

Optimisation et Recherche Opérationnelle

M1 Informatique

Julien Ah-Pine (julien.ah-pine@univ-lyon2.fr)

Université Lyon 2 - ICOM

M1 Informatique 2021/2022

Théorie des graphes et RO

⇒ Pourquoi les graphes et leurs applications en RO ?

- Les graphes permettent de **représenter de nombreux problèmes** de manière intuitive et contribuent donc souvent à **formaliser et résoudre ces derniers** : “Un bon dessin vaut mieux qu’un long discours”.
- Les graphes sont rencontrés dans de **nombreux domaines** et pour de nombreux problèmes d’optimisation : cartographie (plus courts chemins dans des réseaux routiers, réseaux de télécommunication, ...), économie-gestion (planning de livraison, ordonnancement, ...), aide à la décision (aide multicritère à la décision, ...), ...
- Les objectifs du cours sont :
 - ▶ Présenter certains problèmes de la théorie des graphes trouvant des applications en RO.
 - ▶ Etudier et mettre en œuvre des algorithmes résolvant exactement ou de façon approchée ces problèmes.

Recherche Opérationnelle

⇒ Qu’est-ce que la **Recherche Opérationnelle (RO)** ?

- Voici la définition de la ROADEF (Société Française de Recherche Opérationnelle et d’Aide à la Décision) donnée dans son livre blanc en 2019 (https://www.roadef.org/pdf/LIVRE_BLANC_A5_juin.pdf) :
 - ▶ La Recherche Opérationnelle (RO) peut se définir comme la mise en œuvre de méthodes scientifiques, essentiellement mathématiques et algorithmique, en vue de **prendre la meilleure décision possible**. Etablir plus précisément quand entre en jeu la RO dans un processus de prise de décision est une tâche plus difficile. En effet, l’expert en RO exploite des connaissances provenant de sources particulièrement variées qu’on peut classer en trois grandes catégories : les mathématiques appliquées, l’informatique et les modèles métiers.
 - ▶ Exemples d’application : organisation des lignes de production de véhicules, optimisation des portefeuilles bancaires, aide au séquençage de l’ADN, organisation des produits recyclables ...
- Dans ce cours nous aborderons l’utilisation des **graphes en RO** et les problèmes permettant d’être résolus dans ce cadre.

Contenu du cours

- 1 Éléments de la théorie des graphes
- 2 Quelques problèmes importants de l’étude d’un graphe
- 3 Problème du plus court chemin
- 4 Problèmes de Flot

Quelques références faisant partie des sources du cours

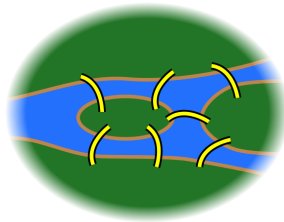
- F. Drosbeke, M. Hallin et Cl. Lefèvre, *Les graphes par l'exemple*, Ellipses, 1987
- M. Gondran et M. Minoux, *Graphes et Algorithmes (4ème édition)*, Lavoisier, 2009
- R. Faure, B. Lemaire et Ch. Picouleau, *Précis de recherche opérationnelle (5ème édition)*, Dunod, 2000
- T. Cormen, Ch. Leiserson, R. Rivest, C. Stein, *Introduction à l'algorithmique (2ème édition)*, Dunod, 2004

Déroulement du cours

- 4 séances CM de 1h45
- 8 séances TD/TP de 1h45 (avec implémentation sur machine à l'aide du langage R)
- Evaluation :
 - ▶ 1 dossier R par groupe de 2
 - ▶ 1 examen individuel sur table
- Les supports de cours peuvent être récupérés à partir de la page moodle du cours CM.

Exemple introductif

- A priori le premier problème résolu par la théorie des graphes (en 1736).



- La ville de Königsberg (aujourd'hui Kaliningrad) est traversée par le Pregel, qui coule de part et d'autre de l'île de Kneiphof et possède 7 ponts, comme le montre la figure ci-dessus. Un piéton désire se promener et visiter tous les quartiers en empruntant une et une seule fois chaque pont.
- Est-ce que le parcours désiré du piéton est réalisable ?

Rappel du Sommaire

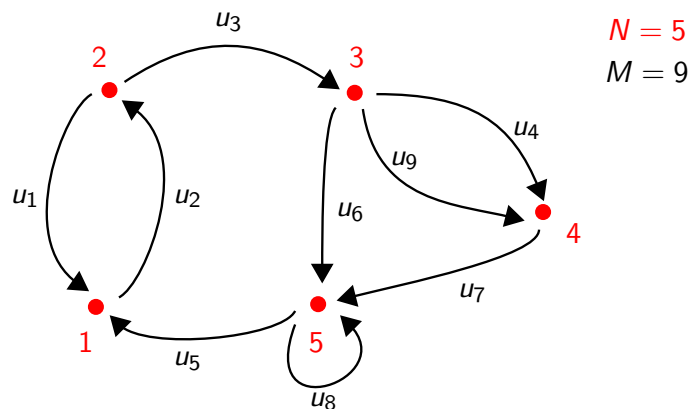
- 1 Éléments de la théorie des graphes
- 2 Quelques problèmes importants de l'étude d'un graphe
- 3 Problème du plus court chemin
- 4 Problèmes de Flot

Rappel du Sommaire

1 Éléments de la théorie des graphes

- Définitions et généralités sur les graphes
- Matrices associées à un graphe
- Connexité

Exemple de graphe orienté



Graphes : concepts orientés

Définition. (Graphe orienté)

Un graphe $G = [X, U]$ est déterminé par la donnée :

- 1 d'un ensemble X dont les éléments sont appelés des **sommets** (ou des **noeuds**). Si $N = |X|$ est le nombre de sommets, on dit que le graphe G est d'ordre N .
- 2 d'un ensemble U dont les éléments $u \in U$ sont des couples **ordonnés** de sommets appelés des **arcs**. Si $u = (i, j)$ est un arc de G , i est l'**extrémité initiale** et j l'**extrémité terminale** de u . On notera $M = |U|$ le nombre d'arcs.

Boucle et p -graphe

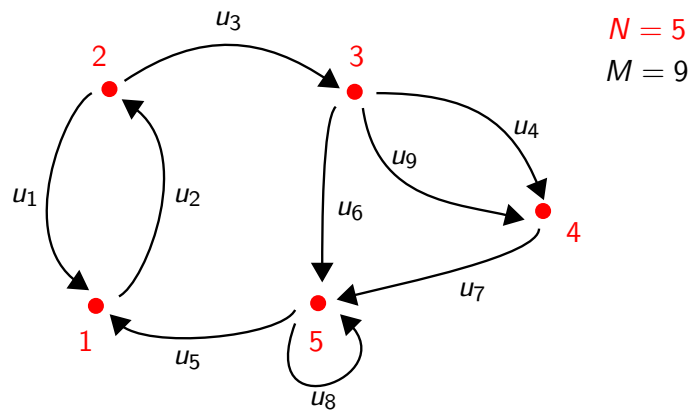
Définition. (Boucle)

Un arc $u = (i, i)$ dont les extrémités coïncident est appelé une **boucle**.

Définition. (p -graphe)

Un **p -graphe** est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (i, j) entre deux sommets quelconques i et j , pris dans cet ordre.

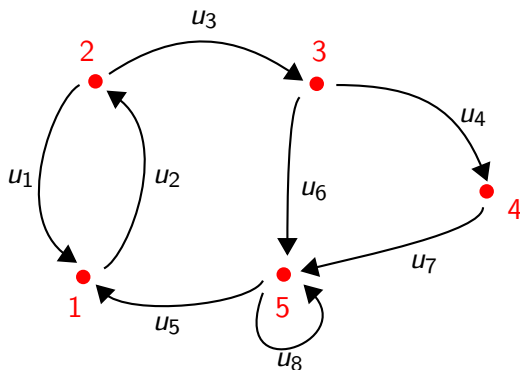
Exemple de boucle et d'un 2-graphe



$N = 5$
 $M = 9$

1-graphe et application multivoque

- Si G est un **1-graphe** alors il est parfaitement déterminé par la donnée de l'ensemble X et de l'application multivoque Γ . On peut donc aussi noter $G = [X, \Gamma]$.



$$\Gamma_1 = \{2\} \quad ; \quad \Gamma_2 = \{1, 3\} \quad ; \quad \Gamma_3 = \{4, 5\} \dots$$

Graphes et applications multivoques

Définition. (Ensemble de successeurs d'un sommet)

j est un **successeur** de i s'il existe un arc de la forme (i, j) . L'**ensemble des successeurs** d'un sommet $i \in X$ est noté $\Gamma(i)$.

Définition. (Application multivoque)

L'application Γ qui, à tout élément de X , fait correspondre une partie de X ($\Gamma : X \rightarrow \mathcal{P}(X)$), est appelée une **application multivoque**.

Définition. (Ensemble de prédécesseurs d'un sommet)

j est un **prédécesseur** de i s'il existe un arc de la forme (j, i) . L'**ensemble des prédécesseurs** d'un sommet $i \in X$ est noté $\Gamma^{-1}(i)$. (Γ^{-1} est alors l'application multivoque réciproque de Γ).

Graphes : concepts **non orientés**

Définition. (Graphe **non orienté**)

Un graphe $G = [X, U]$ dont les éléments $u \in U$ **ne sont pas ordonnés** est dit **non orienté**. Les éléments u sont alors appelés **arêtes**.

- Il s'agit d'un graphe dont on ne s'intéresse pas à l'orientation des arcs. Dans ce cas, $(i, j) \in U$ est équivalent à $(j, i) \in U$.
- Dans la suite du cours :
 - ▶ L'utilisation du terme **arc** sous-entend que le graphe est **orienté**
 - ▶ L'utilisation du terme **arête** sous-entend que le graphe est **non orienté**

Multigraphe et Graphe simple

Définition. (Multigraphe)

Un multigraphe est un graphe pour lequel il peut exister plusieurs arêtes entre deux sommets i et j donnés.

Définition. (Graphe simple)

Un graphe sans boucle et ayant au plus une arête pour toute paire de sommets est appelé graphe simple.

Définitions principales

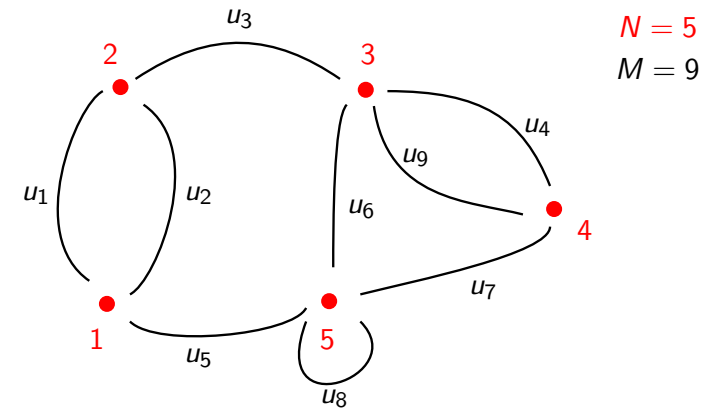
Définition. (Arcs adjacents, arêtes adjacentes)

Deux arcs (deux arêtes) sont dit **adjacents** s'ils ont au moins une extrémité commune.

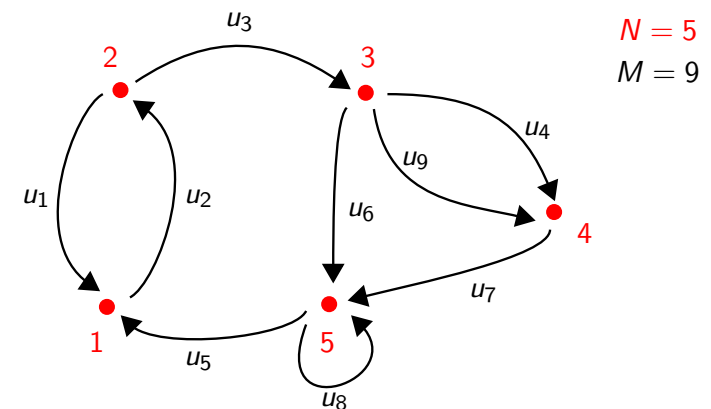
Définition. (Degré et demi-degré)

- le **demi-degré extérieur** du sommet i , noté $d_G^+(i)$ est le nombre d'arcs ayant i comme extrémité **initiale**
- le **demi-degré intérieur** du sommet i , noté $d_G^-(i)$ est le nombre d'arcs ayant i comme extrémité **terminale**
- le **degré** du sommet i , noté $d_G(i)$ est le nombre d'arcs (arêtes) ayant i comme extrémité et on a (pour les arcs) : $d_G(i) = d_G^+(i) + d_G^-(i)$

Exemple de multigraphe



Exemple



Sous-graphes, graphes partiels et graphes complémentaires

Définition. (Sous-graphes, graphes partiels et graphes complémentaires)

- Le **sous-graphe induit** par $A \subset X$ est le graphe G_A dont les sommets sont les éléments de A et dont les arcs sont les arcs de G ayant les deux extrémités dans A
- Soit $G = [X, U]$ et soit $V \subset U$. Le **graphe partiel engendré** par V est le graphe $[X, V]$
- Soit $G = [X, U]$ et soient $A \subset X, V \subset U$. Le **sous-graphe partiel engendré** par A et V est le graphe partiel de G_A engendré par V
- Etant donné un 1-graphe $G = [X, U]$, le **graphe complémentaire** de G , est le graphe $[X, \bar{U}]$ tel que : $(i, j) \in U \Rightarrow (i, j) \notin \bar{U}$ et $(i, j) \notin U \Rightarrow (i, j) \in \bar{U}$

Matrice d'incidence sommets-arcs

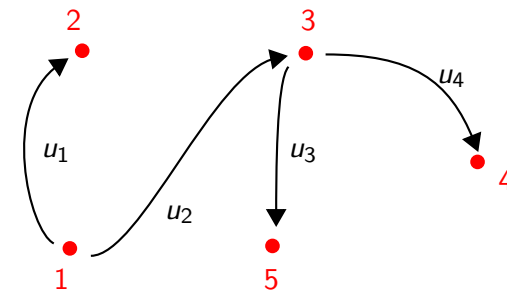
Définition. (Matrice d'incidence sommets-arcs)

Soit $G = [X, U]$ un graphe sans boucle. La matrice d'incidence de G est une matrice $A = (a_{iu})$, $i = 1, \dots, N$, $u = 1, \dots, M$, à coefficients entiers dans $\{0, 1, -1\}$ tel que chaque colonne correspond à un arc de G et chaque ligne à un sommet. Si $u = (i, j) \in U$ alors la colonne u contient des termes nuls sauf pour les suivants : $a_{iu} = 1$ et $a_{ju} = -1$.

Rappel du Sommaire

- 1 Eléments de la théorie des graphes
 - Définitions et généralités sur les graphes
 - Matrices associées à un graphe
 - Connexité

Exemple de matrice d'incidence sommets-arcs



$$A = \begin{matrix} & u_1 & u_2 & u_3 & u_4 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \end{matrix}$$

Matrice d'incidence sommets-arêtes

Définition. (Matrice d'incidence sommets-arêtes)

Soit $G = [X, U]$ un graphe non orienté sans boucle. La matrice d'incidence de G est une matrice $A = (a_{iu})$, $i = 1, \dots, N$, $u = 1, \dots, M$, à coefficients entiers dans $\{0, 1\}$ telle que chaque colonne correspond à une arête de G et chaque ligne à un sommet. Si $u = (i, j) \in U$ alors la colonne u contient des termes nuls sauf pour les suivants : $a_{iu} = 1$ et $a_{ju} = 1$.

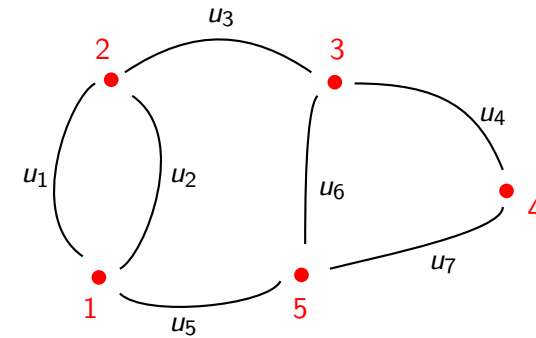
Matrice d'adjacence sommets-sommets

Définition. (Matrice d'adjacence sommets-sommets)

Soit $G = [X, U]$ un 1-graphe comportant éventuellement des boucles. La matrice d'adjacence de G est une matrice carrée $A = (a_{ij})$, $i = 1, \dots, N$, $j = 1, \dots, N$, à coefficients entiers dans $\{0, 1\}$ tel que chaque colonne correspond à un sommet de G et chaque ligne à un sommet de G et de terme général : $a_{ij} = 1 \Leftrightarrow (i, j) \in U$ ($a_{ij} = 0$ sinon).

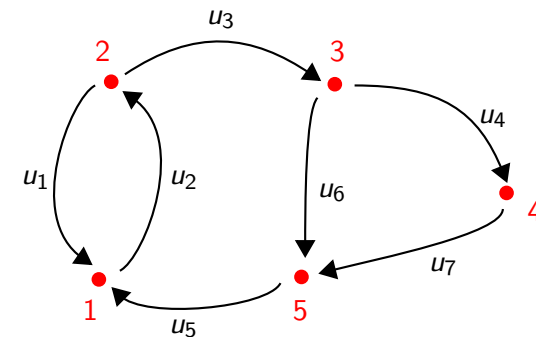
- Dans le cas non orienté, la matrice d'adjacence est symétrique.

Exemple de matrice d'incidence sommets-arêtes



$$A = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

Exemple de matrice d'adjacence



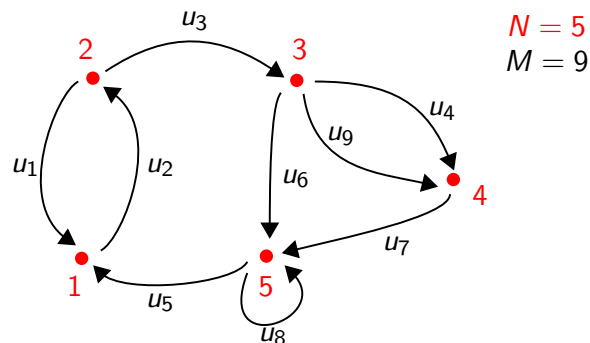
$$A = \begin{matrix} & 1 & 2 & 3 & 4 & 5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Rappel du Sommaire

1 Éléments de la théorie des graphes

- Définitions et généralités sur les graphes
- Matrices associées à un graphe
- Connexité

Exemple de matrice



- La chaîne $L = \{u_2, u_5, u_6\}$ de longueur 3, joint le sommet 2 et 3.
- La chaîne $L = \{u_1, u_5, u_6, u_3\}$ de longueur 4 est un cycle élémentaire.

Chaîne, chaîne élémentaire, cycle, cycle élémentaire

Définition. (Chaîne de longueur q et chaîne élémentaire)

-Une chaîne de longueur q est une séquence de q arcs :

$L = \{u_1, u_2, \dots, u_q\}$ telle que chaque arc u_r de la séquence ($2 \leq r \leq q-1$) ait une extrémité commune avec l'arc u_{r-1} ($u_{r-1} \neq u_r$) et l'autre extrémité commune avec l'arc u_{r+1} ($u_{r+1} \neq u_r$).

-L'extrémité i de u_1 non adjacente à u_2 et l'extrémité j de u_q non adjacente à u_{q-1} sont appelées les **extrémités de la chaîne** L .

-On appelle **chaîne élémentaire** une chaîne telle qu'en la parcourant, on ne rencontre pas deux fois le même sommet.

Définition. (Cycle et cycle élémentaire)

Un cycle est une chaîne dont les extrémités coïncident (chaîne fermée).

Un **cycle élémentaire** est un cycle minimal (pour l'inclusion) c'est-à-dire ne contenant strictement aucun autre cycle. Dans un cycle élémentaire, on ne rencontre pas deux fois le même sommet (sauf l'origine).

Chemin, chemin élémentaire, circuit, circuit élémentaire

Définition. (Chemin de longueur q et chemin élémentaire)

-Un chemin de longueur q est une séquence de q arcs :

$P = \{u_1, u_2, \dots, u_q\}$ avec

$u_1 = (i_0, i_1); u_2 = (i_1, i_2); u_3 = (i_2, i_3); \dots; u_q = (i_{q-1}, i_q)$. (chaîne dont tous les arcs sont **orientés dans le même sens**.)

-Les extrémités i_0 et i_q sont respectivement appelés extrémités **initiale** et **terminale** du chemin P .

-On appelle **chemin élémentaire** un chemin tel qu'en le parcourant, on ne rencontre pas deux fois le même sommet.

Définition. (Circuit et circuit élémentaire)

Un circuit est un chemin dont les extrémités coïncident (chemin fermée).

Un **circuit élémentaire** est un circuit minimal (pour l'inclusion) c'est-à-dire ne contenant strictement aucun autre circuit. Dans un circuit élémentaire, on ne rencontre pas deux fois le même sommet (sauf l'origine).

Parcours, parcours eulérien et hamiltonien

- Un parcours d'un graphe G est une chaîne, un cycle, un chemin ou un circuit.

Définition. (Parcours eulérien)

Un parcours d'un graphe G est dit eulérien s'il passe une et une seule fois par chaque arc ou arête de G (il peut passer plusieurs fois par un même sommet).

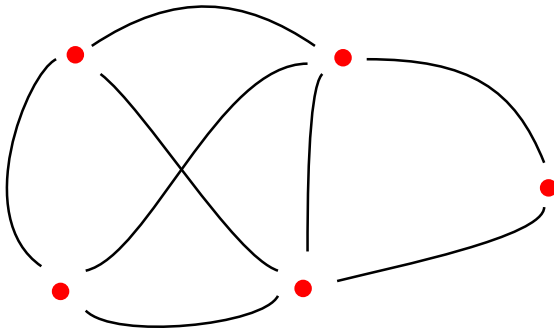
Définition. (Parcours hamiltonien)

Un parcours d'un graphe G est dit hamiltonien s'il passe une et une seule fois par chaque sommet de G (et donc au plus une fois par chaque arc ou arête).

Théorème d'Euler

Théorème. (Théorème d'Euler)

Un multigraphe $G = [X, U]$ connexe admet un parcours eulérien ssi le nombre de sommets de degré impair est 0 ou 2. S'il y en a 0, alors il s'agit d'un cycle d'origine quelconque. S'il y en a 2 alors le parcours est une chaîne reliant ces deux noeuds.



Connexité

Définition. (Graphe connexe)

Un graphe est dit connexe, si pour tout couple de sommets i et j , il existe une chaîne joignant i et j .

Définition. (Graphe orienté fortement connexe)

Un graphe orienté est dit **fortement connexe**, si pour tout couple de sommets ordonnés (i, j) , il existe un **chemin** joignant i et j .

Rappel du Sommaire

- 1 Éléments de la théorie des graphes
- 2 Quelques problèmes importants de l'étude d'un graphe
- 3 Problème du plus court chemin
- 4 Problèmes de Flot

Rappel du Sommaire

2 Quelques problèmes importants de l'étude d'un graphe

- Graphes sans circuit
- Noyau d'un graphe
- Stabilité et absorption d'un graphe non orienté
- Coloration des sommets
- Arbre partiel de poids minimum

Niveaux d'un graphe sans circuit

- L'algorithme permettant de déterminer si un graphe est sans circuit repose sur la définition suivante et la propriété qui suit.

Définition. (Niveaux d'un graphe)

Soit G un graphe à N sommets, considérons les $N + 1$ sous-ensembles de sommets, $X(k)$, $k = 0, \dots, N$, définis de la manière suivante :

- $X(0) = \{i | i \in X, \Gamma(i) = \emptyset\}$
- $X(1) = \{i | i \in X \setminus X(0), \Gamma(i) \subset X(0)\}$
- $X(2) = \{i | i \in X \setminus (X(0) \cup X(1)), \Gamma(i) \subset (X(0) \cup X(1))\}$
- ...
- $X(N) = \{i | i \in X \setminus (X(0) \cup X(1) \cup \dots \cup X(N-1)), \Gamma(i) \subset (X(0) \cup X(1) \cup \dots \cup X(N-1))\}$

Ces $N + 1$ sous-ensembles sont disjoints (par construction) et représentent les niveaux du graphe.

Pourquoi déterminer si un graphe est sans circuit ?

- L'**absence de circuits** est une condition nécessaire pour beaucoup d'algorithmes en théorie des graphes. Il est donc important de disposer de méthodes permettant de tester l'absence de circuit dans un graphe donné.
- Nous allons étudier :
 - ▶ Un algorithme permettant d'obtenir un circuit si le graphe en possède au moins un.
 - ▶ Un algorithme permettant d'obtenir les niveaux d'un graphe sans circuit.

Propriétés des niveaux d'un graphe sans circuit

Propriété.

Si G est **sans circuit** alors les niveaux constituent un recouvrement de X , c'est à dire que $X = X(0) \cup X(1) \cup \dots \cup X(N)$. De plus, nous avons la propriété que $X(k)$ est l'ensemble des sommets i de X tel que le chemin de longueur maximale issu de i contient k arcs.

Exemples

• Exemple 1 : $A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$
 $X(0) = \emptyset; X(1) = X(2) = X(3) = X(4) = X(5) = \emptyset.$

• Exemple 2 : $A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$
 $X(0) = \{5\}; X(1) = \{3, 4\}; X(2) = \{2\}; X(3) = \{1\};$
 $X(4) = X(5) = \emptyset.$

Exemple

• Reprenons l'exemple 2 précédent : $A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix} \end{matrix}$

• Nous voyons que nous pouvons successivement enlever les lignes et colonnes :

- ① 5
- ② 4
- ③ 3
- ④ 2

Nous obtenons finalement $\tilde{A} = (0)$ ce qui indique l'absence de circuit.

Algorithme permettant de tester l'absence de circuit

• Soit A la matrice d'adjacence du graphe $G = [X, U]$.

Input : A

- 1 $\tilde{A} \leftarrow A$
- 2 **Tant que** il existe une ligne i de \tilde{A} ne comportant que des 0 **faire**
- 3 Enlever de \tilde{A} la ligne i et la colonne i
- 4 **Fin Tant que**
- 5 **Si** $\tilde{A} = 0$ **faire**
- 6 **Output :** Absence de circuit
- 7 **Sinon faire**
- 8 **Output :** Existence d'un circuit
- 9 **Fin Si**

Obtention d'un circuit

• Si à l'issue de l'algorithme précédent, $\tilde{A} \neq 0$ alors il existe un circuit et on peut en construire un à l'aide de l'algorithme suivant.

• Soit \tilde{A} la matrice d'adjacence en sortie de l'algorithme précédent.

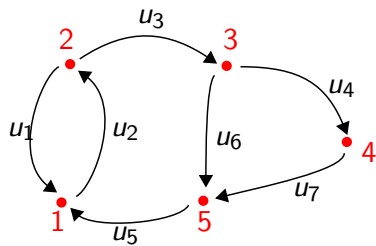
Input : \tilde{A}

- 1 $v \leftarrow i$ (sommet de \tilde{A} pris au hasard)
- 2 $S \leftarrow \{v\}$
- 3 **Tant que** il n'existe pas $j \in S$ tel que $\tilde{a}_{vj} = 1$ **faire**
- 4 Choisir $k \notin S$ tel que $\tilde{a}_{vk} = 1$
- 5 $S \leftarrow \text{append}(S, \{k\})$
- 6 $v \leftarrow k$
- 7 **Fin Tant que**
- 8 **Output :** $\{j, k, \dots, j\}$

• Dans cet algorithme, l'opération $\text{append}(S, \{k\})$ ajoute à la fin de la séquence de sommets S , le sommet k .

Exemple

- Reprenons l'exemple 1 précédent : $A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & \color{red}1 & 0 & 0 & 0 \\ \color{red}1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \color{red}1 \\ 0 & 0 & 0 & 0 & 1 \\ \color{red}1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$
- Dans ce cas $\tilde{A} = A$ car aucune ligne ne comporte que des 0 et il existe donc au moins un circuit.



Prenons au hasard le sommet 3 :

- 1 $S = \{3\}$
- 2 $S = \{3, 5\}$
- 3 $S = \{3, 5, 1\}$
- 4 $S = \{3, 5, 1, 2\}$
- 5 $\tilde{a}_{21} = 1$ avec $1 \in S$ et donc $\{1, 2, 1\}$ est un circuit

Algorithme d'obtention des niveaux d'un graphe sans circuit

- Input :** A
- 1 $k \leftarrow 0$
 - 2 Calculer $X(k)$, l'ensemble des sommets correspondant aux lignes de A non marquées ne contenant que des 1 barrés ou des 0
 - 3 **Tant que** $X(k) \neq \emptyset$ **faire**
 - 4 Marquer les lignes de A correspondant aux éléments de $X(k)$ et barrer les 1 des colonnes correspondantes
 - 5 $k \leftarrow k + 1$
 - 6 Calculer $X(k)$, l'ensemble des sommets correspondants aux lignes de A non marquées ne contenant que des 1 barrés ou des 0
 - 7 **Fin Tant que**
 - 8 **Output :** $X(0), \dots, X(k-1)$ et $X(k) = \dots = X(N) = \emptyset$

Niveaux d'un graphe sans circuit

- On suppose ici que le graphe est sans circuit.

Rappel :

- Dans ce cas, un sommet appartient à $X(k)$ avec $k = 0, \dots, N$, si le chemin de longueur maximale issu de ce sommet est de longueur k .
- De plus, les niveaux $X(k)$ forment une partition de X et tout sommet de $X(k)$ n'admet aucun successeur dans les niveaux $X(l)$ avec $l \geq k$.

Exemple

- Reprenons l'exemple 2 précédent : $A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & \cancel{1} & 0 & \cancel{1} & 0 \\ \cancel{2} & 0 & \cancel{1} & \cancel{1} & 0 \\ 0 & 0 & 0 & 0 & \cancel{1} \\ 0 & 0 & 0 & 0 & \cancel{1} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$

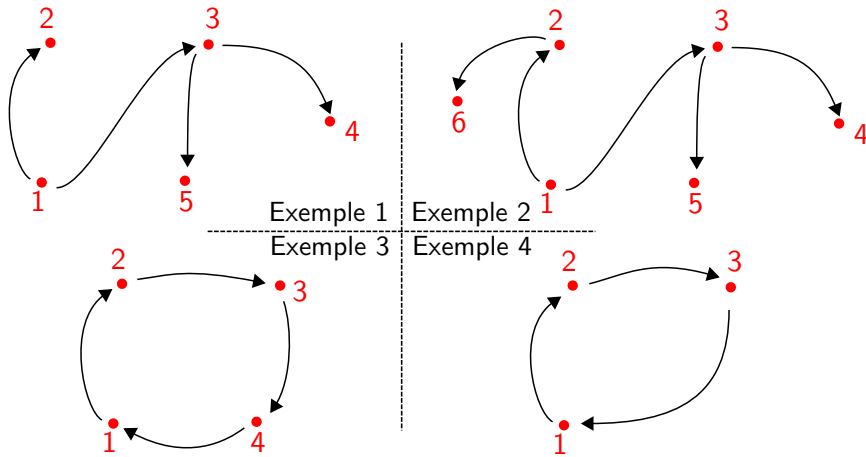
- Nous voyons que nous pouvons successivement déterminer les niveaux suivants :
 - 1 $X(0) = \{5\}$
 - 2 $X(1) = \{3, 4\}$
 - 3 $X(2) = \{2\}$
 - 4 $X(3) = \{1\}$
- $X(4) = \emptyset$ puisque tous les sommets ont été marqués. On arrête donc l'algorithme.

Rappel du Sommaire

2 Quelques problèmes importants de l'étude d'un graphe

- Graphes sans circuit
- Noyau d'un graphe
- Stabilité et absorption d'un graphe non orienté
- Coloration des sommets
- Arbre partiel de poids minimum

Exemples



- Exemple 1 : $Y = \{2, 4, 5\}$; Exemple 2 : $\{1, 4, 5, 6\}$; Exemple 3 : $Y = \{1, 3\}$ et $Y = \{2, 4\}$; Exemple 4 : $Y = \emptyset$.

Définition

Définition. (Noyau d'un graphe)

Soit $G = [X, U]$ un graphe sans boucle, de matrice d'adjacence A . Un sous-ensemble Y de sommets est un **noyau** s'il satisfait aux deux conditions suivantes :

- $\forall i, j \in Y : (i, j) \notin U \wedge (j, i) \notin U$ (un ensemble Y vérifiant cette propriété est dit **stable**)
- $\forall i \in X \setminus Y, \exists j \in Y$ tel que $(i, j) \in U$ (un ensemble Y vérifiant cette propriété est dit **absorbant**)

Théorème d'existence

- De façon générale, il peut exister plusieurs noyaux étant donné un graphe.

Théorème.

Un 1-graphe sans circuit possède un et un seul noyau.

Propriété.

Dans un graphe sans circuit, tous les sommets n'ayant aucun successeur (les feuilles) sont des éléments du noyau (mais pas forcément les seuls!).

Algorithme d'obtention d'un noyau d'un graphe sans circuit

- Input :** A (Matrice d'adjacence d'un graphe sans circuit)
- 1 $Y \leftarrow \emptyset$
 - 2 Marquer une ligne j ne comportant que des 0
 - 3 $Y \leftarrow Y \cup \{j\}$
 - 4 Dans la colonne j , barrer les lignes et colonnes i telles que $a_{ij} = 1$
 - 5 **Tant que** il existe une ligne non marquée et non barrée ne comportant que des 1 barrés ou des 0 **faire**
 - 6 Marquer une ligne j non marquée et non barrée ne comprenant que des 1 barrés ou des 0
 - 7 $Y \leftarrow Y \cup \{j\}$
 - 8 Dans la colonne j , barrer les lignes et colonnes i telles que $a_{ij} = 1$
 - 9 **Fin Tant que**
 - 10 **Output :** Y

Exemple pratique

- Le concept de noyau est important en théorie des graphes tant du point de vue théorique que du point de vue pratique. Il intervient notamment dans les problèmes de théorie des jeux et d'aide à la décision.
- Exemple concret : Un gouvernement a décidé de répartir 500 Millions d'Euros en vue de diminuer la mortalité sur les routes et propose dans ce cas cinq alternatives :
 - 1 Augmentation du nombre de gendarmes
 - 2 Construction d'autoroutes
 - 3 Mise en place de signaux lumineux à tous les carrefours
 - 4 Récompenses aux meilleurs conducteurs
 - 5 Gratuité des transports publics

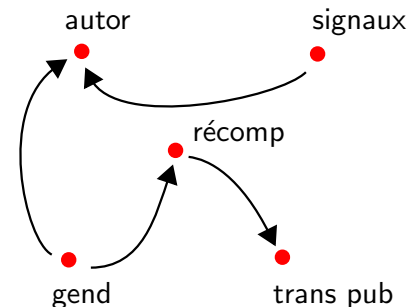
Exemple

$$\begin{matrix} & \bar{1} & 2 & \bar{3} & \bar{4} & 5 \\ \bar{1} & (0 & \bar{1} & 0 & \bar{1} & 0) \\ 2 & (0 & 0 & \bar{1} & \bar{1} & 0) \\ \bar{3} & (0 & 0 & 0 & 0 & \bar{1}) \\ \bar{4} & (0 & 0 & 0 & 0 & \bar{1}) \\ 5 & (0 & 0 & 0 & 0 & 0) \end{matrix}$$

- Reprenons l'exemple 2 précédent : $A =$
- Nous voyons que nous pouvons successivement déterminer les éléments du noyau :
 - 1 Itération 1 : $Y = \{5\}$
 - 2 Itération 2 : $Y = \{2, 5\}$
- Puisque toutes les lignes sont soit marquées (en rouge) soit barrées, on arrête l'algorithme.

Exemple pratique (suite)

- Il charge un groupe de recueillir les préférences du public et ce groupe parvient à établir le graphe suivant où les cinq sommets représentent les cinq alternatives et où un arc (i, j) est tracé s'il est certain que la population ne préfère pas i à j (autrement dit elle préfère j à i).



Détermination du noyau :

$$Y = \{\text{autor}, \text{trans pub}\}$$

Ceci permet de justifier : (i) la non-subvention des éléments hors du noyau, (ii) la subvention des éléments du noyau, (iii) l'incomparabilité entre les deux éléments du noyau (50/50).

Rappel du Sommaire

2 Quelques problèmes importants de l'étude d'un graphe

- Graphes sans circuit
- Noyau d'un graphe
- Stabilité et absorption d'un graphe non orienté
- Coloration des sommets
- Arbre partiel de poids minimum

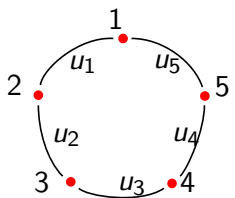
Stabilité et absorption

Définition. (Sous-ensemble stable et absorbant)

Soit $G = [X, U]$ un graphe non orienté.

- Un sous-ensemble $S \subset X$ est **stable** si deux sommets de S ne sont jamais adjacents : $\forall i, j \in S : (i, j) \notin U$
- Un sous-ensemble $T \subset X$ est **absorbant**^a si toute arête de U possède au moins une extrémité dans T : $\forall u = (i, j) \in U : i \in T \vee j \in T$

a. on dit également transversal pour les graphes non orientés



- Exemples de sous-ensembles stables :
 $S_1 = \{1, 3\}, S_2 = \{2, 5\}$.
- Exemples de sous-ensembles absorbants :
 $T_1 = \{2, 4, 5\}, T_2 = \{1, 3, 4\}$.

Rappel et définition

Définition. (Graphe non orienté)

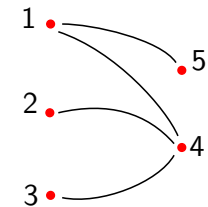
Un graphe $G = [X, U]$ dont les éléments $u \in U$ ne sont pas ordonnés est dit non orienté. Les éléments u sont alors appelés arêtes.

- Il s'agit d'un graphe dont on ne s'intéresse pas à l'orientation des arcs. Dans ce cas, $(i, j) \in U$ est équivalent à $(j, i) \in U$.

Définition. (Graphe biparti)

Un graphe $G = [X, U]$ est biparti s'il est possible de partitionner X en deux sous-ensembles S et \bar{S} tel que chaque arête ait exactement une extrémité dans S et une extrémité dans \bar{S}

- Exemple de graphe biparti :



Propriétés sur la stabilité et l'absorption

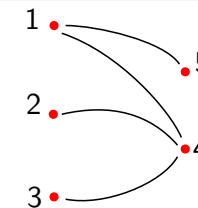
Propriété.

Le complément de tout sous-ensemble stable est absorbant et réciproquement.

- Dans l'exemple précédent on a par exemple : $T_1 = X \setminus S_1$ et $S_2 = X \setminus T_2$.

Propriété.

Dans un graphe biparti, les sous-ensembles X_1 et X_2 sont tout à la fois stables et absorbants.



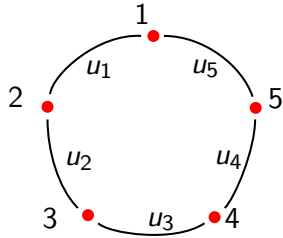
Nombres de stabilité et d'absorption

Définition.

Soit respectivement \mathbb{S} et \mathbb{T} , les familles des sous-ensembles stables et absorbants de X . On définit alors les nombres suivants :

- Nombre de stabilité : $\alpha(G) = \max_{S \in \mathbb{S}} |S|$
- Nombre d'absorption : $\tau(G) = \min_{T \in \mathbb{T}} |T|$

Remarque : $\alpha(G) + \tau(G) = |X|$.



- Dans l'exemple précédent : $\alpha(G) = 2$ et $\tau(G) = 3$.

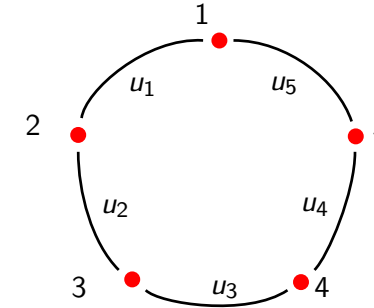
Rappel du Sommaire

2 Quelques problèmes importants de l'étude d'un graphe

- Graphes sans circuit
- Noyau d'un graphe
- Stabilité et absorption d'un graphe non orienté
- Coloration des sommets
- Arbre partiel de poids minimum

Exemple d'application

- Soit une forteresse composée de 5 tours $\{1, \dots, 5\}$ qui sont reliées par 5 murs $\{u_1, \dots, u_5\}$ comme l'indique le graphe suivant :



- Un gardien placé sur une tour peut surveiller les deux murs adjacents à cette tour. Pour des raisons d'économie, on cherche à mobiliser le minimum de gardiens pour garder la forteresse. Quel est donc le nombre minimum de gardiens permettant de garder tous les murs de la forteresse ?

Coloration des sommets d'un graphe

- **Colorer** les sommets d'un graphe $G = [X, U]$, consiste à leur associer des couleurs (ou tout autre identificateur : numéros, noms, ...) de telle sorte que deux sommets adjacents n'aient pas la même couleur. Si k -couleur sont utilisées on parle de **k -coloration**.
- On vérifie directement qu'une k -coloration des sommets de G est équivalent à une partition de X en k sous-ensembles stables, chacun d'entre eux contenant les sommets de même couleur.

Nombre chromatique

Définition. (Nombre chromatique)

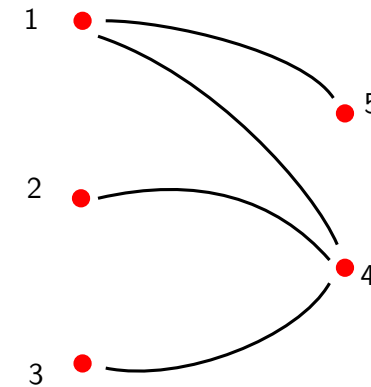
Le **nombre chromatique** d'un graphe G , noté $\gamma(G)$, est la plus petite valeur de k pour laquelle il existe une k -coloration. Une coloration des sommets en $\gamma(G)$ couleur est dite **minimale**.

- De nombreux problèmes concrets, tels que certains problèmes d'horaire et d'allocation de ressources, peuvent se ramener à la recherche d'une coloration minimale des sommets d'un graphe.

Détermination du nombre chromatique d'un graphe

- La détermination du nombre chromatique d'un graphe $G = [X, U]$ ainsi que l'obtention d'une coloration minimale des sommets de G constitue un problème complexe. Naïvement, on peut procéder de la manière suivante :
 - Enumérer tous les ensembles stables maximaux (au sens de l'inclusion).
 - Chercher un recouvrement des sommets formé d'un nombre minimum d'ensembles stables maximaux. Ce nombre est égal à $\gamma(G)$.
 - Déduire de ce recouvrement une partition de X en sous-ensembles stables et donc une coloration minimale de G .
 - Cependant, le nombre d'ensembles stables maximaux sera souvent trop important pour que la procédure soit effectivement applicable.
- ⇒ En pratique on a recours à des algorithmes de coloration **heuristiques** simples mais qui mènent à une coloration des sommets **non nécessairement minimale**.

Exemple d'un graphe biparti



- Nombre chromatique du graphe : $\gamma(G) = 2$.

Propriété.

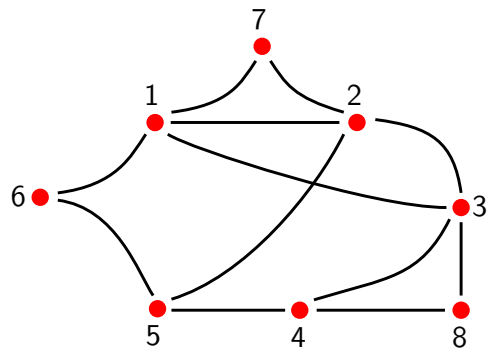
Un graphe G est biparti ssi il est **bichromatique**.

Algorithme de coloration de Welsh et Powell

Input : A (matrice d'adjacence de G)

- 1 Ranger les sommets par ordre de degrés non croissant
- 2 $k \leftarrow 0$
- 3 $B \leftarrow$ liste ordonnée des sommets
- 4 **Tant que** toutes les lignes de B ne sont pas colorées **faire**
- 5 $k \leftarrow k + 1$
- 6 **Tant que** $B \neq \emptyset$ **faire**
- 7 Colorer dans A par la couleur c_k la 1ère ligne non colorée dans B ainsi que la colonne correspondante
- 8 $B \leftarrow$ liste ordonnée des sommets non colorés ayant un zéro dans toutes les colonnes de A de couleur c_k
- 9 **Fin Tant que**
- 10 $B \leftarrow$ liste ordonnée des sommets non colorés
- 11 **Fin Tant que**
- 12 **Output** : k -coloration de G

Exemple



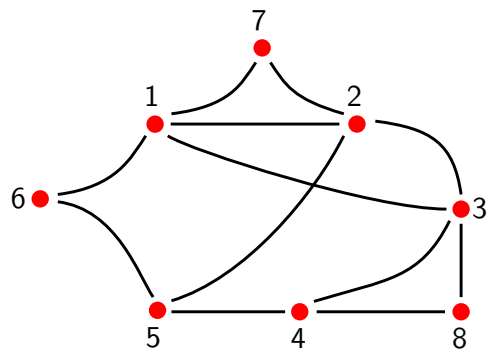
Matrice d'adjacence A :

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	1	0	0	0	1
4	0	0	1	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

Degré des sommets :

	1	2	3	4	5	6	7	8
$d_G(i)$	(4	4	4	3	3	2	2	2)

Exemple (1ère itération : $k = 1$)



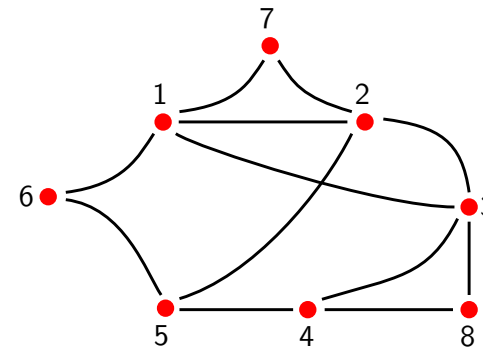
5 $k \leftarrow 1$

Matrice d'adjacence A :

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	1	0	0	0	1
4	0	0	1	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

8 $B \leftarrow \{4, 5, 8\}$
8 $B \leftarrow \emptyset$

Exemple (initialisation)



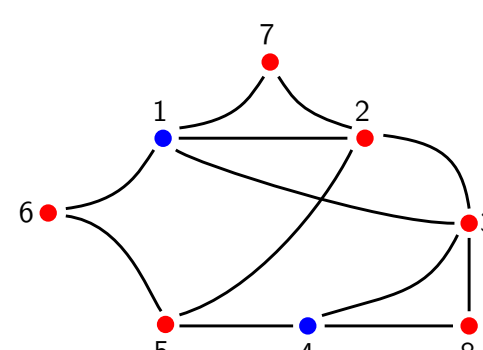
Input : A

- 1 Les sommets sont déjà ordonnés
- 2 $k \leftarrow 0$
- 3 $B \leftarrow \{1, 2, \dots, 8\}$

Matrice d'adjacence A :

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	1	0	0	0	1
4	0	0	1	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

Exemple (2ème itération : $k = 2$)



10 $B \leftarrow \{2, 3, 5, 6, 7, 8\}$
5 $k \leftarrow 2$

Matrice d'adjacence A :

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	1	0	0	0	1
4	0	0	1	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

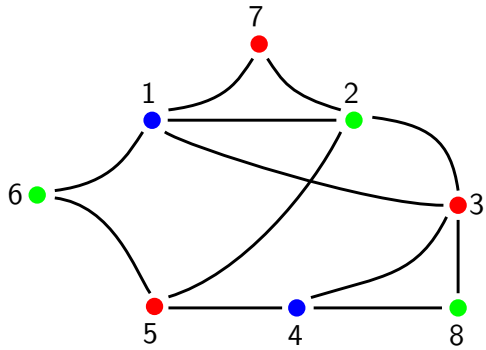
8 $B \leftarrow \{6, 8\}$
8 $B \leftarrow \{8\}$
8 $B \leftarrow \emptyset$

Exemple (3ème itération : $k = 3$)

10 $B \leftarrow \{3, 5, 7\}$
 5 $k \leftarrow 3$

Matrice d'adjacence A :

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	1	0
2	1	0	1	0	1	0	1	0
3	1	1	0	1	0	0	0	1
4	0	0	1	0	1	0	0	1
5	0	1	0	1	0	1	0	0
6	1	0	0	0	1	0	0	0
7	1	1	0	0	0	0	0	0
8	0	0	1	1	0	0	0	0

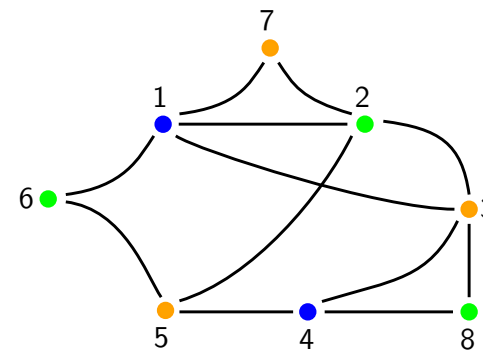


8 $B \leftarrow \{5, 7\}$
 8 $B \leftarrow \{7\}$
 8 $B \leftarrow \emptyset$

Borne pour le nombre chromatique

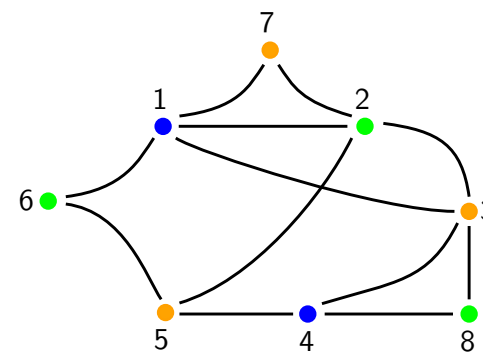
- **Attention** : l'algorithme de Welsh et Powell est une heuristique et donc la k -coloration obtenue n'est pas nécessairement minimale. Le nombre chromatique est en général inconnu.
- En revanche, il existe des bornes pour $\gamma(G)$:
 - ▶ Bornes inférieures :
 - ★ $\gamma(G) \geq \frac{N}{N - \min_i(d_G(i))}$
 - ★ $\gamma(G) \geq \frac{N}{\alpha(G)}$
 - ★ $\gamma(G) \geq \omega(G)$ où $\omega(G)$ est le cardinal de la plus grande clique de G (une clique est un sous-graphe complet càd tel qu'il existe une arête pour toute paire de sommets)
 - ▶ Bornes supérieures :
 - ★ $\gamma(G) \leq \max_i(d_G(i)) + 1$
 - ★ $\gamma(G) \leq N + 1 - \alpha(G)$
 - ★ $\gamma(G) \leq k$ où k est le nombre de couleurs obtenu à l'issue de l'algorithme de Welsh et Powell

Exemple (fin de l'algorithme)



- 4 Toutes les lignes sont colorées ($B = \emptyset$)
 12 On obtient une 3-coloration : $\{1, 4\}, \{2, 6, 8\}, \{3, 5, 7\}$

Exemple



Calcul des bornes :

- Bornes inférieures :
 - ▶ $\gamma(G) \geq \frac{N}{N - \min_i(d_G(i))} = \frac{8}{6}$
 - ▶ $\gamma(G) \geq \frac{N}{\alpha(G)} = \frac{8}{3}$
 - ▶ $\gamma(G) \geq \omega(G) = 3$
- Bornes supérieures :
 - ▶ $\gamma(G) \leq \max_i(d_G(i)) + 1 = 5$
 - ▶ $\gamma(G) \leq N + 1 - \alpha(G) = 6$
 - ▶ $\gamma(G) \leq k = 3$

Degré des sommets :

	1	2	3	4	5	6	7	8
$d_G(i)$	4	4	4	3	3	2	2	2

Donc pour l'exemple : $\gamma(G) = 3$.

Rappel du Sommaire

2 Quelques problèmes importants de l'étude d'un graphe

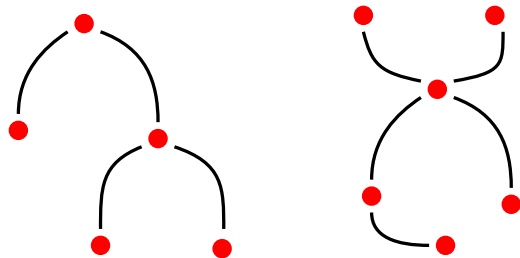
- Graphes sans circuit
- Noyau d'un graphe
- Stabilité et absorption d'un graphe non orienté
- Coloration des sommets
- Arbre partiel de poids minimum

Arbres

Définition. (Arbres et forêts)

Un arbre est un graphe non orienté simple, connexe et sans cycle. Une forêt est un graphe non orienté simple et sans cycle. Une forêt peut être vue comme un ensemble d'arbres.

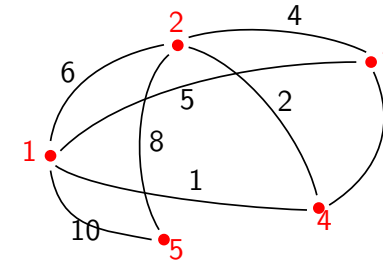
- Exemples d'application : réseaux de télécommunications, hiérarchies, taxonomies...
- Exemples d'arbres :



Préambule : Graphes valués

Définition. (Graphe valué)

Etant donné un graphe $G = [X, U]$, on associe à chaque arête u un nombre $p(u) \in \mathbb{R}$ appelé poids de l'arête. On dit que G est valué ou pondéré par les valeurs $p(u)$. Si $u = (i, j)$, on utilisera également la notation p_{ij} pour désigner le poids de l'arête u .



Poids d'un graphe partiel

Définition. (Poids d'un graphe partiel)

Soit $G = [X, U]$ un graphe valué simple et soit $U' \subset U$. Le poids du graphe partiel $G' = [X, U']$ est le nombre :

$$p(U') = \sum_{u \in U'} p(u)$$

Problème de l'arbre partiel de poids minimum

Définition. (Problème de l'arbre partiel de poids minimum)

Soit $G = [X, U]$ un graphe valué simple et connexe. Un graphe partiel connexe et sans cycles de G (donc un arbre admettant X pour ensemble de sommets) est appelé arbre partiel de G . Notons par \mathbb{A} l'ensemble de tous les arbres partiels de G . L'arbre partiel de poids minimum^a est l'arbre $G' = [X, U']$ qui minimise $p(U')$ sur \mathbb{A} :

$$\forall G'' = [X, U''] \in \mathbb{A} : p(U') \leq p(U'')$$

a. on parle aussi d'arbre couvrant de poids ou de longueur minimum

- La construction d'un arbre partiel de poids minimum dans un graphe a de nombreuses applications notamment dans certains problèmes de distribution et de télécommunication. Par ailleurs, en fouille de données, et en classification automatique en particulier, la résolution du problème aboutit à une méthode de classification hiérarchique.

Obtention d'un arbre partiel de poids minimal : algorithme de Kruskal

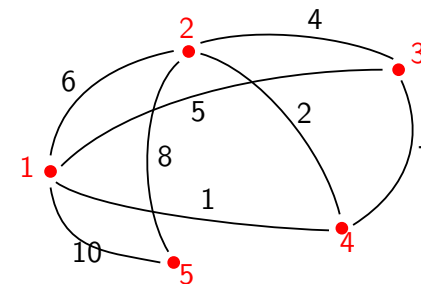
- Input :** $G = [X, U]$
- 1 Ranger les arêtes de U par ordre de poids croissants
 - 2 $U' \leftarrow \{u_1\}$; $k \leftarrow 1$
 - 3 **Tant que** $|U'| \neq N - 1$ **faire**
 - 4 $k \leftarrow k + 1$
 - 5 **Tant que** il existe un cycle dans $U' \cup \{u_k\}$ **faire**
 - 6 $k \leftarrow k + 1$
 - 7 **Fin Tant que**
 - 8 $U' \leftarrow U' \cup \{u_k\}$
 - 9 **Fin Tant que**
 - 10 **Output :** $G' = [X, U']$ est l'arbre partiel de poids minimum

- On ajoute successivement à U' les arêtes dont les poids sont les plus bas pourvu que ces ajouts n'entraînent pas l'apparition d'un cycle.

Obtention d'un arbre partiel de poids minimal : propriétés de l'algorithme de Kruskal

- Il existe plusieurs algorithmes d'obtention d'un arbre partiel de poids minimum dans un graphe valué simple et connexe $G = [X, U]$. Nous présentons ici l'algorithme de Kruskal. Il repose sur les propriétés suivantes :
 - ▶ La solution du problème est unique si les poids des arêtes sont tous distincts. Nous supposons cette condition satisfaite dans la suite.
 - ▶ La solution du problème est invariante pour les transformations monotones croissantes des poids : elle ne dépend donc que du rang des arêtes lorsque celles-ci sont classées par poids croissants.
 - ▶ Notons U_Y avec $Y \subset X$, l'ensemble des arêtes de $G = [X, U]$ ayant une extrémité dans Y et l'autre dans $X \setminus Y$. Si $G' = [X, U']$ est un arbre partiel de poids minimum alors pour tout $Y \subset X$, l'arête de valeur minimum de U_Y appartient à U' .
 - ▶ Le graphe $G = [X, U]$ étant supposé simple et connexe, tout arbre partiel de G est un arbre à $N - 1$ arêtes et N sommets.

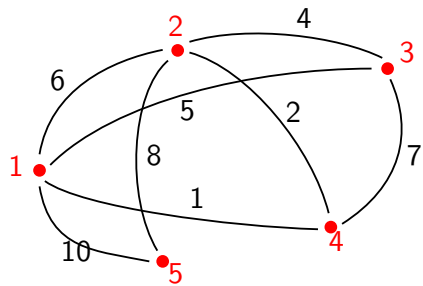
Exemple



- 1 Ranger les arêtes de U par ordre de poids croissants

(i, j)	p_{ij}	u_k
(1, 4)	1	u_1
(2, 4)	2	u_2
(2, 3)	4	u_3
(1, 3)	5	u_4
(1, 2)	6	u_5
(3, 4)	7	u_6
(2, 5)	8	u_7
(1, 5)	10	u_8

Exemple (suite)



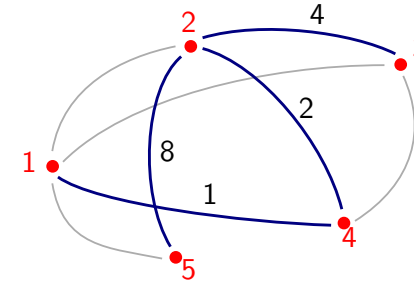
(i, j)	p_{ij}	u_k
(1, 4)	1	u_1
(2, 4)	2	u_2
(2, 3)	4	u_3
(1, 3)	5	u_4
(1, 2)	6	u_5
(3, 4)	7	u_6
(2, 5)	8	u_7
(1, 5)	10	u_8

- 2 $U' = \{u_1\}; k = 1$
- 3 $|U'| = 1$
- 4 $k = 2$
- 8 $U' = \{u_1, u_2\}$
- 3 $|U'| = 2$
- 4 $k = 3$
- 8 $U' = \{u_1, u_2, u_3\}$
- 3 $|U'| = 3$
- 4 $k = 4$
- 6 $k = 5$
- 6 $k = 6$
- 4 $k = 7$
- 8 $U' = \{u_1, u_2, u_3, u_7\}$
- 3 $|U'| = 4$

Rappel du Sommaire

- 1 Éléments de la théorie des graphes
- 2 Quelques problèmes importants de l'étude d'un graphe
- 3 **Problème du plus court chemin**
- 4 Problèmes de Flot

Exemple (suite)



- Arbre partiel de poids minimum : $G' = [X, U']$ avec $U' = \{u_1, u_2, u_3, u_7\}$
 $p(U') = p(u_1) + p(u_2) + p(u_3) + p(u_7) = 15$

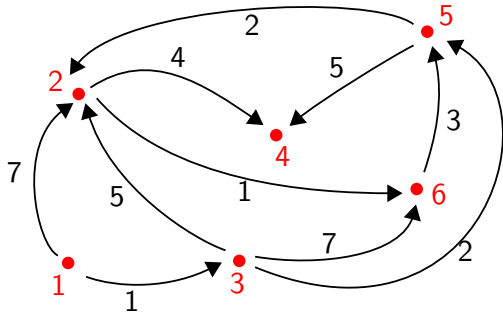
Exemples d'applications

- Les problèmes de cheminement dans les graphes sont parmi les problèmes les plus anciens de la théorie des graphes. Le problème du plus court chemin est, parmi ceux-ci, le plus typique et possède de nombreuses applications :
 - ▶ Problèmes de tournées
 - ▶ Problèmes d'optimisation de réseaux (routiers, télécommunications)
 - ▶ certains problèmes d'investissements et de gestion de stocks
 - ▶ certains problèmes en intelligence artificielle
 - ▶ ...

Préambule : Graphes valués

Définition. (Rappel : Graphe valué)

Etant donné un graphe $G = [X, U]$, on associe à chaque arc u un nombre $l(u) \in \mathbb{R}$ appelé "longueur de l'arc" ou poids de l'arc (comme précédemment). On dit que G est valué par les longueurs $l(u)$. Si $u = (i, j)$, on utilisera également la notation l_{ij} pour désigner la longueur de l'arc u .



Rappel du Sommaire

- 3 Problème du plus court chemin
 - Algorithme de Moore-Dijkstra
 - Algorithme de Ford-Bellman

Problème du plus court chemin

Définition. (Problème du plus court chemin)

Le problème du plus court chemin entre deux sommets i et j sera de trouver un chemin $\mu(i, j)$ de i à j dont la longueur totale :

$$l(\mu) = \sum_{u \in \mu(i, j)} l(u) \text{ soit minimum}$$

- La longueur d'un chemin est la somme des longueurs des arcs le constituant.

Algorithme de Moore-Dijkstra

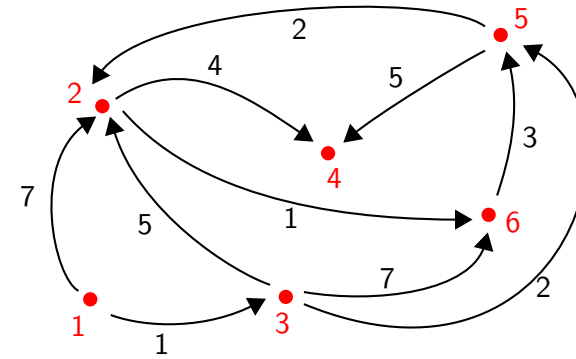
- Détermination du plus court chemin du sommet s aux autres sommets dans un graphe valué dont les longueurs sont positives càd $\forall u \in U : l(u) \geq 0$.
- Exemples de longueurs positives ou nulles :
 - ▶ Temps (en minutes, en heures, ...)
 - ▶ Coûts (en euros, ...), ...
- Posons $X = \{1, 2, \dots, N\}$ et soit l_{ij} la longueur de l'arc $(i, j) \in U$.
- Définissons $\pi^*(i)$ comme la longueur du plus court chemin de s à i . Nous avons $\pi^*(s) = 0$.
- L'algorithme utilise la représentation du graphe par Γ . Il procède en $N - 1$ itérations. Au début de chaque itération, l'ensemble X est partitionné en deux sous-ensembles, S et $\bar{S} = X \setminus S$ avec $s \in S$.
- Chaque sommet i de X est affecté d'une étiquette $\pi(i)$ qui vérifie la propriété suivante :
 - ▶ si $i \in S, \pi(i) = \pi^*(i)$
 - ▶ si $i \in \bar{S}, \pi(i) = \min_{j \in S, j \in \Gamma^{-1}(i)} (\pi(j) + l_{ji})$

Pseudo code de l'algorithme de Moore-Dijkstra

```

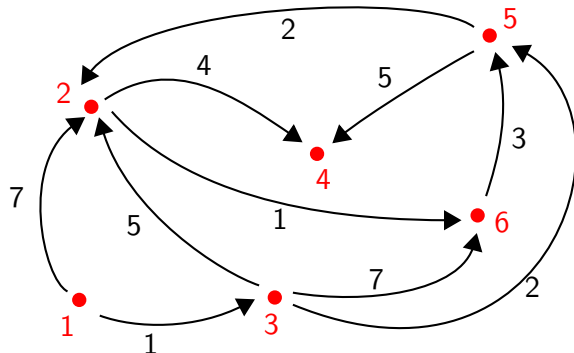
Input :  $G = [X, U], s$ 
1  $\pi(s) \leftarrow 0; \bar{S} \leftarrow \{1, 2, \dots, N\} \setminus \{s\}$ 
2 Pour tout  $i \in \bar{S}$  faire
3   Si  $i \in \Gamma(s)$  faire
4      $\pi(i) \leftarrow l_{si}$ 
5   Sinon faire
6      $\pi(i) \leftarrow +\infty$ 
7   Fin Si
8 Fin Pour
9 Tant que  $\bar{S} \neq \emptyset$  faire
10  Sélectionner  $i^* \in \bar{S}$  tel que  $\pi(i^*) = \min_{i \in \bar{S}} \{\pi(i)\}$ 
11   $\bar{S} \leftarrow \bar{S} \setminus \{i^*\}$ 
12  Pour tout  $i \in \Gamma(i^*) \cap \bar{S}$  faire
13     $\pi(i) \leftarrow \min(\pi(i), \pi(i^*) + l_{i^*i})$ 
14  Fin Pour
15 Fin Tant que
16 Output :  $\pi$ 
    
```

Exemple : cas d'un graphe orienté



- Calcul du plus court chemin entre 1 et les autres sommets.

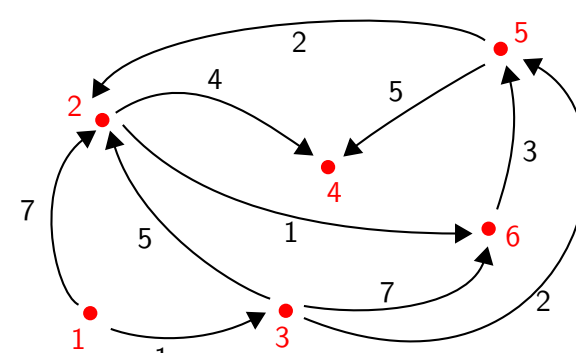
Déroulement de l'algorithme



```

1-8 Initialisation :  $S = \{1\}, \pi = (0, 7, 1, +\infty, +\infty, +\infty)$ 
10  $i^* = 3$ 
11  $S = \{1, 3\}$ 
12-14  $\Gamma_3 \cap \bar{S} = \{2, 5, 6\}$ 
 $\pi(2) = \min(7, 1 + 5) = 6, \pi(5) = \min(+\infty, 1 + 2) = 3,$ 
 $\pi(6) = \min(+\infty, 1 + 7) = 8$ 
    
```

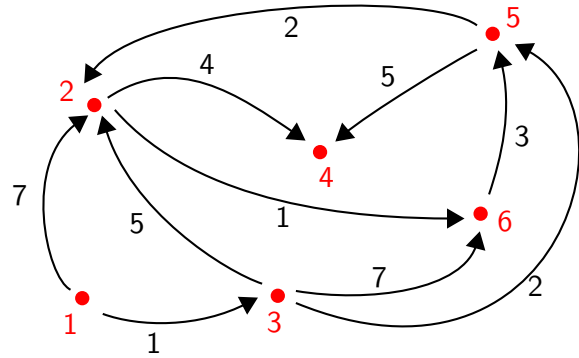
Déroulement de l'algorithme



```

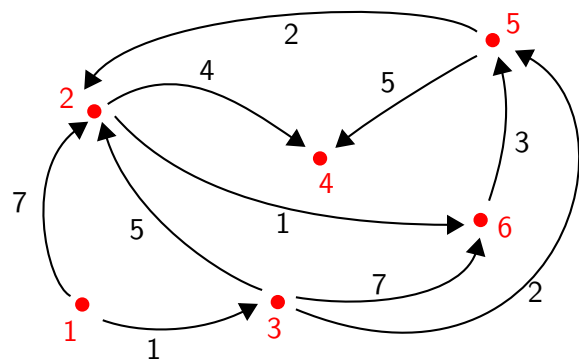
9 Valeur courante :  $S = \{1, 3\}, \pi = (0, 6, 1, +\infty, 3, 8)$ 
10  $i^* = 5$ 
11  $S = \{1, 3, 5\}$ 
12-14  $\Gamma_5 \cap \bar{S} = \{2, 4\}$ 
 $\pi(2) = \min(6, 3 + 2) = 5, \pi(4) = \min(+\infty, 3 + 5) = 8$ 
    
```


Déroulement de l'algorithme



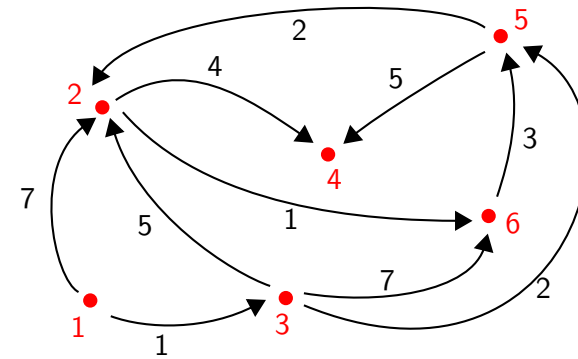
- 9 Valeur courante : $S = \{1, 3, 5\}$, $\pi = (0, 5, 1, 8, 3, 8)$
- 10 $i^* = 2$
- 11 $S = \{1, 2, 3, 5\}$
- 12-14 $\Gamma_2 \cap \bar{S} = \{4, 6\}$
 $\pi(4) = \min(8, 5 + 4) = 8$, $\pi(6) = \min(8, 5 + 1) = 6$

Déroulement de l'algorithme



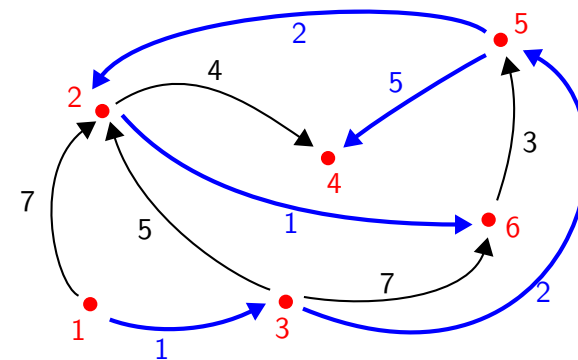
- 9 Valeur courante : $S = \{1, 2, 3, 5, 6\}$, $\pi = (0, 5, 1, 8, 3, 6)$
- 10 $i^* = 4$
- 11 $S = \{1, 2, 3, 4, 5, 6\}$
- 12 $\Gamma_4 \cap \bar{S} = \emptyset$

Déroulement de l'algorithme



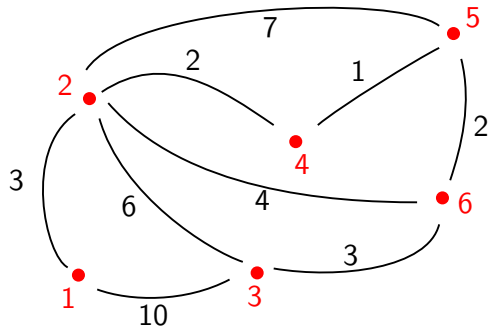
- 9 Valeur courante : $S = \{1, 2, 3, 5\}$, $\pi = (0, 5, 1, 8, 3, 6)$
- 10 $i^* = 6$
- 11 $S = \{1, 2, 3, 5, 6\}$
- 12 $\Gamma_6 \cap \bar{S} = \emptyset$

Résultat de l'algorithme



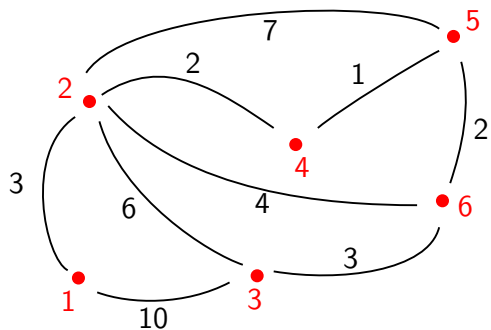
- Les plus courts chemins de 1 vers les autres sommets du graphe sont de longueurs : $\pi^* = (0, 5, 1, 8, 3, 6)$

Exemple : cas d'un graphe non orienté



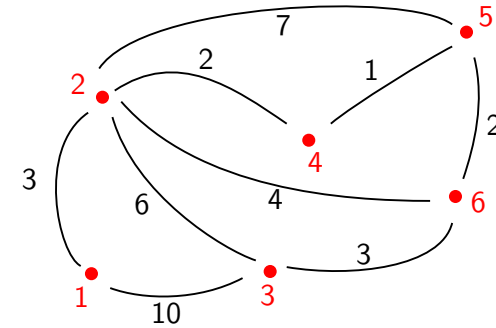
- Calcul du plus court chemin entre 1 et les autres sommets.

Déroulement de l'algorithme



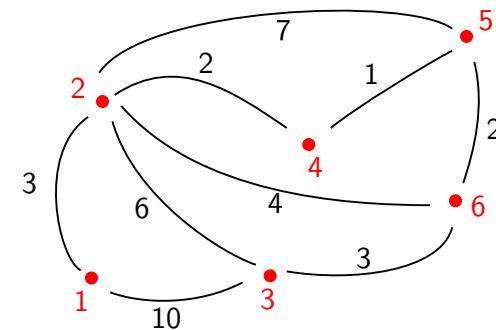
- 9 Valeur courante : $S = \{1, 2\}$, $\pi = (0, 3, 9, 5, 10, 7)$
 10 $i^* = 4$
 11 $S = \{1, 2, 4\}$
 12-14 $\Gamma_4 \cap \bar{S} = \{5\}$
 $\pi(5) = \min(10, 5 + 1) = 6$

Déroulement de l'algorithme



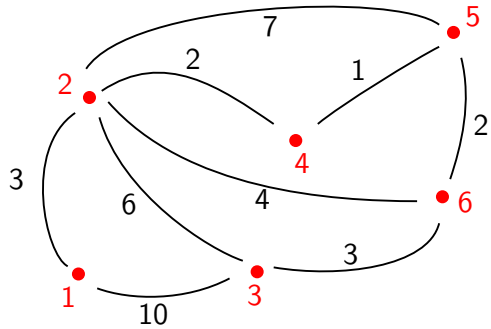
- 1-8 Initialisation : $S = \{1\}$, $\pi = (0, 3, 10, +\infty, +\infty, +\infty)$
 10 $i^* = 2$
 11 $S = \{1, 2\}$
 12-14 $\Gamma_2 \cap \bar{S} = \{3, 4, 5, 6\}$
 $\pi(3) = \min(10, 3 + 6) = 9$, $\pi(4) = \min(+\infty, 3 + 2) = 5$,
 $\pi(5) = \min(+\infty, 3 + 7) = 10$, $\pi(6) = \min(+\infty, 3 + 4) = 7$

Déroulement de l'algorithme



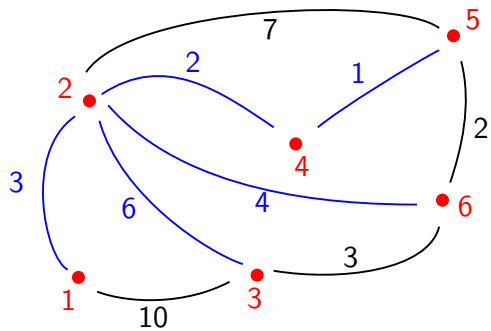
- 9 Valeur courante : $S = \{1, 2, 4\}$, $\pi = (0, 3, 9, 5, 6, 7)$
 10 $i^* = 5$
 11 $S = \{1, 2, 4, 5\}$
 12-14 $\Gamma_5 \cap \bar{S} = \{6\}$
 $\pi(6) = \min(7, 6 + 2) = 7$

Déroulement de l'algorithme



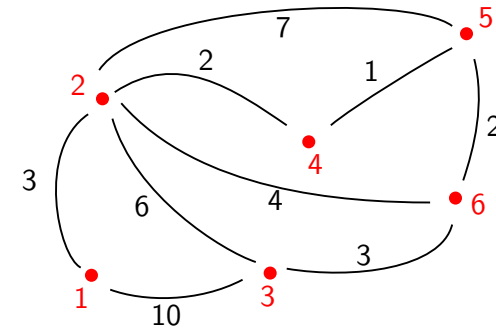
- 9 Valeur courante : $S = \{1, 2, 4, 5\}$, $\pi = (0, 3, 9, 5, 6, 7)$
- 10 $i^* = 6$
- 11 $S = \{1, 2, 4, 5, 6\}$
- 12-14 $\Gamma_5 \cap \bar{S} = \{3\}$
- $\pi(3) = \min(9, 7 + 3) = 9$

Résultat de l'algorithme



- Les plus courts chemins de 1 vers les autres sommets du graphe sont de longueurs : $\pi^* = (0, 3, 9, 5, 6, 7)$.

Déroulement de l'algorithme



- 9 Valeur courante : $S = \{1, 2, 4, 5, 6\}$, $\pi = (0, 3, 9, 5, 6, 7)$
- 10 $i^* = 3$
- 11 $S = \{1, 2, 3, 4, 5, 6\}$
- 12 $\Gamma_5 \cap \bar{S} = \emptyset$

Propriétés des plus courts chemins

Propriété. (Principe d'optimalité)

Tout sous-chemin d'un plus court chemin est un court chemin.

Démonstration.

A faire. □

Bien fondé de l'algorithme de Moore-Dijkstra

Propriété.

L'algorithme calcule successivement les sommets les plus proches de s : le sommet ajouté à l'ensemble S à l'itération $k = 1, \dots, N - 1$ de l'étape (9-15), est le k ème sommet le plus proche de s .

Complexité de l'algorithme de Moore-Dijkstra

- Soit N le nombre de sommets et M le nombre d'arcs. A chaque itération de l'étape (9-15), il y a deux opérations : une opération de sélection et une opération de mise à jour.
- Le nombre d'opérations de sélection à l'itération $k = 1, \dots, N - 1$ est $N - k$ d'où au total $\frac{N(N-1)}{2}$ opérations de sélection ($O(N^2)$).
- Le nombre d'opérations de mise à jour à l'itération k est au plus le nombre de successeurs du sommet sélectionné $d_{i^*}^+$. Donc au total, nous avons au plus $\sum_{i \in X} d_i^+ = M$ opérations de mise à jour ($O(M)$).

Propriété.

Le temps requis par l'algorithme de Moore-Dijkstra est en $O(N^2)$.

- Remarque : si le graphe est peu dense (M petit), on peut utiliser des structures de données particulières pour améliorer la complexité de $O(N^2)$ à $O(M \log N)$.

Démonstration.

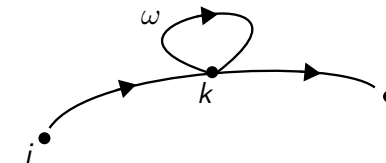
Notons S_k l'ensemble S à l'itération k et $\bar{S}_k = X \setminus S_k$.

- Propriété vraie pour $k = 1$ (S_1 contient le plus proche voisin de s)
- Supposons que la propriété est vraie pour $k > 1$ (S_k contient les k plus proches voisins de s) et montrons qu'elle est vraie pour $k + 1$:
 - Observation : à l'issue de l'itération k , $\forall i \in \bar{S}_k, \pi_k(i) = \min_{j \in S_k, j \in \Gamma^{-1}(i)} (\pi_k(j) + l_{ji})$. $\pi_k(i)$ est le plus court chemin entre s et i parmi ceux passant par des sommets de S_k .
 - A l'itération $k + 1$, on sélectionne i^* tel que $\pi_k(i^*) = \min_{i \in \bar{S}_k} \pi_k(i)$. Montrons que celui-ci est le $k + 1$ ème plus proche voisin de s en montrant que pour tout autre chemin $\mu(s, i^*)$, $l(\mu(s, i^*)) \geq \pi_k(i^*)$.
 - Posons $\mu(s, i^*) = \{\mu(s, h), \mu(h, i^*)\}$ où h est le **premier** sommet rencontré dans $\mu(s, i^*)$ qui appartienne à \bar{S}_k .
 - On a $l(\mu(s, h)) \geq \pi_k(h)$ (par définition), puis $l(\mu(s, h)) + l(\mu(h, i^*)) \geq \pi_k(h)$ car $l(\mu(h, i^*)) \geq 0$ et donc $l(\mu(s, i^*)) \geq \pi_k(h)$. Or $\pi_k(h) \geq \pi_k(i^*)$ (par définition). On en déduit que $l(\mu(s, i^*)) \geq \pi_k(i^*)$.



Cas des graphes valués quelconques

- Soit $G = [X, U]$ un graphe valué tel que $\forall u \in U : l(u) \in \mathbb{R}$.
- **Conditions d'existence de plus courts chemins :**
Soit $\mu(i, j)$ un chemin de i à j comprenant un circuit ω comme illustré ci-dessous :



- Soit $\mu'(i, j)$ un chemin de i à j ne comprenant pas le circuit ω . On a : $l(\mu) = l(\mu') + l(\omega)$. Ainsi :
 - Si $l(\omega) < 0$, il n'existe pas de plus court chemin de i à j (ω est un **circuit absorbant**)
 - Si $l(\omega) \geq 0$ alors $l(\mu') \leq l(\mu)$ et dans la recherche d'un plus court chemin on peut se restreindre aux **chemins élémentaires**.
- Dans la suite on supposera qu'il n'y a pas de circuit de longueur négative.

Rappel du Sommaire

- 3 Problème du plus court chemin
 - Algorithme de Moore-Dijkstra
 - Algorithme de Ford-Bellman

Pseudo code de l'algorithme de Ford-Bellman

```

Input :  $G = [X, U], s$ 
1  $\pi(s) \leftarrow 0$ 
2 Pour tout  $i \in \{1, 2, \dots, N\} \setminus \{s\}$  faire
3    $\pi(i) \leftarrow +\infty$ 
4 Fin Pour
5 Répéter
6   Pour tout  $i \in \{1, 2, \dots, N\} \setminus \{s\}$  faire
7      $\pi(i) \leftarrow \min(\pi(i), \min_{j \in \Gamma^{-1}(i)} \pi(j) + l_{ji})$ ;
8   Fin Pour
9 Tant que une des valeurs  $\pi(i)$  change dans la boucle Pour
10 Output :  $\pi$ 

```

Algorithme de Ford-Bellman

- Détermination du plus court chemin du sommet s aux autres sommets dans un graphe valué dont les longueurs sont quelconques c'ad $\forall u \in U : l(u) \in \mathbb{R}$ mais pour lequel il n'y a pas de circuit de longueur négative.
- Comme précédemment, $\pi^*(i)$ est la longueur du plus court chemin de s à i et nous avons $\pi^*(s) = 0$.
- L'algorithme utilise la représentation du graphe par Γ^{-1} . L'algorithme affine successivement une borne supérieure de la longueur du plus court chemin entre s et tous les autres sommets jusqu'à atteindre la longueur minimale.

Complexité de l'algorithme de Ford-Bellman

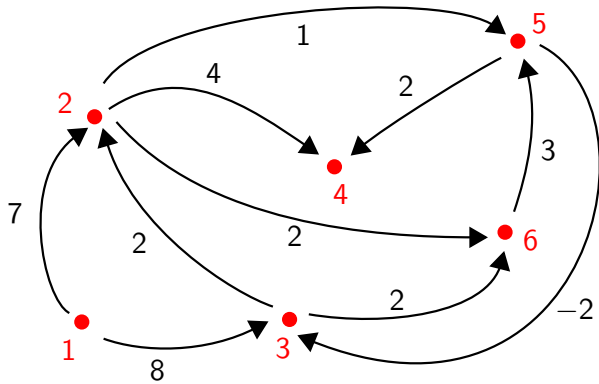
- Soit N le nombre de sommets et M le nombre d'arcs. L'étape (5-9) demande chaque fois M opérations d'additions et de comparaisons. Par ailleurs, il y a au plus N itérations de l'étape (5-9) (car les chemins élémentaires sont de longueur inférieure ou égale à N).

Propriété.

Le temps requis par l'algorithme de Ford-Bellman est en $O(NM)$.

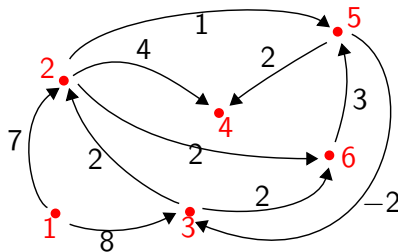
- Remarque : L'ordre dans lequel les sommets sont parcourus à l'étape (5-9) est important puisqu'il influence le nombre de fois que cette étape est itérée (avant convergence).

Exemple d'application : cas d'un graphe valué de longueur quelconque



- Calcul du plus court chemin entre 1 et les autres sommets.

Déroulement de l'algorithme

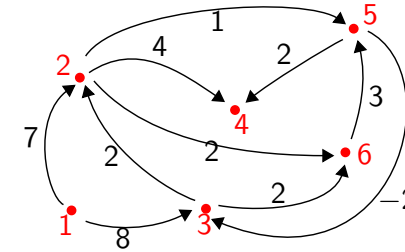


- Avec l'ordre suivant : 2, 3, 4, 5, 6.
- Valeur courante $\pi = (0, 7, 8, 11, 8, 9)$

Itération 2 (6-8)

$$\begin{cases} \pi(2) = \min(\pi(2), \pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(3), \pi(1) + l_{13}, \pi(5) + l_{53}) = 6 \\ \pi(4) = \min(\pi(4), \pi(2) + l_{24}, \pi(5) + l_{54}) = 10 \\ \pi(5) = \min(\pi(5), \pi(2) + l_{25}, \pi(6) + l_{65}) = 8 \\ \pi(6) = \min(\pi(6), \pi(2) + l_{26}, \pi(3) + l_{36}) = 8 \end{cases}$$

Déroulement de l'algorithme

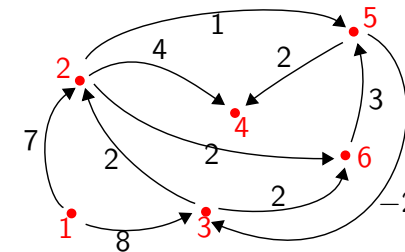


- Avec l'ordre suivant : 2, 3, 4, 5, 6.
- Initialisation (1-4) $\pi = (0, +\infty, +\infty, +\infty, +\infty, +\infty)$

Itération 1 (6-8)

$$\begin{cases} \pi(2) = \min(\pi(2), \pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(3), \pi(1) + l_{13}, \pi(5) + l_{53}) = 8 \\ \pi(4) = \min(\pi(4), \pi(2) + l_{24}, \pi(5) + l_{54}) = 11 \\ \pi(5) = \min(\pi(5), \pi(2) + l_{25}, \pi(6) + l_{65}) = 8 \\ \pi(6) = \min(\pi(6), \pi(2) + l_{26}, \pi(3) + l_{36}) = 9 \end{cases}$$

Déroulement de l'algorithme

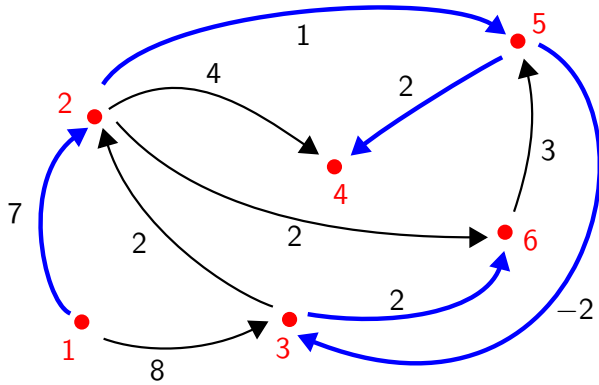


- Avec l'ordre suivant : 2, 3, 4, 5, 6.
- Valeur courante $\pi = (0, 7, 6, 10, 8, 8)$

Itération 3 (6-8)

$$\begin{cases} \pi(2) = \min(\pi(2), \pi(1) + l_{12}, \pi(3) + l_{32}) = 7 \\ \pi(3) = \min(\pi(3), \pi(1) + l_{13}, \pi(5) + l_{53}) = 6 \\ \pi(4) = \min(\pi(4), \pi(2) + l_{24}, \pi(5) + l_{54}) = 10 \\ \pi(5) = \min(\pi(5), \pi(2) + l_{25}, \pi(6) + l_{65}) = 8 \\ \pi(6) = \min(\pi(6), \pi(2) + l_{26}, \pi(3) + l_{36}) = 8 \end{cases}$$

Résultat de l'algorithme



- Les plus courts chemins de 1 vers les autres sommets du graphe sont de longueurs : $\pi^* = (0, 7, 6, 10, 8, 8)$.

Pseudo code de l'identification des chemins de longueurs minimales

- Identification d'un chemin de longueur minimale entre s et i étant donné π^* et les longueurs des arcs du graphe valué G .
- On part de l'extrémité finale i et on remonte progressivement vers s .

Input : $G = [X, U]$ et π^*

1 $k \leftarrow i$;

2 $\mu \leftarrow \{ \}$;

3 **Tant que** $k \neq s$ **faire**

4 Rechercher j tel que $\pi^*(j) = \pi^*(k) - l_{jk}$

5 $\mu \leftarrow (j, k) \cup \mu$

6 $k \leftarrow j$

7 **Fin Tant que**

- Appliquer l'algorithme à l'exemple précédent pour retrouver les chemins de longueurs minimales donnés en bleu.

Identification d'un chemin de longueur minimale

- Etant donné un graphe valué $G = [X, U]$ et le vecteur, π^* , des longueurs des plus courts chemins entre s et les autres sommets. Comment déterminer les chemins correspondants à ces longueurs minimales ?

Rappel du Sommaire

- 1 Eléments de la théorie des graphes
- 2 Quelques problèmes importants de l'étude d'un graphe
- 3 Problème du plus court chemin
- 4 Problèmes de Flot

Exemples d'applications

- Les problèmes de flots constituent un très important domaine de la théorie des graphes. Sous leur forme la plus simple, ils consistent à organiser de façon optimale, sous diverses contraintes, les mouvements de certaines quantités d'un bien dans un **réseau**.
- Ces mouvements concernent par exemple :
 - ▶ l'acheminement d'un produit depuis les centres de production vers les centres de distribution (réseau géographique)
 - ▶ la répartition des communications téléphoniques entre les différents centres de gestion (réseau téléphonique)
 - ▶ l'organisation de la circulation routière entre plusieurs villes (réseau routier)
 - ▶ la distribution des tâches au sein d'un ensemble de personnes (problèmes d'affectation)
 - ▶ ...

Réseaux avec capacités

Définition. (Réseau avec capacités)

Soit $G = [X, U]$ un 1-graphe orienté antisymétrique^a comportant N sommets. Nous supposons qu'il existe dans X deux sommets particuliers s (sommets entrée ou "source") et p (sommets sortie ou "puits") tels que $\Gamma^{-1}(s) = \emptyset$ et $\Gamma(p) = \emptyset$. On associe à chaque arc u une quantité $c(u) \in \mathbb{N}^b$ appelé "capacité de l'arc". Si $u = (i, j)$, on utilisera également la notation c_{ij} pour désigner la capacité de l'arc u et on dénote $C = \{c_{ij}, (i, j) \in U\}$. On dit que $G = [X, U, C]$ est un réseau avec capacités.

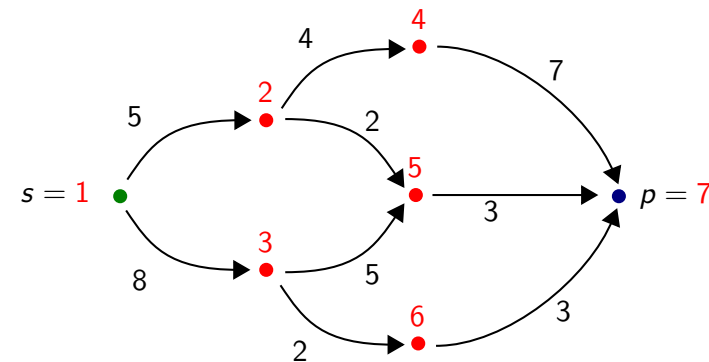
a. $\forall a \neq b (aRb \Rightarrow b\bar{R}a)$

b. Ensemble des entiers naturels

Exemples d'applications (suite)

- Caractéristiques des réseaux selon le domaine d'application :
 - ▶ Caractéristiques physiques : capacité de débit, d'écoulement, et de réception...
 - ▶ Caractéristiques économiques : coût unitaire du mouvement ou de l'attente...
- Les problématiques relatives à ces applications sont :
 - ▶ les quantités maximales pouvant être produites, livrées, stockées; des capacités des lignes et des routes existantes...
 - ▶ les coûts de pénurie, de livraison et de stockage; les frais de communication et de déplacement...

Exemple



Flot et valeur d'un flot

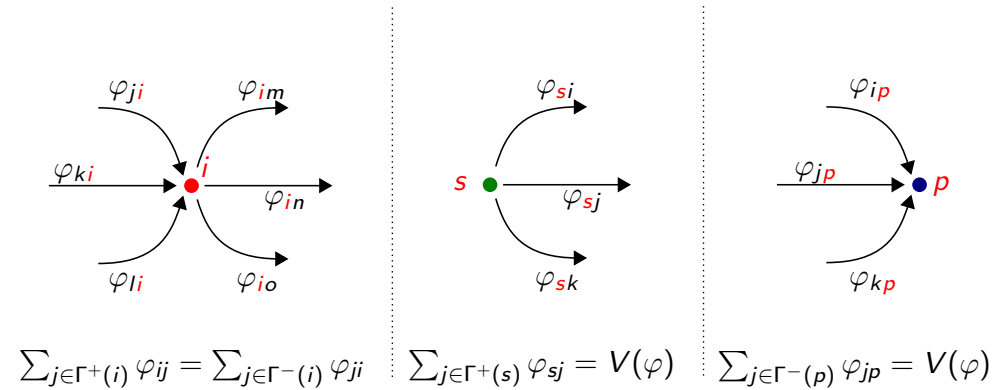
Définition. (Flot dans un réseau avec capacités)

On appelle **flot** dans un réseau avec capacités $G = [X, U, C]$, un graphe $F = [X, U, \varphi]$ tel que $\varphi = \{\varphi_{ij} \geq 0, (i, j) \in U\}$ est un ensemble de quantités non-négatives sur chaque arc $(i, j) \in U$. Ce flot est **réalisable** s'il satisfait aux contraintes suivantes :

- Contraintes de capacités : $\forall (i, j) \in U : \varphi_{ij} \leq c_{ij}$
- Contraintes de conservations du flux :
 - $\forall i \in X \setminus \{s, p\} : \sum_{j \in \Gamma^-(i)} \varphi_{ji} = \sum_{j \in \Gamma^+(i)} \varphi_{ij}$
 - $\sum_{j \in \Gamma^+(s)} \varphi_{sj} = \sum_{j \in \Gamma^-(p)} \varphi_{jp} = V(\varphi)$

où $V(\varphi)$, appelé **valeur du flot**, est le volume total que met en circulation le flot réalisable φ dans le réseau.

Illustrations (lois de Kirchhoff)



Rappel du Sommaire

- 4 Problèmes de Flot
 - Flot de valeur maximale

Problème du flot de valeur maximale

Définition. (Problème du flot de valeur maximale)

Soit $G = [X, U, C]$ un réseau avec capacité. Le **flot de valeur maximale** φ^* est, parmi l'ensemble des flots réalisables \mathbb{F} , celui qui maximise la quantité $V(\varphi)$:

$$\forall \varphi \in \mathbb{F} : V(\varphi^*) \geq V(\varphi)$$

- Remarque :
 - Un flot de valeur maximale n'est pas nécessairement unique.
 - Ce problème peut-être modélisé par programmation linéaire.
 - Mais les caractéristiques de celui-ci permettent d'élaborer des algorithmes de résolutions exactes plus efficaces : algorithmes de Ford-Fulkerson.

Principes de l'algorithme de Ford-Fulkerson

- 1 On part d'un flot réalisable initial $\varphi = \{\varphi_{ij}, (i, j) \in U\}$ de valeur $V(\varphi)$.
- 2 A chaque itération, on essaie d'accroître le flot par le marquage de certains sommets (pas tous, cf conditions ci-après) :
 - ▶ s porte une marque permanente $m_s = (\emptyset, \infty, +)$ (un triplet) indiquant qu'il peut engendrer un flot arbitrairement grand.
 - ▶ A un sommet $j \neq s$ est associé une marque m_j ayant la forme $m_j = (i, \alpha_j, \pm)$ où :
 - ★ $m_j(1) = i$ représente le sommet à partir duquel j a été marqué,
 - ★ $m_j(2) = \alpha_j > 0$ est la modification maximale possible, en valeur absolue, du flot φ_{ij} circulant entre i et j ,
 - ★ $m_j(3) = +$ indique que $i \in \Gamma^{-1}(j)$ dans ce cas $\alpha_j = \min\{\alpha_i, r_{ij} = c_{ij} - \varphi_{ij}\}$
 - ★ $m_j(3) = -$ indique que $i \in \Gamma(j)$ dans ce cas $\alpha_j = \min\{\alpha_i, r_{ij} = \varphi_{ji}\}$.

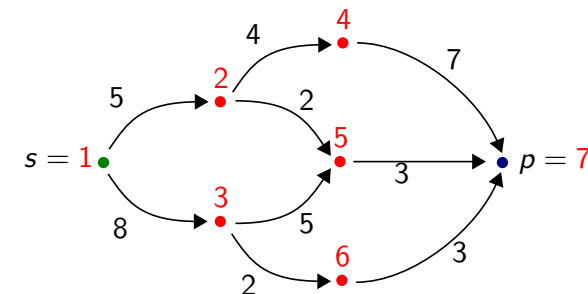
```

Input :  $G = [X, U, C], s, p$  et  $\varphi$  un flot réalisable
 $m_s \leftarrow (\emptyset, \infty, +)$  et  $S = \{s\}$ 
Tant que  $\exists (j \in \bar{S}, i \in S) : (c_{ij} - \varphi_{ij} > 0) \vee (\varphi_{ji} > 0)$  faire
    Si  $c_{ij} - \varphi_{ij} > 0$  faire
         $m_j \leftarrow (i, \alpha_j, +)$  avec  $\alpha_j = \min\{\alpha_i, c_{ij} - \varphi_{ij}\}$ 
    Sinon Si  $\varphi_{ji} > 0$  faire
         $m_j \leftarrow (i, \alpha_j, -)$  avec  $\alpha_j = \min\{\alpha_i, \varphi_{ji}\}$ 
    Fin Si
     $S \leftarrow S \cup \{j\}$ 
    Si  $j = p$  faire
         $V(\varphi) \leftarrow V(\varphi) + \alpha_p$ 
        Aller en 14
    Fin Si
Fin Tant que
Si  $p \in S$  faire
    Tant que  $j \neq s$  faire
        Si  $m_j(3) = +$  faire
             $\varphi_{m_j(1)j} \leftarrow \varphi_{m_j(1)j} + \alpha_p$ 
        Sinon Si  $m_j(3) = -$  faire
             $\varphi_{jm_j(1)} \leftarrow \varphi_{jm_j(1)} - \alpha_p$ 
        Fin Si
         $j \leftarrow m_j(1)$ 
    Fin Tant que
    Aller en 1
Sinon faire
    Output :  $\varphi$ 
    
```

Principes de l'algorithme de Ford-Fulkerson (suite)

- 3 Si le sommet p a été marqué, la quantité α_p représente l'accroissement de la valeur du flot réalisable à cette étape. Pour réaliser cet accroissement :
 - ▶ On construit, grâce aux marques, la chaîne allant de s à p qui a permis le marquage de p . Cette chaîne est constituée d'arcs tels que l'une des deux conditions est vérifiée :
 - a) $(i, j) \in U$ avec $r_{ij} = c_{ij} - \varphi_{ij} > 0$ et $m_j = (i, r_{ij}, +)$
 - b) $(j, i) \in U$ avec $r_{ij} = \varphi_{ji} > 0$ et $m_j = (i, r_{ij}, -)$
 - ▶ On en déduit un nouveau flot réalisable φ de valeur strictement supérieure (de α_p) à celle du flot initial et de composants :
 - ★ $\varphi_{ij} \leftarrow \varphi_{ij} + \alpha_p$ si (i, j) est un arc de la chaîne dans le cas a)
 - ★ $\varphi_{ji} \leftarrow \varphi_{ji} - \alpha_p$ si (j, i) est un arc de la chaîne dans le cas b)
 - ★ φ reste inchangé pour les autres arcs $(i, j) \in U$
 - ▶ On utilise ce nouveau flot comme flot initial pour une nouvelle itération.
- 4 Si p ne peut être marqué, le flot φ obtenu à la fin de l'étape précédente est un flot de valeur maximale.

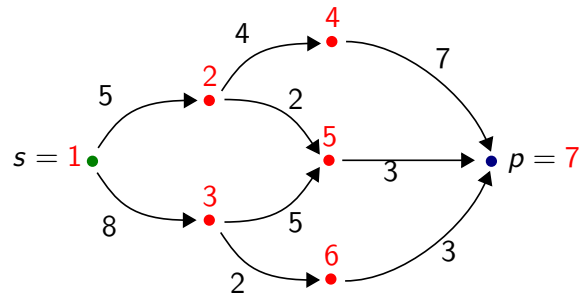
Exemple (1ère itération)



Input : $G = [X, U, C], \varphi = \{\varphi_{ij} = 0, (i, j) \in U\}$

- 1 $m_s = m_1 = (\emptyset, \infty, +)$ et $S = \{1\}$
- 2 $(j = 2 \in \bar{S}, i = 1 \in S)$ est tel que $c_{12} - \varphi_{12} = 5 > 0$
- 4 $m_2 = (1, \alpha_2, +)$ avec $\alpha_2 = \min\{\infty, 5\} = 5$
- 8 $S = \{1, 2\}$

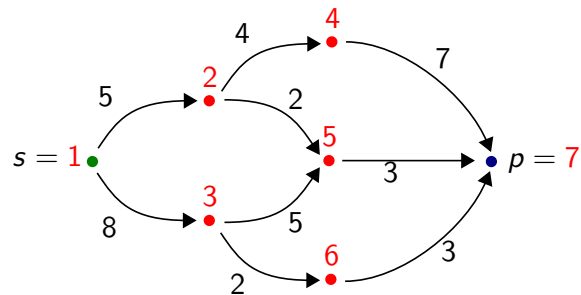
Exemple (suite - 1ère itération)



- 2 $(j = 3 \in \bar{S}, i = 1 \in S)$ est tel que $c_{13} - \varphi_{13} = 8 > 0$
- 4 $m_3 = (1, \alpha_3, +)$ avec $\alpha_3 = \min\{\infty, 8\} = 8$
- 8 $S = \{1, 2, 3\}$

- 2 $(j = 4 \in \bar{S}, i = 2 \in S)$ est tel que $c_{24} - \varphi_{24} = 4 > 0$
- 4 $m_4 = (2, \alpha_4, +)$ avec $\alpha_4 = \min\{5, 4\} = 4$
- 8 $S = \{1, 2, 3, 4\}$

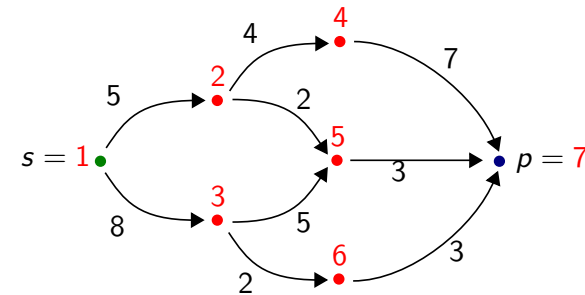
Exemple (suite - 1ère itération)



- 2 $(j = 7 \in \bar{S}, i = 4 \in S)$ est tel que $c_{47} - \varphi_{47} = 7 > 0$
- 4 $m_7 = (4, \alpha_7, +)$ avec $\alpha_7 = \min\{4, 7\} = 4$
- 8 $S = \{1, 2, 3, 4, 5, 6, 7\}$

- 9 $j = 7 = p$
- 10 $V(\varphi) = 4$
- 11 Aller en 14

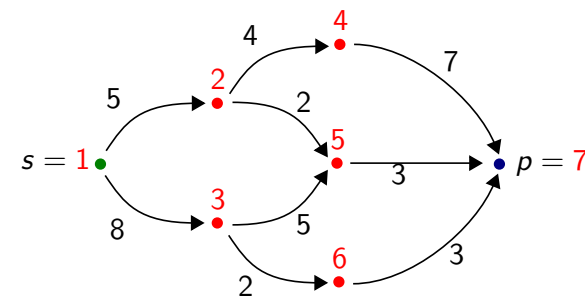
Exemple (suite - 1ère itération)



- 2 $(j = 5 \in \bar{S}, i = 2 \in S)$ est tel que $c_{25} - \varphi_{25} = 2 > 0$
- 4 $m_5 = (2, \alpha_5, +)$ avec $\alpha_5 = \min\{5, 2\} = 2$
- 8 $S = \{1, 2, 3, 4, 5\}$

- 2 $(j = 6 \in \bar{S}, i = 3 \in S)$ est tel que $c_{36} - \varphi_{36} = 2 > 0$
- 4 $m_6 = (3, \alpha_6, +)$ avec $\alpha_6 = \min\{8, 2\} = 2$
- 8 $S = \{1, 2, 3, 4, 5, 6\}$

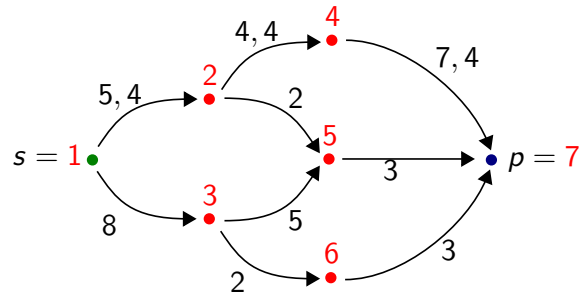
Exemple (suite - 1ère itération)



- 14 $p \in S$
- 15-22 $\varphi_{47} = 4, \varphi_{24} = 4, \varphi_{12} = 4$

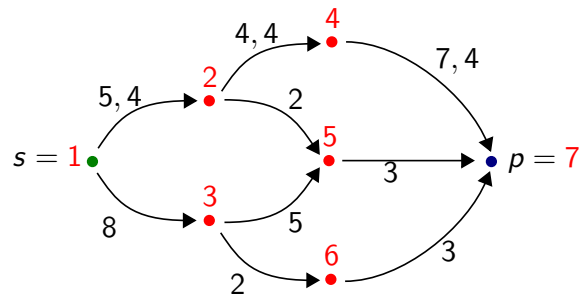
- 23 Aller en 1

Exemple (suite - 2ème itération)



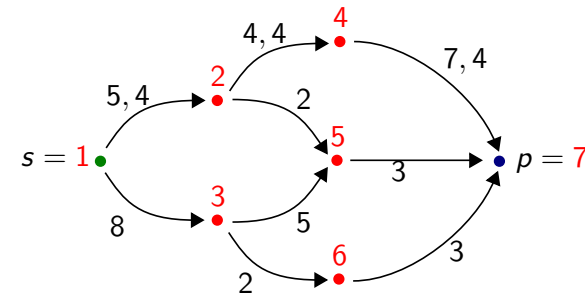
- 1 $m_1 = (\emptyset, \infty, +)$ et $S = \{1\}$
- 2 $(j = 2 \in \bar{S}, i = 1 \in S)$ est tel que $c_{12} - \varphi_{12} = 1 > 0$
- 4 $m_2 = (1, \alpha_2, +)$ avec $\alpha_2 = \min\{\infty, 1\} = 1$
- 8 $S = \{1, 2\}$
- 2 $(j = 3 \in \bar{S}, i = 1 \in S)$ est tel que $c_{13} - \varphi_{13} = 8 > 0$
- 4 $m_3 = (1, \alpha_3, +)$ avec $\alpha_3 = \min\{\infty, 8\} = 8$
- 8 $S = \{1, 2, 3\}$

Exemple (suite - 2ème itération)



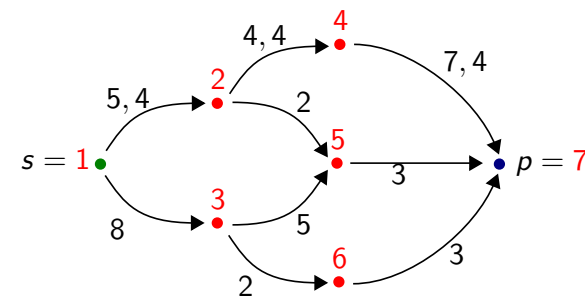
- 2 $(j = 7 \in \bar{S}, i = 5 \in S)$ est tel que $c_{57} - \varphi_{57} = 3 > 0$
- 4 $m_7 = (5, \alpha_7, +)$ avec $\alpha_7 = \min\{1, 3\} = 1$
- 8 $S = \{1, 2, 3, 5, 6, 7\}$
- 9 $j = 7 = p$
- 10 $V(\varphi) = 5$
- 11 Aller en 14

Exemple (suite - 2ème itération)



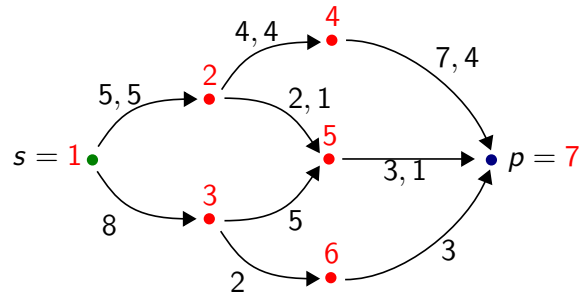
- 2 $(j = 5 \in \bar{S}, i = 2 \in S)$ est tel que $c_{25} - \varphi_{25} = 2 > 0$
- 4 $m_5 = (2, \alpha_5, +)$ avec $\alpha_5 = \min\{1, 2\} = 1$
- 8 $S = \{1, 2, 3, 5\}$
- 2 $(j = 6 \in \bar{S}, i = 3 \in S)$ est tel que $c_{36} - \varphi_{36} = 2 > 0$
- 4 $m_6 = (3, \alpha_6, +)$ avec $\alpha_6 = \min\{8, 2\} = 2$
- 8 $S = \{1, 2, 3, 5, 6\}$

Exemple (suite - 2ème itération)



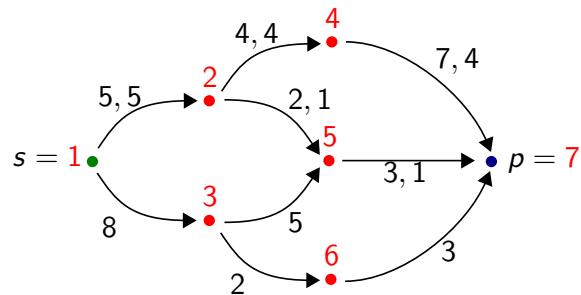
- 14 $p \in S$
- 15-22 $\varphi_{57} = 1, \varphi_{25} = 1, \varphi_{12} = 5$
- 23 Aller en 1

Exemple (suite - 3ème itération)



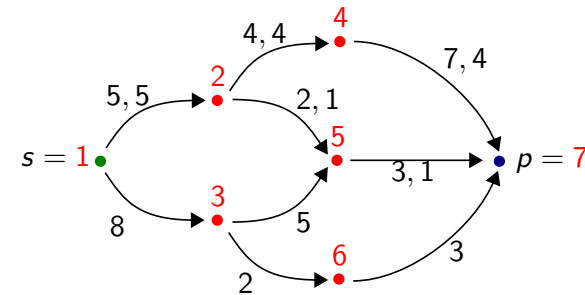
- 1 $m_1 = (\emptyset, \infty, +)$ et $S = \{1\}$
- 2 $(j = 3 \in \bar{S}, i = 1 \in S)$ est tel que $c_{13} - \varphi_{13} = 8 > 0$
- 4 $m_3 = (1, \alpha_3, +)$ avec $\alpha_3 = \min\{\infty, 8\} = 8$
- 8 $S = \{1, 3\}$
- 2 $(j = 5 \in \bar{S}, i = 3 \in S)$ est tel que $c_{35} - \varphi_{35} = 5 > 0$
- 4 $m_5 = (3, \alpha_5, +)$ avec $\alpha_5 = \min\{8, 5\} = 5$
- 8 $S = \{1, 3, 5\}$

Exemple (suite - 3ème itération)



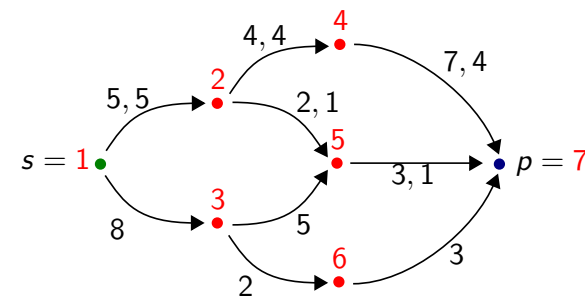
- 2 $(j = 7 \in \bar{S}, i = 5 \in S)$ est tel que $c_{57} - \varphi_{57} = 2 > 0$
- 4 $m_7 = (5, \alpha_7, +)$ avec $\alpha_7 = \min\{5, 2\} = 2$
- 8 $S = \{1, 3, 5, 2, 6, 7\}$
- 9 $j = 7 = p$
- 10 $V(\varphi) = 7$
- 11 Aller en 14

Exemple (suite - 3ème itération)



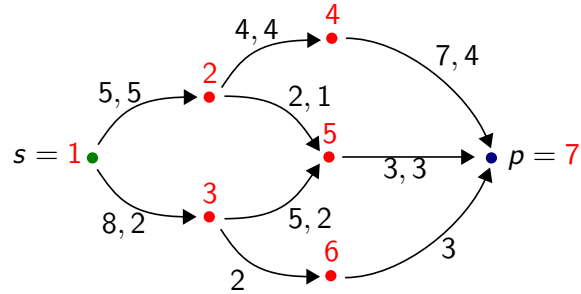
- 2 $(j = 2 \in \bar{S}, i = 5 \in S)$ est tel que $\varphi_{25} = 1 > 0$
- 4 $m_2 = (5, \alpha_2, -)$ avec $\alpha_2 = \min\{5, 1\} = 1$
- 8 $S = \{1, 3, 5, 2\}$
- 2 $(j = 6 \in \bar{S}, i = 3 \in S)$ est tel que $c_{36} - \varphi_{36} = 2 > 0$
- 4 $m_6 = (3, \alpha_6, +)$ avec $\alpha_6 = \min\{8, 2\} = 2$
- 8 $S = \{1, 3, 5, 2, 6\}$

Exemple (suite - 3ème itération)



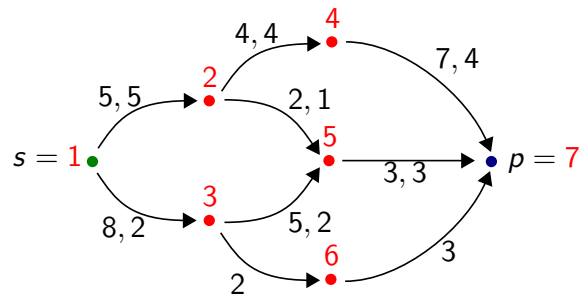
- 14 $p \in S$
- 15-22 $\varphi_{57} = 3, \varphi_{35} = 2, \varphi_{13} = 2$
- 23 Aller en 1

Exemple (suite - 4ème itération)



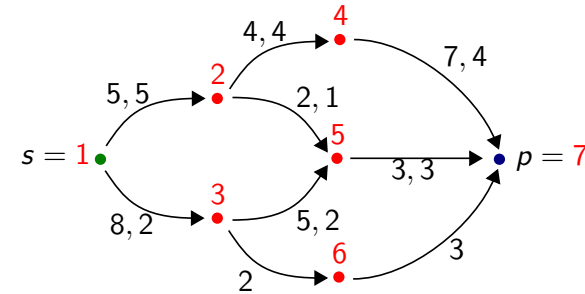
- 1 $m_1 = (\emptyset, \infty, +)$ et $S = \{1\}$
- 2 $(j = 3 \in \bar{S}, i = 1 \in S)$ est tel que $c_{13} - \varphi_{13} = 6 > 0$
- 4 $m_3 = (1, \alpha_3, +)$ avec $\alpha_3 = \min\{\infty, 6\} = 6$
- 8 $S = \{1, 3\}$
- 2 $(j = 5 \in \bar{S}, i = 3 \in S)$ est tel que $c_{35} - \varphi_{35} = 3 > 0$
- 4 $m_5 = (3, \alpha_5, +)$ avec $\alpha_5 = \min\{6, 3\} = 3$
- 8 $S = \{1, 3, 5\}$

Exemple (suite - 4ème itération)



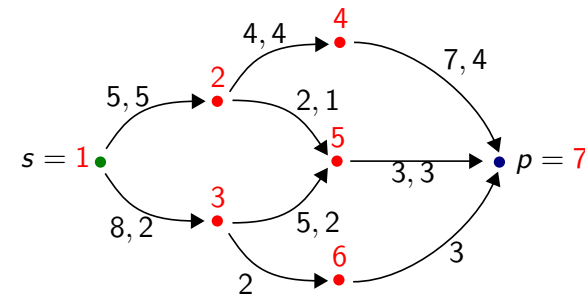
- 2 $(j = 7 \in \bar{S}, i = 6 \in S)$ est tel que $c_{67} - \varphi_{67} = 3 > 0$
- 4 $m_7 = (6, \alpha_7, +)$ avec $\alpha_7 = \min\{2, 3\} = 2$
- 8 $S = \{1, 3, 5, 6, 2, 7\}$
- 9 $j = 7 = p$
- 10 $V(\varphi) = 9$
- 11 Aller en 14

Exemple (suite - 4ème itération)



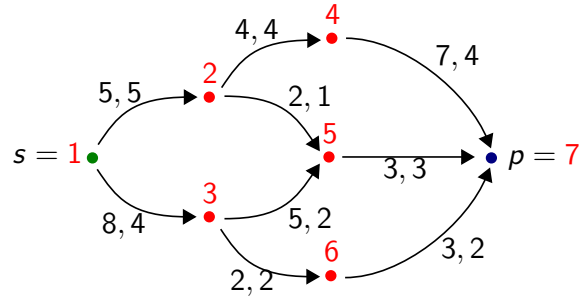
- 2 $(j = 6 \in \bar{S}, i = 3 \in S)$ est tel que $c_{36} - \varphi_{36} = 2 > 0$
- 4 $m_6 = (3, \alpha_6, +)$ avec $\alpha_6 = \min\{6, 2\} = 2$
- 8 $S = \{1, 3, 5, 6\}$
- 2 $(j = 2 \in \bar{S}, i = 5 \in S)$ est tel que $\varphi_{25} = 1 > 0$
- 4 $m_2 = (5, \alpha_2, -)$ avec $\alpha_2 = \min\{3, 1\} = 1$
- 8 $S = \{1, 3, 5, 6, 2\}$

Exemple (suite - 4ème itération)



- 14 $p \in S$
- 15-22 $\varphi_{67} = 2, \varphi_{36} = 2, \varphi_{13} = 4$
- 23 Aller en 1

Exemple (suite - 5ème itération)



- 1 $m_1 = (\emptyset, \infty, +)$ et $S = \{1\}$
- 2 $(j = 3 \in \bar{S}, i = 1 \in S)$ est tel que $c_{13} - \varphi_{13} = 4 > 0$
- 4 $m_3 = (1, \alpha_3, +)$ avec $\alpha_3 = \min\{\infty, 4\} = 4$
- 8 $S = \{1, 3\}$
- 2 $(j = 5 \in \bar{S}, i = 3 \in S)$ est tel que $c_{35} - \varphi_{35} = 3 > 0$
- 4 $m_5 = (3, \alpha_5, +)$ avec $\alpha_5 = \min\{4, 3\} = 3$
- 8 $S = \{1, 3, 5\}$

Bien fondé de l'algorithme de Ford-Fulkerson

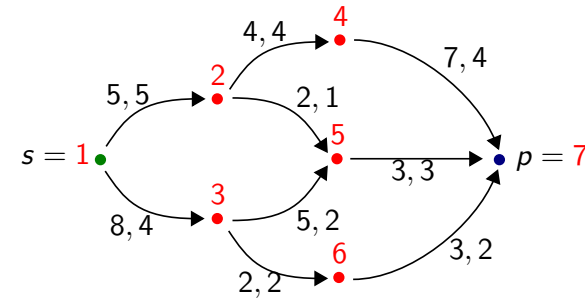
- Pourquoi l'algorithme de Ford-Fulkerson solutionne le problème de manière exacte ?
- Quelques définitions et propriétés en préambule :

Définition. (Coupe)

Une **coupe** $K = \{S, \bar{S}\}$ dans le réseau, est une partition de X en deux sous-ensembles S et \bar{S} tels que $s \in S$ et $p \in \bar{S}$. La capacité d'une coupe est définie par :

$$c(K) = \sum_{(i,j) \in U : i \in S \wedge j \in \bar{S}} c_{ij}$$

Exemple (suite - 5ème itération)



- 2 $(j = 2 \in \bar{S}, i = 5 \in S)$ est tel que $\varphi_{25} = 1 > 0$
- 4 $m_2 = (5, \alpha_2, -)$ avec $\alpha_2 = \min\{3, 1\} = 1$
- 8 $S = \{1, 3, 5, 2\}$
- 25 **Output** : φ

Bien fondé de l'algorithme de Ford-Fulkerson (suite)

Propriété.

Pour toute coupe $K = \{S, \bar{S}\}$ et tout flot φ , on a :

$$V(\varphi) = \sum_{i \in S, j \notin S} \varphi_{ij} - \sum_{i \in S, j \notin S} \varphi_{ji}$$

Démonstration.

$V(\varphi) = \sum_{j \in \Gamma(s)} \varphi_{sj} + \sum_{i \in S: i \neq s} \left(\sum_{j \in \Gamma(i)} \varphi_{ij} - \sum_{j \in \Gamma^{-1}(i)} \varphi_{ji} \right)$
 s n'ayant pas de prédécesseur donc $\sum_{j \in \Gamma^{-1}(s)} \varphi_{js} = 0$ et nous avons :

$$V(\varphi) = \sum_{i \in S} \left(\sum_{j \in \Gamma(i)} \varphi_{ij} - \sum_{j \in \Gamma^{-1}(i)} \varphi_{ji} \right)$$

Considérons maintenant la coupe $K = \{S, \bar{S}\}$, nous avons :

$$V(\varphi) =$$

$$\sum_{i \in S} \left(\sum_{j \in \Gamma(i) \cap S} \varphi_{ij} + \sum_{j \in \Gamma(i) \cap \bar{S}} \varphi_{ij} - \sum_{j \in \Gamma^{-1}(i) \cap S} \varphi_{ji} - \sum_{j \in \Gamma^{-1}(i) \cap \bar{S}} \varphi_{ji} \right)$$

En simplifiant nous obtenons : $V(\varphi) = \sum_{i \in S, j \in \Gamma(i) \cap \bar{S}} \varphi_{ij} - \sum_{i \in S, j \in \Gamma^{-1}(i) \cap \bar{S}} \varphi_{ji}$ □

Bien fondé de l'algorithme de Ford-Fulkerson (suite)

Propriété.

Pour toute coupe $K = \{S, \bar{S}\}$ et tout flot φ , on a :

$$c(K) \geq V(\varphi)$$

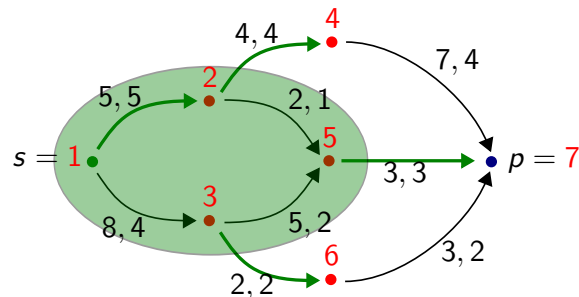
Démonstration.

Pour tout arc $(i, j) \in U$, nous avons : $0 \leq \varphi_{ij} \leq c_{ij}$ d'où :

$$\begin{aligned} c(K) &= \sum_{i \in S, j \notin S} c_{ij} \\ &\geq \sum_{i \in S, j \notin S} \varphi_{ij} \\ &\geq \sum_{i \in S, j \notin S} \varphi_{ij} - \sum_{i \in S, j \notin S} \varphi_{ji} \\ &\geq V(\varphi) \end{aligned}$$

□

Illustration sur l'exemple précédent



- $S = \{1, 3, 5, 2\}$ et $\bar{S} = \{4, 6, 7\}$ (lors de la 5ème itération)
- Tout $(i, j) \in U$ tel que $i \in S \wedge j \in \bar{S}$ sont saturés (car sinon $j \in \bar{S}$ aurait été marqué)
- Tout $(j, i) \in U$ tel que $i \in S \wedge j \in \bar{S}$ sont tels que $\varphi_{ji} = 0$ (idem)
- Des observations précédentes, nous en déduisons que

$$c(\{S, \bar{S}\}) = \sum_{i \in S, j \in \bar{S}} c_{ij} = \sum_{i \in S, j \in \bar{S}} \varphi_{ij} - \sum_{i \in S, j \in \bar{S}} \varphi_{ji} = V(\varphi)$$

Bien fondé de l'algorithme de Ford-Fulkerson (suite)

- Le résultat théorique sur lequel repose la recherche d'un flot de valeur maximale est le **théorème de Ford-Fulkerson** :

Théorème.

Soit $G = [X, U, C]$ un réseau avec capacités, soit \mathbb{F} la famille des flots réalisables de G et soit \mathbb{K} la famille des coupes dans G . Notons :

- ▶ $V_M = V(\varphi^*) = \max_{\varphi \in \mathbb{F}} \{V(\varphi)\}$
- ▶ $c_m = \min_{K \in \mathbb{K}} \{c(K)\}$

Alors $V_M = c_m$.