Seminar Unit 7

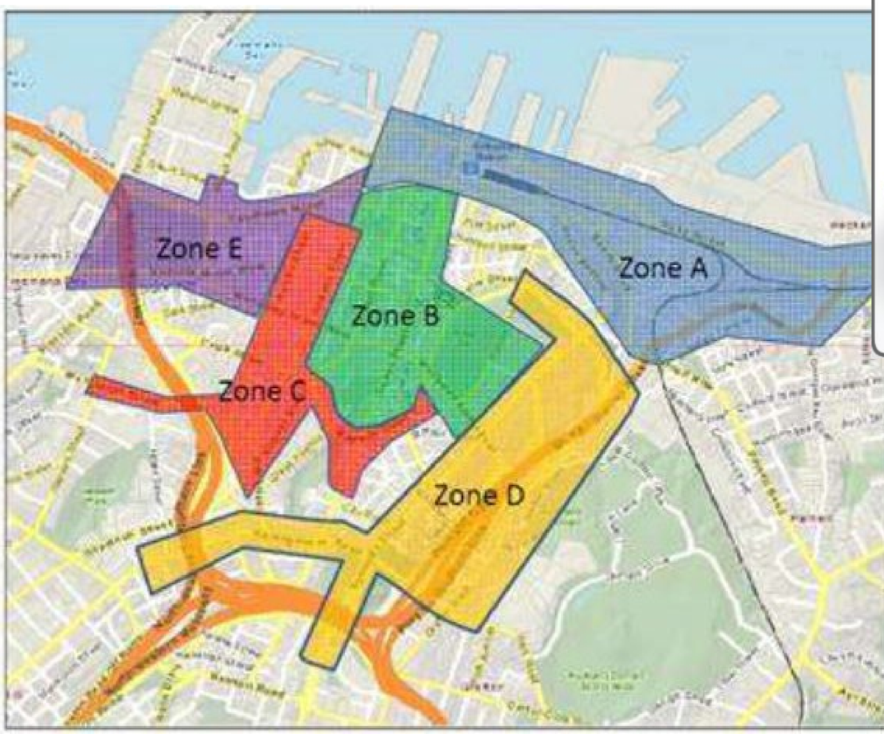
Set covering, set partitioning and set packing problems

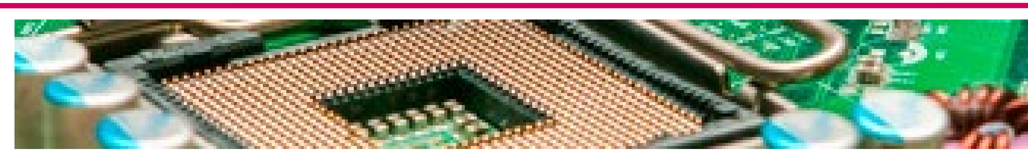
Problemas de recubrimiento, particionamiento y empaquetamiento

Contents Part 1

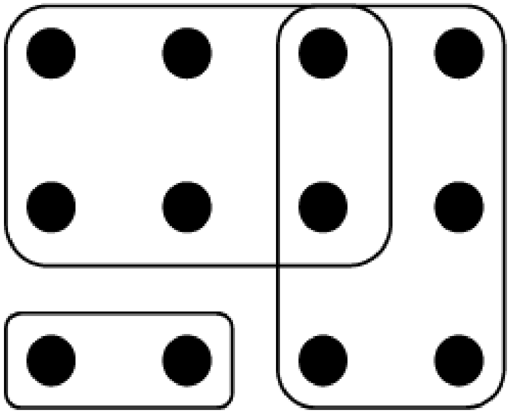
- The set covering problem
- The set partitioning problem
- The set packing problem

The general **goal** is to **find the adequate set of elements** (*aka* resources in general) and perhaps **organise them to cover** a usually bigger set of **necessities** (*aka* as tasks to get our goal)

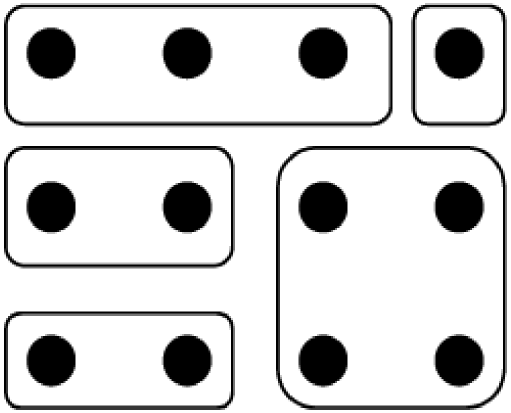




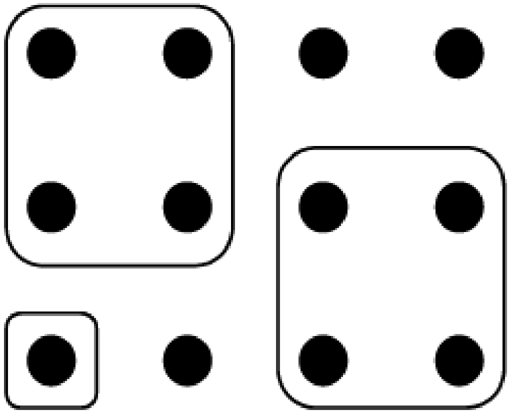
Remember…



Covering



Partitioning



Packing

All items in **at least one** partition

All items **in one and only one** partition

All items in **zero or at most one** partition

∀item i: |covered(i)| ≥ 1

|covered(i)| = 1

|covered(i)| ≤ 1

How to solve (optimise) them?

- Linear programming - **Operational Research** techniques
- Constraint programming
- Search and **heuristic search**
- Metaheuristics:
  - Local search
  - Genetic algorithms
- Dynamic programming
- Particular/specialised algorithms

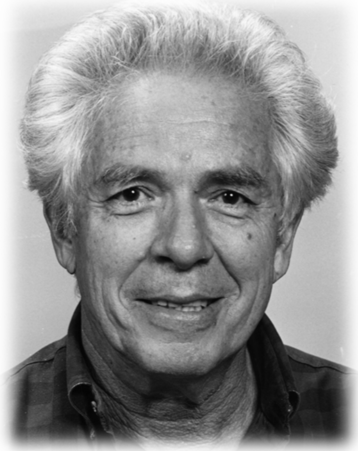
CPAIOR: Integration of AI and OR Techniques in Constraint Programming

Hungarian algorithm

Given a set of resources to be assigned to different tasks, and a cost/time that depends on the pair <resource, task>, which is the optimal assignment of resources to tasks that minimises the total cost/time?

Model

- n resources
- m tasks
- we must create a nonnegative matrix of costs with the same number of rows & columns - we take the max(n,m)



Harold W. Kuhn

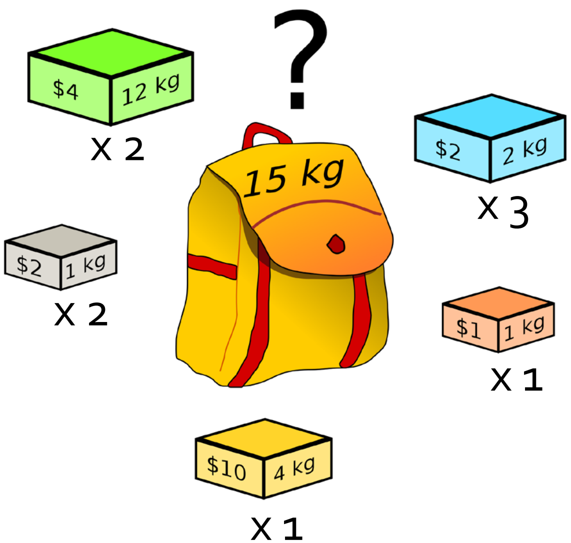
In some terms, similar to **partitioning** or **covering** problems

Knapsack problem as a special case of **resource allocation**

Given a list of items (with a weight and a reward) and a limited capacity collection (knapsack or backpack), which is the set of items for what the total weight is less than or equal to the capacity and the total reward is as large as possible?

Model (for a packing problem)

- n types of items: 1..n
- qi, instances of each type of item ≤ Qi
- ri (reward) and wi (weight) of the i-th type of item
- W, max weight allowed by the knapsack
- xi, integer number of instances of the i-th item type



Knapsack problem

What about having…

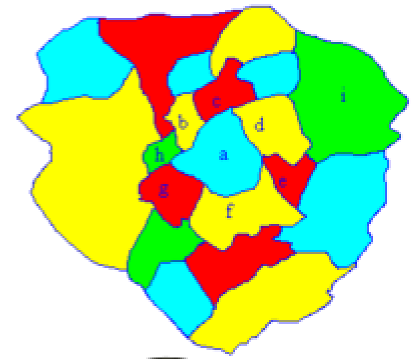
multiple knapsacks to pack all items?

items that provide **different reward depending on** the selected **knapsack**?

Similar to **partitioning** (and perhaps **covering**) problems

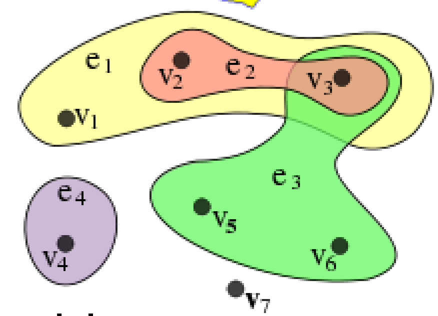
CSP (Constraint Satisfaction Problem)

Given a set of objects (variables) whose state (values) must satisfy a number of constraints or limitations, which is a consistent assignment of values to be considered as a solution?



Model

- n variables: {x1, x2, ..., xn}
- domain for each variable xi∈{d1,d2... dn}
- m contraints that involve any number of variables: {c1, c2, ..., cm} (typically binary constraints)



We can model **covering**, **partitioning** and **packing** problems

CSP (Constraint Satisfaction Problem)

Variants

- **Dynamic** CSPs, where the number of variables and/or constraints is dynamic and changes throughout time

- **Flexible** CSPs, where not all the constraints need to be satisfied – very useful in overconstrained problems; we want to satisfy as many constraints as possible (Constraint Optimization Problems)
  - valued CSPs
  - probabilistic CSPs
  - weighted CSPs, etc.

Task 1. Covering all characteristics

Flight crew assignment

(from Hillier and Lieberman, 2015)

An **airways** company has to **assign** 3 **crews** based in San Francisco **to** the 11 current **flights** listed in the first column of the next table. The other 12 columns show the 12 feasible sequences of flights for a crew. (The numbers in each column indicate the order of the flights)

Exactly three of the **sequences need** to be chosen (one per crew) to cover every flight, although it is permissible to have more than one crew per sequence on a flight (we'd need to pay as if they were working). **Note**: three crews in total, not three crews per flight

The **cost** of **assigning** a **crew** to a particular sequence of **flights** is given in K$ in the bottom row of the table

Task 1. Covering all characteristics – Flight crew assignment

	Feasible Sequence of Flights											
Flight	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1. San Francisco to Los Angeles	1			1			1			1		
2. San Francisco to Denver		1			1			1			1	
3. San Francisco to Seattle			1			1			1			1
4. Los Angeles to Chicago				2			2		3	2		3
5. Los Angeles to San Francisco	2					3				5	5	
6. Chicago to Denver				3	3				4			
7. Chicago to Seattle							3	3		3	3	4
8. Denver to San Francisco		2		4	4				5			
9. Denver to Chicago					2			2			2	
10. Seattle to San Francisco			2				4	4				5
11. Seattle to Los Angeles						2			2	4	4	2
Cost, $1,000's	2	3	4	6	7	5	7	8	9	9	8	9

Goal: **minimise** the **total cost** of the crew assignments that **cover all** the 11 flights

Task 2. Temporal planning for delivering parcels

A **van** is in the depot at **12:00** and needs to deliver five parcels, being back no later than **20:00**. **Moving** from one place to another always **takes 30 minutes**

There are **different** opening hours and duration for delivery

Parcel	Customers' opening hours	Duration for delivery
P1	9:00-13:30; 17:00-20:30	30 minutes
P2	12:00-14:30; 16:30-18:30	60 minutes
P3	14:00-17:00; 18:00-20:30	30 minutes
P4	9:00-17:00; 18:30-20:30	120 minutes
P5	12:00-14:30; 18:30-19:30	60 minutes

Goal: find the (optimal) temporal plan

Model and **solve** the two previous tasks **using Choco**

- Model the variables, constraints and the optimisation function (if necessary)
- Create a Java application and use the services that Choco provides to define all the elements of the subsequent CSP
- Solve the problem and analyse the results