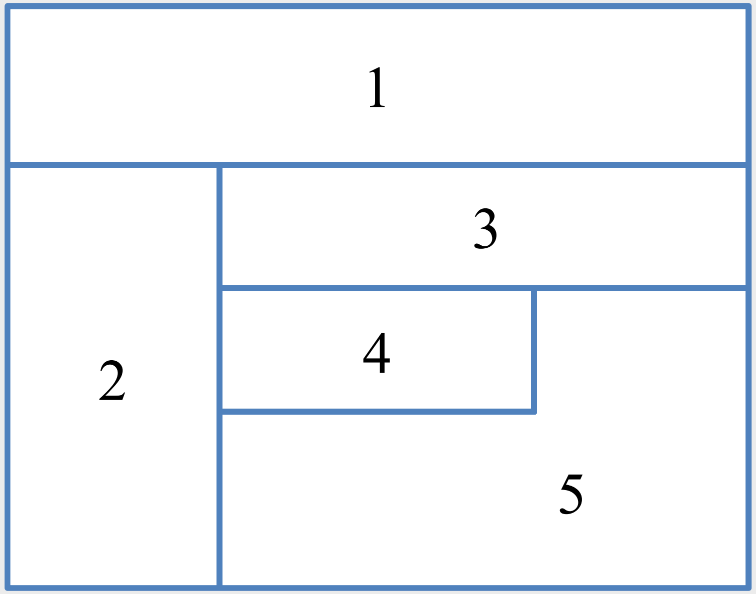
Optimization Group
Dept. of Informatics
University of Bergen

Fall Semester
2018

Agenda

- Set Covering Problem
  - EMS location planning
  - EMS location planning II
- Set Packing Problem
- Set Partitioning Problem
  - Political Districting Problem
  - Workplan Problem
  - Airline Crew Scheduling

EMS Location Planning

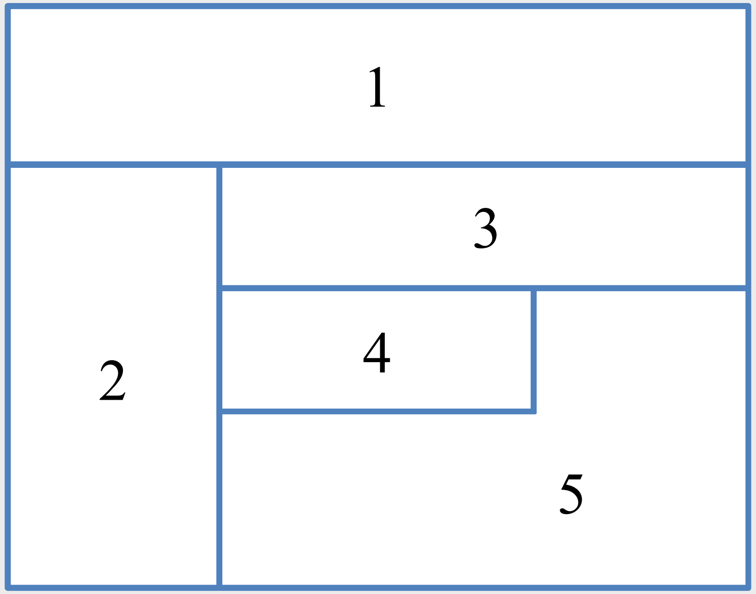


- We want to locate EMS stations so that each district has a station in it, or in a district next to it
- Objective: minimize number of stations required

Decision variables:

$$x_i = \begin{cases} 1 & \text{if station is put in district } i \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, 5$



Objective: *minimize*

$$\sum_{i=1}^{5} x_i$$

EMS Location Planning

Constraints:

Every district has a station in it, or in a district next to it:

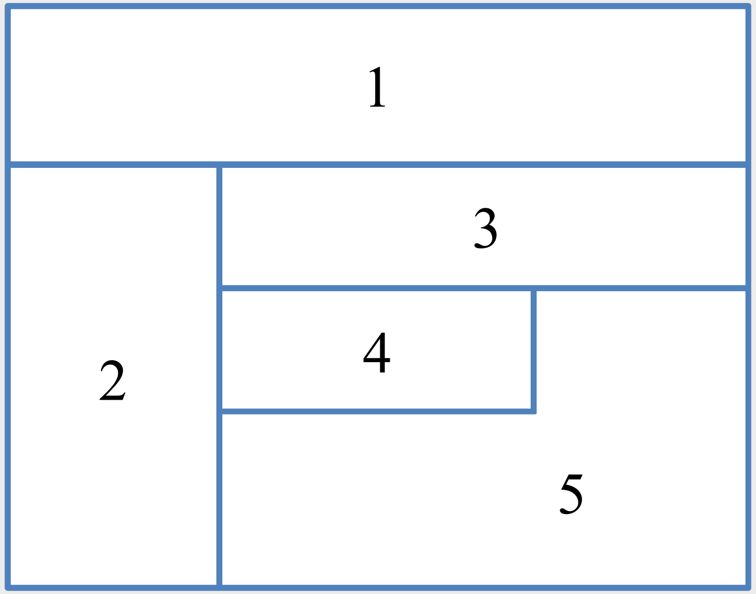
$$x_1 + x_2 + x_3 \geq 1 \quad (D_1)$$

$$x_2 + x_1 + x_3 + x_4 + x_5 \geq 1 \quad (D_2)$$

$$x_3 + x_1 + x_2 + x_4 + x_5 \geq 1 \quad (D_3)$$

$$x_4 + x_2 + x_3 + x_5 \geq 1 \quad (D_4)$$

$$x_5 + x_2 + x_3 + x_4 \geq 1 \quad (D_5)$$



EMS Location Planning

$$\min \sum_{i=1}^{5} x_i$$

s.t.

$$x_1 + x_2 + x_3 \geq 1 \quad (D_1)$$

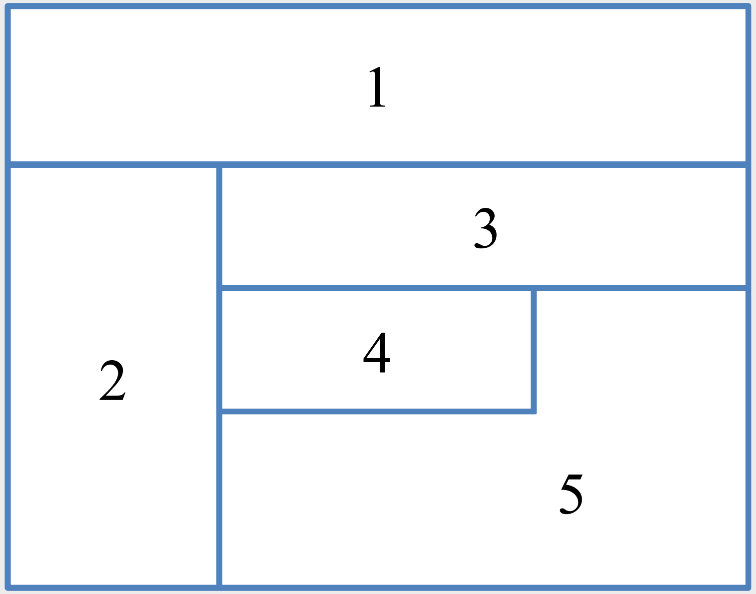
$$x_2 + x_1 + x_3 + x_4 + x_5 \geq 1 \quad (D_2)$$

$$x_3 + x_1 + x_2 + x_4 + x_5 \geq 1 \quad (D_3)$$

$$x_4 + x_2 + x_3 + x_5 \geq 1 \quad (D_4)$$

$$x_5 + x_2 + x_3 + x_4 \geq 1 \quad (D_5)$$

$$x_i \in \{0,1\} \qquad \text{for } i = 1, \ldots, 5$$



Set Covering Problem

- Set of items $N = \{1, 2, \ldots, n\}$
  (e.g. set of all districts)

- Collection of subsets of N: $S_1, S_2, \ldots, S_t$
  (e.g. S_i = set of locations that satisfy district i 's demands)

- Cost c_j for each item $j \in N$

- Choose items from N so that at least one item from each subset $S_1, \ldots, S_t$ is chosen, while minimizing cost

Set Covering Problem

Decision variables:

$$x_j = \begin{cases} 1 & \text{if item } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } j \in N$$

Model:

$$min \sum_{j \in N} c_j \, x_j$$

s.t.

$$\sum_{j \in S_i} x_j \geq 1 \quad \text{for } i = 1, \dots, t$$

$$x_j \in \{0,1\} \qquad \text{for } j \in N$$

EMS Location Planning II

- Suppose now that with each district, there is an associated estimated demand:

District	Demand (1000s)
1	5
2	4
3	7
4	9
5	6

- Suppose now we can only build 2 stations
- Instead of minimizing the number of stations required, what if we wanted to minimize the unsatisfied demand?

EMS Location Planning II

District	Demand (1000s)
1	5
2	4
3	7
4	9
5	6

- Idea: introduce additional decision variables

$$y_i = \begin{cases} 1 & \text{if demand in district } i \text{ is not satisfied} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \ldots, 5$$

- Objective: minimize unsatisfied demand

$$\textit{Minimize} \qquad 5y_1 + 4y_2 + 7y_3 + 9y_4 + 6y_5$$

EMS Location Planning II

- Constraints:
- District 1 needs a station in Districts 1, 2, 3, or its demand is unsatisfied

$$\underbrace{x_1 + x_2 + x_3}_{\boxed{\text{satisfied}}} + \underbrace{y_1}_{\boxed{\text{unsatisfied}}} \geq 1 \qquad (D_1)$$

- Same idea for other districts

$$x_2 + x_1 + x_3 + x_4 + x_5 + y_2 \geq 1 \qquad (D_2)$$

$$x_3 + x_1 + x_2 + x_4 + x_5 + y_3 \geq 1 \qquad (D_3)$$

$$x_4 + x_2 + x_3 + x_5 + y_4 \geq 1 \qquad (D_4)$$

$$x_5 + x_2 + x_3 + x_4 + y_5 \geq 1 \qquad (D_5)$$

- At most 2 stations can be built

$$\sum_{i=1}^{5} x_i \leq 2$$

The model:

$$\textit{Minimize} \qquad 5y_1 + 4y_2 + 7y_3 + 9y_4 + 6y_5$$

s.t.

$$x_1 + x_2 + x_3 + y_1 \geq 1 \qquad (D_1)$$

$$x_2 + x_1 + x_3 + x_4 + x_5 + y_2 \geq 1 \qquad (D_2)$$

$$x_3 + x_1 + x_2 + x_4 + x_5 + y_3 \geq 1 \qquad (D_3)$$

$$x_4 + x_2 + x_3 + x_5 + y_4 \geq 1 \qquad (D_4)$$

$$x_5 + x_2 + x_3 + x_4 + y_5 \geq 1 \qquad (D_5)$$

$$\sum_{i=1}^{5} x_i \leq 2$$

$$x_i \in \{0,1\} \qquad \text{for } i = 1, \ldots, 5$$

$$y_i \in \{0,1\} \qquad \text{for } i = 1, \ldots, 5$$

Set Packing Problem

- Your kitchen contains a collection of different food ingredients
- You have a cookbook with a collection of recipes
- Each recipe requires a subset of the food ingredients.
- You want to prepare the largest possible collection of recipes from the cookbook.
- You are actually looking for a collection of recipes whose sets of ingredients are pairwise disjoint.

SET PACKING PROBLEM

- Set of items $N = \{1, 2, \ldots, n\}$

  (e.g. set of all ingredients)

- Collection of subsets of N: $S_1, S_2, \ldots, S_t$

  (e.g. S_j = set of ingredients that we need for recipe j)

- Maximize the total number of subsets so that the selected sets are pairwise disjoint

Set Packing Problem

Decision variables:

$$x_j = \begin{cases} 1 & \text{if recipe } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \qquad j = 1, \dots, t$$

Model:

$$max \sum_{j=1}^{t} x_j$$

s.t.

$$\sum_{j=1}^{t} a_{ij} x_j \leq 1 \quad \text{for } i \in N$$

$a_{ij} = 1$ if item i is in S_j

$$x_j \in \{0,1\} \qquad \text{for } j = 1, \dots, t$$

Set Packing Problem

- Set of ingredients $N = \{A, B, C, D, E\}$
  - $S_1 = \{A, B\}$
  - $S_2 = \{A, C, E\}$
  - $S_3 = \{B, D, E\}$
  - $S_4 = \{C\}$
  - $S_5 = \{A\}$
  - $S_6 = \{D, E\}$

- Maximize number of subsets so that the selected sets are pairwise disjoint

Set Packing Problem

The model:

$$\text{Maximize} \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

s.t.

$$\begin{array}{lr}
x_1 + x_2 + x_5 \leq 1 & (A) \\
x_1 + x_3 \leq 1 & (B) \\
x_2 + x_4 \leq 1 & (C) \\
x_3 + x_6 \leq 1 & (D) \\
x_2 + x_3 + x_6 \leq 1 & (E)
\end{array}$$

$$x_i \in \{0,1\} \qquad \text{for } i = 1, \dots, 6$$

Set Packing Problem

- English course!

- Set of Non-English languages $N = \{A, B, C, D, E\}$
  - $S_1 = \{A, B\}$ //person 1 can speak in languages A and B
  - $S_2 = \{A, C, E\}$
  - $S_3 = \{B, D, E\}$
  - $S_4 = \{C\}$
  - $S_5 = \{A\}$
  - $S_6 = \{D, E\}$

- A maximum set packing will choose the largest possible number of people under the desired constraint. Maximize the size of the group!

Set Packing Problem

- Camera refurbishing!
- Consider a business that refurbishes cameras by assembling together the good parts of defective products.
- There are many models of cameras, but some models share parts.
- Some camera parts are available, we should decide which partly-done camera should get the needed parts to get completed.
- We want to maximize the number of completed cameras

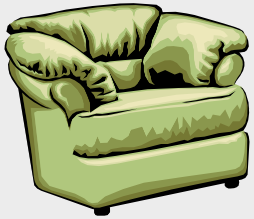
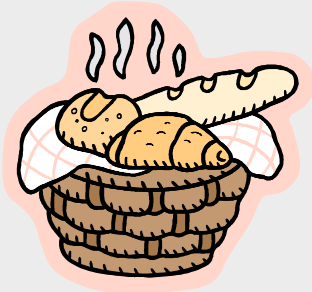
Set Packing Problem

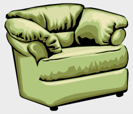
- Set of available camera parts $N = \{A, B, C, D, E\}$
  - $S_1 = \{A, B\}$ // camera 1 need parts A and B to be fixed
  - $S_2 = \{A, C, E\}$
  - $S_3 = \{B, D, E\}$
  - $S_4 = \{C\}$
  - $S_5 = \{A\}$
  - $S_6 = \{D, E\}$

- A maximum set packing will choose the largest possible number of cameras that can be fixed.

SET PACKING PROBLEM

- Combinatorial Auction!
- Set of Products:

- Each customer can bid:
- $700 for {  AND  }
- $1200 for {  AND  }
- $60 for {  AND  }
- $3 for {  }

Set Packing Problem

Decision variables:

$$x_j = \begin{cases} 1 & \text{if } bid\ j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \qquad j = 1, \dots, t$$

Model:

$$max \sum_{j=1}^{t} p_j x_j$$

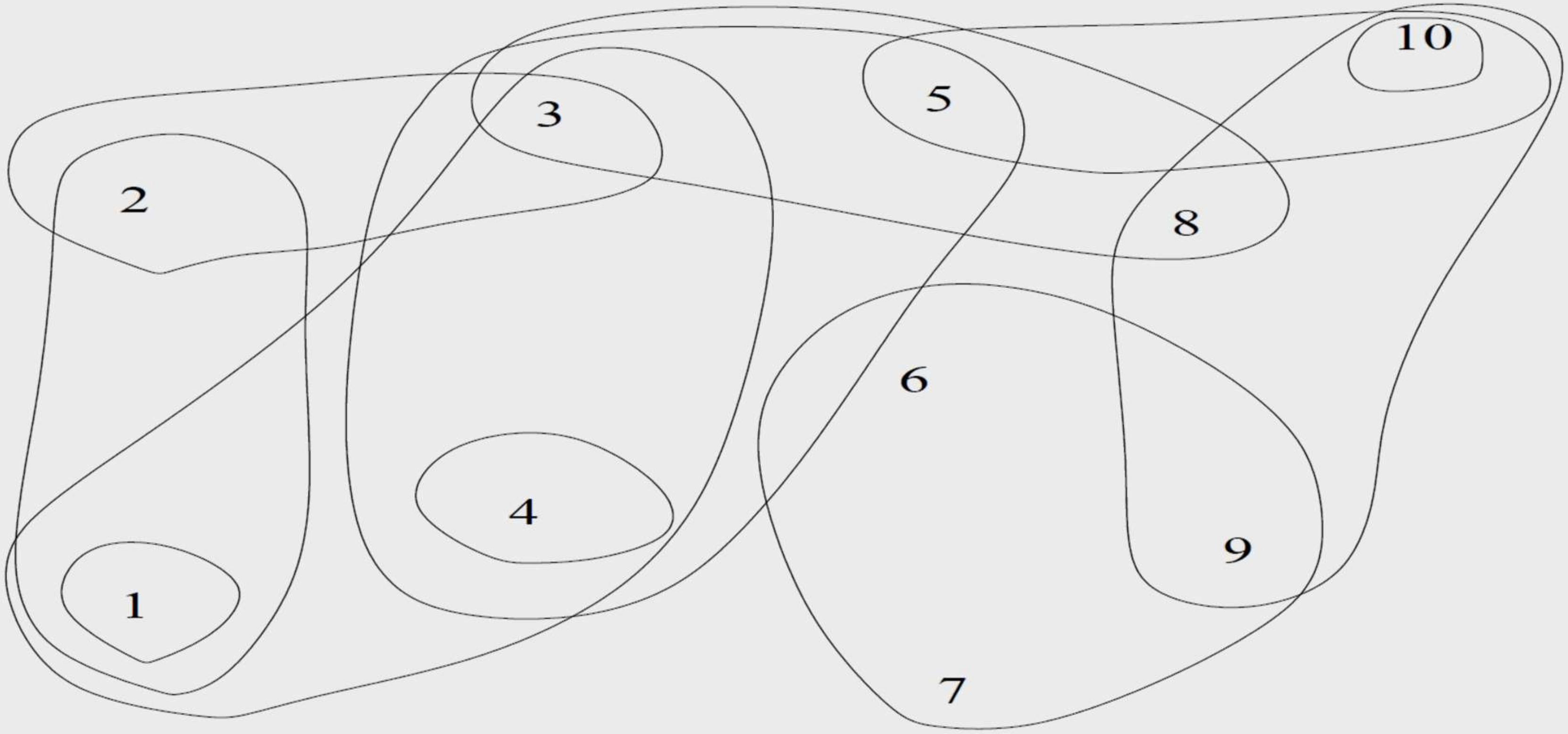
s.t.

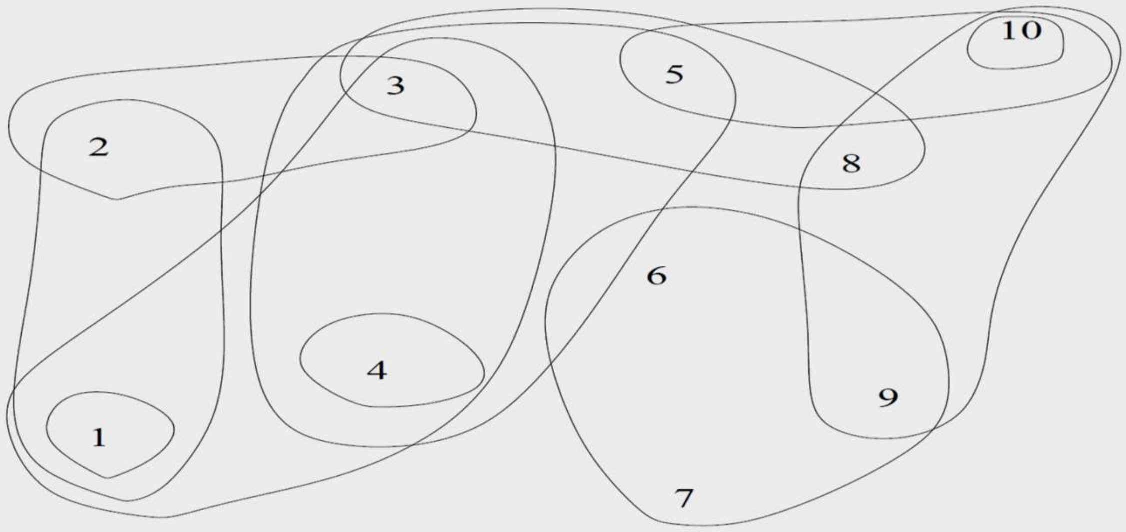
$$\sum_{j=1}^{t} a_{ij} x_j \leq 1 \quad \text{for } i \in N$$

$a_{ij} = 1$ if item i is in bid_j

$$x_j \in \{0,1\} \qquad \text{for } j = 1, \dots, t$$

Set Partitioning Problem





1	1	1	0	0	1	0	0	0	0	0	0		0	=	1
2	0	1	1	0	0	0	0	0	0	0	0		0	=	1
3	0	0	1	0	1	1	1	0	0	0	0		0	=	1
4	0	0	0	1	1	0	0	0	0	0	0		0	=	1
5	0	0	0	0	0	1	1	1	0	0	0		0	=	1
6	0	0	0	0	0	0	0	0	0	1	0		0	=	1
7	0	0	0	0	0	0	0	0	0	1	0	$\cdots$	0	=	1
8	0	0	0	0	0	0	1	0	1	0	0		0	=	1
9	0	0	0	0	0	0	0	0	1	1	0		0	=	1
10	0	0	0	0	0	0	0	1	1	0	1		0	=	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
m	0	0	0	0	0	0	0	0	0	0	0		1	=	1
	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}		c_n		

Set Partitioning Problem

- Set of items: $I = \{1, 2, \ldots, m\}$
- Set of Collections: C ($C_1, C_2, \ldots, C_n$ are subsets of I)
- Cost p_j for each collection $j \in \{C_1, C_2, \ldots, C_n\}$
- Choose collections from C so that each item is assigned to <u>exactly one</u> collection, while minimizing cost

Set Partitioning Problem

Decision variables:

$$x_j = \begin{cases} 1 & \text{if collection } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } j \in C$$

Model:

$$min \sum_{j \in C} p_j \, x_j$$

s.t.

$$\sum_{j \in C} a_{ij} x_j = 1 \qquad \text{for } i \in I$$

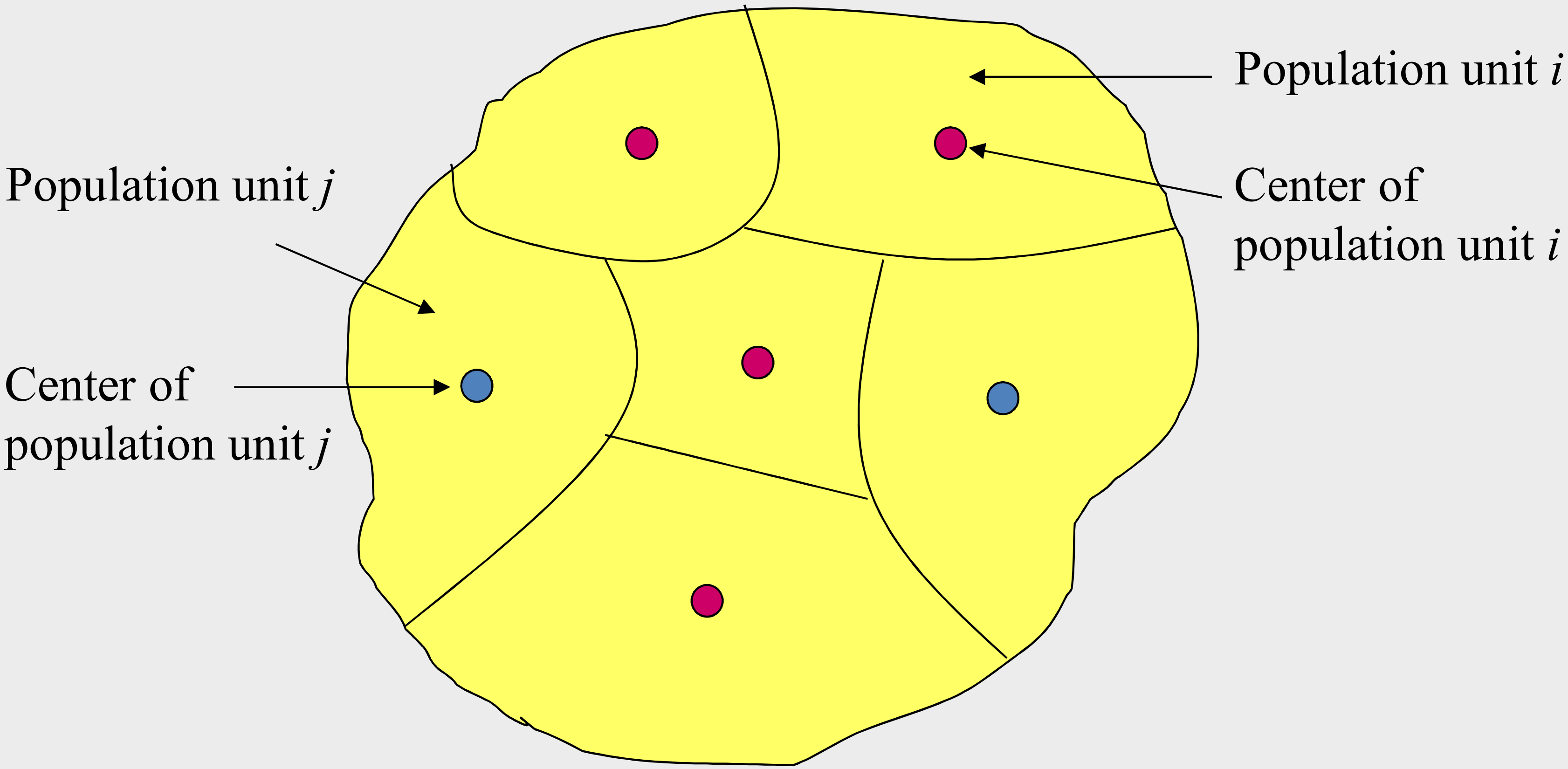
$$x_j \in \{0,1\} \qquad \text{for } j \in C$$

$a_{ij} = 1$ if item i is included in collection j

Political Districting Problem

- The political districting problem consists in partitioning an area into electoral constituencies (districts)
- Each population unit is assigned to one district;
- The number of districts is usually known (N districts);
- All districts must have approximately the same number of voters for better equity

POLITICAL DISTRICTING PROBLEM



Political Districting Problem

Parameters:

I : set of population units

N : number of district centers

p_i : population of the i^{th} population unit

a : minimum population allowed for a district (as a ratio of average district population)

b : maximum population allowed for a district (as a ratio of average district population)

d_{ij} : distance between the centers of population units i and j.

a and b can be considered as deviations from the average population of all population units which is given by

$$\sum_{i \in I} \frac{P_i}{N}$$

Political Districting Problem

Variables:

$$x_{ij} = \begin{cases} 1 & \text{population unit } i \text{ is assigned to district } j \\ 0 & \text{otherwise} \end{cases} \quad \text{for } \; i\,,\, j \in I$$

Objective function:

$$min \sum_{i \in N} \sum_{j \in N} d_{ij}^2 \, P_i \, x_{ij}$$

Political Districting Problem

Model:

$$min \sum_{i\in N}\sum_{j\in N} d_{ij}^2\, P_i\, x_{ij}$$

s.t.

$$\sum_{j\in N} x_{ij} = 1 \quad \text{for } i \in I$$

$$\sum_{j\in N} x_{jj} = N$$

$$\sum_{i\in N} P_i\, x_{ij} \geq a \left(\frac{\sum_{i\in I} P_i}{N}\right) x_{jj} \quad \text{for } j \in I$$

$$\sum_{i\in N} P_i\, x_{ij} \leq b \left(\frac{\sum_{i\in I} P_i}{N}\right) x_{jj} \quad \text{for } j \in I$$

$$x_{ij} \in \{0,1\} \qquad \text{for } i, j \in I$$

Workplan Problem

- We have 6 assignments A, B, C, D, E and F, that needs to be carried out. For every assignment we have a start time and a duration (in hours).

Assignment	A	B	C	D	E	F
Start	0.0	1.0	2.0	2.5	3.5	5.0
Duration	1.5	2.0	2.0	2.0	2.0	1.5

Workplan Problem

A Workplan:

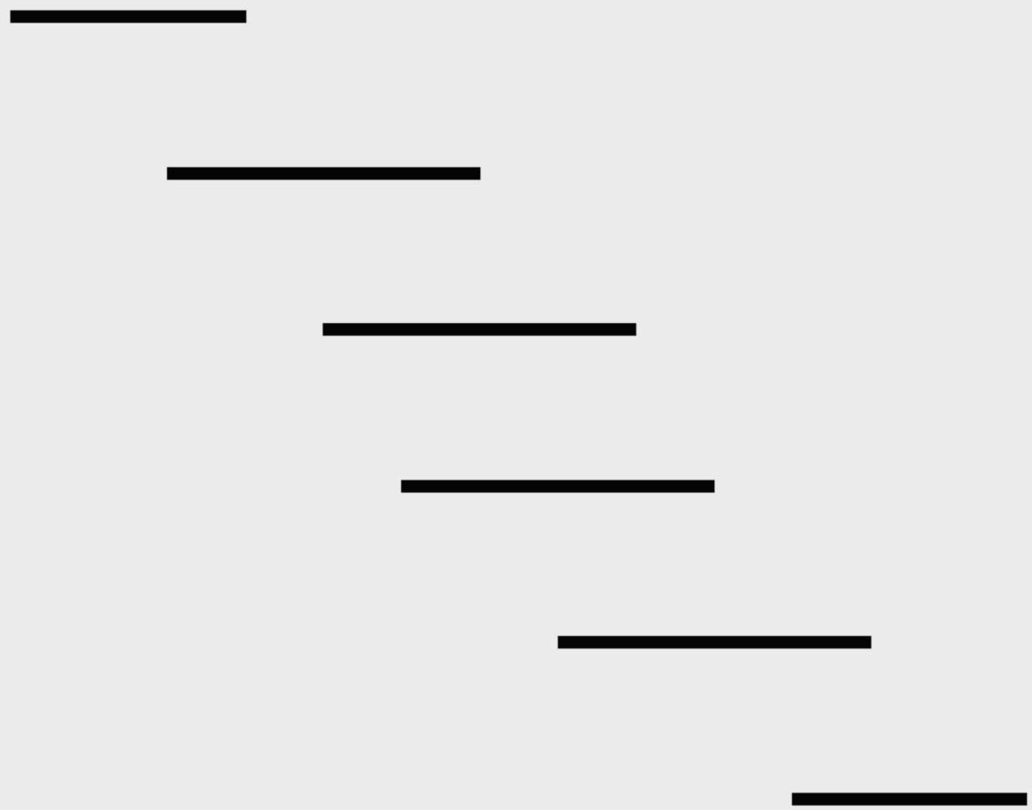
- A workplan is a set of assignments. Now we want to formulate a mathematical model that finds the cheapest set of workplans that fullfills all the assignments.

Workplan rules:

- A workplan can not consist of assignments that overlap each other.
- The length L of a workplan is equal to the finish time of the last assignment minus start of the first assignments plus 30 minutes for checking in and checking out.
- The cost of a workplan is *max* (4.0, L).

WORKPLAN PROBLEM

	Start	Duration
A	0.0	1.5
B	1.0	2.0
C	2.0	2.0
D	2.5	2.0
E	3.5	2.0
F	5.0	1.5



Workplan Problem

- Workplans starting with assignment A

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
A	1	1	1	1	1	1	1
B	0	0	0	0	0	0	0
C	0	1	1	0	0	0	0
D	0	0	0	1	1	0	0
E	0	0	0	0	0	1	0
F	0	0	1	0	1	0	1

Assignment	A	B	C	D	E	F
Start	0.0	1.0	2.0	2.5	3.5	5.0
Duration	1.5	2.0	2.0	2.0	2.0	1.5

Workplan Problem

- Workplans starting with assignment A

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
A	1	1	1	1	1	1	1
B	0	0	0	0	0	0	0
C	0	1	1	0	0	0	0
D	0	0	0	1	1	0	0
E	0	0	0	0	0	1	0
F	0	0	1	0	1	0	1

	4	$4\frac{1}{2}$	7	5	7	6	7
A	1	1	1	1	1	1	1
B	0	0	0	0	0	0	0
C	0	1	1	0	0	0	0
D	0	0	0	1	1	0	0
E	0	0	0	0	0	1	0
F	0	0	1	0	1	0	1

Assignment	A	B	C	D	E	F
Start	0.0	1.0	2.0	2.5	3.5	5.0
Duration	1.5	2.0	2.0	2.0	2.0	1.5

- Completing the model

	4	$4\frac{1}{2}$	7	5	7	6	7	4	5	6	4	5	4	$4\frac{1}{2}$	4	4
A	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
C	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
D	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0
E	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0
F	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1

Workplan Problem

Decision variables:

$$x_j = \begin{cases} 1 & \text{if we use workplan } j \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } j \in C$$

Model:

$$min \sum_{j \in C} c_j \, x_j$$

s.t.

$$\sum_{j \in C} a_{ij} x_j = 1 \qquad \text{for } i \in I$$

$$x_j \in \{0,1\} \qquad \text{for } j \in C$$

$a_{ij} = 1$ if assignment i is included in workplan j

A feasible solution is:

- $x_3 = 1, x_9 = 1, x_{13} = 1$ with a solution value of 16.

	4	$4\frac{1}{2}$	7	5	7	6	7	4	5	6	4	5	4	$4\frac{1}{2}$	4	4
A	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
C	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
D	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0
E	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0
F	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1

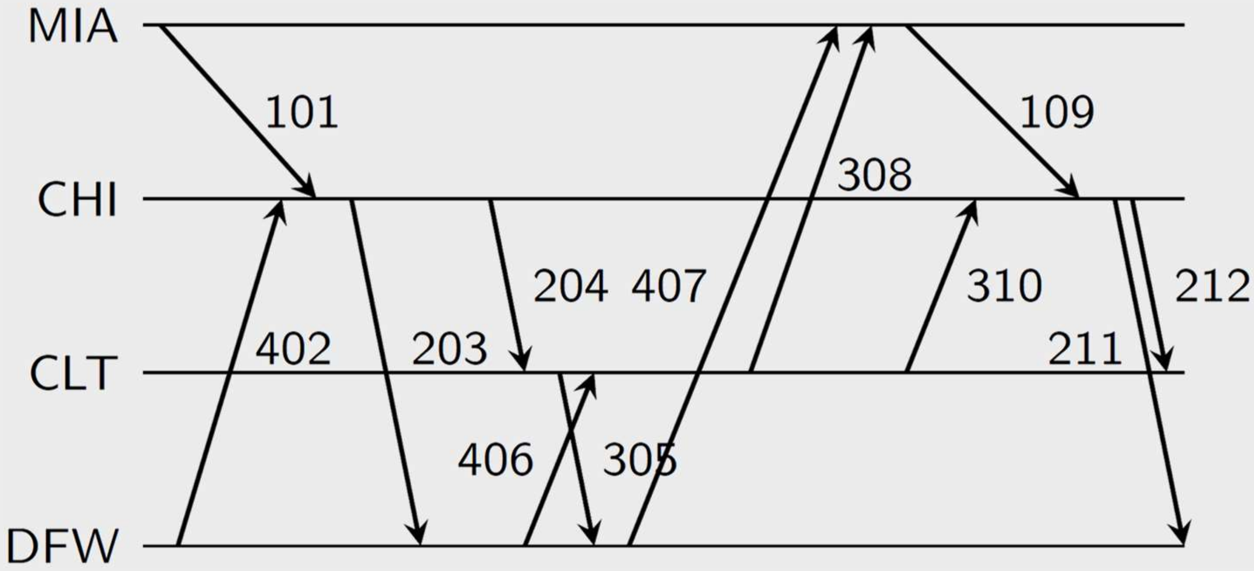
Workplan Problem

An optimal solution is:

✓ $x_2 = 1$, $x_9 = 1$, $x_{14} = 1$ with a solution value of 13.

	4	$4\frac{1}{2}$	7	5	7	6	7	4	5	6	4	5	4	$4\frac{1}{2}$	4	4
A	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
C	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
D	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0
E	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0
F	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1

AIRLINE CREW SCHEDULING



- Horizontal axis is time
  - ➢ e.g. Flight 101 from MIA to CHI, Flight 402 from DFW to CHI
- A single crew (typically pilots and attendants) is assigned to a pairing: a sequence of flights over a 2 to 3 day period
  - ➢ Pairing must begin and end at the same city

Airline Crew Scheduling

- Typically, lots of government and union rules to consider
- Reasonable pairings usually generated by complex algorithms
- Below, all possible pairings of 3 or 4 flights with associated costs

	Flight sequence	Cost		Flight sequence	Cost
1	101-203-406-308	2900	**9**	305-407-109-212	2600
2	101-203-407	2700	**10**	308-109-212	2050
3	101-204-305-407	2600	**11**	402-204-305	2400
4	101-204-308	3600	**12**	402-204-310-211	3600
5	203-406-310	2600	**13**	406-308-109-211	2550
6	203-407-109	3150	**14**	406-310-211	2650
7	204-305-407-109	2550	**15**	407-109-211	2350
8	204-308-109	2500			

- Want to find a collection of pairings that cover each flight exactly once, at minimum cost
- Decision variables → selecting pairings

Airline Crew Scheduling

	Flight sequence	Cost		Flight sequence	Cost
1	101-203-406-308	2900	**9**	305-407-109-212	2600
2	101-203-407	2700	**10**	308-109-212	2050
3	101-204-305-407	2600	**11**	402-204-305	2400
4	101-204-308	3600	**12**	402-204-310-211	3600
5	203-406-310	2600	**13**	406-308-109-211	2550
6	203-407-109	3150	**14**	406-310-211	2650
7	204-305-407-109	2550	**15**	407-109-211	2350
8	204-308-109	2500			

Decision variables:

$$x_i = \begin{cases} 1 & \text{if pairing } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \dots, 15$$

Objective: $min \quad 2900\, x_1 + 2700\, x_2 + 2600\, x_3 + 3600\, x_4 + 2600\, x_5 + 3150\, x_6 + 2550\, x_7 + 2500\, x_8$

$$+ 2600\, x_9 + 2050\, x_{10} + 2400\, x_{11} + 3600\, x_{12} + 2550\, x_{13} + 2650\, x_{14} + 2350\, x_{15}$$

Airline Crew Scheduling

	Flight sequence	Cost		Flight sequence	Cost
1	101-203-406-308	2900	**9**	305-407-109-212	2600
2	101-203-407	2700	**10**	308-109-212	2050
3	101-204-305-407	2600	**11**	402-204-305	2400
4	101-204-308	3600	**12**	402-204-310-211	3600
5	203-406-310	2600	**13**	406-308-109-211	2550
6	203-407-109	3150	**14**	406-310-211	2650
7	204-305-407-109	2550	**15**	407-109-211	2350
8	204-308-109	2500			

Constraints: each flight needs to be covered by a pairing exactly once

- For example, flight 101:

$$x_1 + x_2 + x_3 + x_4 = 1$$

- Same idea for each flight

Airline Crew Scheduling

The full model:

Objective: $\min \quad 2900\,x_1 + 2700\,x_2 + 2600\,x_3 + 3600\,x_4 + 2600\,x_5 + 3150\,x_6 + 2550\,x_7 + 2500\,x_8 + 2600\,x_9 + 2050\,x_{10} + 2400\,x_{11} + 3600\,x_{12} + 2550\,x_{13} + 2650\,x_{14} + 2350\,x_{15}$

s.t.

$$
\begin{array}{ll}
x_1 + x_2 + x_3 + x_4 = 1 & (101)\\
x_6 + x_7 + x_8 + x_9 + x_{10} + x_{13} + x_{15} = 1 & (109)\\
x_1 + x_2 + x_5 + x_6 = 1 & (203)\\
x_3 + x_4 + x_7 + x_8 + x_{11} + x_{12} = 1 & (204)\\
x_{12} + x_{13} + x_{14} + x_{15} = 1 & (211)\\
x_9 + x_{10} = 1 & (212)\\
x_3 + x_7 + x_9 + x_{11} = 1 & (305)\\
x_1 + x_4 + x_8 + x_{10} + x_{13} = 1 & (308)\\
x_5 + x_{12} + x_{14} = 1 & (310)\\
x_{11} + x_{12} = 1 & (402)\\
x_1 + x_5 + x_{13} + x_{14} = 1 & (406)\\
x_2 + x_3 + x_6 + x_7 + x_9 + x_{15} = 1 & (407)\\
x_1, \dots, x_{15} \in \{0,1\} &
\end{array}
$$

LECTURE #11:

KNAPSACK AND BINPACKING PROBLEM