Foundations of Operations Research

Master of Science in Computer Engineering

Roberto Cordone
`roberto.cordone@unimi.it`

Tuesday 13.15 - 15.15

Thursday 10.15 - 13.15

`http://homes.di.unimi.it/~cordone/courses/2014-for/2014-for.html`

Motivation

Many decision-making problems can be formulated in terms of networks

- deciding which links to activate in order to connect all towns
- determining the fastest route from origin to target
- finding the largest group of related users in a social network
- finding the shortest closed path to complete mail delivery

And so on:

- service, facility, plant location
- project or production planning
- resource management
- activity scheduling
- ...

Graphs

Any binary relation on a finite ground set $V = \{v_1, \dots, v_n\}$
can be described listing the pairs of elements of V which are related

$$E = \{[i,j] : i \in V, j \in V, i \text{ and } j \text{ are related } \} \Rightarrow E \subseteq V \times V$$

A standard way to represent a binary relation on a ground set is named graph $G = (V, E)$, i. e. a pair of sets:

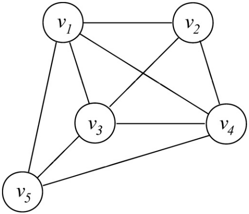
- a set V of elementary objects named vertices
- a set E of unordered pairs of objects from V named edges

The graphical representation of a graph depicts vertices as points (or round shapes) and edges as lines

$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

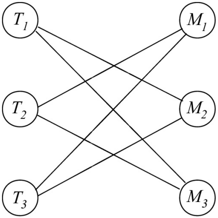
$$E = \{[v_1, v_2], [v_1, v_3], [v_1, v_4], [v_1, v_5], [v_2, v_3], [v_2, v_4], [v_3, v_4], [v_3, v_5], [v_4, v_5]\}$$

Notice the square brackets:
no order between the elements of a pair



Examples

- road networks: the vertices stand for cities, the edges for roads
- electric grids: the vertices stand for plants, stations or users, the edges for power lines
- telecommunication networks: the vertices stand for transmitters, transponders and receivers, the edges for links
- social networks: the vertices stand for users, the edges for human relations
- games: the vertices stand for positions, the edges for moves
- (in)compatibility relations; the vertices stand for objects/persons, the edges for (in)compatible object/person pairs



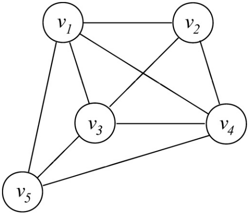
V collects tasks and machines
$V = \{T_1, T_2, T_3, M_1, M_2, M_3\}$

Edge $[i, j]$ indicates that task i can be performed by machine j

A task cannot be performed on a task, nor a machine on a machine

Graph topology

- i and j are the extreme vertices of edge $[i, j]$
- two vertices i and j are adjacent if edge $[i, j]$ exists
- edge $[i, j]$ is incident to vertices i and j
- the degree δ_v of a vertex v is the number of incident edges

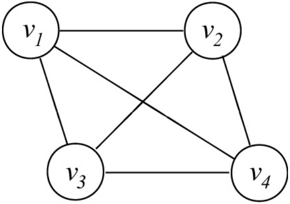


- v_1 and v_2 are the extreme vertices of $[v_1, v_2]$
- v_3 and v_4 are adjacent (edge $[v_3, v_4]$ exists)
- edge $[v_3, v_5]$ is incident to vertices v_3 and v_5
- the degree of vertex v_3 is $\delta_{v_3} = 4$

Complete graphs

A graph is complete when all pairs of vertices correspond to an edge

$$E = \{[v_i, v_j] : v_i \in V, v_j \in V, i < j\}$$



A complete graph with n vertices includes

$$m = \frac{n\,(n-1)}{2} \text{ edges}$$

Why?

All graphs have $m \leq \dfrac{n\,(n-1)}{2}$ edges

Subgraphs

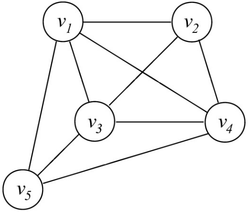
$H = (U, X)$ is a subgraph of $G = (V, E)$ if

- it is a graph
- $U \subseteq V$ and $X \subseteq E$

It is a spanning subgraph when $U = V$

It is an induced subgraph when $X = E_U = \{[u, v] \in E : u, v \in U\}$

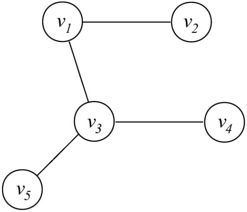
$G = (V, E)$



$V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{[v_1, v_2], [v_1, v_3], [v_1, v_4], [v_1, v_5], [v_2, v_3], [v_2, v_4], [v_3, v_4], [v_3, v_5], [v_4, v_5]\}$

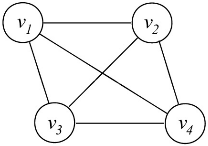
$H_1 = (U_1, X_1)$



$U_1 = \{v_1, v_2, v_3, v_4, v_5\} = V$

$X_1 = \{[v_1, v_2], [v_1, v_3], [v_3, v_4], [v_3, v_5]\}$

$H_2 = (U_2, X_2)$



$U_2 = \{v_1, v_2, v_3, v_4\}$

$X_2 = \{[v_1, v_2], [v_1, v_3], [v_1, v_4], [v_2, v_3], [v_2, v_4], [v_3, v_4]\} = E_{U_2}$

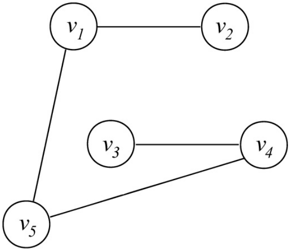
- A path is a sequence of edges, each sharing one extreme with the previous and the other with the next edge (*if previous and next exist*)

$$P = \left([v_{\pi_0}, v_{\pi_1}], [v_{\pi_1}, v_{\pi_2}], \ldots, [v_{\pi_{k-1}}, v_{\pi_k}] \right)$$

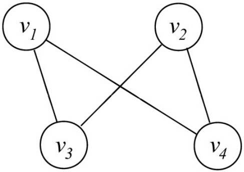
The extreme vertices v_{π_0} and v_{π_k} are connected

- A cycle is a path whose first and last extreme vertices coincide

$$v_{\pi_k} = v_{\pi_0}$$



$P = ([v_1, v_2], [v_1, v_5], [v_4, v_5], [v_3, v_4])$



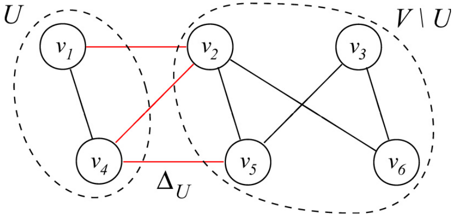
$P = ([v_1, v_3], [v_2, v_3], [v_2, v_4], [v_1, v_4])$

In a connected graph all pairs of vertices are connected by a path

Cuts

- Given a subset of vertices $U \subset V$, the induced cut Δ_U is the subset of edges with one extreme in U and the other in $V \setminus U$

$$\Delta_U = \{[u, v] \in E : |[u, v] \cap U| = |[u, v] \cap (V \setminus U)| = 1\}$$



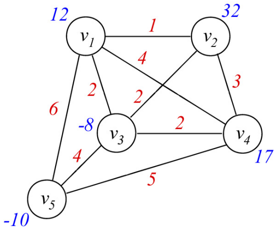
$$U = \{v_1, v_4\}$$

$$\Delta_U = \{[v_1, v_2], [v_2, v_4], [v_4, v_5]\}$$

Weighted graphs

One or more vertex/edge weight function can be defined

- A vertex-weighted graph (V, E, w) is a graph $G = (V, E)$ whose vertices are associated to quantitative information $w : V \to \mathbb{R}$
- An edge-weighted graph (V, E, c) is a graph $G = (V, E)$ whose edges are associated to quantitative information $c : E \to \mathbb{R}$



Application	Vertices	Edges
road networks	trips generated or attracted	lengths, travel times or travel costs
electric grids	power produced or consumed	link building cost
telecommunication networks	traffic demand	link capacity or cost
social networks	individual value	relation strength
games	position quality	move probability or cost
(in)compatibility relations	element utility	strength of (in)compatibility

Modelling with graphs (1)

What is the largest set of people I can contact by way of introduction?

The vertex set V includes all individuals (I am vertex $i \in V$); the edge set E includes all acquaintancies (pairs of individuals who know each other)

Find the maximum cardinality subset $U \subseteq V$ which includes only vertices u such that there exist a path P_{iu} between i and u

$$U = \left\{ u \in V : \exists P_{iu} = \left(\left[v_{\pi_0}, v_{\pi_1} \right], \ldots, \left[v_{\pi_{k-1}}, v_{\pi_k} \right] \right) \text{ with } v_{\pi_0} = i, v_{\pi_k} = u \right\}$$

Is it true that everyone is six steps away from any other person in the world, by way of introduction?

The vertex set V includes all individuals; the edge set E all acquaintancies

Find for each individual $v \in V$ the maximum cardinality subset $U_v^6 \subseteq V$ which includes only vertices u such that there exist a path P_{vu}^6 of at most 6 edges between v and u

$$U_v^6 = \left\{ u \in V : \exists P_{vu}^6 = \left(\left[v_{\pi_0}, v_{\pi_1} \right], \ldots, \left[v_{\pi_{k-1}}, v_{\pi_k} \right] \right) \text{ with } v_{\pi_0} = v, v_{\pi_k} = u, k \leq 6 \right\}$$

If $U_v^6 = V$ for all $v \in V$, the "six-degrees-of separation" statement is correct

Modelling with graphs (2)

Compute the Erdős number of a mathematician

The vertex set V includes all mathematicians (the given one is u, Erdős is v); the edge set E includes all pairs with a published joint work

Find the minimum cardinality path P_{uv} between u and v

$$\min |P_{uv}| \text{ such that } P_{uv} = \left([u, v_{\pi_1}], \ldots, \left[v_{\pi_{k-1}}, v \right] \right)$$

A museum consists of a set of corridors, crossing each other in halls.
Where should they be positioned to have a guard close to each corridor?
How many guards are required to control the whole museum?

The vertex set V includes all halls, the edge set E all corridors

Find the minimum cardinality subset of vertices $U \subseteq V$ such that each edge of the graph is adjacent to at least one vertex of U

$$\min |U| \text{ such that } X = \{[u, v] \in E : [u, v] \cap U \neq \emptyset\} = E$$

Modelling with graphs (3)

Which railway tracks should be bombed in order to destroy any connection between an enemy industrial centre and the battlefront?

The vertex set V includes all stations (v is the industrial centre, V^* collects the stations on the battlefront), the edge set includes all rail tracks

$$\begin{aligned} &\min |\Delta_U| \\ &\quad \Delta_U = \{[u,v] \in E : |[u,v] \cap U| = |[u,v] \cap (V \setminus U)| = 1\} \\ &\quad U \ni v \\ &\quad U \subseteq V \setminus V^* \end{aligned}$$

Given a set of possible financial investments, their expected return on investment (ROI) and the correlation matrix between any two of them, what is the most profitable subset of pairwise uncorrelated investments?

The vertex set V includes all investments, the weight w_v provides the ROI of investment $v \in V$, the edge set includes all correlated pairs

$$\max \sum_{v \in U} w_v \text{ such that } U \subseteq V \text{ and } E_U = \emptyset$$

What is the shortest chain of one-letter exchanges from `HAND` *to* `FOOT`*?*

. . . do it yourselves

Combinatorial models

What about the decision variables, the objective function, the inequality constraints...?

Combinatorial models concern subgraphs (more in general, subsets), instead of numerical variables and inequalities

They can always be reduced to mathematical programming models with a suitable definition of binary variables (cfr. the assignment problem)

We shall see how in a future lesson. . .

Directed graphs

If the binary relation is asymmetric, the order in element pairs is relevant

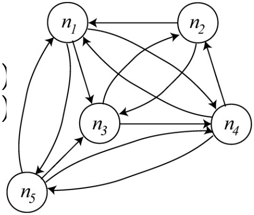
Its model is a pair of sets $G = (N, A)$ named directed graph (digraph):

- a set of elementary objects named nodes
- a set of ordered pairs of nodes named arcs

The graphical representation of a directed graph depicts nodes as points (or round shapes), arcs as lines and their directions as arrows

$$N = \{n_1, n_2, n_3, n_4, n_5\}$$

$$\begin{aligned} A = \{&(n_1, n_3), (n_1, n_4), (n_1, n_5), (n_2, n_1), (n_2, n_3) \\ &(n_3, n_2), (n_3, n_4), (n_4, n_1), (n_4, n_2), (n_4, n_5) \\ &(n_5, n_1), (n_5, n_3), (n_5, n_4)\} \end{aligned}$$



Notice the round parenthesis...

Directed paths, cycles and cuts

- i is the tail and j is the head of arc (i, j)
- arc (i, j) is an outgoing arc for i, an ingoing arc for j
- the outdegree δ_i^+ of a node $i \in N$ is the number of outgoing arcs
- the indegree δ_i^- of a node $i \in N$ is the number of ingoing arcs
- a directed path is a sequence of arcs whose head coincides with the tail of the following one (*except the last arc*)

$$P = \left(\left(i_{\pi_0}, i_{\pi_1} \right), \left(i_{\pi_1}, i_{\pi_2} \right), \ldots, \left(i_{\pi_{k-1}}, i_{\pi_k} \right) \right)$$

  Nodes i_{π_0} and i_{π_k} are strongly connected and in a strongly connected graph all pairs of nodes are strongly connected
- a directed cycle (circuit) is a directed path whose first and last node coincide

$$i_{\pi_k} = i_{\pi_0}$$

- given a subset of nodes $U \subset N$, the outgoing (ingoing) cut Δ_U^+ (Δ_U^-) is the subset of edges with tail (head) in U and head (tail) in $N \setminus U$

$$\Delta_U^+ = \{(i, j) \in A : i \in U, j \in N \setminus U\}$$
$$\Delta_U^- = \{(i, j) \in A : i \in N \setminus U, j \in U\}$$

Modelling with directed graphs

Some social networks consider directed relations between members (followers and "leaders")

In most games, positions can evolve irreversibly into other positions (e. g. captures in chess, ordinary pieces in draughts/checkers, tic-tac-toe. . .)

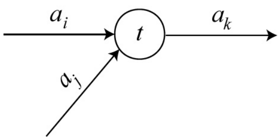
In urban road networks, several streets are one-way only

Consider a project, composed of a set of activities subject to a binary precedence relation, which requires one activity to be terminated before starting another one: $a_i \prec a_k$ *and* $a_j \prec a_k$

Activity-on-arc model (AOA)

Node $\leftrightarrow$ "milestone" event

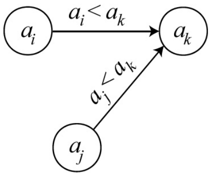
Arc $\leftrightarrow$ activity



Activity-on-node model (AON)

Node $\leftrightarrow$ activity

Arc $\leftrightarrow$ precedence



Graph representations

Nodes are put into one-to-one correspondence with natural numbers

$$N \leftrightarrow \{1, \dots, |N|\}$$

Node weight functions are represented as vectors

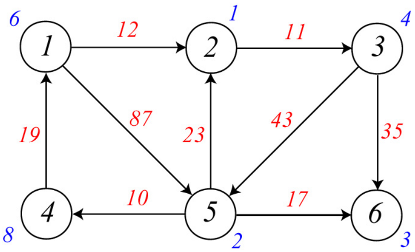
Three representations are common for arcs:

1. arc list: a simple list/vector including all arcs
2. adjacency matrix: a square matrix whose cells correspond to node pairs
3. forward (backward) star: a list/vector including for each node $i \in N$ a list/vector of outgoing (ingoing) arcs

Arc weight functions can be easily included in all three representations

In undirected graphs, forward and backward stars merge into incidence lists

Arc list



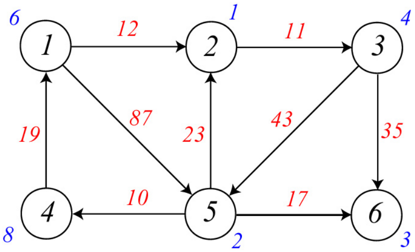
$$N \rightarrow \{1, 2, 3, 4, 5, 6\}$$

$$w \rightarrow [6\ 1\ 4\ 8\ 2\ 3]$$

$$A, c \rightarrow \big((1,2,12), (1,5,87), (2,3,11), (3,5,43), (3,6,35), (4,1,19), (5,2,23), (5,4,10), (5,6,17)\big)$$

- Advantage: compact representation (proportional to $|A|$)
- Disadvantage: inefficient search for a given arc (proportional to $|A|$)

Adjacency matrix



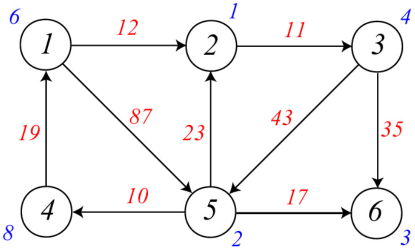
$$N \rightarrow \{1, 2, 3, 4, 5, 6\}$$
$$w \rightarrow [6\ 1\ 4\ 8\ 2\ 3]$$

$$A, c \rightarrow \begin{bmatrix} - & 12 & - & - & 87 & - \\ - & - & 11 & - & - & - \\ - & - & - & - & 43 & 35 \\ 19 & - & - & - & - & - \\ - & 23 & - & 10 & - & 17 \\ - & - & - & - & - & - \end{bmatrix}$$

where "–" is a conventional numerical value to signify "no arc"

- Advantage: very efficient search for a given arc (constant time)
- Disadvantage: huge memory occupation (proportional to $|N|^2$)

Forward star



$$N \rightarrow \{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$$
$$w \rightarrow [6\ 1\ 4\ 8\ 2\ 3]$$

$$A, c \rightarrow \begin{bmatrix} (2,12)\,,(5,87) \\ (3,11) \\ (5,43)\,,(6,35) \\ (1,19) \\ (2,23)\,,(4,10)\,,(6,17) \\ - \end{bmatrix}$$

where "–" marks the end of the list

- Advantage: fairly efficient search for a given arc (proportional to $\left|\delta_v^+\right|$)
- Disadvantage: no information on ingoing arcs (unless accompanied by the backward star)

Basics on algorithm complexity

An algorithm is a finite sequence of formally defined instructions which in a finite time turns an instance of a problem into its corresponding solution

The time required depends on the instance, but also on the computer

To discuss complexity in general terms, we define the asymptotic worst-case complexity

1. we measure time as the number of elementary operations performed (this is to make it independent from the computer)
2. we identify a significant number characterizing the size of the instance (e. g., the number of nodes or arcs of a graph, the number of variables or constraints of a mathematical programming formulation)
3. for each size n, we consider the instance requiring the largest time (this reduces the complexity to a function of n)
4. we approximate the function from above with a simpler one and group into a general class all functions with the same approximation
5. since large values of n characterize the behaviour of an algorithm more than small ones, the approximation focuses on $n \to +\infty$

Big-O notation

$$f(n) \in O(g(n))$$

formally means that

$$\exists c > 0, n_0 \in \mathbb{N} : f(n) \leq c\, g(n) \ \text{ for all } n \geq n_0$$

where c and n_0 are independent from n

Informally, it means that for large values of n, function $f(n)$ assumes values which are at most proportional to the values of function $g(n)$

Therefore, $g(n)$ is a worst-case asymptotic upper bound on $f(n)$

Examples

- $f(n) = 3n^3 + n^2 + 10 \in O\left(n^3\right)$
- $f(n) = 6n^2 + 7 \in O\left(n^2\right)$
- $|E| \leq \dfrac{|V|(|V|-1)}{2}$, so that $|E| \in O\left(|V|^2\right)$

Examples

Searching for a number x in a vector V of n elements

Sequential search (any vector)

$i := 1$; Found $:= False$;
While (Found $= False$) and ($i \leq n$) do
    If ($x = V[i]$) then Found $:= True$;
    $i := i + 1$;
EndWhile
Return Found;

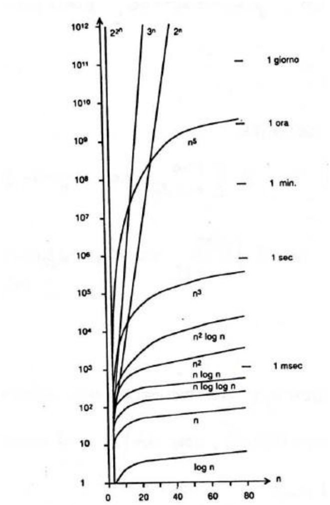
$O(n)$ operations

Dychotomic search (only for sorted vectors!)

$l := 1$; $r := n$;
While ($l < r$) do
    $m := (l + r)/2$;
    If ($x \leq V[m]$)
        then $r := m$
        else $l := m + 1$;
EndWhile
If ($x = V[l]$)
    then Return $True$
    else Return $False$;

$O(\log_2 n)$ operations

Polynomial versus exponential growth



The capital distinction is between

- polynomial complexity:
  $f(n) \in O(n^d)$ for some constant d
- exponential complexity:
  $f(n) \in O(2^n)$

Assume 1 operation/μsec

n	n^2 ops.	2^n ops.
1	1μ sec	2μ secs
10	0.1 msecs	1 msec
20	0.4 msecs	1 sec
30	0.9 msecs	17.9 mins
40	1.6 msecs	12.7 days
50	2.5 msecs	35.7 years
60	3.6 msecs	366 centuries