Network problems

A generalisation leading to graphical models

Hugues Talbot

CentraleSupélec
Centre de Vision Numérique (CVN)

May 11, 2018

Outline

Generalisation of transport problems

- Many important optimization problems can best be analyzed by means of a graphical or network representation.
- we consider three specific network models—shortest-path problems, maximum-flow problems, and minimum-spanning tree problems—for which efficient solution procedures exist.
- We also discuss minimum-cost network flow problems (MCNFPs), of which transportation, assignment, transshipment, shortestpath, and maximum-flow problem are all special cases.
- Finally, we discuss a generalization of the transportation simplex, the network simplex, which can be used to solve MCNFPs. We begin the chapter with some basic terms used to describe graphs and networks.

Set notations

- Let E be a finite set
- A set S is called a *subset of E* if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$
- The set of all subsets of E is denoted by $\mathcal{P}(E)$

Exemple

- *If* $E = \{1, 2, 3\}$
- *Then* $\mathcal{P}(E) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$

- *Remark.* $S \in \mathcal{P}(E)$ means that S is a subset of E
- The proposition $S \in \mathcal{P}(E)$ can thus be equivalently written as $S \subseteq E$

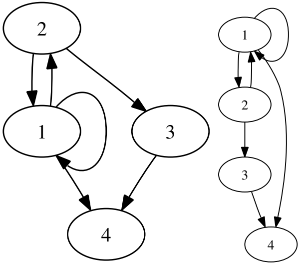
Graph

Definition

- *A* graph *is a pair* $G = (E, \Gamma)$ *where* E *is a finite set and where* Γ *is a map from* E *to* $\mathcal{P}(E)$

Exemple

- $G = (E, \Gamma)$
- *with* $E = \{1, 2, 3, 4\}$ *and*
- Γ *defined by*
  - $\Gamma(1) = \{1, 2, 4\}$
  - $\Gamma(2) = \{3, 1\}$
  - $\Gamma(3) = \{4\}$
  - $\Gamma(5) = \emptyset$



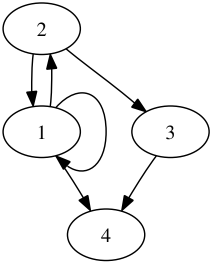
Representation by arrows

Usual terminology

- Any element of E is called a *vertex (of the graph G)*
- Let x and y be two vertices of E, if $y \in \Gamma(x)$,
  - y is a *successor of x* and x is a *predecessor of y*
  - the ordered pair (x, y) is called an *arc (of the graph G)*

Exemple

- *1 is a vertex of G*
- *4 is a successor of 3*
- *2 is a predecessor of 3*
- *Thus, $(3, 4)$ and $(2, 3)$ are two arcs of G*



Directed and undirected graphs

- Sometimes vertices (plural of vertex) are called *nodes*.
- An *arc* is always directed. Bidirectional arcs are called *edges*.

An example: shortest paths

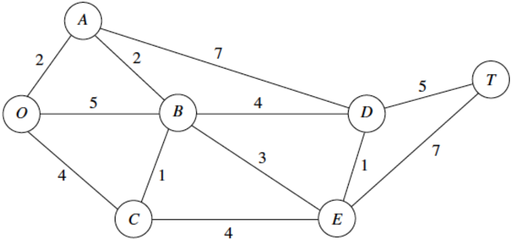


Figure: An undirected graph

An example: shortest paths (2)

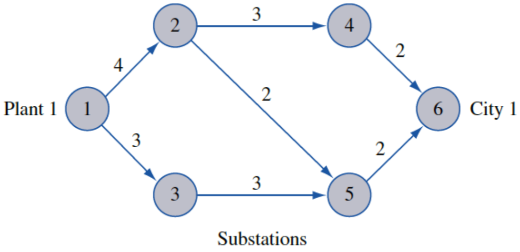


Figure: A directed graph

Shortest path as as transport problem

	Node					
Node	2	3	4	5	6	Supply
1	[4] 1	[3]	[M]	[M]	[M]	1
2	[0]	[M]	[3]	[2] 1	[M]	1
3	[M]	[0] 1	[M]	[3]	[M]	1
4	[M]	[M]	[0] 1	[M]	[2]	1
5	[M]	[M]	[M]	[0]	[2] 1	1
Demand	1	1	1	1	1	

Figure: Transport formulatio of the shortest path problem

Minimum spanning tree

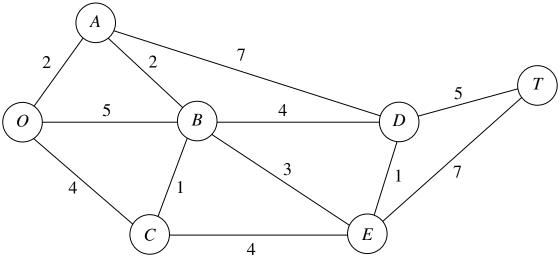


Figure: Example for the Minimum Spanning Tree

Minimum spanning tree

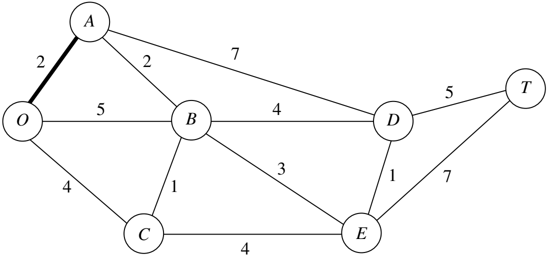


Figure: Example for the Minimum Spanning Tree

Minimum spanning tree

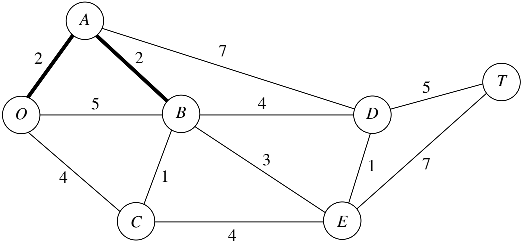


Figure: Example for the Minimum Spanning Tree

Minimum spanning tree

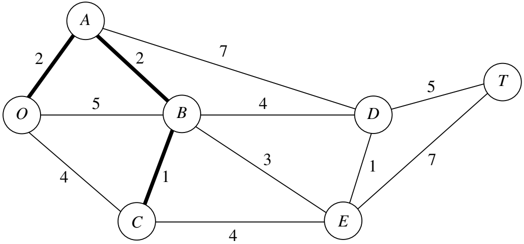


Figure: Example for the Minimum Spanning Tree

Minimum spanning tree

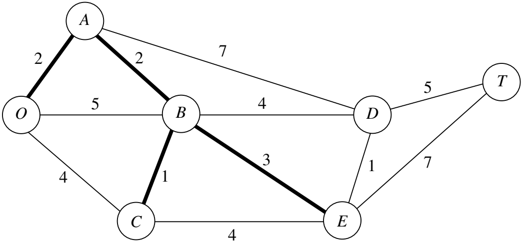


Figure: Example for the Minimum Spanning Tree

Minimum spanning tree

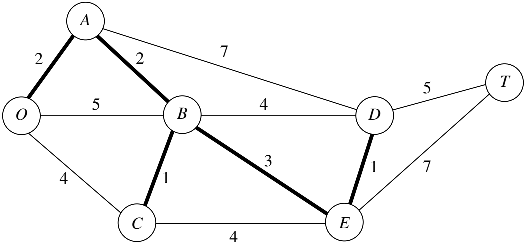


Figure: Example for the Minimum Spanning Tree

Minimum spanning tree

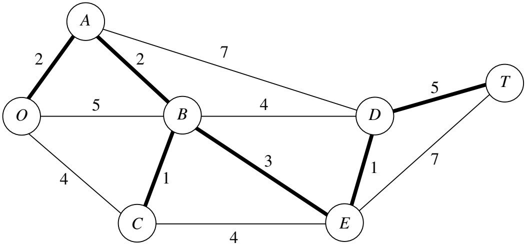


Figure: Example for the Minimum Spanning Tree

MST as a integer program

- Let x_{ij} be 1 if edge ij is in the tree T
- Need constraint to ensure
  - $n-1$ edges in T
  - no cycle in T
- First constraint:

$$\sum_{ij \in E} x_{ij} = n-1$$

- Second constraint: Subtour elimination constraint: any subset of k vertices must have at most $k-1$ edges contained in that subset:

$$\sum_{ij \in E; i \in S, j \in S} x_{ij} \leq |S| - 1, \forall S \subseteq V$$

MST formulation as an IP

$$\text{min.} \sum_{ij \in E} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{ij \in E} x_{ij} = n - 1 \tag{2}$$

$$\sum_{ij \in E; i \in S, j \in S} x_{ij} \leq |S| - 1, \forall S \subseteq V \tag{3}$$

$$x_{ij} \in \{0, 1\} \tag{4}$$

Note: this formulation has an exponential number of constraints. The LP relaxation solves the MST exactly.

Maximum flow problem

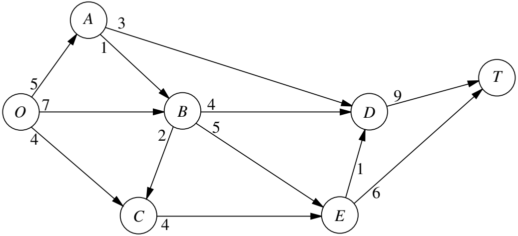


Figure: Example for the Maximum flow problem

Maxflow formulation as an LP

Given $\Gamma = (V, E), s, t, c$ respectively the graphe, source, sink and costs, the MF LP is

$$\text{max.} \sum_{s,j \in E} f_{sj} \tag{5}$$

$$\text{s.t.} \sum_{ij \in E} f_{ij} = \sum_{jk \in E} f_{jk}, \forall j \in V - \{s, t\} \tag{6}$$

$$f_{ij} \leq c_{ij} \tag{7}$$

$$f_{ij} \geq 0 \tag{8}$$

Dual problem and algorithms

- The dual problem is the minimum cut problem. See handout.
- Algorithms: Ford-Fulkerson, Edmonds-Karp, Push-relabel...
- Important problem, leading to *graph cuts*, Boykov algorithm and efficient solution to *Markov Random Fields*.

Maximum flow example solution

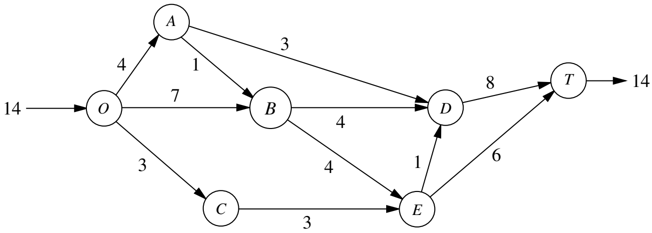


Figure: Solution for the Maximum flow problem

Mincut example solution

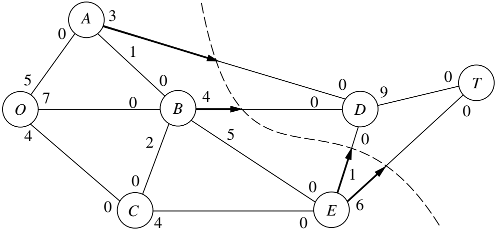


Figure: Dual mincut solution