Algorithmic Methods for Mathematical Models (AMMM)

Greedy Algorithms (for Combinatorial Optimization)

Luis Velasco

(lvelasco @ ac.upc.edu)

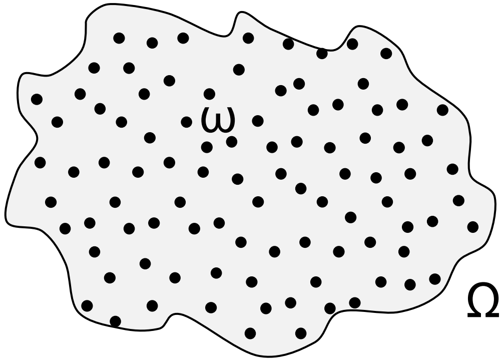
Campus Nord D6-107

Combinatorial Optimization

- A combinatorial optimization problem is defined by:
  - N: finite **ground set** of elements, index *i*
  - Ω: set of **feasible solutions** of *N*
  - c_i: **cost** of the element *i*

$$min_{\omega\subseteq N}\sum_{i\in\omega}c_i$$

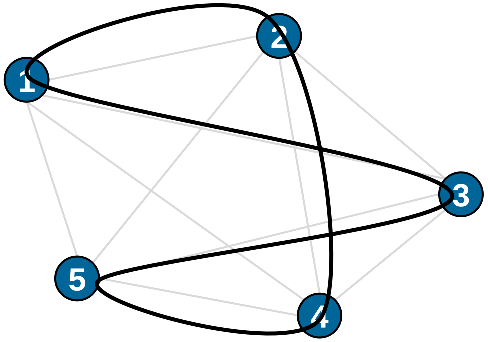
$$\text{s.t.}\quad \omega\in\Omega$$



- Combinatorial problems can be modeled using binary variables $x_i\in\{0,1\}$, one per element.

Example: The Travelling Salesman Problem (TSP)

- Given a graph G(*V*,*E*) with a set of cities (*V*) and their pairwise distances
- The task is to find a **shortest** possible **tour** that visits each city exactly once (Hamiltonian cycle).



Example: The Travelling Salesman Problem (TSP)

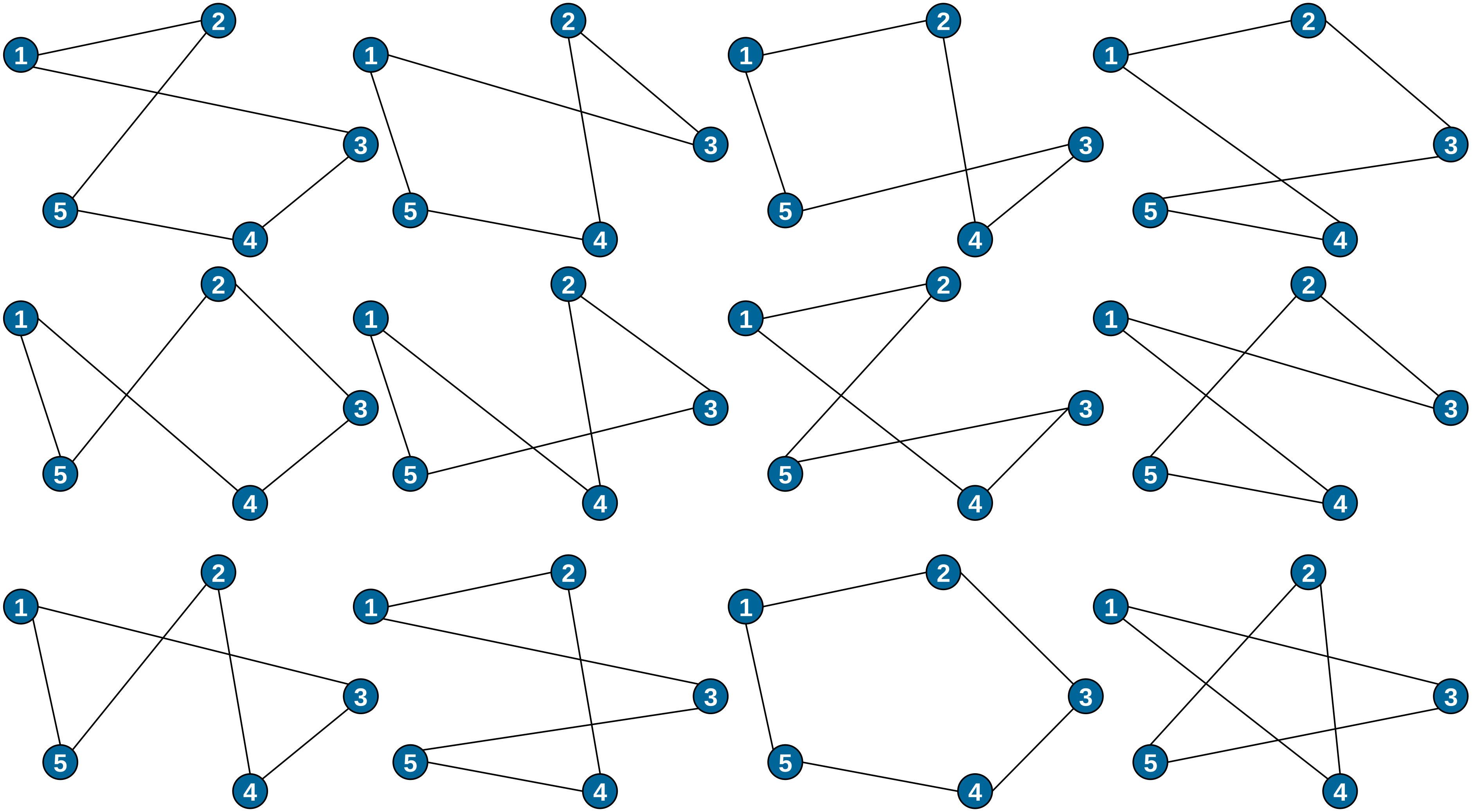
- TSP is a **combinatorial** problem. Its **search space** is (n-1)!/2
  - the ground set is that of all edges connecting the cities to be visited,
  - *F* is formed by all edge subsets that determine a Hamiltonian cycle.
  - *f*(*S*) is the sum of the distances of all edges in each Hamiltonian cycle
- What **factorial** means?

n	search space
3	1
4	3
5	12
6	60
7	360
8	2,520
9	20,160
10	181,440

n	search space
20	6.08E+16
30	4.42E+30
40	1.02E+46
50	3.04E+62
60	6.93E+79
70	8.56E+97
80	4.47E+116
90	8.25E+135
100	4.67E+155

Information on the largest TSP instances solved to date can be found in:
http://www.math.uwaterloo.ca/tsp/optimal/index.html

The (n-1)!/2 combinations (n=5)



Greedy algorithm

- A greedy algorithm builds the solution in an iterative manner.
    - At each iteration, **the best element** from a **candidate list** is added to the **partial solution**
- In general they have **five pillars**:
    - A **candidate set** C, from which a solution is created
    - A **selection function**, which chooses the best candidate *c* to be added
    - A **feasibility function**, to determine if a candidate can be used
    - An **objective function** $f(\omega)$, which assigns a value to a (partial) solution
    - A **solution function**, which indicate when we have a complete solution

Greedy Algorithm for Combinatorial Problems

C: Candidate set, index c

$\omega \subseteq C$: (partial) solution

$q(c,\omega)$: quality of element c *given a partial solution* ω (*greedy function*).

$$q(c,\omega) = \begin{cases} {}^{value}\!/_{cost} & (i.e., \boldsymbol{added\ value}) \\ \infty & if\ \omega \cup \{c\}\ is\ \boldsymbol{Infeasible} \end{cases}$$

```
Initialize $C$
$\omega \leftarrow \{\}$
while $\boxed{\omega \text{ is not a solution}}$ do
        evaluate $q(c, \omega)\ \forall c \in C$
        $c_{best} \leftarrow \boxed{\text{argmax}\{q(c, \omega) \mid c \in C\}}$
        $\omega \leftarrow \omega \cup \{c_{best}\}$
        update $C$, e.g., $C \leftarrow C \setminus \{c_{best}\}$ ($\boxed{\text{you might want to exclude infeasible } c}$)
return $<f(\omega), \omega>$
```

solution function

selection function

feasibility function

Example: TSP

The nearest neighbor (NN) algorithm

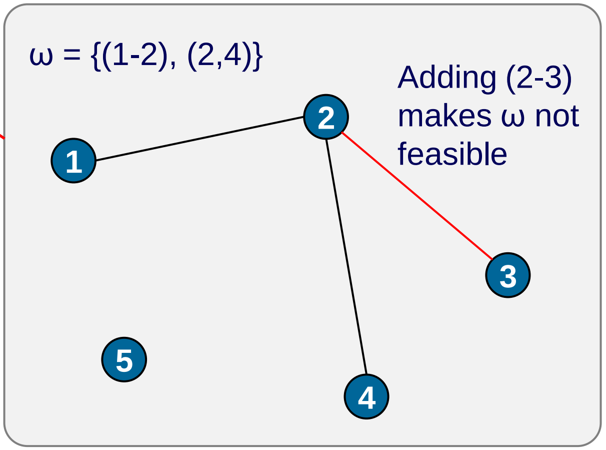
Partial Solution

```
Given graph G($V$,$E$)
$\boxed{\omega} \leftarrow \{v\}$, where $v$ is an arbitrary node (starting point) from $V$
while Tour $\neq \boxed{V}$ do        V is the candidate set
    $C \leftarrow$ set of $\boxed{\text{feasible}}$ links w.r.t. $\omega \subseteq E$
    $e_{best} = (u \in \omega, v \notin \omega) \leftarrow \text{argmin}\{d(e, \omega) \mid e \text{ in } C\}$
    $\omega \leftarrow \omega \cup \{v\}$
return Tour
```

For |*V*| cities randomly distributed on a plane, the algorithm yields:

length = 1.25 * shortest (optimal) length

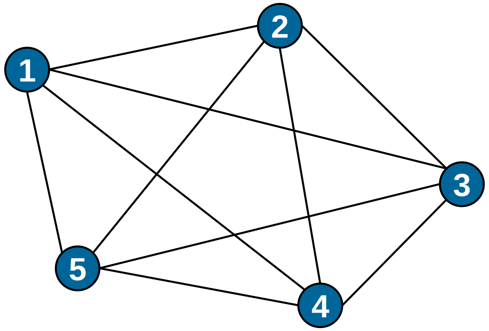
on average.



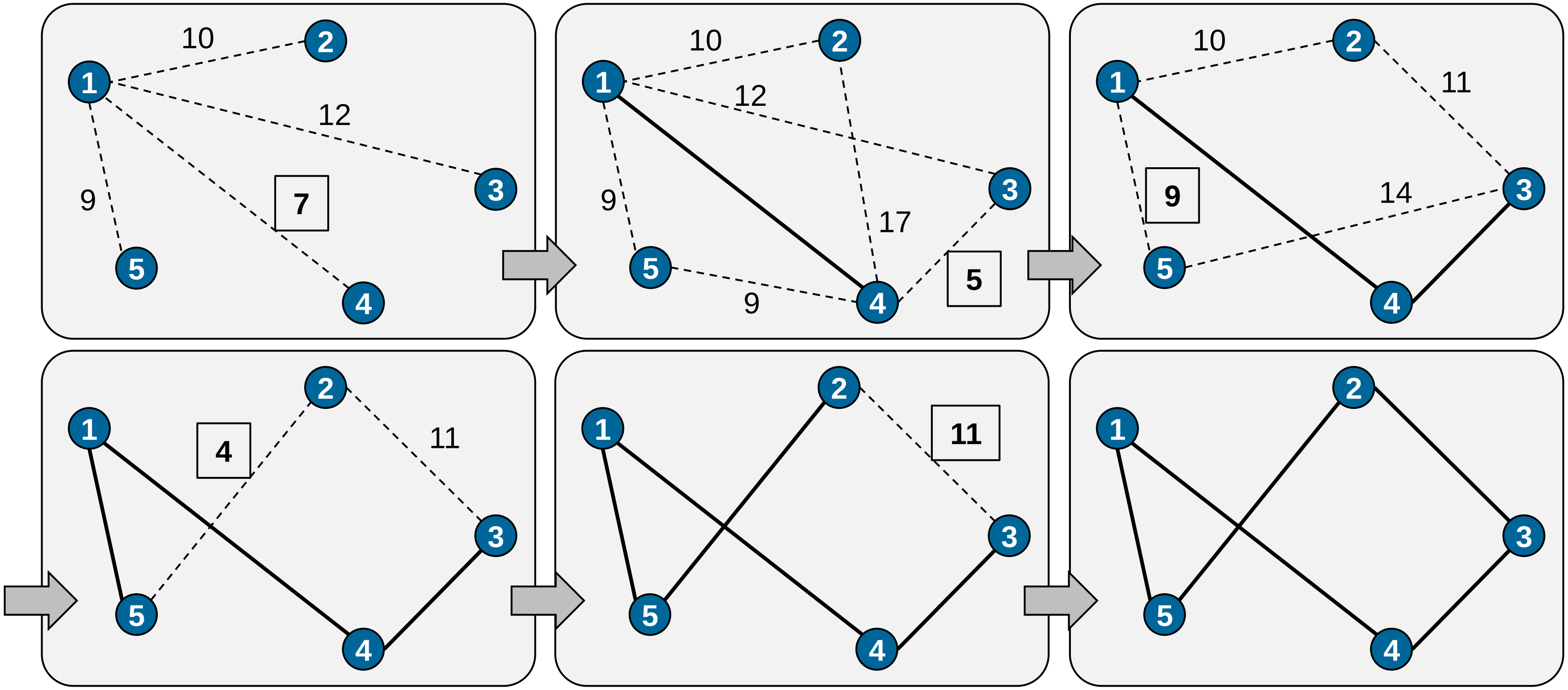
ω = {(1-2), (2,4)}

Adding (2-3) makes ω not feasible

Example: TSP

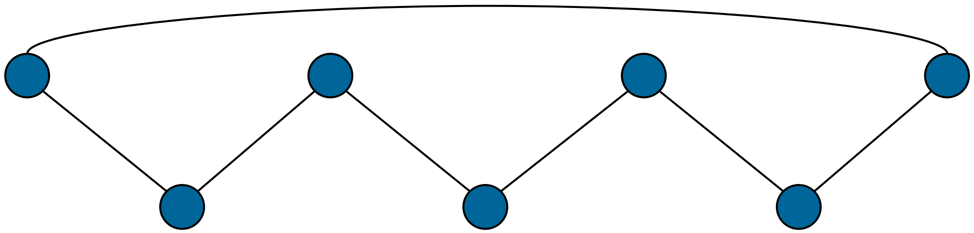


	1	2	3	4	5
1	-	10	12	7	9
2		-	11	17	4
3			-	5	14
4				-	9
5					-



Example: TSP

- A greedy algorithm suffers from myopia.
  - It looks for the best candidate at each iteration.

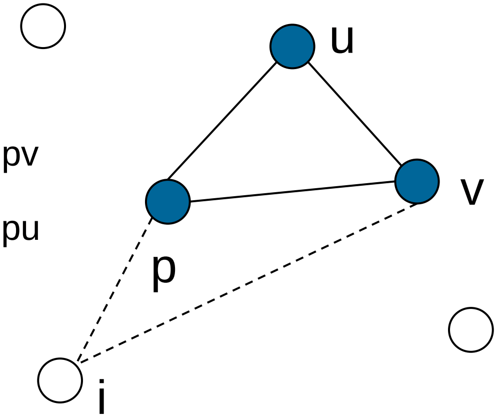


Other algorithms for the TSP

- **Nearest Insertion (greedy)**: From a small cycle, the algorithm expands the cycle by adding the nearest vertex.

$c_{ip}+c_{iv}-c_{pv}$

$c_{ip}+c_{iu}-c_{pu}$



- **Christofides algorithm**: Produces solutions within 3/2 of an optimal solution, i.e., $Z_{heuristic} \leq 3/2\ Z^*$.

- Create the minimum spanning tree MST *T* of G.
- Denote *O* the set of vertices with odd degree in *T*
- Find a perfect matching *M* with minimal weight in the complete graph over the vertices from *O*.
- Combine the edges of *M* and *T* to form a multigraph *H*.
- Form an Eulerian path in *H* (*H* is Eulerian because it is connected, with only even-degree vertices).
- Transform the path found in last step to be Hamiltonian by skipping visited nodes (*shortcutting*).

Algorithmic Methods for Mathematical Models (AMMM)

Greedy Algorithms

Luis Velasco

(lvelasco @ ac.upc.edu)

Campus Nord D6-107