IE303
Operations Research I

Week 12
Spring, 2025
Dr. Ozge Buyukdagli

Making Choices When the Decision Variables Are Continuous

The Research and Development Division of the GOOD PRODUCTS COMPANY has developed three possible new products.

- **Restriction 1:** From the three possible new products, *at most two should be* chosen to be produced. Each of these products can be produced in either of two plants.
- **Restriction 2:** Just one of the two plants should be chosen to be the sole producer of the new products.

GOOD PRODUCTS COMPANY Data

	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	(thousands of dollars)
Sales potential	7	5	9	(units per week)

The objective is to choose the products, the plant, and the production rates of the chosen products so as to maximize total profit.

The Model

Maximize $\quad Z = 5x_1 + 7x_2 + 3x_3,$

subject to

$$
\begin{aligned}
3x_1 + 4x_2 + 2x_3 &\leq 30 \\
4x_1 + 6x_2 + 2x_3 &\leq 40 \\
x_1 &\leq 7 \\
x_2 &\leq 5 \\
x_3 &\leq 9
\end{aligned}
$$

and

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

restriction 1

The number of strictly positive decision variables (x_1, x_2, x_3) must be ≤ 2.

restriction 2

$$
\begin{aligned}
&\text{Either} \quad 3x_1 + 4x_2 + 2x_3 \leq 30 \\
&\text{or} \quad 4x_1 + 6x_2 + 2x_3 \leq 40
\end{aligned}
$$

Formulation with Auxiliary Binary Variables

To deal with requirement 1

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \text{ can hold (can produce product } j) \\ 0 & \text{if } x_j = 0 \text{ must hold (cannot produce product } j), \end{cases}$$

for $j = 1, 2, 3.$

$$x_1 \leq My_1$$
$$x_2 \leq My_2$$
$$x_3 \leq My_3$$
$$y_1 + y_2 + y_3 \leq 2$$
$$y_j \text{ is binary}, \qquad \text{for } j = 1, 2, 3.$$

Formulation with Auxiliary Binary Variables

To deal with requirement 2

$$y_4 = \begin{cases} 1 & \text{if } 4x_1 + 6x_2 + 2x_3 \le 40 \text{ must hold (choose Plant 2)} \\ 0 & \text{if } 3x_1 + 4x_2 + 2x_3 \le 30 \text{ must hold (choose Plant 1).} \end{cases}$$

$$3x_1 + 4x_2 + 2x_3 \le 30 + My_4$$
$$4x_1 + 6x_2 + 2x_3 \le 40 + M(1 - y_4)$$
y_4 is binary.

The complete model

$$\text{Maximize} \quad Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned}
x_1 &\le 7 \\
x_2 &\le 5 \\
x_3 &\le 9 \\
x_1 - My_1 &\le 0 \\
x_2 - My_2 &\le 0 \\
x_3 - My_3 &\le 0 \\
y_1 + y_2 + y_3 &\le 2 \\
3x_1 + 4x_2 + 2x_3 - My_4 &\le 30 \\
4x_1 + 6x_2 + 2x_3 + My_4 &\le 40 + M
\end{aligned}$$

and

$$x_1 \ge 0, \qquad x_2 \ge 0, \qquad x_3 \ge 0$$
$$y_j \text{ is binary}, \qquad \text{for } j = 1, 2, 3, 4.$$

This now is an MIP model,

with 3 variables *not required to be integer and*

4 binary variables

MIP

Some of the Most Famous IP Problems

- Knapsack
- The Cutting Stock
- Project Selection
- Traveling Salesman

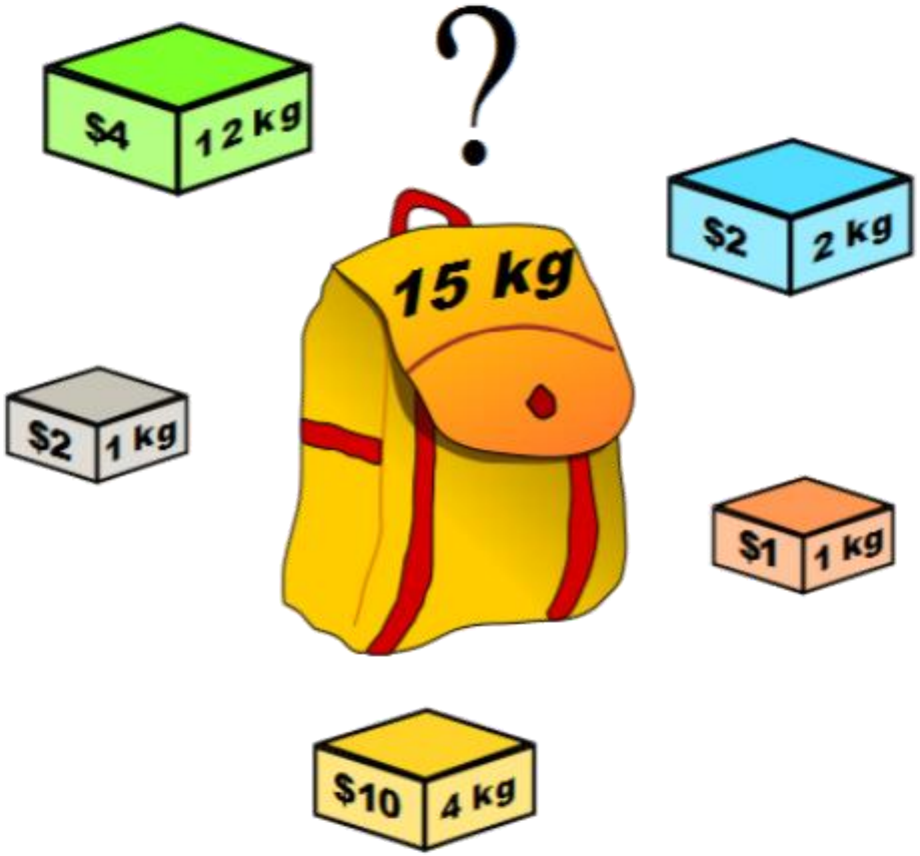
Knapsack Problem

You are going on a trip.

Your suitcase has a capacity of **b kg's**.

- You have n different items that you can pack into your suitcase 1, . . . , n.
- Item i has a weight of w_i and gives you a utility of v_i.

How should you pack your suitcase to maximize total utility?



Alternative

We have computed n data files that we want to store, and we have available W bytes of storage. File i has size w_i bytes and takes w_i minutes to recompute.

We want to avoid as much recomputing as possible, so we want to find a subset of files to store such that

- The files have combined size at most W.
- The total computing time of the stored files is as large as possible.

We can not store parts of files, it is the whole file or nothing.

How should we select the file?

Knapsack Problem

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

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0,1\}.$$

Here x_i represents the number of instances of item i to include in the knapsack. Informally, the problem is to maximize the sum of the values of the items in the knapsack so that the sum of the weights is less than or equal to the knapsack's capacity.

The **knapsack problem** is NP*-Hard, meaning it is computationally very challenging to solve

**non-deterministic polynomial-time*

Lets pack for the Mars

12 - 8 - 5 → 0

15 - 5 - 5 → 0

17 - 8 → 0

5

$X_1 = ?$

$N_1 = 0$

40

35

75

The Cutting Stock Problem

A paper company manufactures and sells rolls of paper of fixed width in 5 standard lengths: 5, 8, 12, 15 and 17 m. Demand for each size is given below.

It produces 25 m length rolls and cuts orders from its stock of this size. What is the min # of rolls to be used to meet the total demand?

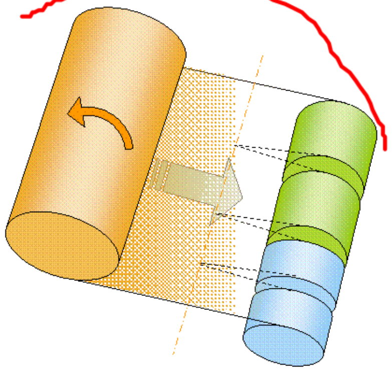
200 8 . 15

30 → 25m

12 . 8 . 5

5 → 15 . 5 . 5

Length (meters)	5	8	12	15	17
Demand	40	35	30	25	20
				20	15



x_i = # of patterns used

$$\boxed{\min \sum_i x_i}$$

The Cutting Stock Problem

First, identify the cutting patterns:

$$1 \cdot x_2 + 2 \cdot x_4 + x_6 + 2x_7 + x_9 + 3x_{10} + 5x_{11} \geq 40$$

	Pattern	Cuts	
x_1	1	17, 8	
x_2	2	17, 5	$\rightarrow 1$
x_3	3	15, 8	$\rightarrow 0$
	4	15, 5, 5	$\rightarrow 2$
	5	12, 12	$\rightarrow 0$
	6	12, 8, 5	1
	7	12, 5, 5	2
	8	8, 8, 8	0
	9	8, 8, 5	1
	10	8, 5, 5, 5	3
x_{11}	11	5, 5, 5, 5, 5	5

if $x_{11} = 1$

The Cutting Stock Problem

Then how do you define the decision variable?

x_j : # of rolls cut according to pattern *j*.

What about the constraints?

i.e. How do you write the constraint for the demand of 12 m rolls of paper?

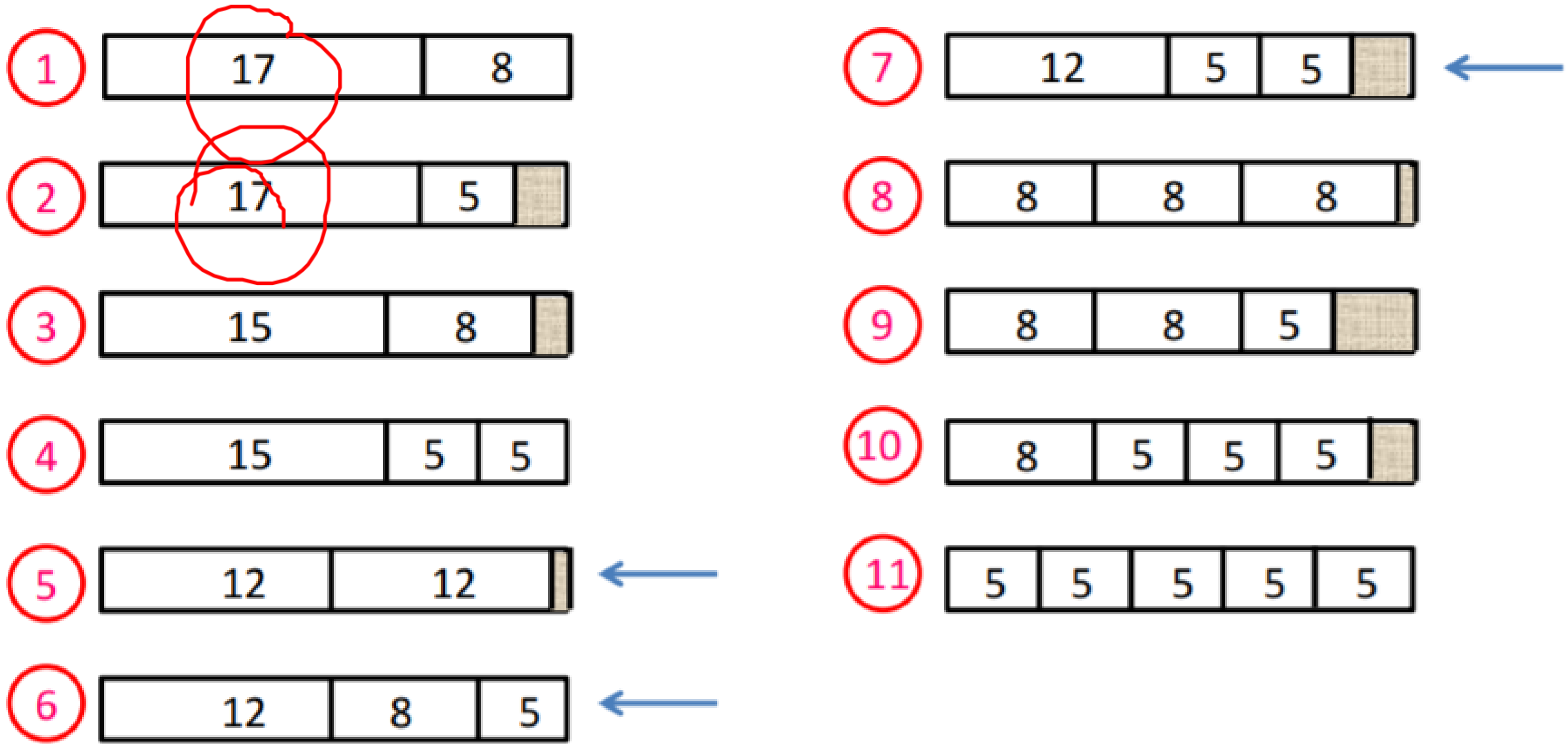
We look at the patterns that contain 12 m pieces, and then write the constraint based on x_j.

Length (meters)	5	8	12	15	17
Demand	40	35	30	25	20

5

The Cutting Stock Problem

The cutting patterns:



$$2x_5 + x_6 + x_7 \geq 30$$

The Cutting Stock Problem

Then how do you define the decision variable?

x_j : # of rolls cut according to pattern *j*.

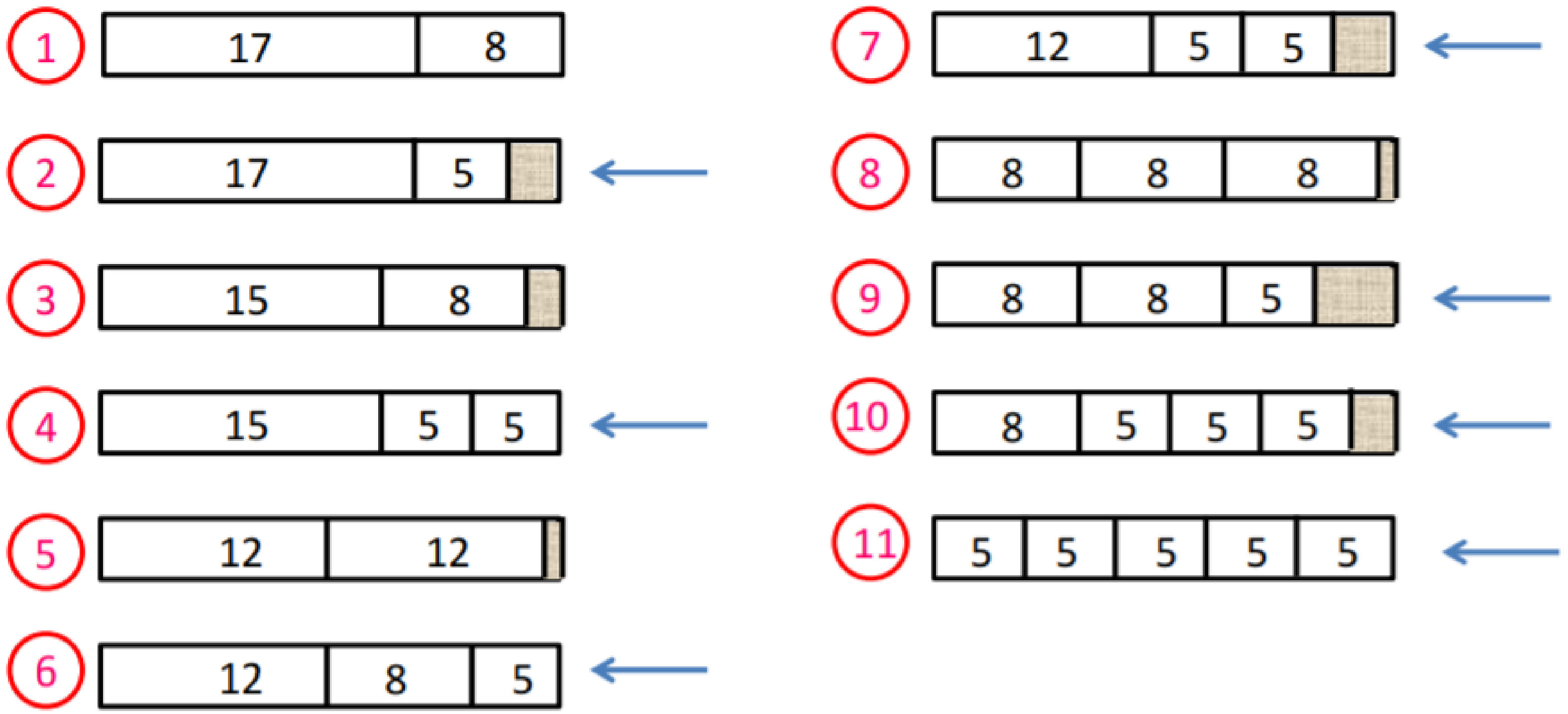
What about the constraints?

i.e. What about demand for 5 m?

Length (meters)	5	8	12	15	17
Demand	40	35	30	25	20

The Cutting Stock Problem

The cutting patterns:



$$x_2 + 2x_4 + x_6 + 2x_7 + x_9 + 3x_{10} + 5x_{11} \geq 40$$

The Cutting Stock Problem

The problem is formulated as follows:

75

$$\min \sum_{j=1}^{11} x_j$$

$$s.t.$$

$$x_2 + 2x_4 + x_6 + 2x_7 + x_9 + 3x_{10} + 5x_{11} \geq 40$$

$$x_1 + x_3 + x_6 + 3x_8 + 2x_9 + x_{10} \geq 35$$

$$2x_5 + x_6 + x_7 \geq 30$$

$$x_3 + x_4 \geq 25$$

$$x_1 + x_2 \geq 20$$

$$x_j \geq 0,\ j = 1, \ldots, 11,\ x_j \text{ integer.}$$

Project Selection

A manager should choose from among 10 different projects:

p_j : profit if we invest in project j , j = 1, . . . , 10
c_j : cost of project j , j = 1, . . . , 10
q: total budget available

There are also some additional requirements:

i. Projects 3 and 4 cannot be chosen together
ii. Exactly 3 projects are to be selected
iii. If project 2 is selected, then so is project 1
iv. If project 1 is selected, then project 3 should not be selected
v. Either project 1 or project 2 is chosen, but not both
vi. Either both projects 1 and 5 are chosen or neither are chosen
vii. At least two and at most four projects should be chosen from the set {1, 2, 7, 8, 9, 10}
viii. Neither project 5 nor 6 can be chosen unless either 3 or 4 is chosen
ix. At least two of the investments 1, 2, 3 are chosen, or at most three of the investments 4, 5, 6, 7, 8 are chosen.
x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

Project Selection

Under these requirements, the objective is to maximize the Profit.

Decision Variables

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is chosen} \\ 0, & \text{otherwise} \end{cases}$$

$$j = 1,\ldots,10$$

Objective function is:

$$\max \sum_{j=1}^{10} p_j x_j$$

Project Selection

First the budget constraint:

$$\sum_{j=1}^{10} c_j x_j \le q$$

i. Projects 3 and 4 cannot be chosen together:

$$x_3 + x_4 \le 1$$

ii Exactly 3 projects are to be chosen:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 3$$

Project Selection

iii. If project 2 is chosen then so is project 1:

$$x_2 \leq x_1$$

iv If project 1 is selected, then project 3 should not be selected

$$x_1 \leq 1 - x_3$$

v. Either project 1 or project 2 is chosen, but not both

$$x_1 + x_2 = 1$$

Project Selection

vi. Either both projects 1 and 5 are chosen or neither are chosen

$$x_1 = x_5 \quad \text{or} \quad x_1 - x_5 = 0$$

vii. At least two and at most four projects should be chosen from the set {1, 2, 7, 8, 9, 10}:

$$x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} \geq 2$$

Project Selection

viii. Neither project 5 nor 6 can be chosen unless either 3 or 4 is chosen:

Equivalently, If $x_3 + x_4 = 0$ then $x_5 = x_6 = 0$

Another possibility,

$$x_5 \leq x_3 + x_4$$

$$x_6 \leq x_3 + x_4$$

$$x_5 + x_6 \leq 2(x_3 + x_4)$$

Project Selection

ix. At least two of the investments 1, 2, 3 are chosen, or at most three of the investments 4, 5, 6, 7, 8 are chosen:

$$x_1 + x_2 + x_3 \geq 2 \qquad \text{or} \qquad x_4 + x_5 + x_6 + x_7 + x_8 \leq 3$$

$$\text{(I)} \qquad\qquad\qquad\qquad \text{(II)}$$

$$2 - (x_1 + x_2 + x_3) \leq M \text{ y}$$

$$-3 + x_4 + x_5 + x_6 + x_7 + x_8 \leq M(1 - y)$$

$$\text{y} \in \{0,1\}$$

Project Selection

x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

Both projects 1 and 2 are chosen if $x_1 + x_2 = 2$.

At least one of projects 9 and 10 is chosen if $x_9 + x_{10} \geq 1$

Hence, we have $x_1 + x_2 = 2 \Rightarrow x_9 + x_{10} \geq 1$

So, how to express this as a linear constraint?

Project Selection

$$x_1 + x_2 = 2 \Rightarrow x_9 + x_{10} \geq 1$$

Use equivalent representation of if-then statement.

$$x_1 + x_2 - 1 \leq Mz$$
$$1 - (x_9 + x_{10}) \leq \mathrm{M}(1 - z)$$
$$z \in \{0,1\}$$

Project Selection

$$\max \sum_{j=1}^{10} \mathrm{p}_j x_j$$

$$\sum_{j=1}^{10} \mathrm{c}_j x_j \le \mathrm{q}$$

$$x_3 + x_4 \le 1 \qquad \text{(i)}$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 3 \qquad \text{(ii)}$$

$$x_2 \le x_1 \qquad \text{(iii)}$$

$$x_1 \le 1 - x_3 \qquad \text{(iv)}$$

$$x_1 + x_2 = 1 \qquad \text{(v)}$$

$$x_1 - x_2 = 0 \qquad \text{(vi)}$$

$$\left.\begin{array}{l} x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} \ge 2 \\ x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} \le 4 \end{array}\right\} \qquad \text{(vii)}$$

$$\left.\begin{array}{l} x_5 \le x_3 + x_4 \\ x_6 \le x_3 + x_4 \end{array}\right\} \qquad \text{(viii)}$$

$$\left.\begin{array}{l} x_1 + x_2 + x_3 \ge 2(1 - \mathrm{y}) \\ x_4 + x_5 + x_6 + x_7 + x_8 \le 5 - 2y \end{array}\right\} \qquad \text{(ix)}$$

$$\left.\begin{array}{l} x_1 + x_2 \le 1 + z \\ x_9 + x_{10} \ge z \end{array}\right\} \qquad \text{(x)}$$

$$x_1, x_2, \ldots, x_{10} \in \{0,1\}$$

$$\mathrm{y}, \mathrm{z} \in \{0,1\}$$

Traveling Salesman Problem

A sales person lives in city 1. He has to visit each of the cities *2,…,n* exactly once and return home.

Let $\mathbf{c}_{ij}$ be the travel time from city i to city j, $i = 1, . . . , n$; $j = 1, . . . , n$.

What is the order in which she should make her tour to finish as quickly as possible?

What should be the decision variables here?

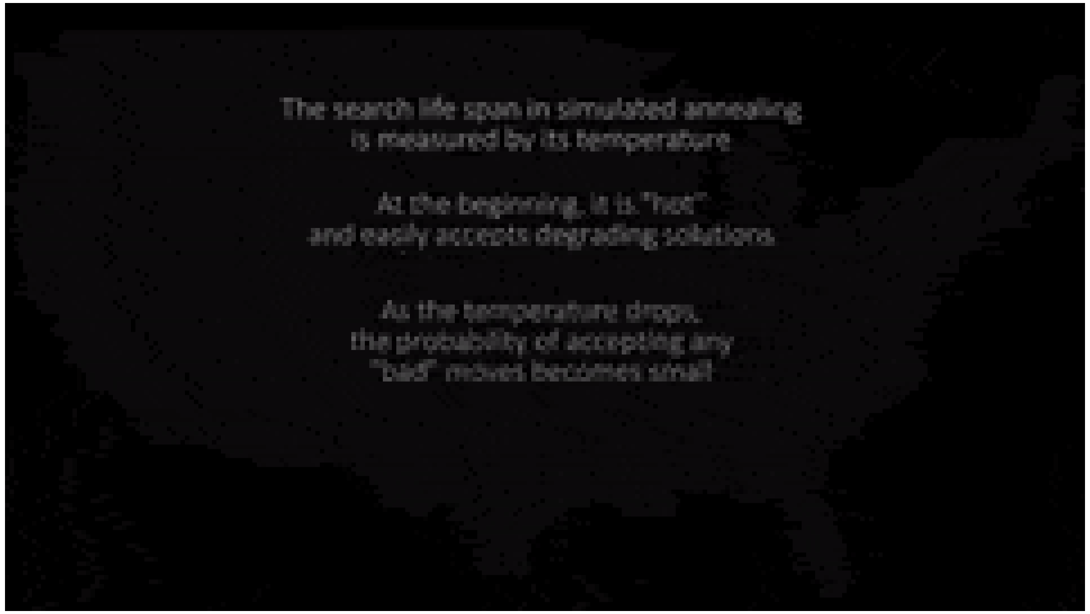
Traveling Salesman Problem

$$x_{ij} = \begin{cases} 1, & \text{if he goes directly from city } i \text{ to city } j \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1, 2, \ldots, n\ ;\ j = 1, 2, \ldots, n$$

Connectivity

Since each city has to be visited only once, the formulation is:



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

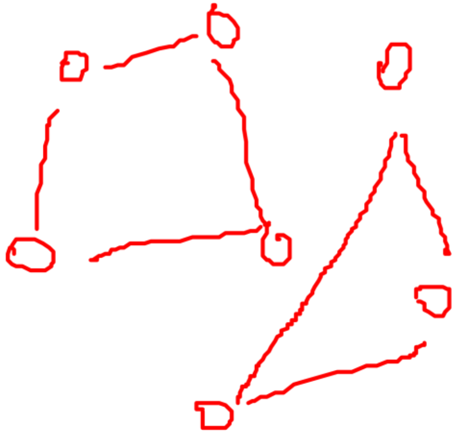
$$s.t.$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1, \ldots, n$$

$$x_{ii} = 0, \qquad i = 1, \ldots, n$$

$$x_{ij} \in \{0, 1\}, \qquad \forall\ i, j = 1, \ldots, n$$



Traveling Salesman Problem

{1,2,3}

{4,5}

S

With this formulation, a feasible solution is:

$$x_{12} = x_{23} = x_{31} = 1,$$
$$x_{45} = x_{54} = 1.$$

But this is a "subtour"! How can we eliminate "subtours"?

We need to impose the constraints:

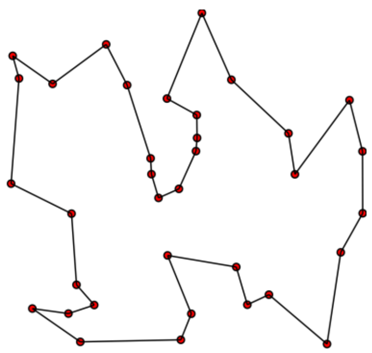
$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \qquad \forall S \subseteq N,\ S \neq \varnothing$$

all

For example, for n=5 and S={1,2,3} this constraint implies:

$$x_{14} + x_{15} + x_{24} + x_{25} + x_{34} + x_{35} \geq 1$$

Traveling Salesman Problem



The difficulty is that there are a total of $2^n - 1$ such constraints. And that is a lot!

- For n = 100, $2^n - 1 \approx 1.27 \times 10^{30}$.

How about the total number of valid tours? That is $(n - 1)!$

- For n = 100, $(n - 1)! \approx 9.33 \times 10^{155}$.

TSP is one of the hardest problems. But several efficient algorithms exist that can find "good" solutions.