SST problem and Kruskal's algorithm

G. Ferrari Trecate

Dipartimento di Ingegneria Industriale e dell'Informazione
Università degli Studi di Pavia

Industrial Automation

Shortest Spanning Tree (SST)

Example: design of a transport network

A subway must connect several districts

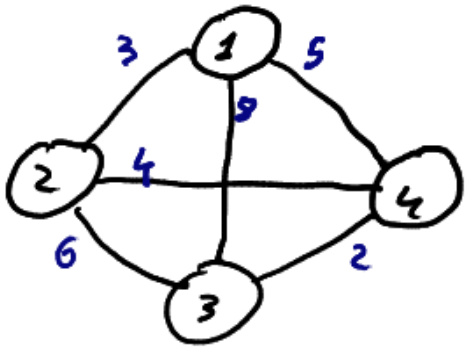
- bidirectional paths
- each straight connection between two stations has a realization cost

Problem: design a minimum-cost network where each pair of districts is connected by a unique path

Network model

Districts = vertices
Possible links = undirected edges
Realization costs = weights on edges

} graph $G=(V,E,c)$



"each pair of districts is connected by a unique path"
↳ feasible solutions are spanning trees
"minimum cost network"
↳ compute a SHORTEST SPANNING TREE (SST)

Rmk. Shortest = of minimal cost
↳ the cost of $G=(V,E,c)$ is $\sum_{e \in E} c(e)$

SST problem as binary LP

Variables: $x_e = \begin{cases} 1 \text{ if } e \in E \text{ belongs to the tree } T \\ 0 \text{ otherwise} \end{cases}$

Binary LP

$$\min_{x_e,\, e \in E} \sum_{e \in E} c(e)\, x_e$$

$$\sum_{e \in E} x_e = n - 1 \qquad (1)$$

$$\sum_{e \in E(U)} x_e \leq |U| - 1, \quad \forall U \subseteq V, \; U \neq \phi \qquad (2)$$

$$x_e \in \{0, 1\} \qquad (3)$$

where $E(U) = \{(i, j) \in E : i, j \in U\}$

$$\min_{x_e,\, e \in E} \sum_{e \in E} c(e)\, x_e$$

$$\sum_{e \in E} x_e = n-1 \qquad (1)$$

$$\sum_{e \in E(U)} x_e \leq |U|-1, \quad \forall U \subseteq V, \; U \neq \phi \qquad (2)$$

$$x_e \in \{0, 1\} \qquad (3)$$

Constraints

(1) : feasible solutions have $n-1$ edges (necessary condition for a spanning tree)

(2) : subtour elimination = feasible solutions are acyclic graphs

↳ (1) + (2) : feasible solutions are spanning trees

$$\min_{x_e,\, e \in E} \sum_{e \in E} c(e)\, x_e$$

$$\sum_{e \in E} x_e = n - 1 \qquad (1)$$

$$\sum_{e \in E(U)} x_e \leq |U| - 1, \quad \forall U \subseteq V, \; U \neq \phi \qquad (2)$$

$$x_e \in \{0, 1\} \qquad (3)$$

Key problems

-) the number of sets U is $2^n - 1$
-) variables x_e are integer and this makes the LP problem NP-hard

Kruskal's algorithm

Direct method - not based on optimization

Input: the undirected network $G = (V, E, c)$

1) Initialization. Build a list L of edges sorted according to non-decreasing costs. Set $\tilde{E} = \emptyset$

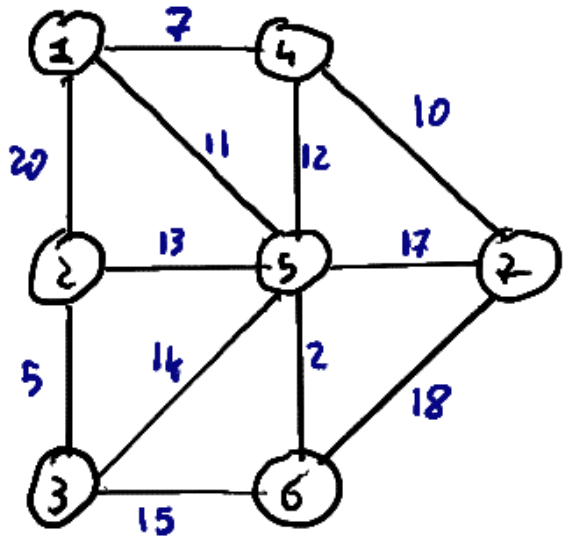
2) Edge selection. Select the first edge $e \in L$. If $\tilde{E} \cup \{e\}$ does not contain a cycle, set $\tilde{E} \leftarrow \tilde{E} \cup \{e\}$. Cut e from L

3) Stop condition. Stop if either

   1) $|\tilde{E}| = n-1 \rightarrow T = (V, \tilde{E})$ is an SST
   2) L is empty $\rightarrow G$ is not connected

   Otherwise go to step 2

Example



List L

Pos.	Edge	Cost	Added to $\tilde{E}$?
1	(5,6)	2	
2	(2,3)	5	
3	(1,4)	7	
4	(4,7)	10	
5	(1,5)	11	
6	(4,5)	12	
7	(2,5)	13	
8	(3,5)	14	
9	(3,6)	15	
10	(5,7)	17	
11	(6,7)	18	
12	(1,2)	20	

Remarks

- Kruskal's algorithm is greedy = at each step the most favorable choice is made
  - ↳ Several optimization problems on network CAN NOT be solved by greedy algorithms → SST is an exception
- There is an implementation of the algorithm with complexity

$$O(m \log n), \quad m = |E|, n = |V|$$

  - ↳ SST is polynomial!

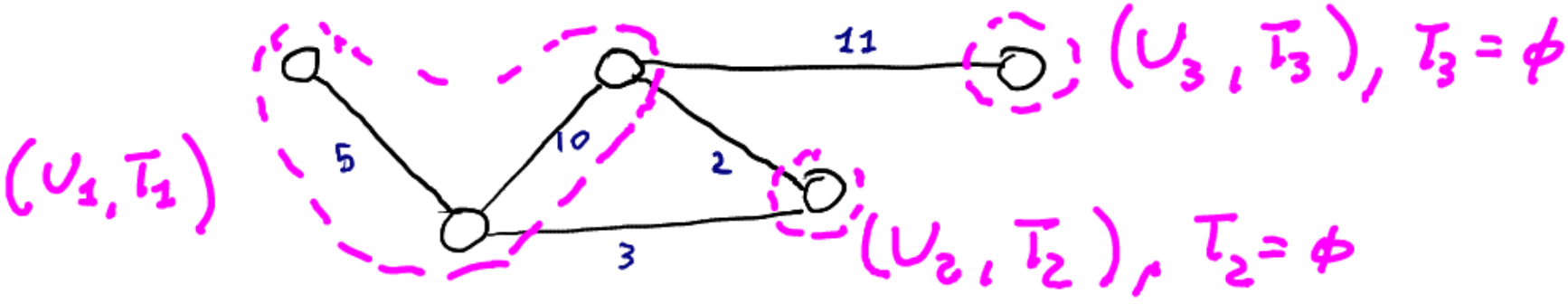
Proof of the correctness

Lemma. Let $G = (V, E, c)$ be an undirected network and let

$$(U_1, T_1), \ldots, (U_r, T_r)$$

be r trees composing a maximal forest.
Let $(u, v) \in E$ be an edge of minimal cost not in $\tilde{E} = \bigcup_{i=1}^{r} T_i$

Then, among all spanning trees containing all edges in $\tilde{E}$
there is one of minimal cost containing (u, v)

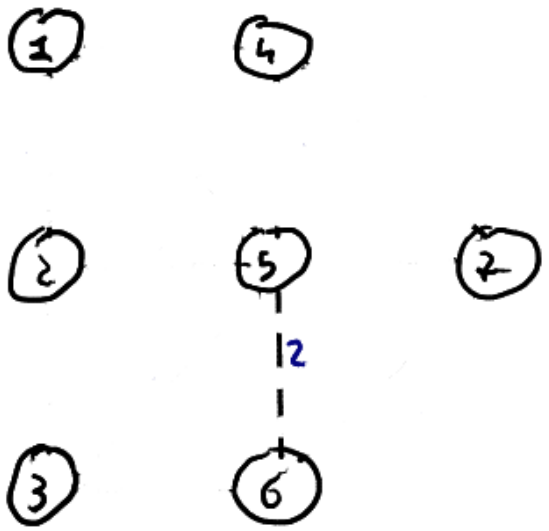


- First iteration

  The maximal forest is (V, ϕ).

  ↳ Trees are isolated nodes $(\{u_1\}, \phi), \ldots, (\{u_n\}, \phi)$

  ↳ Lemma ⇒ ok to include (u, v) of minimal cost



- At the beginning of all other iterations

  The edges in $\tilde{E}$ define a maximal forest by construction

  ↳ Lemma ⇒ it is still ok to add the edge (u, v) with minimal cost