Prise de décision avec graphes

Professeur: Mohammed Saddoune

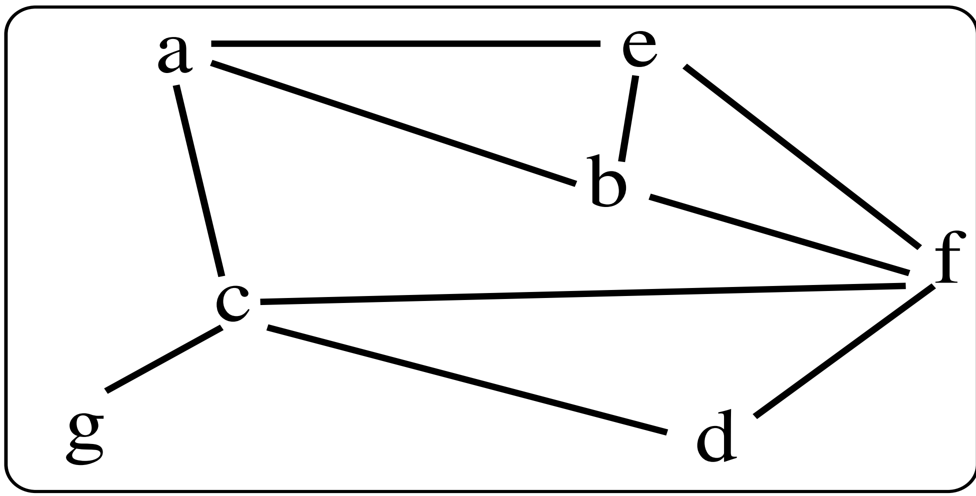
Définition:

C'est quoi un graphe ?

Définition:

Un graphe est un couple G = (V, E) où V est un ensemble fini appelé ensemble des sommets et E est un sous-ensemble de V×V (non ordonné) appelé ensemble des relations entre les sommets.

GRAPHES NON ORIENTÉS



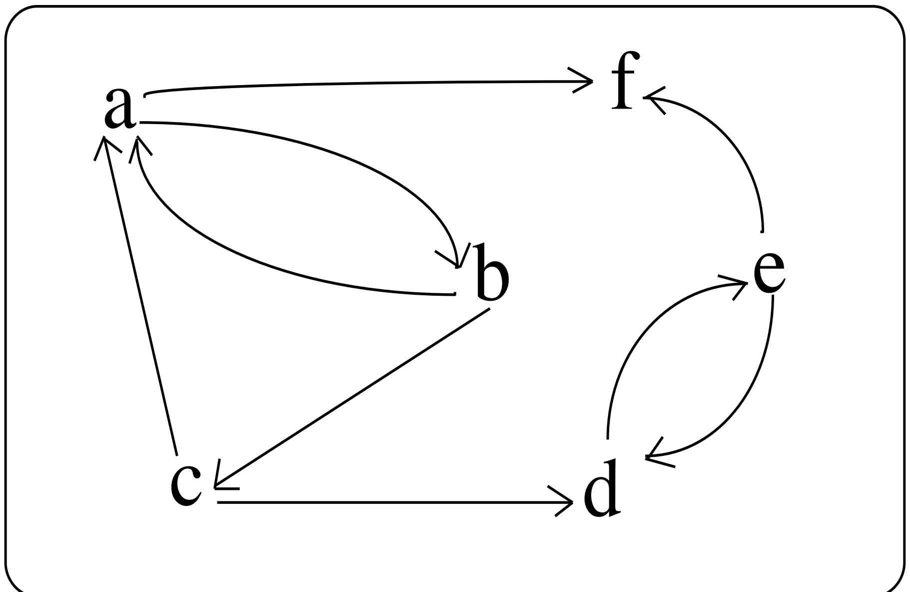
G2=(X2,U2)

noeuds

X2 = {sommets}

U2 = {arêtes} =
{[a,e],[a,b],
[a,c],[b,e],..}

GRAPHES ORIENTÉS



G1 = (X1 , U1)

X1 = {<u>sommets</u>}

= {a,b,c,d,e,f}

U1 = {<u>arcs</u>}

$U1 \subseteq X1 \times X1$

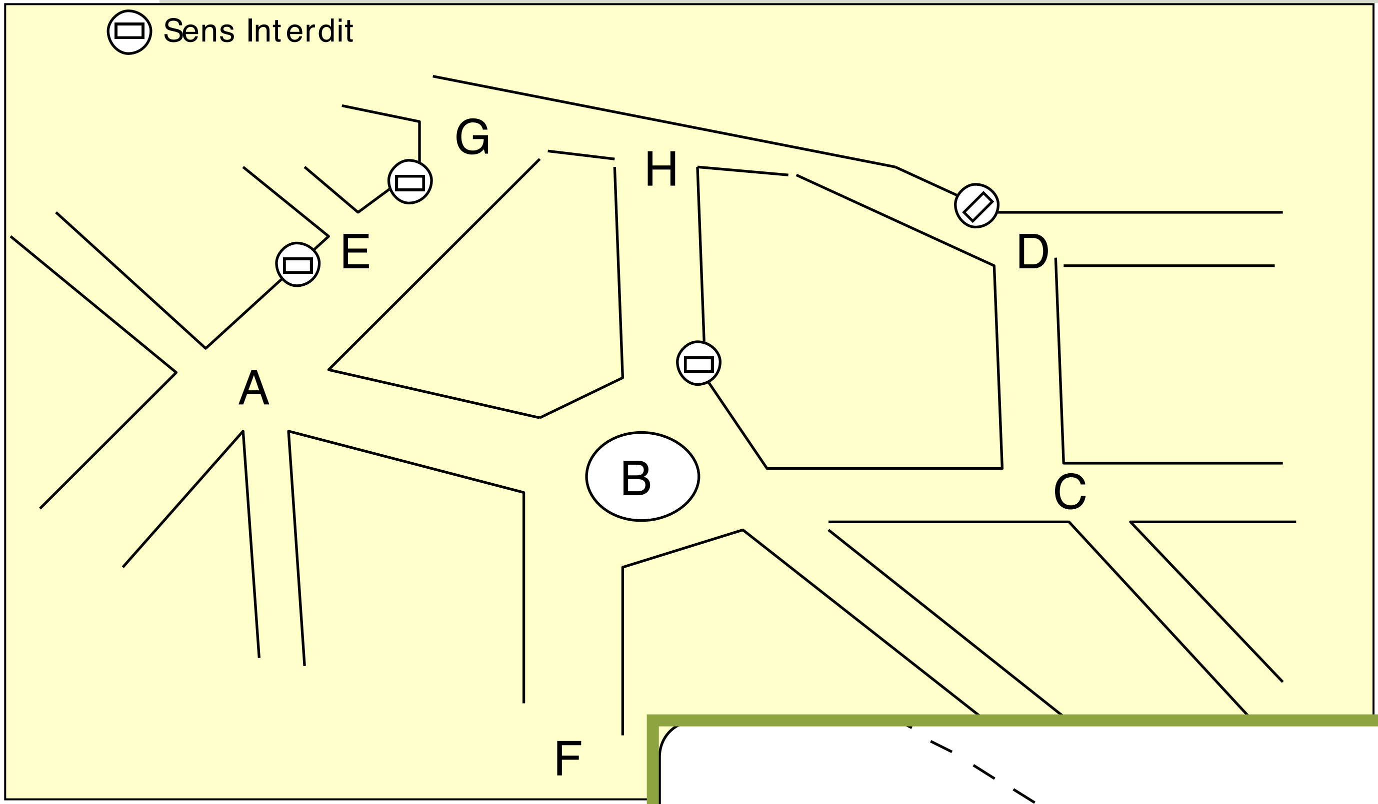
U1 =

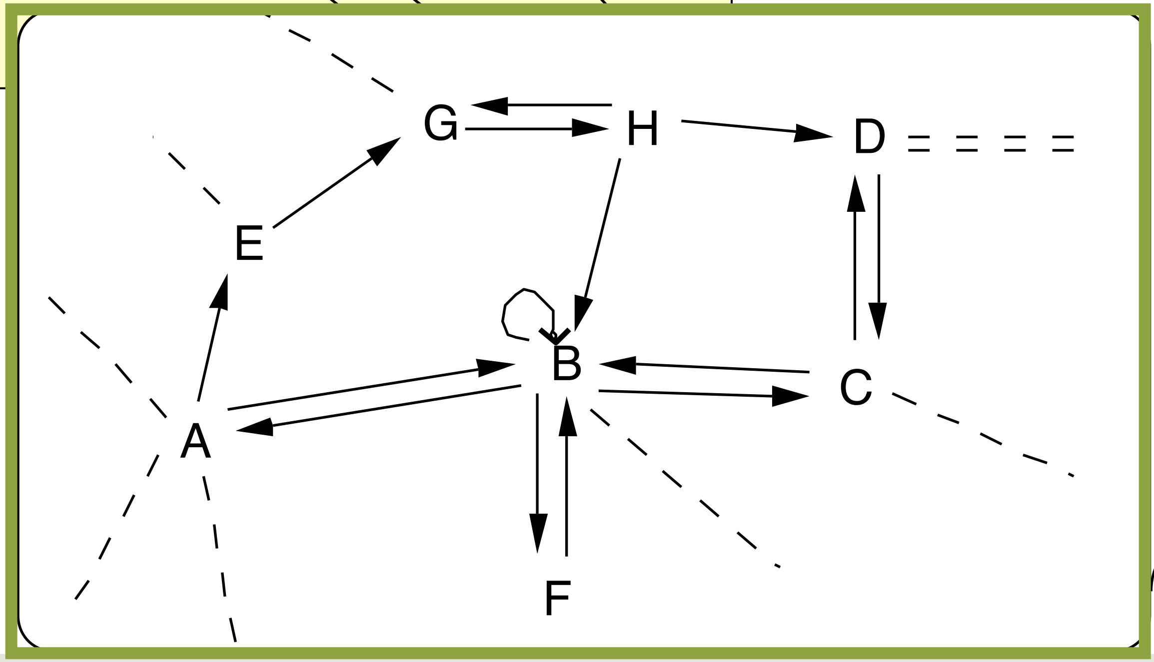
{(a,f),(a,b),(b,c),..}

Degré d’un sommet ?

Degré d’un graphe ?

Circuler





G

Quelques propriétés

- Chaîne ?
- Chemin ?
- Cycle, co-cycle ?
- Circuit, co-circuit ?
- Graphe connexe ?
- Arbre ?
- …

Quelques algorithmes pertinents

Comment déterminer une connexité minimale sur un graphe?

Quelques algorithmes pertinents

Comment déterminer une connexité minimale sur un graphe?

Algorithme Kruskal

(1) classer les arêtes par distance non décroissante

(2) Parcourir la liste des arêtes et rajouter une arête à l'arbre si celle-ci ne crée pas de cycle avec les arêtes déjà rajoutées

Complexité : $O(|V||E|)$

Quelques algorithmes dans la prise de décision

Algorithme de Dijkstra

Recherche de plus court chemin entre un sommet donné (sans perte de généralité on considère sommet 1) et les autres sommets du graphe.

Algorithme de Dijkstra

Hypothèse : tous les arcs ont des longueurs non négatives

(1) $\bar{S} := \{2,\ldots,N\}$; $\pi(1) := 0$; pour tout $x \neq 1$ faire $\pi(x) := \begin{cases} d_{1x} & \text{si } x \in N^+(1) \\ \infty & \text{sinon} \end{cases}$

(2) Déterminer x tel que $\pi(x) \leq \pi(y)$ pour tout y dans $\bar{S}$ et poser $\bar{S} := \bar{S} - \{x\}$

Si $\bar{S} = \emptyset$ alors STOP

(3) Pour tout y dans $\bar{S} \cap N^+(x)$ faire $\pi(y) := \min\{\pi(y), \pi(x) + d_{xy}\}$ et retourner à (2)

Algorithme de Moore

Il s'agit d'une extension naturelle de l'algorithme de Dijkstra au cas où des longueurs peuvent être négatives.

Hypothèse : les longueurs peuvent être négatives, mais il n'existe pas de circuit de longueur négative

(1) $\bar{S}$:={2,…,N}; $\pi(1)$:=0; pour tout $x\neq 1$ faire $\pi(x) := \begin{cases} d_{1x} & \text{si } x \in N^{+}(1) \\ \infty & \text{sinon} \end{cases}$

(2) Déterminer x tel que $\pi(x)\leq\pi(y)$ pour tout y dans $\bar{S}$ et poser $\bar{S} := \bar{S} - \{x\}$;

(3) Pour tout y dans $N^{+}(x)$ faire

$\pi^{*} := \min\{\pi(y), \pi(x)+d_{xy}\}$

si $\pi^{*} < \pi(y)$ alors poser $\pi(y) := \pi^{*}$ et rajouter y dans $\bar{S}$ s'il ne s'y trouve pas déjà.

Si $\bar{S} = \emptyset$ alors STOP sinon aller à (2)

Algorithme de Bellman

Algorithme de Bellman

Hypothèse : aucune (c'est-à-dire que les longueurs peuvent être négatives, et il peut y avoir des circuits de longueur négative)

(1) $\pi^0(1):=0$; $\pi^0(x):=\infty$ pour tout $x\neq 1$; $k:=1$;

(2) $\pi^k(1):=0$; $\pi^k(x):=\min\{\pi^{k-1}(x), \min\limits_{y\in N^-(x)} \{ \pi^{k-1}(y)+d_{yx}\}\}$

(3) Si $\pi^k(x)=\pi^{k-1}(x)$ pour tout x alors STOP

Si $k\leq N-1$ poser $k:=k+1$ et aller à (2)

Si $k=N$ STOP : il existe un circuit de longueur négative

Complexité : $O(MN)$

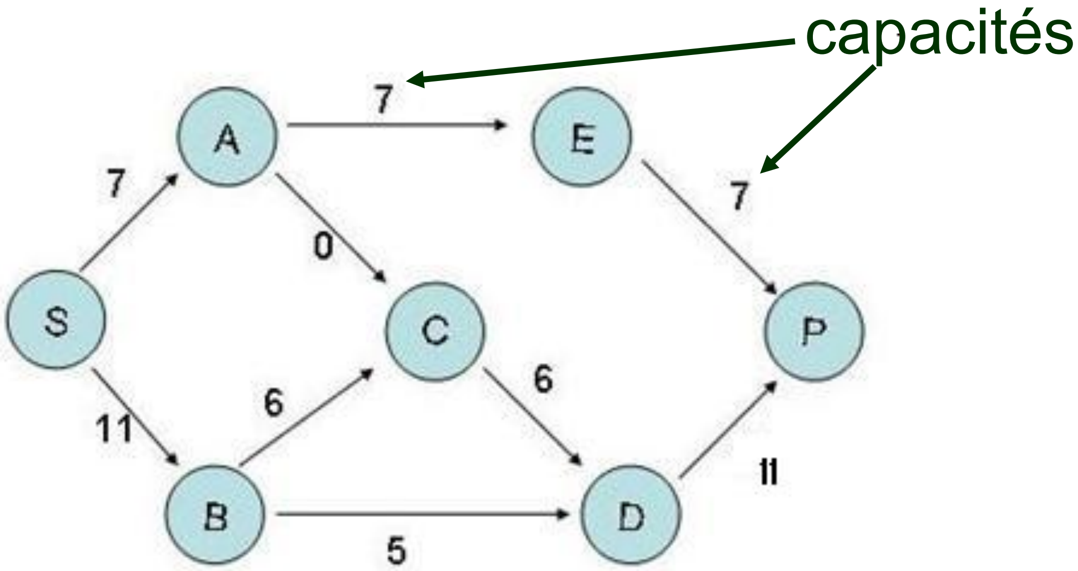
Algorithme de flot maximum

Comment déterminer la capacité maximale qu'on peut imaginer sur un réseau entre une source et une destination à condition:

Tout le flot qui sort de la source doit arriver à la destination (aucune perte, aucune production à l'intérieur du réseau)

Algorithme de flot Maximum

Comment déterminer la capacité maximale qu'on peut imaginer sur un réseau ?



Quelques concepts

- Soit un réseau N = (V, A). V l'ensemble des sommets et A l'ensemble des arcs où chaque arc (x,y) a une capacité c(x,y).
- Un flot représente l'acheminement d'un flux de matière depuis une source s vers une destination t
- Il n'est pas possible de stocker ou de produire de la matière première aux nœuds intermédiaires: un flot vérifie localement une loi de conservation de flux

Le problème du flot maximum consiste à trouver un flot F_{max} de valeur maximale sur le réseau N

Quelques concepts (suite)

- Un arc (x,y) est saturé par un flot F si la valeur du flot sur l'arc égale sa capacité F(x,y) = c(x,y). Un chemin est saturé si l'un de ses arcs est saturé.

- La capacité résiduelle d'un arc (x,y) est la quantité c(x,y)-F(x,y) de flot pouvant encore transiter par lui. La capacité résiduelle d'un chemin est la plus petite capacité résiduelle de ses arcs.

- Saturer un chemin p de s à t consiste à augmenter le flot de ses arcs de la capacité résiduelle du chemin

Algorithme de flot maximum

Algorithme Ford-Fulkerson

ALGORITHME: Saturation

ENTREES : Réseau N= (V,A), s, t des sommets de V

SORTIE: F un flot entre s et t

Initialiser F := 0 //On part d'un flot possible entre s et t

Tant que il existe un chemin p de s à t non saturé par le flot F

Augmenter le flot F en saturant p

Fin TantQue

Fin