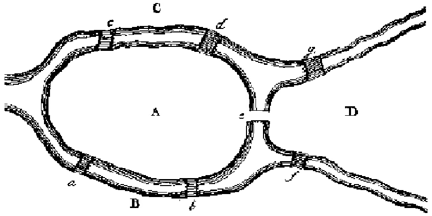
Optimisation et recherche opérationnelle
Partie 2
Eléments de théorie des graphes



Didier MAQUIN

Professeur
Ecole Nationale Supérieure d'Electricité et de Mécanique
Institut National Polytechnique de Lorraine
+
Ecole Nationale Supérieure des Mines de Nancy

Thème de recherche
Surveillance et diagnostic
de fonctionnement des processus

Contact : bureau 116 jaune
03 83 59 56 83
Didier.Maquin@ensem.inpl-nancy.fr

Auto-apprentissage

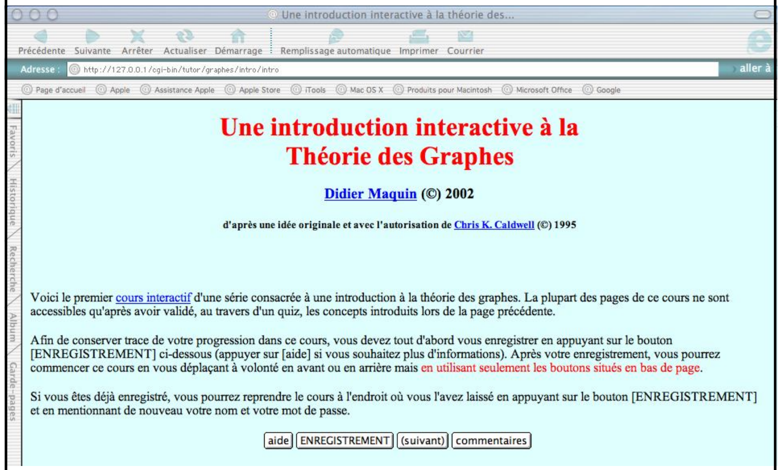
4 séances de 20 mn

d'auto-formation à l'aide d'un cours interactif accessible par le Web

http://cours.ensem.inpl-nancy.fr/cours-dm

ATTENTION

- **Chaque cours nécessite un enregistrement**
  (choisir le même login et le même mot de passe pour toutes les leçons)
- **Le login doit être construit selon : GSI*nom***
  (pour moi, cela serait GSIMAQUIN)



Une introduction interactive à la théorie des...

Précédente Suivante Arrêter Actualiser Démarrage Remplissage automatique Imprimer Courrier

Adresse : http://127.0.0.1/cgi-bin/tutor/graphes/intro/intro aller à

Page d'accueil Apple Assistance Apple Apple Store iTools Mac OS X Produits pour Macintosh Microsoft Office Google

Favoris Historique Recherche Album Garde-pages

Une introduction interactive à la Théorie des Graphes

Didier Maquin (©) 2002

d'après une idée originale et avec l'autorisation de Chris K. Caldwell (©) 1995

Voici le premier cours interactif d'une série consacrée à une introduction à la théorie des graphes. La plupart des pages de ce cours ne sont accessibles qu'après avoir validé, au travers d'un quiz, les concepts introduits lors de la page précédente.

Afin de conserver trace de votre progression dans ce cours, vous devez tout d'abord vous enregistrer en appuyant sur le bouton [ENREGISTREMENT] ci-dessous (appuyer sur [aide] si vous souhaitez plus d'informations). Après votre enregistrement, vous pourrez commencer ce cours en vous déplaçant à volonté en avant ou en arrière mais en utilisant seulement les boutons situés en bas de page.

Si vous êtes déjà enregistré, vous pourrez reprendre le cours à l'endroit où vous l'avez laissé en appuyant sur le bouton [ENREGISTREMENT] et en mentionnant de nouveau votre nom et votre mot de passe.

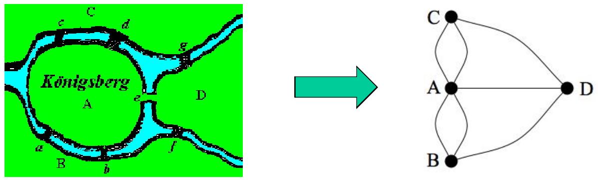
aide | ENREGISTREMENT | (suivant) | commentaires

Plan du cours

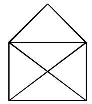
- Bref historique
- Introduction
- Modes de représentation d'un graphe
- Etude de la connexité
- Parcours eulériens et hamiltoniens
- Méthodes de recherche de chemins
- Réseaux, réseaux de transport et problèmes de flots
- Couplages - problèmes d'affectation
- Problèmes d'ordonnancement
- Graphes planaires
- Coloration d'un graphe

Bref historique de la théorie des graphes

- **Naissance en 1736, communication d'Euler où il propose une solution au problème des ponts de Königsberg**

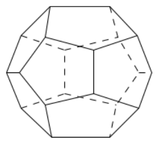


Dessin comportant des sommets (points) et des arêtes reliant ces sommets



Bref historique de la théorie des graphes

- **1847 Kirchhoff**
  théorie des arbres (analyse de circuits électriques)
- **1860 Cayley**
  énumération des isomères saturés des hydrocarbures C_nH_{2n+2}
- **A la même époque, énoncé de problèmes importants**
  - Conjecture des quatre couleurs (1879)
    (Möbius, De Morgan, Cayley, solution trouvée en 1976)
  - Existence de chemins Hamiltoniens (1859)
- **1936 König**
  premier ouvrage sur les graphes
- **1946 ⇒ Kuhn, Ford, Fulkerson, Roy**
  Berge



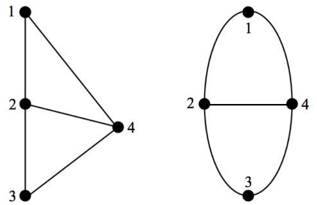
Introduction

Qu'est-ce qu'un graphe ?

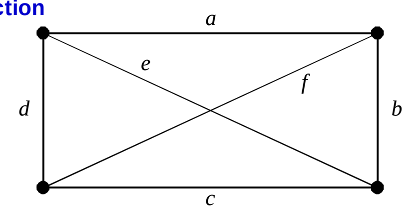
- ***Définition 1***
  - On appelle graphe $G=(X, A)$ la donnée d'un ensemble X dont les éléments sont appelés sommets et d'une partie de A symétrique $(x, y) \in A \Leftrightarrow (y, x) \in A$ dont les éléments sont appelés arêtes.
  - En présence d'une arête $a=(x,y)$ qui peut être notée simplement xy, on dit que x et y sont les extrémités de a, que a est incidente en x et en y, et que y est un successeur ou voisin de x (et vice versa).
  - On dit qu'un graphe est sans boucle si A ne contient pas d'arête de la forme (x, x), c'est-à-dire joignant un sommet à lui-même.
  - Le nombre de sommets est appelé ordre du graphe.
  - Un graphe ne possédant pas de boucle ni d'arêtes parallèles (deux arêtes distinctes joignant la même paire de sommets) est appelé graphe simple ou 1-graphe.

Introduction

Deux graphes identiques



Arêtes *e* et *f* sans intersection



Graphes non-orientés

Introduction

- ***Définition 2***
  - On appelle graphe orienté ou digraphe $G=(X, A)$ la donnée d'un ensemble X dont les éléments sont appelés sommets et d'une partie de A de $X \times X$ dont les éléments sont appelés arcs ou arêtes.
  - En présence d'un arc $a=(x, y)$ qui peut être noté simplement xy, on dit que x est l'origine (ou extrémité initiale) et y l'extrémité (terminale) de a, que a est sortant en x et incident en y. On dit aussi que x et y sont adjacents.

Introduction

Graphes et applications multivoques

- L'ensemble des successeurs d'un sommet $x \in X$ est noté $\Gamma(x)$.
- L'application Γ qui, à tout élément de X, fait correspondre une partie de X (un élément de $\mathcal{P}$(X)) est appelée une application multivoque.
- L'ensemble des prédécesseurs d'un sommet $x \in X$ peut alors être noté $\Gamma^{-1}(x)$ où Γ^{-1} est l'application (multivoque) réciproque de Γ.
- Si le graphe G est un 1-graphe, on constate qu'il est parfaitement déterminé par la donnée de l'ensemble X et de l'application multivoque Γ de $X \rightarrow \mathcal{P}$(X). Un tel graphe peut donc aussi être noté : $G=(X, \Gamma)$.

Introduction

Principales définitions (contexte graphes orientés)

- ***Définition 3***
  - On appelle degré sortant ou demi-degré extérieur d'un sommet x le nombre d'arcs de la forme $a=(x, y)$ avec $y \neq x$, c'est-à-dire le nombre d'éléments de $\Gamma(x) \setminus \{x\}$. On note $d_s(x)$ ce degré.
  - On appelle degré entrant ou demi-degré intérieur d'un sommet x le nombre d'arcs de la forme $a=(y, x)$ avec $y \neq x$, c'est-à-dire le nombre d'éléments de $\Gamma^{-1}(x) \setminus \{x\}$. On note $d_e(x)$ ce degré.
  - On appelle degré de x (ou valence) la somme du degré entrant et du degré sortant.
  - Un sommet de degré entrant non nul et de degré sortant nul est appelé puits, tandis qu'un sommet de degré entrant nul et de degré sortant non nul est appelé source.
  - Un sommet n'ayant pas d'arcs incidents est appelé sommet isolé ; ces sommets ont un degré nul

Introduction

Principales définitions (contexte graphes orientés)

- ***Définition 4***
  - On appelle graphe réflexif un graphe possédant une boucle sur chaque sommet.
  - Un graphe est symétrique si, pour tout arc $a_1=(x, y)$ appartenant à A, l'arc $a_2=(y, x)$ appartient également à A.
  - Un graphe est antisymétrique si, pour tout arc $a_1=(x, y)$ appartenant à A, l'arc $a_2=(y, x)$ n'appartient pas à A.
  - Enfin, un graphe est transitif si, quelque soit deux arcs adjacents $a_1=(x, y)$ et $a_2=(y, z)$ appartenant à A, alors l'arc $a_3=(x, z)$ appartient également à A.

Le concept de graphe symétrique est très proche de celui des graphes non orientés. En fait, à tout graphe symétrique, on peut associer un graphe non orienté en substituant aux arcs $a_1=(x, y)$ et $a_2=(y, x)$, une arête $a=(y, x)$.

Introduction

Principales définitions (contexte graphes orientés)

- ***Définition 5***
  - Un graphe $G=(X, A)$ est dit complet si, pour toute paire de sommets (x, y), il existe au moins un arc de la forme (x, y) ou (y, x).
  - Un graphe simple complet d'ordre n est noté K_n. Un sous-ensemble de sommets $C \subset X$ tel que deux sommets quelconques de C sont reliés par une arête est appelé une clique.
- ***Définition 6***
  - Soit un graphe $G=(X, A)$ et $X' \subset X$. Le sous-graphe engendré par X' est $G'=(X', A')$, A' étant formé des arêtes dont les deux extrémités sont dans X'.
  - Si l'on se donne un sous-ensemble A_1 de A, le graphe partiel engendré par A_1 est $G_1=(X, A_1)$.

D'après la définition précédente, une clique d'un graphe G est donc un sous-graphe complet de G.

Modes de représentation d'un graphe

- **Matrice d'adjacence**

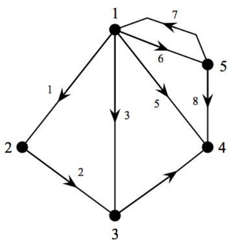
  Utilisation des outils d'algèbre linéaire
- ***Définition 7***

  Considérons un graphe $G=(X, A)$ comportant n sommets. La matrice d'adjacence de G est égale à la matrice $U=(u_{ij})$ de dimension $n \times n$ telle que :

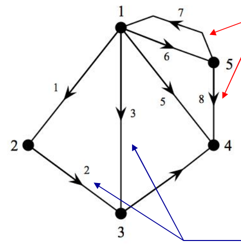
$$u_{ij} = \begin{cases} 1 & \text{si } (i,j) \in A \text{ (c'est-à-dire } (i,j) \text{ est une arête)} \\ 0 & \text{sinon} \end{cases}$$

Une telle matrice, ne contenant que des « 0 » et des « 1 » est appelée matrice booléenne.

$$U = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



Modes de représentation d'un graphe



$$\begin{array}{cc} U = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} & \to 2 \\ \quad\quad\quad\downarrow 2 & \end{array}$$

Propriétés

- la somme des éléments de la $i^{ème}$ ligne de U est égale au degré sortant $d_s(x_i)$ du sommet x_i de G.
- la somme des éléments de la $j^{ème}$ colonne de U est égale au degré entrant $d_e(x_j)$ du sommet x_j de G.
- U est symétrique si, et seulement si, le graphe G est symétrique.

Modes de représentation d'un graphe

- **Matrice d'incidence**

  Incidence entre arêtes et sommets

- ***Définition 8***

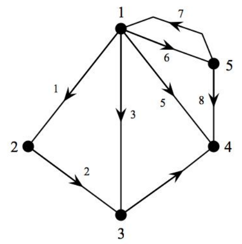
  Considérons un graphe orienté sans boucle $G=(X, A)$ comportant n sommets $x_1, \ldots, x_n$ et m arêtes $a_1, \ldots, a_m$. On appelle matrice d'incidence (aux arcs) de G la matrice $M=(m_{ij})$ de dimension $n \times m$ telle que :

$$m_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est l'extrémité initiale de } a_j \\ -1 & \text{si } x_i \text{ est l'extrémité terminale de } a_j \\ 0 & \text{si } x_i \text{ n'est pas une extrémité de } a_j \end{cases}$$

  Pour un graphe non orienté sans boucle, la matrice d'incidence (aux arêtes) est définie par :

$$m_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est une extrémité de } a_j \\ 0 & \text{sinon} \end{cases}$$

Modes de représentation d'un graphe



$$\begin{array}{cc} & \begin{array}{cccccccc} \text{Arc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ M = & \begin{pmatrix} -1 & 0 & -1 & 0 & -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{pmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \\ & \text{Sommet} \end{array}$$

Etude de la connexité

Chaînes et cycles, élémentaires et simples

- ***Définition*** **9**
  - Une chaîne est une séquence finie et alternée de sommets et d'arêtes, débutant et finissant par des sommets, telle que chaque arête est incidente avec les sommets qui l'encadre dans la séquence.
  - Le premier et le dernier sommet sont appelés (sommets) extrémités de la chaîne.
  - La longueur de la chaîne est égale au nombre d'arêtes qui la composent.
  - Si aucun des sommets composant la séquence n'apparaît plus d'une fois, la chaîne est dite chaîne élémentaire.
  - Si aucune des arêtes composant la séquence n'apparaît plus d'une fois, la chaîne est dite chaîne simple.

Etude de la connexité

Chaînes et cycles, élémentaires et simples

- Un cycle est une chaîne dont les extrémités coïncident.
- Un cycle élémentaire (tel que l'on ne rencontre pas deux fois le même sommet en le parcourant) est un cycle minimal pour l'inclusion, c'est-à-dire ne contenant strictement aucun autre cycle.

Etude de la connexité

Chemins et circuits, élémentaires et simples

➢ ***Définition* 10**

- Un chemin est une séquence finie et alternée de sommets et d'arcs, débutant et finissant par des sommets, telle que chaque arc est sortant d'un sommet et incident au sommet suivant dans la séquence (cela correspond à la notion de chaîne « orientée »).
- Si aucun des sommets composant la séquence n'apparaît plus d'une fois, le chemin est dit chemin élémentaire.
- Si aucune des arêtes composant la séquence n'apparaît plus d'une fois, le chemin est dit chemin simple.
- Un circuit est un chemin dont les extrémités coïncident.
- En parcourant un circuit élémentaire, on ne rencontre pas deux fois le même sommet.

Etude de la connexité

Graphes et sous-graphes connexes

➢ ***Définition* 11**

Un graphe G est connexe s'il existe au moins une chaîne entre une paire quelconque de sommets de G.

La relation :

$$x_i \; \mathrm{R} \; x_j \Leftrightarrow \begin{cases} \text{soit } \; x_i = x_j \\ \text{soit il existe une chaîne joignant } \; x_i \text{ à } x_j \end{cases}$$

est une relation d'équivalence (réflexivité, symétrie, transitivité). Les classes d'équivalence induites sur X par cette relation forment une partition de X en $X_1, X_2, \ldots X_p$.

Le nombre p de classes d'équivalence distinctes est appelé nombre de connexité du graphe.

Un graphe est dit connexe si et seulement si son nombre de connexité est égal à 1.

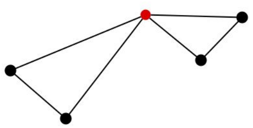
Les sous-graphes G_i engendrés par les sous-ensembles X_i sont appelés les composantes connexes du graphe. Chaque composante connexe est un graphe connexe.

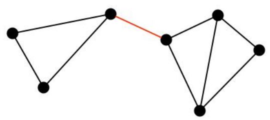
Etude de la connexité

Graphes et sous-graphes connexes

➢ ***Définition* 12**

- Un point d'articulation d'un graphe est un sommet dont la suppression augmente le nombre de composantes connexes.
- Un isthme est une arête dont la suppression a le même effet.
- Un ensemble d'articulation d'un graphe connexe G est un ensemble de sommets tel que le sous-graphe G' déduit de G par suppression des sommets de E, ne soit plus connexe.





Etude de la connexité

Graphes et sous-graphes fortement connexes

➢ ***Définition* 13**

Un graphe orienté est fortement connexe s'il existe un chemin joignant deux sommets quelconque.

La relation :

$$x_i \; \mathrm{R} \; x_j \Leftrightarrow \begin{cases} \text{soit } \; x_i = x_j \\ \text{soit il existe à la fois un chemin joignant } \; x_i \text{ à } x_j \text{ et un chemin joignant } \; x_j \text{ à } x_i \end{cases}$$

est une relation d'équivalence et les classes d'équivalence induites sur X par cette relation forment une partition de X en $X_1, X_2, \ldots X_q$.

Les sous-graphes engendrés par les sous-ensembles $G_1, G_2, \ldots G_q$ sont appelés les composantes fortement connexes du graphe.

Etude de la connexité

Graphes et sous-graphes fortement connexes

- ***Définition* 14**

  On appelle graphe réduit le quotient du graphe G par la relation de forte connexité $G_r = G/R$

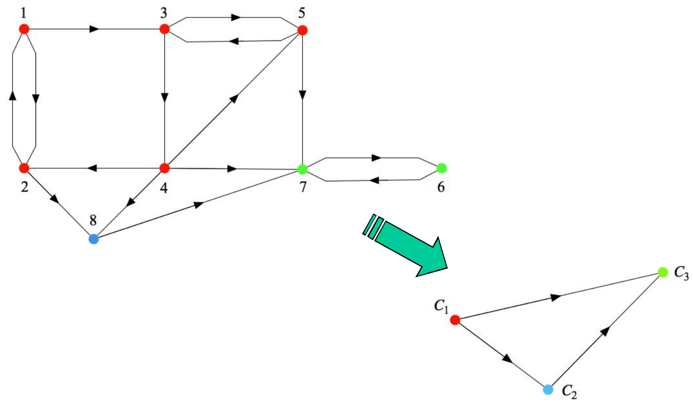
  Les sommets de G_r sont donc les composantes fortement connexes et il existe un arc entre deux composantes fortement connexes si et seulement s'il existe au moins un arc entre un sommet de la première composante et un sommet de la seconde.

  On vérifie que le graphe G_r est sans circuit.

25

Etude de la connexité

Graphes et sous-graphes fortement connexes



26

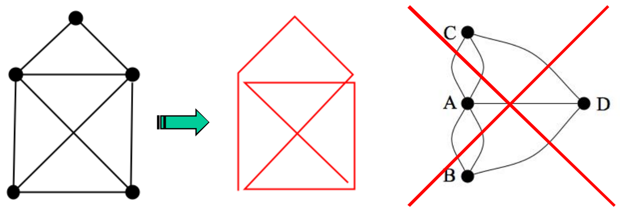
Parcours eulériens et hamiltoniens

Chaînes et cycles eulériens

- ***Définition* 16**

  Soit un graphe orienté $G=(X, A)$.

  - Une chaîne eulérienne est une chaîne empruntant une fois et une fois seulement chaque arête de G.
  - Un cycle eulérien est une chaîne eulérienne dont les extrémités coïncident.
  - Un graphe possédant un cycle eulérien est appelé graphe eulérien.



27

Parcours eulériens et hamiltoniens

Chaînes et cycles eulériens

- ***Théorème* 1**

  - Un graphe non orienté connexe possède une chaîne eulérienne si et seulement si le nombre de sommets de degré impair est égal à 0 ou 2.
  - Il admet un cycle eulérien si et seulement si tous ses sommets ont un degré pair.

Condition nécessaire

En chaque sommet, le nombre d'arcs incidents doit être égal au nombre d'arcs sortants, les sommets doivent donc être de degré pair.

Dans le cas d'une chaîne, les deux extrémités font exception ; on part ou on arrive une fois de plus, d'où un degré impair pour ces deux sommets extrémités.

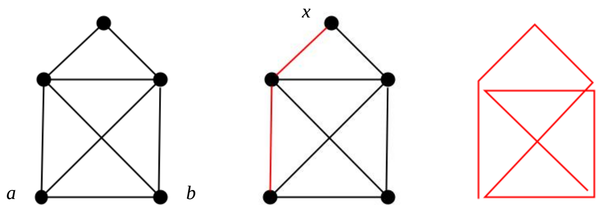
28

Parcours eulériens et hamiltoniens

Chaînes et cycles eulériens

Condition nécessaire

Soit a et b deux sommets de degré impair (s'il n'y en a pas a et b sont confondus).
Soit L la chaîne parcourue en partant de a (avec l'interdiction d'emprunter deux fois la même arête).
Si l'on arrive à un sommet $x \neq b$, on a utilisé un nombre impair d'arêtes incidentes à x. On peut donc « repartir » par une arête non déjà utilisée.
Quand on ne peut plus bouger, c'est qu'on est en b. Si toutes les arêtes ont été utilisées, on a parcouru une chaîne eulérienne.

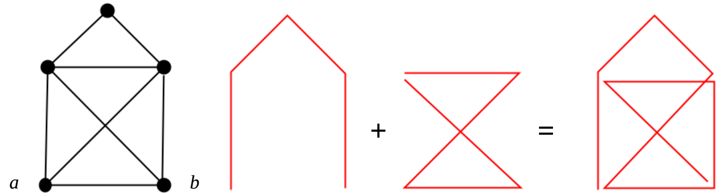


Parcours eulériens et hamiltoniens

Chaînes et cycles eulériens

Condition nécessaire

Si toutes les arêtes n'ont pas été utilisées, on greffe, dans la chaîne, des cycles eulériens



Parcours eulériens et hamiltoniens

Chemins et circuits eulériens

- ***Définition* 17**

  Soit un graphe orienté $G=(X, A)$.

  - Un chemin dans un graphe orienté est dit eulérien s'il passe exactement une fois par chaque arête.
  - Un graphe orienté est dit eulérien s'il admet un circuit eulérien.

- ***Théorème* 2**

  - Un graphe orienté connexe admet un chemin eulérien (mais pas de circuit eulérien) si, et seulement si, pour tout sommet sauf deux (a et b), le degré entrant est égal au degré sortant et

    $d_e(a)=d_s(a)-1$ et $d_e(b)=d_s(b)+1$

  - Un graphe orienté connexe admet un circuit eulérien si, et seulement si, pour tout sommet, le degré entrant est égal au degré sortant.

Parcours eulériens et hamiltoniens

Problèmes « prototypes » classiques : problème du postier chinois

parcourir les rues d'une ville en passant <u>au moins</u> une fois dans chaque rue, le graphe n'étant pas nécessairement eulérien ; on cherche bien sûr à minimiser la longueur totale du parcours.

Application : tournées de distribution de courrier, de ramassage d'ordures, d'inspection de réseaux de distribution.

Dans le cas orienté où chaque arc doit être emprunté dans un sens privilégié, le problème se ramène à la recherche d'un flot à coût minimum.

- ***Théorème* 3**

  - Un graphe non orienté admet un cycle chinois si, et seulement si, il est connexe.
  - Un graphe orienté admet un circuit chinois si, et seulement si, il est fortement connexe.

Parcours eulériens et hamiltoniens

Chaînes et cycles hamiltoniens

- ***Définition* 18**

  Soit G=(X, A) un graphe connexe d 'ordre n.

  - On appelle chemin hamiltonien (chaîne hamiltonienne) un chemin (une chaîne) passant une fois, et une fois seulement, par chacun des sommets de G.
  - Un chemin hamiltonien (une chaîne hamiltonienne) est donc un chemin (une chaîne) élémentaire de longueur n–1.
  - Un circuit hamiltonien (un cycle hamiltonien) est un circuit (un cycle) qui passe une fois, et une seule fois, par chacun des sommets de G.
  - On dit qu'un graphe G est hamiltonien s'il contient un cycle hamiltonien (cas non orienté) ou un circuit hamiltonien (cas orienté).

Parcours eulériens et hamiltoniens

Chaînes et cycles hamiltoniens : exemples classiques

- **Problème du voyageur de commerce**
  - Un représentant de commerce doit rendre visite à n clients x_1, x_2, … x_n en partant d'une ville x_0 et revenir à son point de départ. Il connaît des distances d_{0j} qui séparent le dépôt x_0 de chacun de ses clients, ainsi que la distance d_{ij} entre deux clients quelconques x_i et x_j.
  - Dans quel ordre doit-il rendre visiste à ses clients pour que la distance totale parcourue soit minimale ?
  - Ce problème revient à chercher un cycle hamiltonien de longueur totale minimale dans le graphe complet G construit sur l'ensemble des sommets, les arêtes étant munies des longueurs d_{ij}.
  - Lorsque le point d'arrivée est différent du point de départ, le problème revient à rechercher une chaîne hamiltonienne de longueur totale minimale.

Parcours eulériens et hamiltoniens

Chaînes et cycles hamiltoniens : exemples classiques

- **Ordonnancement de tâches**
  - On cherche un ordre dans lequel on peut effectuer n tâches données (deux tâches quelconques ne pouvant être effectuées simultanément) tout en respectant un certain nombre de contraintes d'antériorité.
  - Si l'on construit le graphe G dont l'ensemble des sommets correspond à l'ensemble des tâches, et où il existe un arc (i, j) si la tâche i peut être effectuée avant la tâche j, le problème revient à déterminer un chemin hamiltonien de G.

Chaînes et cycles hamiltoniens : extension

On appelle cycle (circuit) préhamiltonien d'un graphe G, un cycle (un circuit) passant au moins une fois par chaque sommet de G. Un graphe G qui admet un tel cycle (ou circuit) est appelé graphe préhamiltonien et une condition nécessaire et suffisante pour qu'il en soit ainsi est que G soit connexe (fortement connexe).

Méthodes de recherche de chemins

Le problème du plus court chemin

- **Nombreuses applications**
  - les problèmes de tournées,
  - certains problèmes d'investissement et de gestion de stocks,
  - les problèmes de programmation dynamique à états discrets et temps discret,
  - les problèmes d'optimisation de réseaux (routiers, télécommunications),
  - certaines méthodes de traitement numérique du signal, de codage et de décodage de l'information
  - les problèmes de labyrinthe et de récréations mathématiques.

Méthodes de recherche de chemins

Le problème du plus court chemin

- **Position du problème**
  - Graphe orienté $G=(X, A)$
  - Pour $a=(i,j) \in A$, on associe $l(a)=l_{ij} \in \mathbf{R}$ (longueur de l'arc)
  - Problème du plus court chemin entre deux sommets i et j

    trouver un chemin $\mu(i,j)$ de i à j dont la longueur totale

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

Longueur : coût de transport, dépense de construction, temps de parcours, etc.

37

Méthodes de recherche de chemins

Le problème du plus court chemin

- **Algorithme itératif**
  - L'ensemble des sommets est partitionné en deux sous-ensembles S et $\overline{S}$
  - Le sous-ensemble S (initialisé à $\{1\}$) contient les sommets définitivement marqués, c'est-à-dire les sommets pour lesquels la marque $\pi(i)$ représente effectivement la longueur du plus court chemin entre le sommet 1 et le sommet i.
  - Le complémentaire contient tous les sommets k ayant une marque provisoire définie par :

$$\forall k \in \overline{S} : \pi(k) = \min_{i \in S \cap \Gamma_k^{-1}} \left(\pi(i) + l_{ik}\right)$$

  - Le sommet j de marque provisoire minimale est marqué définitivement.

$$\pi(j) = \min_{k \in \overline{S}} \left(\pi(k)\right)$$

38

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives

- Initialisations

$$\overline{S} = \{2,\ 3,\ \ldots,\ n\} \qquad \pi(1) = 0 \qquad \pi(i) = \begin{cases} l_{ij} & \text{si } i \in \Gamma(1) \\ +\infty & \text{sinon} \end{cases}$$

- Sélectionner j tel que pour $k \in \overline{S}$ $\quad \pi(j) = \min(\pi(k))$

  Faire $\overline{S} \leftarrow \overline{S} \setminus \{j\}$

  Si $|\overline{S}| = 0$ alors FIN

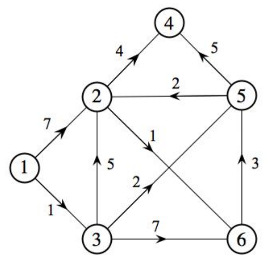
- Faire pour tout $i \in \overline{S} \cap \Gamma(j)$

  $\pi(i) \leftarrow \min\left(\pi(i), \pi(j) + l_{ji}\right)$

39

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives

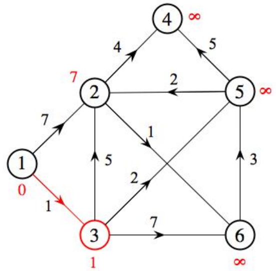


$$\overline{S} = \{2,3,4,5,6\}, \quad \pi(1) = 0, \pi(2) = 7, \pi(3) = 1, \pi(4) = \infty, \pi(5) = \infty, \pi(6) = \infty$$

40

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives

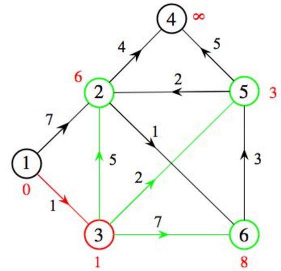


$$\bar{S} = \{2,3,4,5,6\}, \quad \pi(1)=0, \pi(2)=7, \pi(3)=1, \pi(4)=\infty, \pi(5)=\infty, \pi(6)=\infty$$

$$j=3, \quad \bar{S} = \{2,4,5,6\}$$

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives



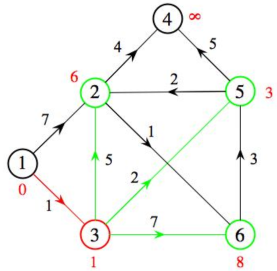
$$\bar{S} = \{2,3,4,5,6\}, \quad \pi(1)=0, \pi(2)=7, \pi(3)=1, \pi(4)=\infty, \pi(5)=\infty, \pi(6)=\infty$$

$$j=3, \quad \bar{S} = \{2,4,5,6\}$$

$$\bar{S} \cap \Gamma(3) = \{2,5,6\}, \quad \pi(2)=\min(7,1+5)=6, \pi(5)=\min(\infty,1+2)=3, \ \pi(6)=\min(\infty,1+7)=8$$

Méthodes de recherche de chemins

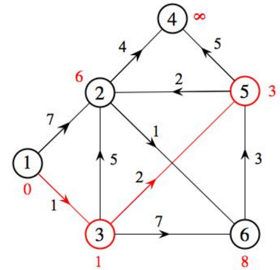
Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives



$$\bar{S} = \{2,4,5,6\}, \quad \pi(1)=0, \pi(2)=6, \pi(4)=\infty, \pi(5)=3, \pi(6)=8$$

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives

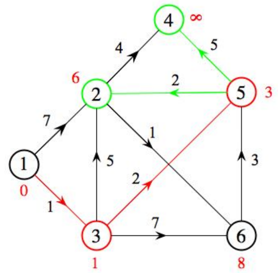


$$\bar{S} = \{2,4,5,6\}, \quad \pi(1)=0, \pi(2)=6, \pi(4)=\infty, \pi(5)=3, \pi(6)=8$$

$$j=5, \quad \bar{S} = \{2,4,6\}$$

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives



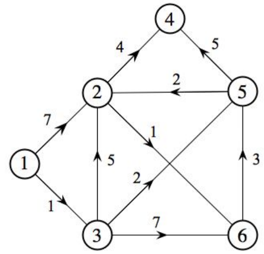
$$\bar{S} = \{2,4,5,6\}, \quad \pi(1)=0, \pi(2)=6, \pi(4)=\infty, \pi(5)=3, \pi(6)=8$$

$$j = 5, \quad \bar{S} = \{2,4,6\}$$

$$\bar{S} \cap \Gamma(5) = \{2,4\}, \quad \pi(2)=\min(6,3+2)=5, \pi(4)=\min(\infty,3+5)=8$$

Méthodes de recherche de chemins

Algorithme 1 (Moore-Dijkstra)
Plus court chemin entre deux sommets dans un graphe à longueurs positives



$$\pi(1)=0, \pi(2)=5, \pi(3)=1, \pi(4)=8, \pi(5)=3, \pi(6)=6$$

Réseaux, réseaux de transport et flots

➢ ***Définition* 22**

- Un graphe fortement connexe, sans boucle et ayant plus d'un sommet, est appelé un réseau.
- On appelle nœud d'un réseau un sommet qui a plus de deux arcs incidents. Les autres sommets sont appelés antinœuds.
- On appelle branche tout chemin pour lequel seuls les premiers et derniers sommets sont des nœuds.
- Dans un graphe orienté G, un flot est l'affectation d'une valeur réelle à chaque arc de G, représentant une quantité transportée sur cet arc, de telle sorte que, en chaque sommet, la somme des flots entrants soit égale à la somme des flots sortants (loi de Kirchhoff : conservation des flux en chaque sommet).

Problème classique : recherche d'un flot maximal.

On se donne une capacité maximale sur chaque arc. Le problème du flot maximal consiste à déterminer un flot dont la valeur en un certain lieu est maximale. On peut, de plus, se donner un coût de transport d'une unité de flot sur chaque arc et chercher le flot maximal de coût minimal.

Réseaux, réseaux de transport et flots

➢ ***Définition* 23**

- On appelle réseau de transport un graphe orienté antisymétrique valué $G=(X, A, C)$, sans boucle et dans lequel il existe :

  – un sommet x_1 sans prédécesseur nommé entrée ou source du réseau,
  – un sommet x_n sans successeur nommé sortie ou puits du réseau,

  et tel qu'au moins un chemin unisse x_1 à x_n dans G.

- La fonction de pondération C est supposée positive et l'on nomme capacité de l'arc a le nombre $C(a)$.

➢ ***Définition* 24**

- Une fonction $\varphi(a)$ définie sur A et à valeurs réelles est un flot pour le réseau de transport si :

  – il est positif : $\varphi(a) > 0, \quad \forall\, a \in A$
  – il vérifie la loi des nœuds de Kirchhoff $\forall\, x \neq x_1$ et $x \neq x_2$

- Il ne dépasse pas la capacité des arcs : $\varphi(a) \leq C(a), \forall\, a \in A$.

Réseaux, réseaux de transport et flots

Si x n'est ni x_1 ni x_n, la quantité entrante en x doit être égale à la quantité sortante que nous désignons par :

$$\varphi_x = \sum_{a \in A_x^-} \varphi(a) = \sum_{a \in A_x^+} \varphi(a)$$

Si φ est un flot sur un réseau de transport G, alors on a $\varphi_{x1} = \varphi_{xn}$
Cette quantité s'appelle la valeur du flot.

Recherche d'un flot complet

➢ ***Définition* 25**

- Pour un flot φ dans un réseau de transport $G=(X, A, C)$, on dit qu'un arc est saturé si l'on a $\varphi(a) = C(a)$.
- Le flot est dit complet si tout chemin allant de x_1 à x_n contient au moins un arc saturé.

Réseaux, réseaux de transport et flots

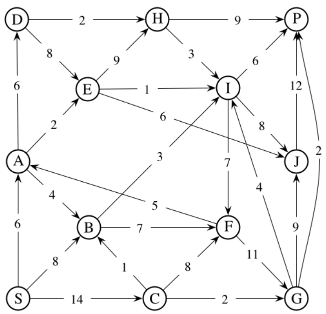
Recherche d'un flot complet

- On considère le graphe partiel engendré par les arcs non saturés par le flot.
- Si le flot n'est pas complet, il existe nécessairement un chemin μ allant de l'entrée à la sortie.
- On peut alors définir un nouveau flot pour le réseau en augmentant de 1 le flot de chacun des arcs constituant le chemin μ.
- La valeur du flot est alors également augmentée de 1.

- On peut donc progressivement augmenter la valeur d'un flot incomplet jusqu'à ce qu'il soit complet.
- En tenant compte des différences entre les capacités et la valeur du flot sur les arcs de μ, on peut connaître d'avance l'augmentation possible du flot.
- Cependant, le flot complet ainsi obtenu n'est pas, en général, le flot maximal.

Réseaux, réseaux de transport et flots

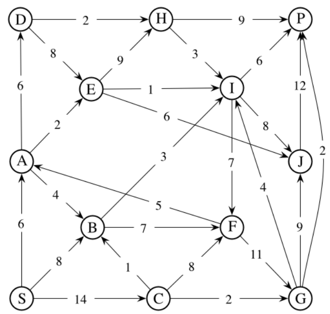
Recherche d'un flot complet : exemple



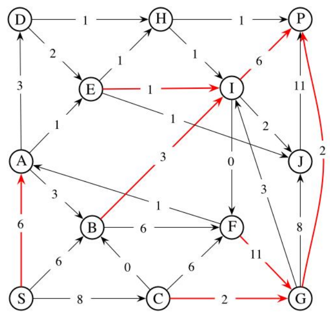
Capacités

Réseaux, réseaux de transport et flots

Recherche d'un flot complet : exemple



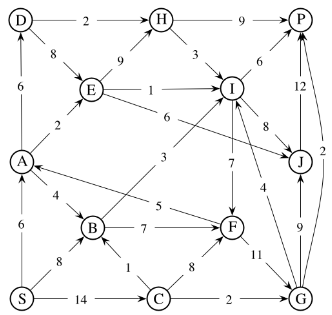
Capacités



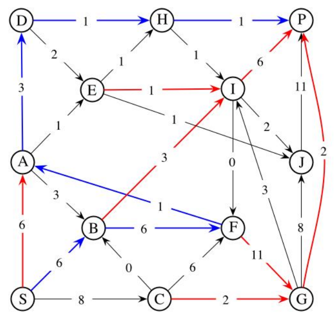
Flot au jugé

Réseaux, réseaux de transport et flots

Recherche d'un flot complet : exemple



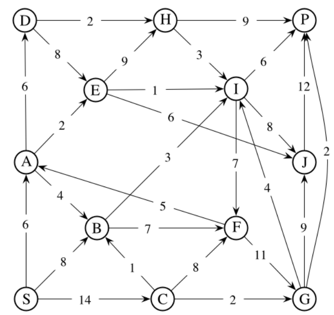
Capacités



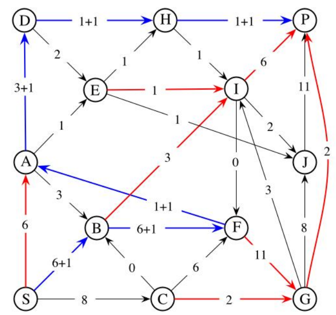
Amélioration du flot

Réseaux, réseaux de transport et flots

Recherche d'un flot complet : exemple



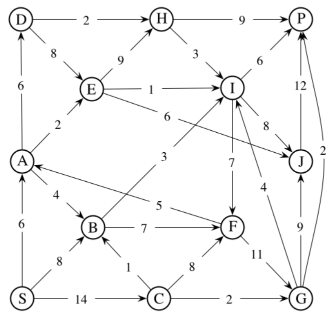
Capacités



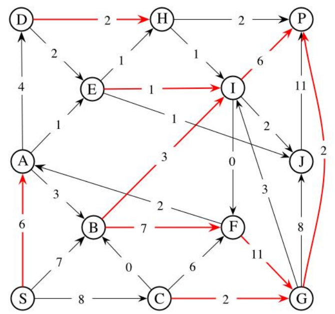
Amélioration du flot

Réseaux, réseaux de transport et flots

Recherche d'un flot complet : exemple



Capacités



Flot amélioré

Réseaux, réseaux de transport et flots

Amélioration du flot

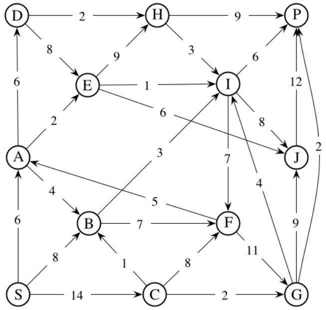
- Processus d'étiquetage à partir du flot complet
  On place une étiquette $\boxed{+}$ en x_1 ; soit x un sommet déjà marqué
  - On marque avec une étiquette $\boxed{+x}$ tout successeur y non marqué de x pour lequel le flot n'est pas à son maximum
  - On marque avec une étiquette $\boxed{-x}$ tout prédécesseur y non marqué de x pour lequel le flot n'est pas nul
- Rôle de l'étiquette
  indiquer si le flot peut être augmenté dans le sens de parcours (étiquette $\boxed{+x}$) ou diminué s'il est dans le sens contraire (étiquette $\boxed{-x}$).

  Si l'on parvient jusqu'au marquage du sommet x_n avec cette procédure, c'est qu'il existe une chaîne μ de x_1 à x_n dont tous les sommets sont marqués avec l'indice du sommet précédent au signe près.

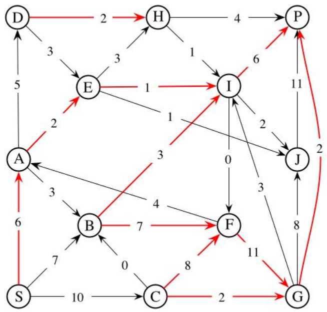
  Le flot peut alors être augmenté d'une unité.

Réseaux, réseaux de transport et flots

Recherche d'un flot maximal : exemple



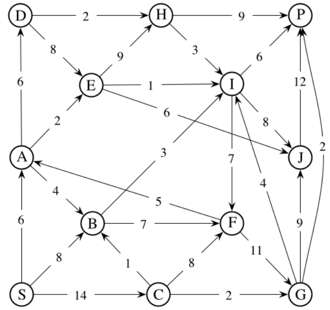
Capacités



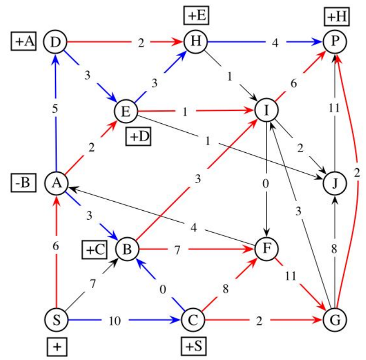
Flot complet

Réseaux, réseaux de transport et flots

Recherche d'un flot maximal : exemple



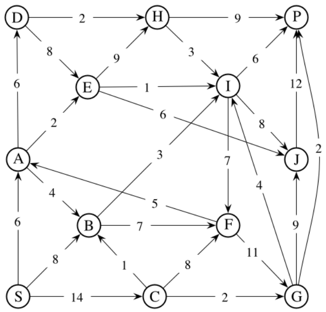
Capacités



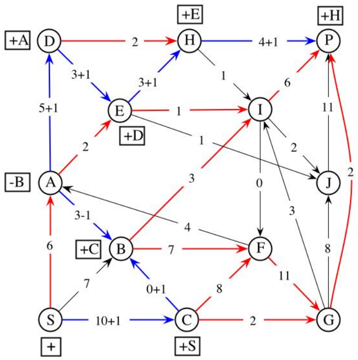
Amélioration du flot

Réseaux, réseaux de transport et flots

Recherche d'un flot maximal : exemple



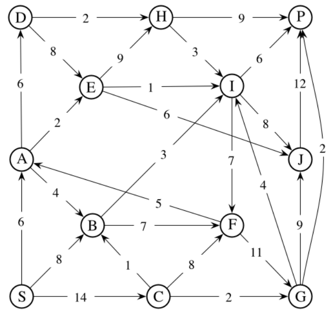
Capacités



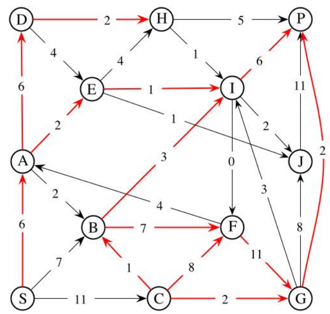
Amélioration du flot

Réseaux, réseaux de transport et flots

Recherche d'un flot maximal : exemple



Capacités



Flot maximal

Réseaux, réseaux de transport et flots

Amélioration du flot : graphe d'écart ou réseau résiduel

➢ ***Définition* 26**

- Soit un réseau de transport G=(X, A, C) possédant un flot complet φ. On appelle graphe d'écart ou réseau résiduel, le réseau $G'(\varphi)=(X, A', C')$ tel que :
  - si $a \in A$ et $\varphi(a) < C(a)$ alors $a \in A'$ et $C'(a) = C(a) - \varphi(a)$
  - si $a \in A$ et $\varphi(a) = C(a)$ alors $a \notin A'$
  - si $a = (x, y) \in A$ et $\varphi(a) > 0$ alors $a^{-1} = (y, x) \in A'$ et $C'(a^{-1}) = \varphi(a)$

Le réseau résiduel indique le long de quels arcs on peut augmenter ou diminuer le flot. L'intérêt du graphe d'écart apparaît dans le théorème suivant.

➢ ***Théorème* 4**

- Soit φ un flot de G (de x_1 à x_n) et $G'(\varphi)$ le réseau résiduel associé à φ. Une condition nécessaire et suffisante pour que le flot φ soit maximal est qu'il n'existe pas de chemin de x_1 à x_n dans $G'(\varphi)$.

Réseaux, réseaux de transport et flots

Algorithme 4 (Ford-Fulkerson)
Recherche d'un flot maximum

- Itération $k = 0$

  Partir d'un flot initial φ^0 compatible avec les contraintes de capacité

  par exemple $\varphi^0 = (0\ 0\ 0\ 0\ ..\ 0)$

- A l'itération k, soit φ^k le flot courant

  Rechercher un chemin μ^k de x_1 à x_n dans le graphe d'écart $G'(\varphi^k)$.

  S'il n'en existe pas, FIN : le flot φ^k est maximal

- Soit ε^k la capacité résiduelle du chemin μ^k (minimum des capacités résiduelles)

  Définir le flot φ^{k+1} par :

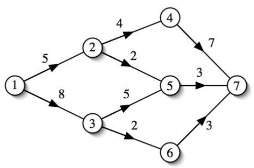
  $\varphi_a^{k+1} = \varphi_a^{k} + \varepsilon^k$ si $a \in \mu^k$ et si a est orientée dans le sens de μ^k

  $\varphi_a^{k+1} = \varphi_a^{k} - \varepsilon^k$ si $a \in \mu^k$ et si a est orientée dans le sens contraire de μ^k

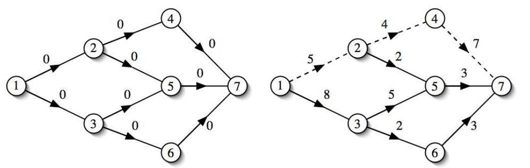
  Faire $k \leftarrow k + 1$

Réseaux, réseaux de transport et flots

Exemple de recherche d'un flot maximum par l'algorithme de Ford et Fulkerson



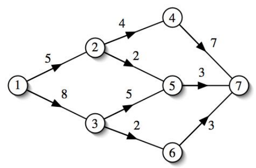
Capacités des arcs



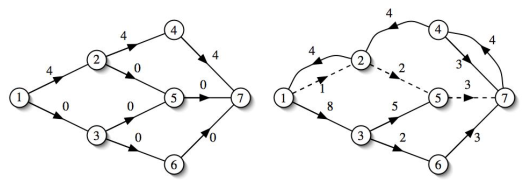
Flot initial et graphe d'écart

Réseaux, réseaux de transport et flots

Exemple de recherche d'un flot maximum par l'algorithme de Ford et Fulkerson



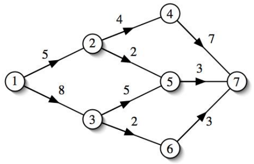
Capacités des arcs



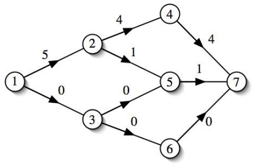
Flot à l'itération numéro 1 et graphe d'écart

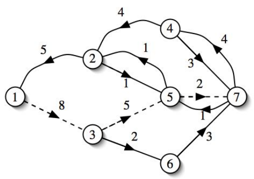
Réseaux, réseaux de transport et flots

Exemple de recherche d’un flot maximum par l’algorithme de Ford et Fulkerson



Capacités des arcs

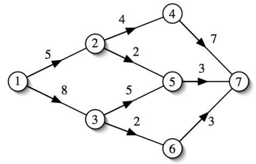




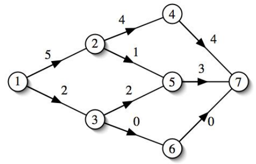
Flot à l’itération numéro 2 et graphe d’écart

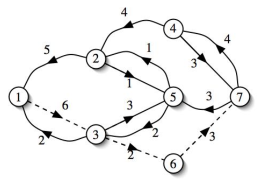
Réseaux, réseaux de transport et flots

Exemple de recherche d’un flot maximum par l’algorithme de Ford et Fulkerson



Capacités des arcs

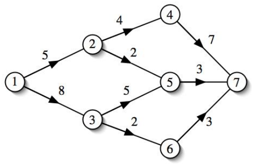




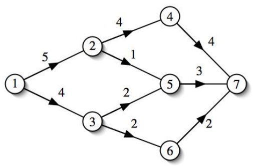
Flot à l’itération numéro 3 et graphe d’écart

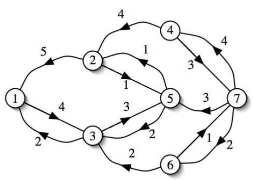
Réseaux, réseaux de transport et flots

Exemple de recherche d’un flot maximum par l’algorithme de Ford et Fulkerson



Capacités des arcs





Flot à l’itération numéro 4 et graphe d’écart

Réseaux, réseaux de transport et flots

Recherche d’un flot maximal à coût minimal

- En pratique, on associe à la plupart des réseaux de transport, une notion de coût unitaire du transport qui se traduit par l’association, à chaque arc a du réseau, d’un nombre réel $w(a)$ représentant ce coût unitaire.

- Le coût total d’un flot φ est défini comme la somme des coûts associés aux flots élémentaires
$$W(\varphi) = \sum_{a \in A} w(a)\varphi(a)$$

- $\Rightarrow$ Recherche, parmi les flots à valeur maximale, de celui qui est à coût minimal.

- Adaptation de l’algorithme de Ford et Fulkerson.

Réseaux, réseaux de transport et flots

Recherche d'un flot maximal à coût minimal

Algorithme de Busaker-Gowen

- A l'itération numéro k, le flot courant φ^k est supposé être un flot de valeur $\varphi_0{}^k$ et de coût minimal parmi l'ensemble de tous les flots de valeurs $\varphi_0{}^k$.
- Soit $G'(\varphi^k)$ le graphe d'écart relatif à φ^k ; on attribue aux arcs de $G'(\varphi^k)$ les coûts w' et les capacités c' suivantes :

  si $a = (x_i, x_j) \in A$ et $\varphi(a) < c(a)$,
  l'arc $a^+ = (x_i, x_j)$ a un coût $w'(a) = w(a)$ et une capacité $c'(a) = c(a) - \varphi(a) > 0$

  si $a = (x_i, x_j) \in A$ et $\varphi(a) > 0$,
  l'arc $a^- = (x_j, x_i)$ a un coût $w'(a) = -w(a)$ et une capacité $c'(a) = \varphi(a)$.
- Soit alors μ^k un chemin de coût minimal relativement aux coûts w' de x_1 à x_n sur le graphe d'écart $G'(\varphi^k)$; on note ε^k la capacité résiduelle de ce chemin.
- On définit alors le flot φ^{k+1} comme dans l'algorithme de Ford et Fulkerson.
  Si μ est le vecteur associé au chemin μ^k, on a $\varphi^{k+1} = \varepsilon^k \mu$

Couplages

Problèmes d'affectation

Etant donné n tâches à réaliser et n machines pour les réaliser et sachant que l'on connaît le coût de réalisation C_{ij} de la tâche t_i par la machine m_j,

on cherche une permutation σ de {1, 2, …, n} conduisant à un coût total

$$\sum_{i=1}^{n} C_{i,\sigma(i)}$$

minimal (sur l'ensemble de toutes les n! permutations σ possibles).

- Cas particulier d'un problème de transport (sans capacité)
- Cas particulier d'un problème de couplage parfait de poids minimum

Couplages

Problèmes d'affectation

- Constat : on ne change pas la ou les solutions optimales en augmentant ou en diminuant d'une même quantité tous les éléments d'une même ligne (ou d'une même colonne) de la matrice des coûts.
- Si l'on fait apparaître, par ce type de transformation, suffisamment de zéros dans le tableau (mais pas de coût négatifs) et qu'il existe n zéros indépendants (c'est-à-dire un seul zéro dans chaque ligne et dans chaque colonne), on aura trouvé l'affectation optimale.
- Algorithme Hongrois (algorithme itératif comportant trois phases)

Couplages

Problèmes d'affectation : algorithme Hongrois

	1	2	3	4	5
A	7	3	5	7	10
B	6	∞	∞	8	7
C	6	5	1	5	∞
D	11	4	∞	11	15
E	∞	4	5	2	10

- Première phase : obtention de valeurs nulles
  A tous les éléments de chacune des colonnes, on enlève le plus petit élément de cette colonne, puis on effectue la même opération sur les lignes

	1	2	3	4	5
A	1	0	4	5	3
B	0	∞	∞	6	0
C	0	2	0	3	∞
D	5	1	∞	9	8
E	∞	1	4	0	3

	1	2	3	4	5
A	1	0	4	5	3
B	0	∞	∞	6	0
C	0	2	0	3	∞
D	4	0	∞	8	7
E	∞	1	4	0	3

Couplages

Problèmes d'affectation : algorithme Hongrois

- Deuxième phase : recherche du maximum d'affectations possibles
  On cherche à former une solution à coût nul.
  On considère la ligne ayant un nombre minimal de zéros, on encadre l'un des zéros de cette ligne, puis on barre les zéros qui se trouvent sur la même ligne ou la même colonne.
  On procède de même pour toutes les lignes.

	1	2	3	4	5
A	1	$\boxed{0}$	4	5	3
B	$\boxed{0}$	∞	∞	6	$\cancel{0}$
C	$\cancel{0}$	2	$\boxed{0}$	3	∞
D	4	$\cancel{0}$	∞	8	7
E	∞	1	4	$\boxed{0}$	3

73

Couplages

Problèmes d'affectation : algorithme Hongrois

- Troisième phase : détermination des affectations intéressantes
  - On marque toutes les lignes qui ne contiennent aucun zéros encadré
  - On marque les lignes qui ont un zéro encadré dans une colonne marquée
  - On marque les colonnes qui ont un ou plusieurs zéros barrés dans une ligne marquée
  - On répète jusqu'à ce que l'on ne puisse plus faire de nouveaux marquages

	1	2	3	4	5	
A	1	$\boxed{0}$	4	5	3	*
B	$\boxed{0}$	∞	∞	6	$\cancel{0}$	
C	$\cancel{0}$	2	$\boxed{0}$	3	∞	
D	4	$\cancel{0}$	∞	8	7	*
E	∞	1	4	$\boxed{0}$	3	
		*				

74

Couplages

Problèmes d'affectation : algorithme Hongrois

- Troisième phase : détermination des affectations intéressantes
  - Les affectations intéressantes à considérer sont celles issues des sommets correspondant aux lignes marquées ; on barre donc les lignes non marquées
  - Les sommets correspondant au colonnes marquées ne sont pas intéressants car il ne peuvent être ré-affectés ; on barre donc les colonnes marquées

	1	2	3	4	5	
A	1	$\boxed{0}$	4	5	3	*
B	$\boxed{0}$	∞	∞	6	$\cancel{0}$	
C	$\cancel{0}$	2	$\boxed{0}$	3	∞	
D	4	$\cancel{0}$	∞	8	7	*
E	∞	1	4	$\boxed{0}$	3	
		*				

75

Couplages

Problèmes d'affectation : algorithme Hongrois

- Troisième phase : détermination des affectations intéressantes
  - Dans le tableau réduit obtenu (cases non barrées), on recherche l'élément le plus petit (nécessairement non nul par construction)
  - On retranche sa valeur aux colonnes non barrées
  - Et on l'ajoute aux lignes barrées

	1	2	3	4	5	
A	1	$\boxed{0}$	4	5	3	*
B	$\boxed{0}$	∞	∞	6	$\cancel{0}$	
C	$\cancel{0}$	2	$\boxed{0}$	3	∞	
D	4	$\cancel{0}$	∞	8	7	*
E	∞	1	4	$\boxed{0}$	3	
		*				

	1	2	3	4	5
A	0	0	3	4	2
B	-1	∞	∞	5	-1
C	-1	2	-1	2	∞
D	3	0	∞	7	6
E	∞	1	3	-1	2

	1	2	3	4	5
A	0	0	3	4	2
B	0	∞	∞	6	0
C	0	3	0	3	∞
D	3	0	∞	7	6
E	∞	2	4	0	3

76

Couplages

Problèmes d'affectation : algorithme Hongrois

- Deuxième phase : recherche du maximum d'affectations possibles
  On cherche à former une solution à coût nul.
  On considère la ligne ayant un nombre minimal de zéros, on encadre l'un des zéros de cette ligne, puis on barre les zéros qui se trouvent sur la même ligne ou la même colonne.
  On procède de même pour toutes les lignes.

	1	2	3	4	5
A	$\boxed{0}$	$\cancel{0}$	3	4	2
B	$\cancel{0}$	∞	∞	6	$\boxed{0}$
C	$\cancel{0}$	3	$\boxed{0}$	3	∞
D	3	$\boxed{0}$	∞	7	6
E	∞	2	4	$\boxed{0}$	3

Couplages

Problèmes d'affectation : algorithme Hongrois

- Solution optimale

	1	2	3	4	5
A	$\boxed{0}$	$\cancel{0}$	3	4	2
B	$\cancel{0}$	∞	∞	6	$\boxed{0}$
C	$\cancel{0}$	3	$\boxed{0}$	3	∞
D	3	$\boxed{0}$	∞	7	6
E	∞	2	4	$\boxed{0}$	3

- Permutation

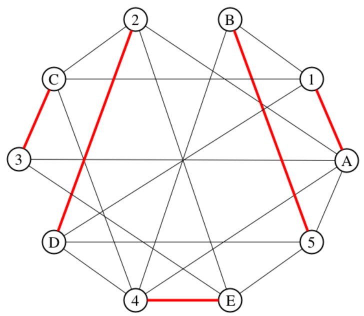
$$\sigma = \begin{pmatrix} A & B & C & D & E \\ 1 & 5 & 3 & 2 & 4 \end{pmatrix}$$

Couplages

Problèmes d'affectation : algorithme Hongrois

- Solution optimale
  $C_{A1} + C_{B5} + C_{C3} + C_{D2} + C_{E4} = 7 + 7 + 1 + 4 + 2 = 21$

	1	2	3	4	5
A	$\boxed{7}$	3	5	7	10
B	6	∞	∞	8	$\boxed{7}$
C	6	5	$\boxed{1}$	5	∞
D	11	$\boxed{4}$	∞	11	15
E	∞	4	5	$\boxed{2}$	10



Problèmes d'ordonnancement

Objectif

- Faciliter la mise en œuvre et guider l'exécution d'un ensemble complexe de tâches.
- Etant donné un objectif qu'on se propose d'atteindre et dont la réalisation suppose l'exécution préalable de multiples tâches, soumises à de nombreuses contraintes, déterminer l'ordre et le calendrier des diverses tâches.
- Le critère d'optimalité peut porter sur la minimisation de la durée et/ou du coût de la réalisation du projet.

Problème central de l'ordonnancement

- Le projet est décomposable en un nombre fini de tâches, caractérisées par leur durée d'exécution (éventuellement également par leur coût d'exécution)
- Ces tâches sont soumises à des contraintes de postériorité stricte (une tâche ne peut commencer que si certaines autres tâches sont complètement terminées)

Problèmes d'ordonnancement

Le graphe potentiels-tâches

- A partir de la description des tâches du projet, on construit le graphe suivant
  - A chaque tâche i, on associe un sommet i du graphe
  - On définit un arc entre i et j de longueur d_i (la durée de la tâche i) si la tâche i doit précéder la tâche j
  - On introduit deux sommets correspondant à deux tâches fictives de durée nulle, la tâche de début des travaux α, antérieure à toutes les tâches et la tâche de fin des travaux ω postérieure à toutes les tâches
  - Le graphe ainsi défini est sans circuit
  - Le travail commençant à la date 0, on cherche un ordonnancement qui minimise la durée totale du projet
  - Pour qu'une tâche puisse commencer, il est nécessaire que toutes les tâches qui la relient à la tâche début du projet soient réalisées

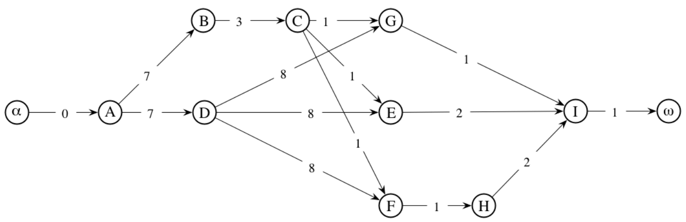
Problèmes d'ordonnancement

Le graphe potentiels-tâches

Tâche		Durée	Tâches ant.
A	Maçonnerie	7	-
B	Charpente	3	A
C	Toiture	1	B
D	Sanitaire et électricité	8	A
E	Façade	2	D, C
F	Fenêtres	1	D,C
G	Aménagement jardin	1	D,C
H	Peinture	2	F
I	Emménagement	1	E,G,H

Problèmes d'ordonnancement

Le graphe potentiels-tâches



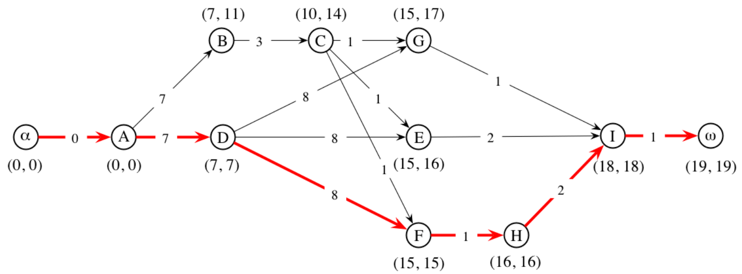
Problèmes d'ordonnancement

Le graphe potentiels-tâches

- ***Définition* 32**
  - La date au plus tôt, t_i de début de la tâche i est égale à $\max(t_j+d_j)$ pour $j \in \Gamma_i^{-1}$ c'est-à-dire que t_i est égal à la longueur du plus long chemin de α à i.
  - La durée minimale du projet est égale à t_ω.
  - La date au plus tard T_i pour commencer la tâche i est égale à $\min(T_j-d_j)$ pour $j \in \Gamma_i$ avec $T_\omega = t_\omega$
    c'est-à-dire $T_i = t_\omega - l(i,\omega)$ où $l(i,\omega)$ est la longueur du plus long chemin de i à ω.
  - La marge totale M_i de la tâche i est définie comme la différence entre la date au plus tôt et au plus tard : $M_i = T_i - t_i$
  - Les tâches dont la marge est nulle sont appelées tâches critiques. Si un quelconque retard est pris sur la réalisation de l'une de ces tâches, la durée minimale du projet sera augmentée d'autant.
  - La marge libre m_i de la tâche i est le délai dont on peut retarder cette tâche sans affecter les dates de début au plus tôt des tâches postérieures :
    $m_i = \min(t_j - t_i - d_i)$ pour $j \in \Gamma_i$

Problèmes d'ordonnancement

Le graphe potentiels-tâches



Chemin critique

X
(date au plus tôt, date au plus tard)

Problèmes d'ordonnancement

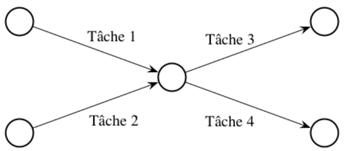
Le graphe potentiels-étapes ou graphe PERT

- Construction du graphe
  - Les tâches sont matérialisées par les arcs du graphe
  - La longueur de chaque arc est égale à la durée d_i de la tâche correspondante
  - Le début et la fin d'une tâche constituent les étapes du projet
  - On introduit deux étapes de début et de fin du projet
  - Si une tâche j doit succéder à une tâche i, on confond l'extrémité initiale de l'arc a_j (tâche j) et l'extrémité de l'arc a_i (tâche i)
  - Le graphe est sans circuit
  - Les sommets s'interprètent comme des étapes

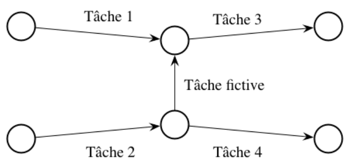
Problèmes d'ordonnancement

Le graphe potentiels-étapes ou graphe PERT

- Construction du graphe
  - Nécessité d'introduire parfois des tâches fictives de durée nulle pour traduire des contraintes de postériorité



Les tâches 3 et 4 doivent succéder aux tâches 1 et 2



La tâche 3 doit succéder aux tâches 1 et 2
La tâche 4 doit succéder à la tâche 2

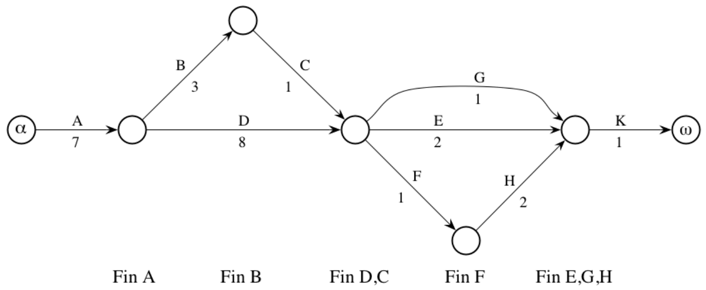
Problèmes d'ordonnancement

Le graphe potentiels-étapes

Tâche		Durée	Tâches ant.
A	Maçonnerie	7	-
B	Charpente	3	A
C	Toiture	1	B
D	Sanitaire et électricité	8	A
E	Façade	2	D, C
F	Fenêtres	1	D,C
G	Aménagement jardin	1	D,C
H	Peinture	2	F
I	Emménagement	1	E,G,H

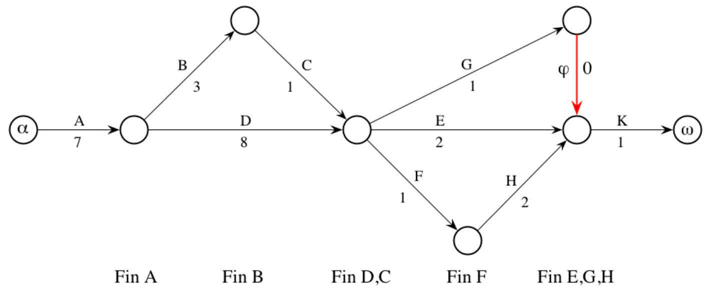
Problèmes d'ordonnancement

Le graphe potentiels-étapes



Problèmes d'ordonnancement

Le graphe potentiels-étapes



Problèmes d'ordonnancement

Résolution

- Un ordonnancement est un programme d'exécution qui fixe la date t_i de début d'exécution de chacune des tâches (dans la méthode du potentiel) ou de début de réalisation de chacune des étapes (dans la méthode PERT).
- Un ordonnancement est optimal s'il est réalisable, c'est-à-dire si $t_j - t_i \geq d_i$ et s'il minimise la durée de réalisation t_ω du projet.
- La durée minimale du projet est égale à la longueur du plus long chemin du sommet α au sommet ω. Ce chemin est appelé chemin critique.
- Chaque $t_j = \max(t_i + d_i)$ pour $i \in \Gamma_j^{-1}$ est la longueur du plus long chemin de α à i et représente la date au plus tôt de début d'exécution de la tâche ou de l'étape j.
- En définissant $T_j = \min(T_i - d_i)$ pour $i \in \Gamma_j$ avec $T_\omega = t_\omega$,
  chaque $T_\omega - T_i$ est la longueur du plus long chemin de i à ω
  et représente la date au plus tard de début d'exécution de la tâche ou de l'étape j.

Problèmes d'ordonnancement

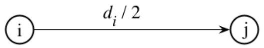
Compléments

- Contraintes additives (par rapport au problème central)
  - Type potentiel
    la tâche *j* doit commencer après la fin de *i* ou bien après la moitié de la réalisation de *i*, ou bien un certain temps après la fin de *i*, etc.
  - Type disjonctif
    les tâches *i* et *j* ne peuvent être réalisées en même temps
  - Type cumulatif
    les moyens nécessaires à l'exécution d'un certain nombre de tâches sont, à chaque instant, limités ; par exemple, le nombre de camions utilisables ou la somme dépensée pour les travaux réalisés à l'instant *t* ne doit pas dépasser un certain seuil.
- Prise en compte
  - Contraintes difficiles à prendre en compte dans le graphe potentiels-étapes (introduction d'un grand nombre d'étapes fictives)
  - Adjonction d'arcs supplémentaires dans le graphe potentiels-tâches.

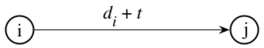
Problèmes d’ordonnancement

Compléments

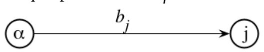
- j ne doit pas commencer avant la moitié de la réalisation de la tâche i



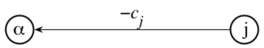
- j ne peut commencer qu’un temps t après la fin de la tâche i



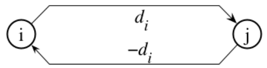
- j ne peut commencer qu’après la date b_j



- j doit commencer avant la date c_j



- j doit suivre immédiatement la tâche i



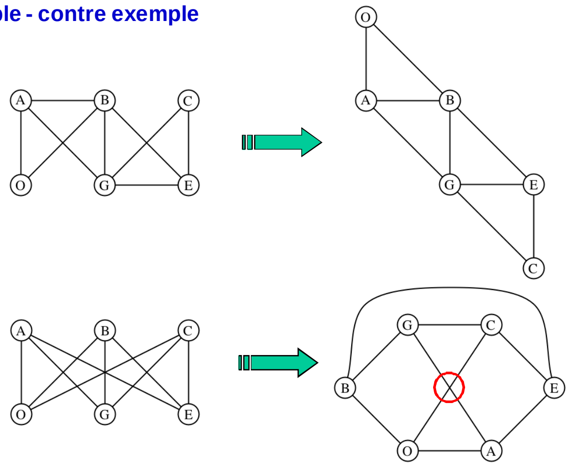
Graphes planaires

➢ ***Définition* 33**

- On dit qu’un graphe (simple ou multigraphe) est planaire si on peut le « dessiner » dans le plan, en matérialisant les arêtes par des arcs continus qui ne se croisent pas en dehors de leurs extrémités.
- Deux graphes que l’on peut amener à coïncider par déformation élastique du plan sont dits isomorphes.
- Les arêtes d’un graphe planaire G délimitent des surfaces planes nommées faces de G, telles que deux points arbitrairement choisis dans une même région peuvent toujours être reliés par un trait continu ne rencontrant ni sommet, ni arêtes.
- La frontière d’une face de G est l ’ensemble des arêtes qui touchent une face z.
- Deux faces z et z' sont dites adjacentes si leurs frontières ont une arête commune.
- On appelle contour de la face z le cycle élémentaire constitué de toutes les arêtes de la frontière de la face z.
- Dans un graphe, il existe toujours une face illimitée et une seule qui constitue la face infinie. On ne définit pas de contour pour cette face.

Graphes planaires

Exemple - contre exemple



Graphes planaires

Test de planarité

➢ ***Définition* 34**

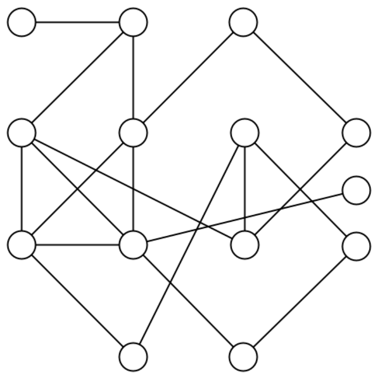
- Soit un graphe connexe G. On dit qu’un graphe H se déduit par contraction s’il a été obtenu par application répétée des règles suivantes :
  - x étant un sommet de degré 1 de G, H est le sous-graphe de G obtenu en supprimant x.
  - x étant un sommet de degré 2 de G, de voisins y et z, H est le sous-graphe de G obtenu en supprimant x et auquel on ajoute une arête (y, z).

➢ ***Théorème* 8 (Kuratowski)**

- La condition nécessaire et suffisante pour qu’un graphe G soit planaire est qu’il n’admette pas de sous-graphe partiel se contractant en un graphe isomorphe à $K_{3,3}$ ou K_5.

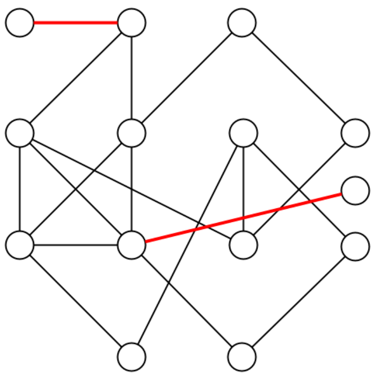
Graphes planaires

Test de planarité



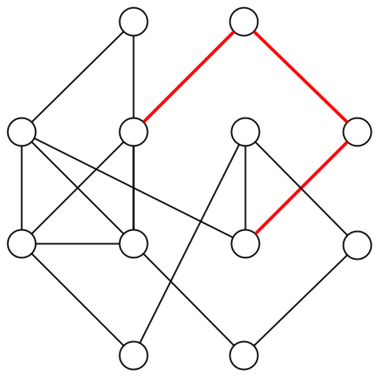
Graphes planaires

Test de planarité



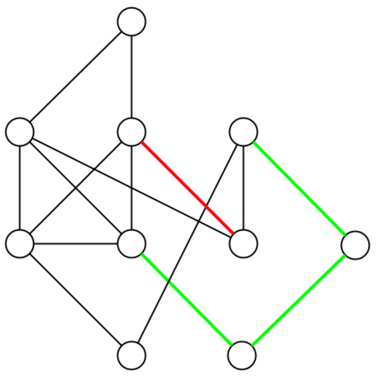
Graphes planaires

Test de planarité



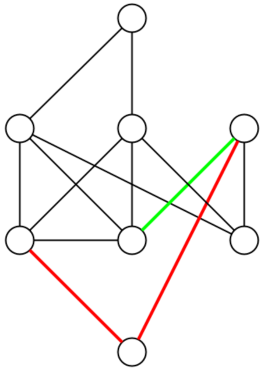
Graphes planaires

Test de planarité



Graphes planaires

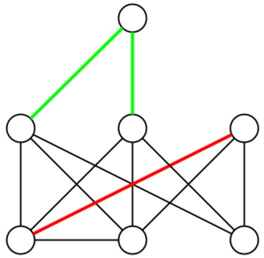
Test de planarité



101

Graphes planaires

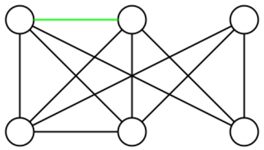
Test de planarité



102

Graphes planaires

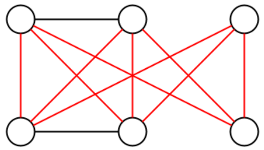
Test de planarité



103

Graphes planaires

Test de planarité



$K_{3,3}$

Graphe non planaire

104

Coloration d'un graphe

- On définit deux types de coloration pour un graphe $G=(X, A)$; la coloration des arêtes et la coloration des sommets.
- La coloration des sommets (resp. des arêtes) d'un graphe G correspond à l'affectation d'une couleur à chacun des sommets (resp. des arêtes) de telle sorte que deux sommets (resp. arêtes) adjacents ne soient pas porteur de la même couleur.
- Un graphe est dit p-chromatique si ses sommets admettent une coloration en p couleurs.
- On appelle nombre chromatique $\gamma(G)$ (resp. indice chromatique $q(G)$) le nombre minimum de couleurs distinctes nécessaires pour effectuer une coloration des sommets (resp. des arêtes) de G.

105

Coloration d'un graphe

Coloration des sommets

➢ ***Définition* 34**

- Un sous-ensemble de sommets $I \subset X$ est un ensemble stable s'il ne comprend que des sommets non adjacents deux à deux.
- Les sommets coloriés de la même couleur dans une coloration des sommets forment donc un ensemble stable.
- Une coloration des sommets est donc une partition des sommets en ensembles stables.
- Soit $\alpha(G)$ le nombre de stabilité, c'est-à-dire le cardinal maximum d'un ensemble stable, alors si $\gamma(G)$ est le nombre chromatique, puisque chaque ensemble de sommets de même couleur a un cardinal inférieur ou égal à $\alpha(G)$, on a :

$$\alpha(G)\ \gamma(G)\ \geq\ N(G)$$

où $N(G)$ est le nombre de sommets du graphe G.

On en déduit : $\gamma(G)\ \geq\ N(G)\ /\ \alpha(G)$

106

Coloration d'un graphe

Coloration des sommets

- La détermination du nombre chromatique d'un graphe ainsi que l'obtention d'une coloration minimale des sommets constituent un problème assez complexe..
- En pratique, on utilise souvent des algorithmes de coloration heuristiques.
- Le nombre chromatique $\gamma(G)$ possède des bornes inférieures et supérieures

Pour un graphe à n sommets et m arêtes, on a :

$\gamma(G)\ \geq\ n\ /\ (n - d_{min})$ avec d_{min} degré minimum des sommets de G
$\gamma(G)\ \geq$ cardinal de la plus grande clique de G
$\gamma(G)\ \geq\ n^2\ /\ (n^2 - 2m)$

$\gamma(G)\ \leq\ n + 1 - \alpha(G)$ avec $\alpha(G)$ nombre de stabilité du graphe G
$\gamma(G)\ \leq d_{max} + 1$ avec d_{max} degré maximum des sommets de G

$\alpha(G)$ nombre de stabilité : maximum des cardinals des ensembles stables d'un graphe
un sous-ensemble S de sommets est stable si deux sommets quelconques de S ne sont pas adjacents

107

Coloration d'un graphe

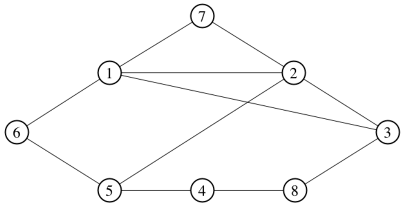
➢ **Algorithme de Welsh et Powell**

- Etape 0 : Initialisation
  M : matrice d'adjacence du graphe dont les sommets sont rangés par ordre de degré décroissants
  $k = 1$
- Etape 1
  $N = M$
- Etape 2
  Colorer par la couleur c_k la première ligne non encore colorée de N ainsi que la colonne correspondante
  N = ensemble des lignes non encore colorées ayant un zéro dans les colonnes de couleur c_k
  si N $\neq \varnothing$ aller à l'étape 2 sinon aller à l'étape 3
- Etape 3
  Si toutes les lignes sont colorées alors STOP (on a une k-coloration)
  $k = k + 1$; (changer de couleur)
  aller à l'étape 1

108

Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**

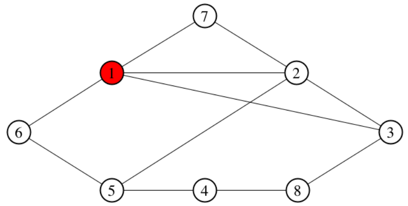


$$\begin{bmatrix} . & 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 & . \\ 1 & 1 & . & 1 & . & . & . & 1 \\ . & . & 1 & . & 1 & . & . & 1 \\ . & 1 & . & 1 & . & 1 & . & . \\ 1 & . & . & . & 1 & . & . & . \\ 1 & 1 & . & . & . & . & . & . \\ . & . & 1 & 1 & . & . & . & . \end{bmatrix}$$

109

Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**

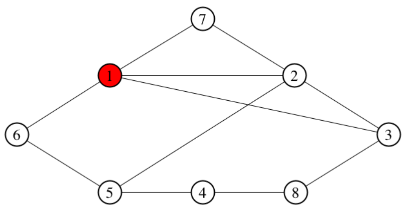


$$\begin{bmatrix} . & 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 & . \\ 1 & 1 & . & 1 & . & . & . & 1 \\ . & . & 1 & . & 1 & . & . & 1 \\ . & 1 & . & 1 & . & 1 & . & . \\ 1 & . & . & . & 1 & . & . & . \\ 1 & 1 & . & . & . & . & . & . \\ . & . & 1 & 1 & . & . & . & . \end{bmatrix}$$

110

Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**

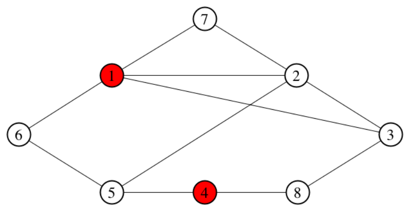


$$\begin{bmatrix} . & 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 & . \\ 1 & 1 & . & 1 & . & . & . & 1 \\ . & . & 1 & . & 1 & . & . & 1 \\ . & 1 & . & 1 & . & 1 & . & . \\ 1 & . & . & . & 1 & . & . & . \\ 1 & 1 & . & . & . & . & . & . \\ . & . & 1 & 1 & . & . & . & . \end{bmatrix}$$

111

Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**

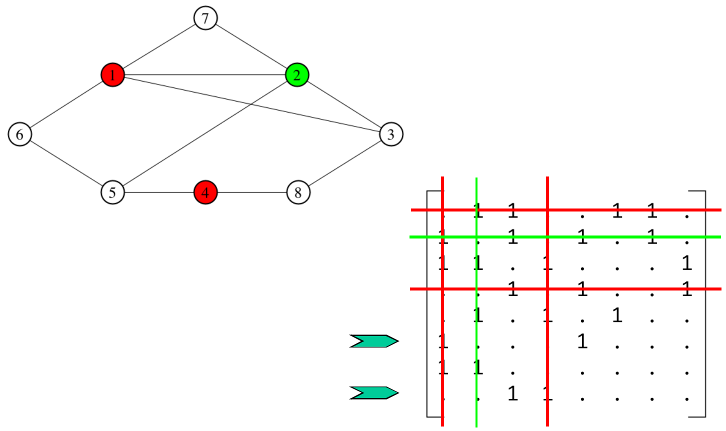


$$\begin{bmatrix} . & 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 & . \\ 1 & 1 & . & 1 & . & . & . & 1 \\ . & . & 1 & . & 1 & . & . & 1 \\ . & 1 & . & 1 & . & 1 & . & . \\ 1 & . & . & . & 1 & . & . & . \\ 1 & 1 & . & . & . & . & . & . \\ . & . & 1 & 1 & . & . & . & . \end{bmatrix}$$

112

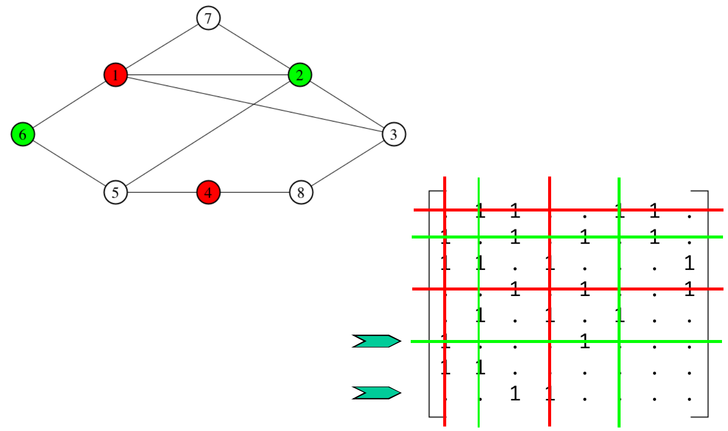
Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**



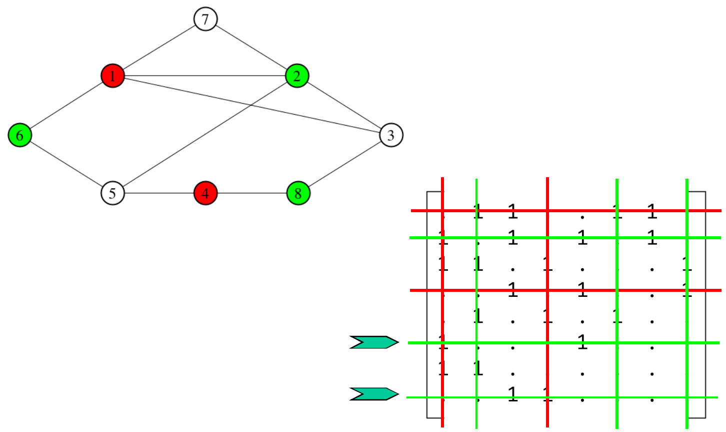
Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**



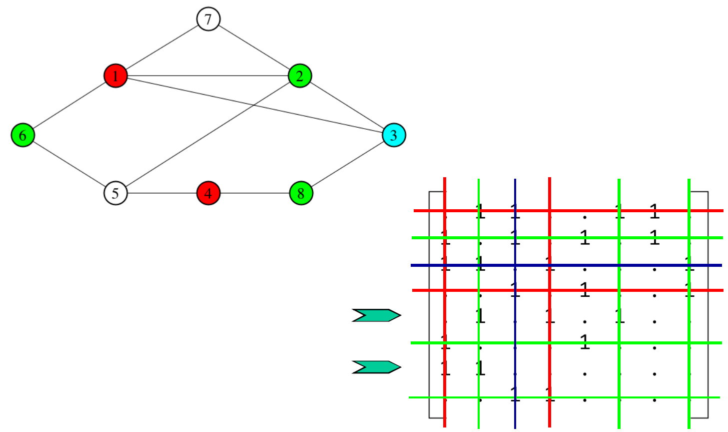
Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**



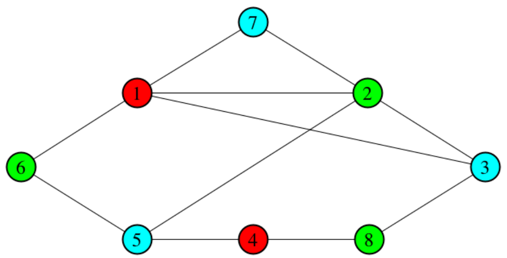
Coloration d'un graphe

➢ **Algorithme de Welsh et Powell : exemple**



Coloration d'un graphe

- **Algorithme de Welsh et Powell : exemple**



117