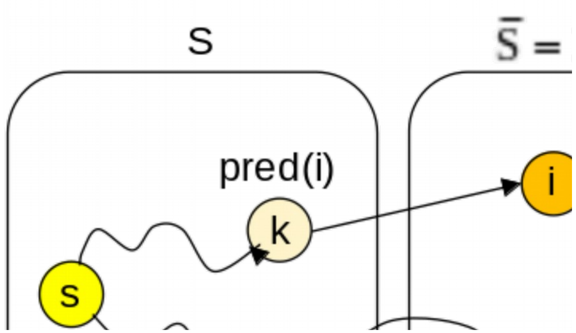
Optimisation Numérique

II – Plus Courts Chemins

Eric Pinson



Institut de Mathématiques Appliquées
Université Catholique de l'Ouest
Angers - France

UCO ANGERS UNIVERSITÉ CATHOLIQUE DE L'OUEST

L2 MIASHS

IMA Mathématiques Appliquées

Plus Courts Chemins

Définition : **Réseau (Graphe Valué) – Longueur d'un chemin**

Un réseau est un graphe $G = (X;U)$ auquel on associe une fonction $c : U \rightarrow IR$ qui à chaque arc $u=(i,j)$ fait correspondre sa "longueur" $c_u=c(i,j)=c_{ij}$. On note $R = (X,U,c)$ un tel réseau.
La longueur d'un chemin (d'une chaîne, d'un circuit ou d'un cycle) est la somme des longueurs de chaque arc qui le compose. Par convention, un chemin (une chaîne, un circuit ou un cycle) qui ne contient pas d'arc est de longueur nulle.

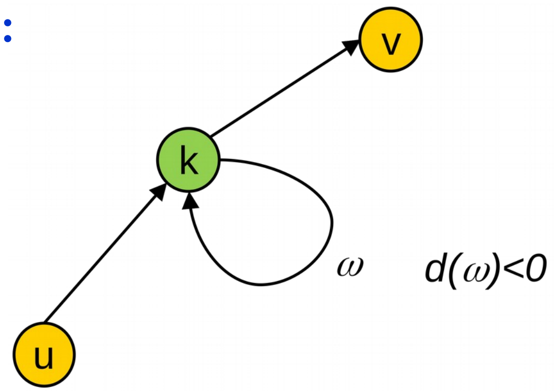
Trois types de problèmes de plus courts chemins :

1. Rechercher un plus court chemin allant d'un sommet s à un autre sommet p
2. Rechercher les plus courts chemins allant d'un sommet s à tous les autres sommets du graphe
3. Rechercher les plus courts chemins entre toutes paires de sommets

Plus Courts Chemins

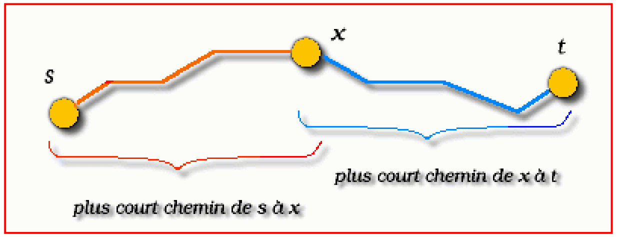
Propriété : Condition d'existence d'un plus court chemin :

S'il existe un chemin entre deux sommets u et v contenant un circuit de coût négatif, alors il n'existe pas de plus court chemin entre u et v. Un circuit négatif est appelé un circuit absorbant.



Propriété : Principe de sous-optimalité

- Les plus courts chemins vérifient le principe de **sous-optimalité** : si **p=(s,…,t)** est un plus court chemin entre **s** et **t**, alors pour tout sommet **x** sur ce chemin, le sous-chemin de **p** jusqu'à **x**, **(s,…,x)**, est un plus court chemin de **s** à **x**
- Un plus court chemin entre 2 sommets est *élémentaire*.



Plus Courts Chemins

Dans ce qui suit, on se focalise sur la recherche du PCC d'un sommet s fixé de G à tous les autres sommets

Propriété : **Conditions d'optimalité**

Associons à tout sommet j d'un graphe valué G=(X,U,c) une valeur (étiquette, potentiel) d(j).

Proposition 1

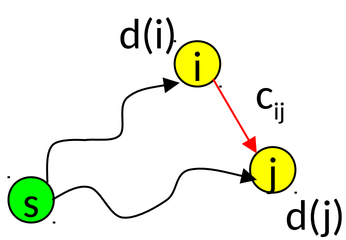
Les valeurs d(j) représentent les longueurs des plus courts chemins d'origine s si et seulement si :

$$\forall (i,j) \in U, d(j) \leq d(i) + c_{ij}$$

Proposition 2

Soient s et t deux sommets de G. Un chemin μ est un plus court chemin de s à t si et seulement si :

$$\forall (i,j) \in \mu, d(j) = d(i) + c_{ij}$$



→ *Arborescence des Plus Courts Chemins*

Plus Courts Chemins

Graphe G=X,U,c) à valuations positives : $\forall(i,j)\in U,\ c_{ij} \geq 0$

Algorithme de Dijsktra

d(s) =0, pred(s)=s
$S=\emptyset$, $\bar{S}=X$
Pour tout $x\in X-\{s\}$
  $d(x) = +\infty$
FinPour
Tant que $S\neq X$ faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j\in\bar{S}} d(j)$
  $S=S\cup\{i\}$, $\bar{S}=\bar{S}-\{i\}$
  Pour tout $j\in\Gamma^{+}(i)\cap\bar{S}$
    Si $d(j) > d(i) + c_{ij}$ alors
      $d(j) = d(i) + c_{ij}$, pred(j)=i
    FinSi
  FinPour
Finttq

- G=(X,U) : graphe valué : longueur de l'arc (i,j) : $c_{ij} \geq 0$
- s : sommet de départ
- d(x) : longueur du plus court chemin de s au sommet x
- pred(x) : prédécesseur de x dans le Plus Court Chemin associé
- S : sommets déjà traités

Plus Courts Chemins

Graphe G=X,U,c) à valuations positives : $\forall(i,j)\in U$, $c_{ij} \geq 0$

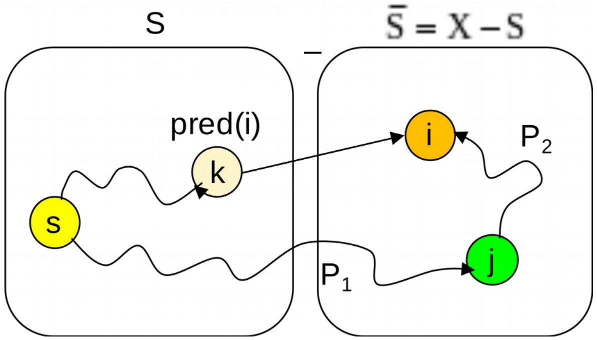
Convergence de l'algorithme de Dijkstra :

Montrons qu'à chaque étape de la procédure, les potentiels associés aux sommets de S sont optimaux, et correspondent à des PCC empruntant uniquement des sommets de S.

A une étape quelconque de l'algorithme, soit i le dernier sommet de X-S inséré dans S. On a :

$$d(i) \leq \min_{j\in X\text{-}S} d(j)$$

Montrons alors que la longueur de tout chemin allant de s à i et empruntant des sommets de X-S est supérieure ou égale à d(i). Soit P un tel chemin. Il peut en évidence être partitionné en deux sous-chemins P_1 et P_2, P_1 partant de s et aboutissant en $j\in X\text{-}S$ en empruntant que des sommets de S (à l'exception de j). On a alors $c(P_1) \geq d(j)$.



Plus Courts Chemins

Graphe G=X,U,c) à valuations positives : $\forall(i,j)\in U,\ c_{ij} \geq 0$

Convergence de l'algorithme de Dijkstra :

Les valuations étant par ailleurs toutes positives, on a $c(P_2) \geq 0$. Comme $d(i) \leq d(k)$, $\forall k\in X\text{-}S$, on en déduit que $c(P) = c(P_1) + c(P_2) \geq d(i)$, et $d(i)$ est la longueur d'un PCC joignant s à i.

Ces potentiels correspondent d'autre part à des PCC en constatant (un simple raisonnement par récurrence en atteste) qu'après introduction de i dans S, les arcs (i,j) sont scannés et on impose $d(j) = d(i) + c_{ij}$ dès lors que $d(j) > d(i) + c_{ij}$, ce qui satisfait la proposition 2.

Plus Courts Chemins

Graphe G=X,U,c) à valuations positives : $\forall(i,j)\in U,\ c_{ij} \geq 0$

Complexité

La complexité de l'algorithme de Dijkstra est $O(n^2)$.

d(s) =0, pred(s)=s
S=$\emptyset$, $\bar{S}$=X
Pour tout x$\in$X-{s}
 d(x) =+∞
FinPour
Tant que S$\neq$X faire
 Sélectionner un sommet i de S tel que : $d(i) = \min_{j\in\bar{S}} d(j)$
 S=S$\cup${i}, $\bar{S}$=$\bar{S}$-{i}
 Pour tout j$\in\Gamma^+$(i)$\cap\bar{S}$
  Si d(j) > d(i) + c_{ij} alors
   d(j) = d(i) + c_{ij} , pred(j)=i
  FinSi
 FinPour
Finttq

O(n)

O(n)

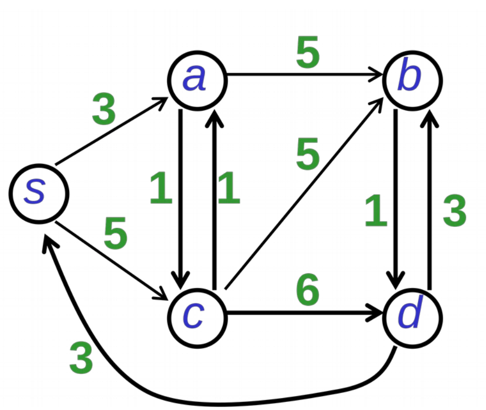
Plus Courts Chemins

Algorithme : **Dijkstra**

d(s) =0, pred(s)=s
S=∅, $\bar{S}$=X
Pour tout x∈X-{s}
  d(x) =+∞
FinPour
Tant que S≠X faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j \in \bar{S}} d(j)$
  S=S∪{i}, $\bar{S}$=$\bar{S}$-{i}
  Pour tout j∈Γ^+(i)∩$\bar{S}$
    Si d(j) > d(i) + c_{ij} alors
      d(j) = d(i) + c_{ij} , pred(j)=i
    FinSi
  FinPour
Finttq

Exemple

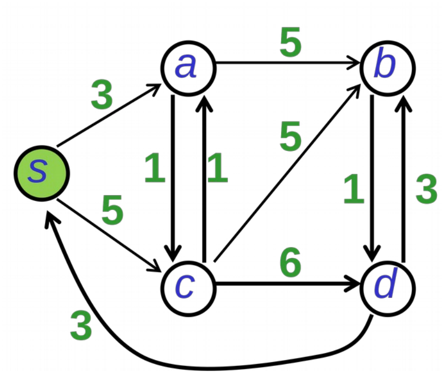
Itération 0



S=∅
d(*s*) = 0
d(*a*) = +∞
d(*b*) = +∞
d(*c*) = +∞
d(*d*) = +∞

pred(*a*) = *nil*
pred(*b*) = nil
pred(*c*) = *nil*
pred(*d*) = nil

Itération 1



S = {s}
d(*s*) = 0
d(*a*) = *3*
d(*b*) = +∞
d(*c*) = *5*
d(*d*) = +∞

pred(*a*) = *s*
pred(*b*) = nil
pred(*c*) = *s*
pred(*d*) = nil

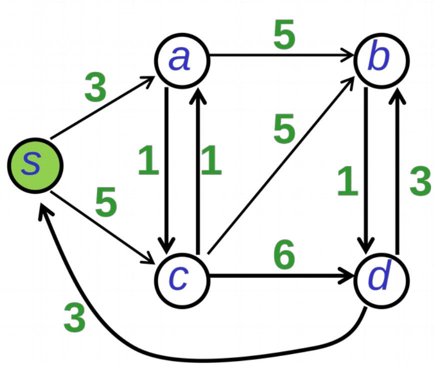
Plus Courts Chemins

Algorithme : **Dijkstra**

Exemple

d(s) =0, pred(s)=s
S=∅, $\bar{S}$=X
Pour tout x∈X-{s}
  d(x) =+∞
FinPour
Tant que S≠X faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j \in \bar{S}} d(j)$
  S=S∪{i}, $\bar{S}=\bar{S}$-{i}
  Pour tout j∈Γ^+(i)∩$\bar{S}$
    Si d(j) > d(i) + c_{ij} alors
      d(j) = d(i) + c_{ij} , pred(j)=i
    FinSi
  FinPour
Finttq

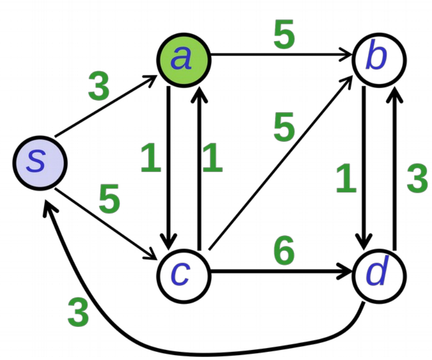
Itération 1



$S = \{s\}$
$d(s) = 0$
$d(a) = 3$
$d(b) = +\infty$
$d(c) = 5$
$d(d) = +\infty$

$pred(a) = s$
$pred(b) = \text{nil}$
$pred(c) = s$
$pred(d) = \text{nil}$

Itération 2



$S = \{s,a\}$
$d(s) = 0$
$d(a) = 3$
$d(b) = 8$
$d(c) = 4$
$d(d) = +\infty$

$pred(a) = s$
$pred(b) = a$
$pred(c) = a$
$pred(d) = \text{nil}$

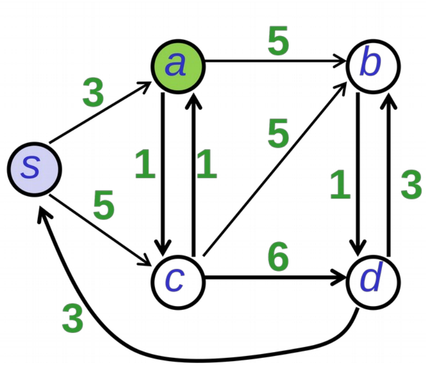
Plus Courts Chemins

Algorithme : **Dijkstra**

Exemple

d(s) =0, pred(s)=s
S=∅, $\bar{S}$=X
Pour tout x∈X-{s}
  d(x) =+∞
FinPour
Tant que S≠X faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j \in \bar{S}} d(j)$
  S=S∪{i}, $\bar{S}$=$\bar{S}$-{i}
  Pour tout j∈Γ^+(i)∩$\bar{S}$
    Si d(j) > d(i) + c_{ij} alors
      d(j) = d(i) + c_{ij} , pred(j)=i
    FinSi
  FinPour
Finttq

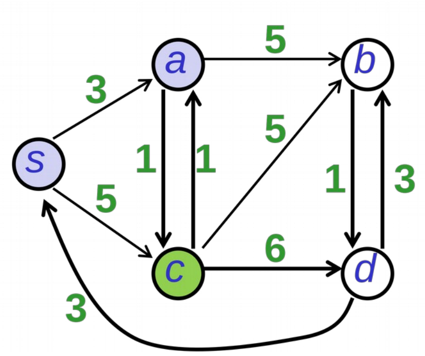
Itération 2



S = {s,a}
d(s) = 0
d(a) = 3
d(b) = 8
d(c) = 4
d(d) = +∞

pred(a) = s
pred(b) = a
pred(c) = a
pred(d) = nil

Itération 3



S = {s,a,c}
d(s) = 0
d(a) = 3
d(b) = 8
d(c) = 4
d(d) = 10

pred(a) = s
pred(b) = a
pred(c) = a
pred(d) = c

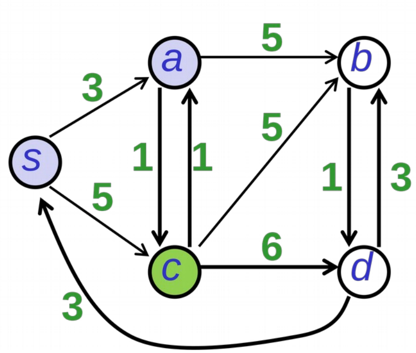
Plus Courts Chemins

Algorithme : **Dijkstra**

Exemple

d(s) =0, pred(s)=s
S=∅, $\bar{S}$=X
Pour tout x∈X-{s}
  d(x) =+∞
FinPour
Tant que S≠X faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j\in\bar{S}} d(j)$
  S=S∪{i}, $\bar{S}=\bar{S}$-{i}
  Pour tout j∈Γ^+(i)∩$\bar{S}$
    Si d(j) > d(i) + c_{ij} alors
      d(j) = d(i) + c_{ij} , pred(j)=i
    FinSi
  FinPour
Fintq

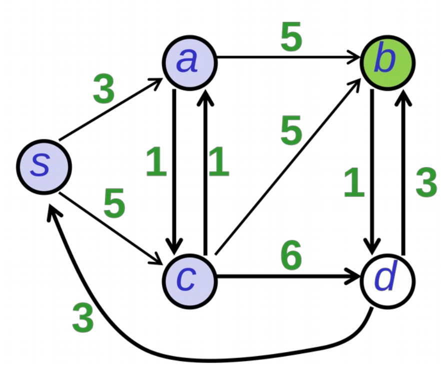
Itération 3



S = {s,*a*,*c*}
d(*s*) = 0
d(*a*) = *3*
d(*b*) = 8
d(*c*) = *4*
d(*d*) = 10

pred(*a*) = *s*
pred(*b*) = a
pred(*c*) = *a*
pred(*d*) = c

Itération 4



S = {s,*a*,*c*,*b*}
d(*s*) = 0
d(*a*) = *3*
d(*b*) = 8
d(*c*) = *4*
d(*d*) = 9

pred(*a*) = *s*
pred(*b*) = a
pred(*c*) = *a*
pred(*d*) = b

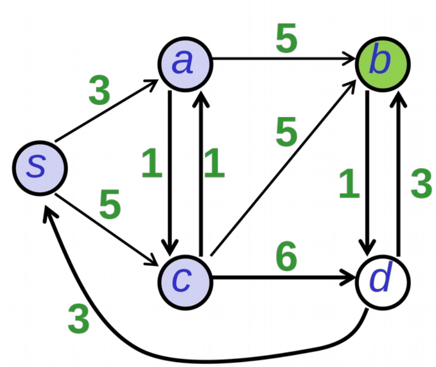
Plus Courts Chemins

Algorithme : **Dijkstra**

$d(s) = 0$, $pred(s)=s$
$S=\emptyset$, $\bar{S}=X$
Pour tout $x \in X-\{s\}$
  $d(x) = +\infty$
FinPour
Tant que $S \neq X$ faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j \in \bar{S}} d(j)$
  $S=S \cup \{i\}$, $\bar{S}=\bar{S}-\{i\}$
  Pour tout $j \in \Gamma^+(i) \cap \bar{S}$
    Si $d(j) > d(i) + c_{ij}$ alors
      $d(j) = d(i) + c_{ij}$, $pred(j)=i$
    FinSi
  FinPour
Finttq

Exemple

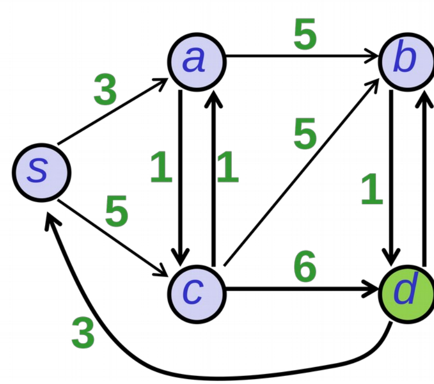
Itération 4



$S = \{s,a,c,b\}$
$d(s) = 0$
$d(a) = 3$
$d(b) = 8$
$d(c) = 4$
$d(d) = 9$

$pred(a) = s$
$pred(b) = a$
$pred(c) = a$
$pred(d) = b$

Itération 5



$S = \{s,a,c,b,d\}$
$d(s) = 0$
$d(a) = 3$
$d(b) = 8$
$d(c) = 4$
$d(d) = 9$

$pred(a) = s$
$pred(b) = a$
$pred(c) = a$
$pred(d) = b$

Plus Courts Chemins

Exercice : **Résolution d'un système linéaire (6)**

Soit P le système linéaire suivant :

$$\begin{aligned} x_3 - x_4 &\leq 5 \\ x_4 - x_1 &\leq 6 \\ x_1 - x_2 &\leq 8 \\ x_2 - x_1 &\leq 7 \\ x_3 - x_2 &\leq 2 \end{aligned}$$

L'objectif est de trouver une solution réalisable de P.

(a) Rappeler la condition d'optimalité des plus courts chemins.

(b) Formaliser ce problème comme un problème de plus courts chemins.

(c) Résoudre le système suivant en lien avec la question précédente :

$$\begin{aligned} x_3 - x_4 &\leq 2 \\ x_4 - x_1 &\leq 6 \\ x_1 - x_2 &\leq 8 \\ x_2 - x_1 &\leq -11 \\ x_3 - x_2 &\leq 2 \end{aligned}$$

Plus Courts Chemins

Exercice : **Résolution d'un système linéaire (6)**

(a) $\forall (i, j) \Sigma\, U, d(j) \quad d(i) + c_{ij}$

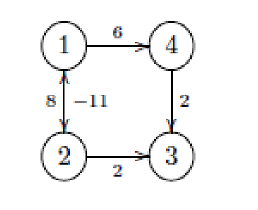
(b) Soit C l'ensemble des contraintes du système linéaire. Chaque contrainte $x_j - x_i \leq c_{ij}$ avec $(i,j) \in C$, peut s'écrire : $x_j \leq x_i + c_{ij}$

Nous en déduisons que résoudre le système linéaire P revient à affecter une valeur aux variables x_j ($j \in]n]$), tel que : $\forall (i,j) \in C,\ x_j \leq x_i + c_{ij}$

Cette relation correspondant à la définition de la condition d'optimalité des plus courts chemins. Par conséquent, les x_j ont pour valeurs les longueurs des plus courts chemins dans un graphe défini comme suit :

- un sommet j par variable x_j ,
- un arc (i,j) de valuation c_{ij} entre les sommets i et j si il existe une contrainte entre x_i et x_j .

(c) La question précédente permet de définir le graphe ci-dessous pour résoudre le système linéaire proposé. Or celui-ci contient un circuit négatif, donc il n'existe pas de plus courts chemins. Par conséquent le système linéaire proposé n'admet pas de solution.



$$\begin{aligned} x_3 - x_4 &\leq 2 \\ x_4 - x_1 &\leq 6 \\ x_1 - x_2 &\leq 8 \\ x_2 - x_1 &\leq -11 \\ x_3 - x_2 &\leq 2 \end{aligned}$$

Plus Courts Chemins

Exercice : **Horaires de service (8)**

Le tableau ci-dessous illustre les heures de service de 8 agents d'une compagnie de bus. Par exemple, l'agent 1 prend son service à 9H et le termine à 13H, le coût associé étant de 30. Si un agent travaille, il effectue son créneau horaire entièrement et le coût associé est toujours le même. L'entreprise souhaite s'assurer au moindre coût que, chaque heure entre 9H et 17H, au moins un agent soit en service.

Agent	1	2	3	4	5	6	7	8
Heures de service	9-13	9-12	12-15	12-17	14-17	13-16	16-17	12-14
Coût	30	18	21	38	20	22	9	17

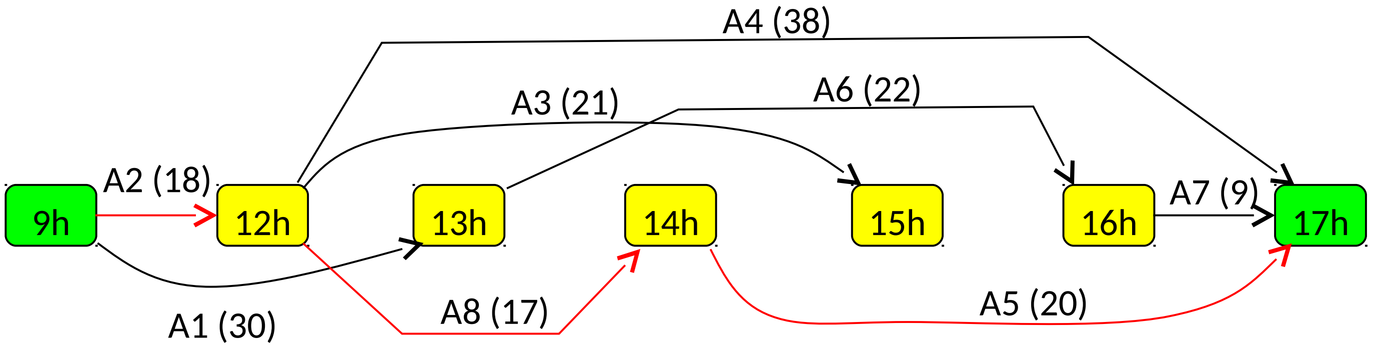
(a) Formaliser ce problème comme un problème de plus courts chemins.

(b) Résoudre l'exemple donné dans le tableau ci-dessus.

Plus Courts Chemins

Exercice : **Horaires de service (8)**

Agent	1	2	3	4	5	6	7	8
Heures de service	9-13	9-12	12-15	12-17	14-17	13-16	16-17	12-14
Coût	30	18	21	38	20	22	9	17



Plus Courts Chemins

Graphes à valuations quelconques

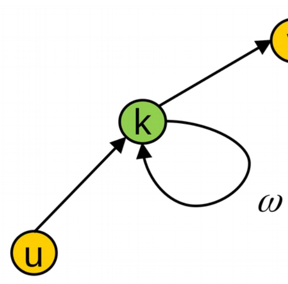
Condition d'existence

Théorème 1 (admis)

La recherche des plus courts chemins dans un graphe est un problème polynomial si et seulement si il n'existe pas de circuits de longueur négative dans ce graphe.

Théorème 2

Il existe des plus courts chemins de longueur finie dans un graphe si et seulement si il n'existe pas de circuits de longueur négative dans ce graphe.



$d(\omega)<0$ *Circuit absorbant*

Plus Courts Chemins

Graphes à valuations quelconques

Algorithme de Berge

d(s) 0, pred(s)=s
Pour tout $x \in X-\{s\}$
$\quad$ $d(x) = +\infty$
FinPour
Tant que $\exists (i,j) \in U, d(j) > d(i) + c_{ij}$ faire
$\quad$ $d(j) = d(i) + c_{ij}$
$\quad$ pred(j) = i
Finttq

Complexité

En l'absence de circuit absorbant, l'algorithme de Berge est de complexité $O(n^2C)$ avec :

$$C = \max_{(i,j) \in U} |c_{ij}|$$

Plus Courts Chemins

Graphes à valuations quelconques

Convergence de l'algorithme de Berge :

En l'absence de circuit absorbant, l'algorithme de Berge converge en un nombre fini d'étapes vers un système de potentiels optimal. En effet, à chaque étape de l'algorithme, d(i) est un majorant du PCC de s à i. G étant supposé sans circuit absorbant, d(i) est borné inférieurement par la longueur d'un PCC se s à i. Comme d(i) décroit tout au long de l'algorithme et est borné inférieurement, il admet une limite et l'algorithme converge. Il reste à montrer que cette limite est la longueur d'un PCC de s à i. Soit i_1 le dernier sommet utilisé pour réviser d(i). On a : $d(i)=d(i_1)+c_{i,i1}$. De la même façon, soit i_2 le dernier sommet utilisé pour réviser i_1. On a : $d(i_1)=d(i_2)+c_{i2,i1}$. De proche en proche, on met ainsi en évidence une suite C de sommets le dernier sommet étant s. Cette suite est non nécessairement élémentaire, mais comme G est sans circuit absorbant, il est possible d'extraire de C une suite de sommets deux à deux distincts $(i=i_0,i_1,i_2,\ldots,i_q=s)$ vérifiant :

$$d(i) = c_{i,i_1} + c_{i_1,i_2} + \ldots + c_{i_{q-1},i_q}$$ car $i_q=s$ et $d(s)=0$.

Plus Courts Chemins

Graphes à valuations quelconques

Convergence de l’algorithme de Berge :

d(i) représente donc bien la longueur d’un chemin de s à i. C’est également un PCC. En effet, soit $\mu=(j_0=i,j_1,\ldots,j_p=s)$ un autre chemin de s à i dans G. En sortie d’algorithme, on a :

$$\forall (x,y)\in U,\ d(y)-d(x) \leq c_{xy}.$$

On obtient alors :

$$\begin{cases} d(i)-d(j_1) \leq c_{j_1,s} \\ d(j_1)-d(j_2) \leq c_{j_2,j_1} \\ \ldots \\ d(j_{p-1})-d(j_p) \leq c_{j_p,j_{p-1}} \end{cases}$$

$$\text{soit } d(i) \leq c(\mu) = c_{j_p,j_{p-1}} + \ldots + c_{j_2,j_1} + c_{j_1,s} \text{ car } j_p = s \text{ et } d(s)=0$$

Plus Courts Chemins

Exercice : **Regroupement d'éléments (7)**

Considérons un ensemble de nombres $a_1, a_2, \cdots, a_n$ tels que $a_1 \leq a_2 \leq \cdots \leq a_n$. L'objectif est de partitionner ces éléments en groupes tels que :

- chaque groupe contient au moins p nombres,
- chaque groupe contient des nombres consécutifs de la liste $a_1, a_2, \cdots, a_n$,
- la somme des carrés des écarts entre chaque nombre et la moyenne de son groupe soit minimale. Soit $\bar{a}(S) = \frac{\sum_{i \in S} a_i}{|S|}$ la moyenne d'un ensemble S. La somme des carrés des écarts entre chaque nombre de S et la moyenne de S est égale à

$$\sum_{i \in S} (a_i - \bar{a}(S))^2$$

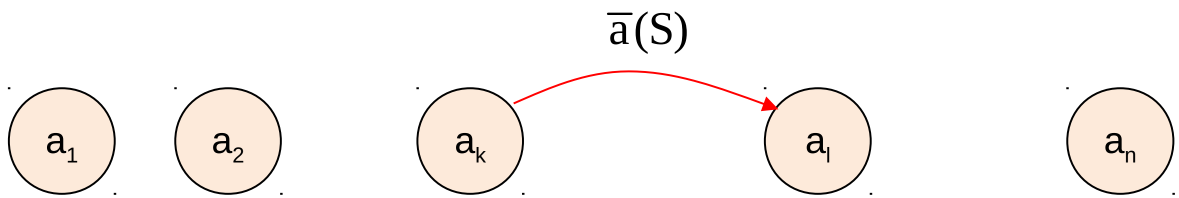
(a) Formaliser ce problème comme un problème de plus court chemin.

(b) Illustrer cette formalisation sur l'exemple suivant :

$$p = 2, n = 6, a_1 = 0.5, a_2 = 0.8, a_3 = 1.1, a_4 = 1.5, a_5 = 1.6, a_6 = 2$$

Plus Courts Chemins

Exercice : **Regroupement d'éléments (7)**



$$S = \{k, k+1, \ldots, l\}$$

$$\boxed{l-k+1 \geq p}$$

Plus Courts Chemins

Graphes à valuations quelconques

Algorithme de programmation dynamique reposant sur la relation : $d(i) = \min\left\{ d(i), \min_{(j,i)\in U} d(j) + c_{ji} \right\}$

Algorithme de Bellman

```
Pour tout x∈X faire
  d(x) = +∞, e(x)=0
FinPour
d(s)=0, pred(s)=s
Tant que ¬Q.Vide() faire
  Q.défiler(i), e(i)=e(i)+1
  Pour (i,j)∈U faire
    Si d(j) > d(i) + c_ij alors
      d(j) = d(i) + c_ij
      Si e(j) ≥ n alors
        Circuit Négatif : Stop
      FinSi
      pred(j) = i
      Si j∉Q alors
        Q.enfiler(j)
    FinSi
  FinPour
Fin ttq
```

Q : File = Liste FIFO

- G=(X,U) : graphe valué : longueur de l'arc (i,j) : c_{ij} quelconque
- s : sommet de départ
- d(x) : longueur du plus court chemin de s au sommet x
- pred(x) : prédécesseur de x dans le Plus Court Chemin associé
- e(x) : nombre d'évaluations du sommet x
- S : sommets déjà traités

Plus Courts Chemins

Graphes à valuations quelconques

Convergence

En l'absence de circuit absorbant, un sommet est évalué au plus n-1 fois. Les potentiels d(i) sont donc stabilisés en au plus n-1 itérations.

Complexité

La complexité de l'algorithme de Bellman est O(mn)

Détection de circuits négatifs

$$C = \max_{(i,j)\in U} |c_{ij}|$$

Plusieurs résultats existent. Deux méthodes sont proposées ici :

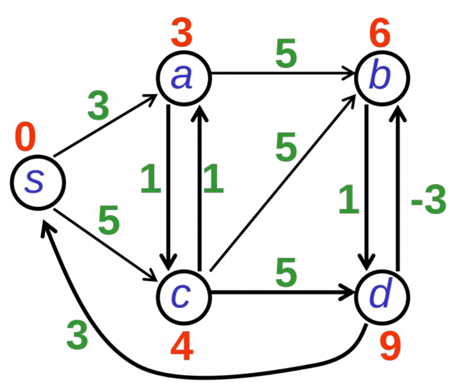
1. Un plus court chemin P comporte au plus n arcs $\rightarrow$ l(P) $\geq$ -nC. Donc, si d(i) <-nC, alors il existe un circuit négatif.
2. sur un graphe sans circuits négatifs, l'algorithme de Bellman traite au plus (n-1) fois chaque sommet. Par conséquent, le traitement pour la $n^{ième}$ fois d'un sommet montre l'existence d'un circuit négatif.

Plus Courts Chemins

Algorithme : **Bellman-Ford**

Exemple

1




```
Pour tout x∈X faire
| d(x) = +∞, e(x)=0
FinPour
d(s)=0, pred(s)=s
Tant que ¬Q.Vide() faire
| Q.défiler(i), e(i)=e(i)+1
| Pour (i,j)∈U faire
| | Si d(j) > d(i) + c_ij alors
| | | d(j) = d(i) + c_ij
| | | Si e(j) ≥n alors
| | | | Circuit Négatif : Stop
| | | FinSi
| | | pred(j) = i
| | | Si j∉Q alors
| | | | Q.enfiler(j)
| | FinSi
| FinPour
Fin ttq
```

Circuit absorbant
(b,d,b) → longueur -2

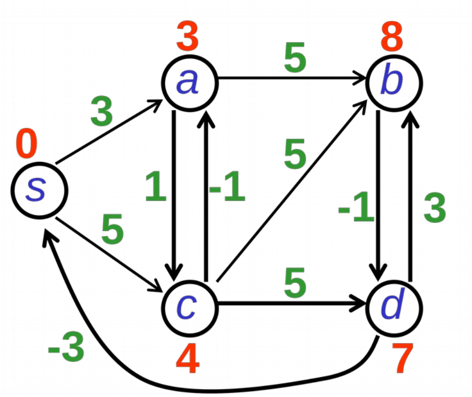
File Q	Nœud traité i	Arc (i,j)	Potentiel d(j)
s	s	(s,a)	3
a		(s,c)	5
a,c	a	(a,b)	8
c,b		(a,c)	4
c,b	c	(c,a)	3
b		(c,b)	8
b		(c,d)	9
b,d	b	(b,d)	9
d	d	d,s)	0
-		(d,b)	6
b	b	(b,d)	7
d	d	(d,b)	4
b	b	(b,d)	5
d	d	(d,b)	2
b	b	(b,d)	3

Plus Courts Chemins

Algorithme : **Bellman-Ford**

Exemple

2




```
Pour tout x∈X faire
  d(x) = +∞, e(x)=0
FinPour
d(s)=0, pred(s)=s
Tant que ¬Q.Vide() faire
  Q.défiler(i), e(i)=e(i)+1
  Pour (i,j)∈U faire
    Si d(j) > d(i) + c_ij alors
      d(j) = d(i) + c_ij
      Si e(j) ≥n alors
        Circuit Négatif : Stop
      FinSi
      pred(j) = i
      Si j∉Q alors
        Q.enfiler(j)
    FinSi
  FinPour
Fin ttq
```

Pas de réduction possible : coûts corrects

File Q	Nœud traité i	Arc (i,j)	Potentiel d(j)
s	s	(s,a)	3
a		(s,c)	5
a,c	a	(a,b)	8
c,b		(a,c)	4
c,b	c	(c,a)	3
b		(c,b)	8
b		(c,d)	9
b,d	b	(b,d)	7
d	d	(d,s)	0
-		(d,b)	8

Plus Courts Chemins

Exercice : **Cycle de coût moyen minimum (2)**

Soient $G = (X, U)$ un graphe et c_{ij} le coût d'un arc $(i, j) \in U$. Un second paramètre τ_{ij} est associé à chaque arc $(i, j) \in U$. L'objectif du problème est de trouver un cycle de coût moyen minimum, le coût moyen d'un cycle μ étant

$$c(\mu) = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}}$$

Nous supposons que toutes les données du problème sont entières et que $\forall (i, j) \in U, \tau_{ij} > 0$. Soit λ^* le coût moyen du cycle de coût moyen minimal.

Pour toute valeur de λ, nous définissons une longueur $l_{ij} = c_{ij} - \lambda\tau_{ij}$ associée à chaque arc $(i, j) \in U$.

(a) Montrer que dans le graphe G muni des longueurs l_{ij}, si le circuit μ de longueur minimale est de longueur strictement négative alors $\lambda > \lambda^*$.

(b) Montrer que dans le graphe G muni des longueurs l_{ij}, si le circuit μ de longueur minimale est de longueur nulle alors le cycle μ est le cycle optimal de coût moyen λ.

(c) Montrer que dans le graphe G muni des longueurs l_{ij}, si le circuit μ de longueur minimale est de longueur strictement positive alors $\lambda < \lambda^*$.

(d) En déduire un algorithme permettant de trouver un cycle de coût moyen minimal.

Plus Courts Chemins

Exercice : **Cycle de coût moyen minimum (2)**

$$c(\mu) = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}}$$

$$\forall \mu, c(\mu) = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}} \geq c(\mu^*) = \lambda^*$$

$$\sum_{(i,j)\in\mu} (c_{ij} - \lambda\tau_{ij}) < 0 \Leftrightarrow \sum_{(i,j)\in\mu} c_{ij} - \lambda \sum_{(i,j)\in\mu} \tau_{ij} < 0 \Leftrightarrow \lambda^* \leq c(\mu) = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}} < \lambda$$

$$\sum_{(i,j)\in\mu} (c_{ij} - \lambda\tau_{ij}) > 0 \Leftrightarrow \sum_{(i,j)\in\mu} c_{ij} - \lambda \sum_{(i,j)\in\mu} \tau_{ij} > 0 \Leftrightarrow \lambda < \lambda^* \leq c(\mu) = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}}$$

$$\sum_{(i,j)\in\mu} (c_{ij} - \lambda\tau_{ij}) = 0 \Leftrightarrow \sum_{(i,j)\in\mu} c_{ij} - \lambda \sum_{(i,j)\in\mu} \tau_{ij} = 0 \Leftrightarrow \lambda = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}}$$

$$\sum_{(i,j)\in\mu} (c_{ij} - \lambda\tau_{ij}) = 0 \Rightarrow \begin{cases} \sum_{(i,j)\in\mu} (c_{ij} - \lambda\tau_{ij}) \geq 0 \Rightarrow \lambda = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}} = \lambda^* \\ \sum_{(i,j)\in\mu} (c_{ij} - \lambda\tau_{ij}) \leq 0 \Rightarrow \lambda = \frac{\sum_{(i,j)\in\mu} c_{ij}}{\sum_{(i,j)\in\mu} \tau_{ij}} \geq \lambda^* \end{cases}$$

Détailler algorithme

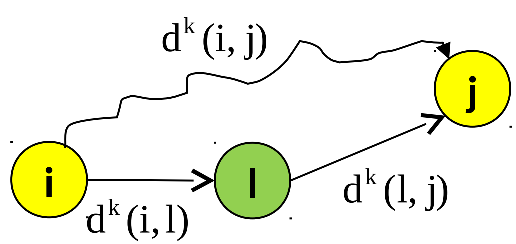
Plus Courts Chemins

Recherche des plus courts chemins entre chaque paire de sommets

Algorithme matriciel utilisant une matrice des distances d où d(i,j) représente la longueur du plus court chemin allant du sommet i au sommet j, et exploitant la propriété fondamentale suivante :

$$\forall (i,j) \in U, d^{k+1}(i,j) = \min_{l \in \Gamma_i^+ \cap \Gamma_j^-} \left\{ d^k(i,j), d^k(i,l) + d^k(l,j) \right\}$$

$d^k(i,j)$ = longueur d'un plus court chemin allant de i à j passant uniquement par les k premiers sommets du graphe.



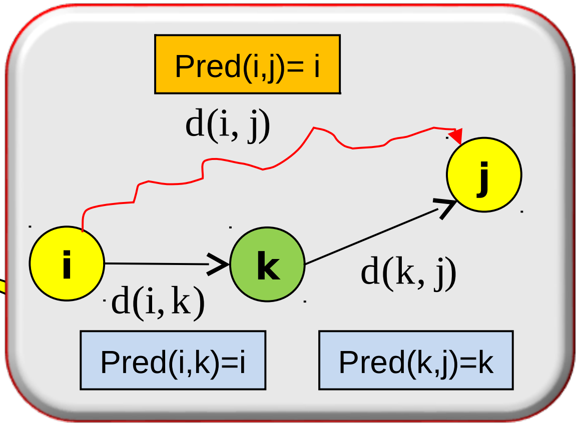
Plus Courts Chemins

Recherche des plus courts chemins entre chaque paire de sommets

Algorithme de Floyd-Warshall

```
Pour tout (i,j)∈U faire
   d(i,j) = c_ij
   pred(i,j) = i
FinPour
Pour tout (i,j)∉U faire
   d(i,j) = +∞
FinPour
Pour tout i∈X faire
   d(i,i) = 0
FinPour
```

```
Pour k=1 à n faire
   Pour i ∈X faire
      Pour j∈X faire
         Si d(i,j) > d(i,k) + d(k,j) alors
            d(i,j) = d(i,k) + d(k,j)
            pred(i,j) = pred(k,j)
         FinSi
      FinPour
   Fin Pour
FinPour
```

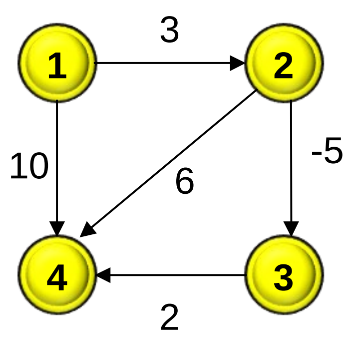


- $G=(X,U)$: graphe valué : longueur de l'arc (i,j) : c_{ij} quelconque
- $d(i,j)$: longueur du plus court chemin séparant les sommet s i et j
- $pred(i,j)$: prédécesseur de j dans le PCC de i à j
- S : sommets déjà traités

Plus Courts Chemins

Recherche des plus courts chemins entre chaque paire de sommets

Exemple



Pour k=1 à n faire
  Pour i ∈X faire
    Pour j∈X faire
      Si d(i,j) > d(i,k) + d(k,j) alors
        d(i,j) = d(i,k) + d(k,j)
        pred(i,j) = pred(k,j)
      FinSi
    FinPour
  Fin Pour
FinPour

d_{ij}	1	2	3	4
1	0	3	$+\infty$	10
2	$+\infty$	0	-5	6
3	$+\infty$	$+\infty$	0	2
4	$+\infty$	$+\infty$	$+\infty$	0

d^1_{ij}	1	2	3	4
1	0	3	$+\infty$	10
2	$+\infty$	0	-5	6
3	$+\infty$	$+\infty$	0	2
4	$+\infty$	$+\infty$	$+\infty$	0

d^2_{ij}	1	2	3	4
1	0	3	-2	9
2	$+\infty$	0	-5	6
3	$+\infty$	$+\infty$	0	2
4	$+\infty$	$+\infty$	$+\infty$	0

d^3_{ij}	1	2	3	4
1	0	3	-2	0
2	$+\infty$	0	-5	-3
3	$+\infty$	$+\infty$	0	2
4	$+\infty$	$+\infty$	$+\infty$	0

Plus Courts Chemins

Recherche des plus courts chemins entre chaque paire de sommets

Exemple

Solution Optimale :

d_{ij}^4	1	2	3	4
1	0	3	-2	0
2	$+\infty$	0	-5	2
3	$+\infty$	$+\infty$	0	-3
4	$+\infty$	$+\infty$	$+\infty$	0

Plus Courts Chemins

Exercice : **Propriétés des plus courts chemins (9)**

Soit $G = (X, U)$ un graphe orienté.
Soit d le vecteur des distances des plus courts chemins.

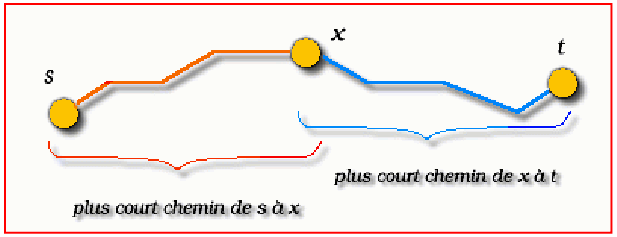
(a) Montrer que si $\mu = \{s, i_1, i_2, \cdots, i_h, k\}$ est un plus court chemin du sommet s au sommet k alors pour tout $q \in]h]$, le sous chemin $\mu' = \{s, i_1, i_2, \cdots, i_q\}$ est un plus court chemin du sommet s au sommet i_q ;

(b) Montrer qu'un chemin μ du sommet s au sommet k est un plus court chemin si et seulement si

$$\forall (i, j) \in \mu, d(j) = d(i) + c_{ij}$$

Plus Courts Chemins

Exercice : **Propriétés des plus courts chemins (9)**



CN $\mu = (s = i_o, i_1, \dots, i_q = k)$ PCC de s à k $\quad \exists (i,j) \in \mu, d(j) < d(i) + c_{ij} \qquad \forall (i,j)\, \Sigma\, U, d(j) \quad d(i) + c_{ij}$

$$d(j) < d(i) + c_{ij} \Rightarrow \begin{cases} \mu_1 = PCC(s,j) \Rightarrow c(\mu_1) = d(j) \\ \mu_2 = PCC(j,k) \Rightarrow c(\mu_2) \end{cases} \qquad c(\mu_1) + c(\mu_2) = d(j) + c(\mu_2) < c(\mu) = d(i) + c_{ij} + c(\mu_2)$$

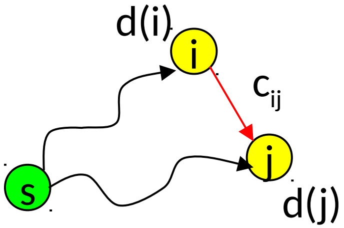
CS $d(k) = c_{i_q, i_{q-1}} + \dots + c_{i_2, i_1} + c_{i_1, s}$ car $i_o = s$ et $d(s) = 0$ $\rightarrow$ Longueur d'un chemin de s à k $\rightarrow$ PCC ?

En effet, soit $\mu' = (j_0 = s, j_1, \dots, j_p = k)$ un autre chemin de s à k dans G. En sortie d'algorithme, on a :

$$\forall (x,y) \in U,\ d(y) - d(x) \leq c_{xy}.$$

On obtient alors :

$$\begin{cases} d(j_o) - d(j_1) \leq c_{j_1, s} \\ d(j_1) - d(j_2) \leq c_{j_2, j_1} \\ \dots \\ d(j_{p-1}) - d(j_p) \leq c_{j_p, j_{p-1}} \end{cases}$$



soit $d(k) = d(j_p) \leq c(\mu') = c_{j_p, j_{p-1}} + \dots + c_{j_2, j_1} + c_{j_1, s}$ car $j_p = s$ et $d(s) = d(j_o) = 0$

Plus Courts Chemins

Exercice : **Propriétés de l'algorithme de Dijkstra (10)**

(a) Montrer que chaque nouveau sommet i inséré dans l'ensemble S vérifie :

$$d(i) \geq Max_{j \in S} d(j)$$

(b) A une itération donnée, soit $K = \{i \in \bar{S}\ d(i) = Min_{j \in \bar{S}} d(j)\}$. Montrer que la longueur des plus courts chemins de s à chaque sommet de K est égale à $Min_{j \in S} d(j)$.

(c) En déduire une amélioration de l'algorithme de Dijkstra.

Plus Courts Chemins

Exercice : **Propriétés de l'algorithme de Dijkstra (10)**

d(s) =0, pred(s)=s
S=∅, $\bar{S}$=X
Pour tout x∈X-{s}
 | d(x) =+∞
FinPour
Tant que S≠X faire
 | Sélectionner un sommet i de S tel que : $d(i) = \min_{j \in \bar{S}} d(j)$
 | S=S∪{i}, $\bar{S}=\bar{S}$-{i}
 | Pour tout j∈Γ^+(i)∩$\bar{S}$
 | | Si d(j) > d(i) + c_{ij} alors
 | | | d(j) = d(i) + c_{ij} , pred(j)=i
 | | FinSi
 | FinPour
Finttq

a) Sinon i aurait été sélectionné plutôt
b) Chaque sommet de K réalise sa marque optin
c) Evident

Plus Courts Chemins

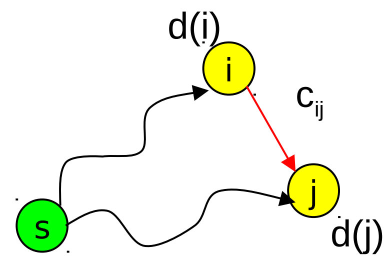
Exercice : **Chemin de capacité maximale (11)**

Soit c_{ij} la capacité d'un arc dans un graphe $G = (X, U)$. La capacité d'un chemin μ allant de s à k est égale à $min_{(i,j)\in\mu} c_{ij}$. Le problème du chemin de capacité maximale est de déterminer un chemin de capacité maximale d'un sommet s à tous les autres sommets.

(a) Quelques modifications sont à effectuer sur l'algorithme de Dijkstra pour résoudre ce problème.
(b) Justifier cet algorithme.

Plus Courts Chemins

Exercice : **Chemin de capacité maximale (11)**



$$c(\mu) = \min_{(i,j)\in\mu} c_{ij}$$

$$d(j) = \max\{d(j), \min(d(i), c_{ij})\}$$

d(s) =0, pred(s)=s
S=∅, $\bar{S}$=X
Pour tout x∈X-{s}
  d(x) =+∞ → d(x)=0
FinPour
Tant que S≠X faire
  Sélectionner un sommet i de S tel que : $d(i) = \min_{j\in\bar{S}} d(j)$ → d(i) = max $_{j\in X\text{-}S}$ d(j)
  S=S∪{i}, $\bar{S}$=$\bar{S}$-{i}
  Pour tout j∈Γ^+(i)∩$\bar{S}$
    Si d(j) > d(i) + c_{ij} alors → d(j) < min{d(i),c_{ij}}
      d(j) = d(i) + c_{ij} , pred(j)=i → d(j) = min{d(i),c_{ij}}
    FinSi
  FinPour
Finttq

Plus Courts Chemins

Exercice : **Chemin de capacité maximale (11)**

Chemins de capacité maximale

Convergence de l'algorithme de Dijkstra :

Montrons qu'à chaque étape de la procédure, les potentiels associés aux sommets de S sont optimaux, et correspondent à des PCC empruntant uniquement des sommets de S

A une étape quelconque de l'algorithme, soit i le dernier sommet de X-S inséré dans S. On a :

capacité

$d(i) \leq \min_{j \in X\text{-}S} d(j)$

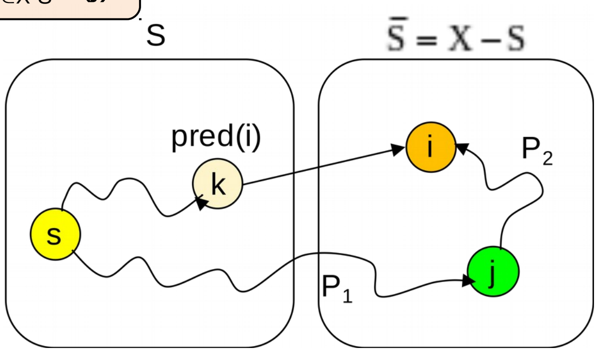
$d(i) \geq \max_{j \in X\text{-}S} d(j)$

Montrons alors que la longueur de tout chemin allant de s à i et empruntant des sommets de X-S est supérieure ou égale à d(i). Soit P un tel chemin. Il peut en évidence être partitionné en deux sous-chemins P_1 et P_2, P_1 partant de s et aboutissant en $j \in X\text{-}S$ en empruntant que des sommets de S (à l'exception de j). On a alors

$c(P_1) \geq d(j)$

$c(P_1) \leq d(j)$

inférieure



$$d(i) = \max\left\{ d(i), \min(d(j), c_{ji}) \right\}$$

40

Plus Courts Chemins

Exercice : **Chemin de capacité maximale (11)**

$d(i) \geq d(k)$

$\min_{(i,j)\in P_2} c_{ij}$

Convergence de l'algorithme de Dijkstra :

Les valuations étant par ailleurs toutes positives, on a $c(P_2) \geq 0$ Comme $d(i) \leq d(k)$, $\forall k \in X-S$, on en déduit que $c(P) = c(P_1) + c(P_2) \geq d(i)$, et d(i) est la longueur d'un PCC joignant s à i. Ces potentiels correspondent d'autre part à des PCC en constatant (un simple raisonnement par récurrence en atteste) qu'après introduction de i dans S, les arcs (i,j) sont scannés et on impose $d(j) = d(i) + c_{ij}$ dès lors que $d(j) > d(i) + c_{ij}$, ce qui satisfait la proposition 2.

Chemin de capacité maximale

$d(j) = \min\{d(i),c_{ij}\}$

$c(P) = \mathrm{Min}\{c(P_1);c(P_2)\} \leq d(i)$

$d(j) < \min\{d(i),c_{ij}\}$

Chemins de capacité maximale

Plus Courts Chemins

Exercice : **Chemin de fiabilité maximale (12)**

Dans un graphe $G = (X, U)$, à chaque arc (i, j) est associée une probabilité d'existence, $0 < \mu_{ij} \leq 1$. La probabilité d'existence d'un chemin est le produit des probabilités d'existence des arcs qui le compose ($\mu(P) = \prod_{(i,j)\in P} \mu_{ij}$). L'objectif est de déterminer les chemins les plus probables allant du sommet s à tous les autres sommets.

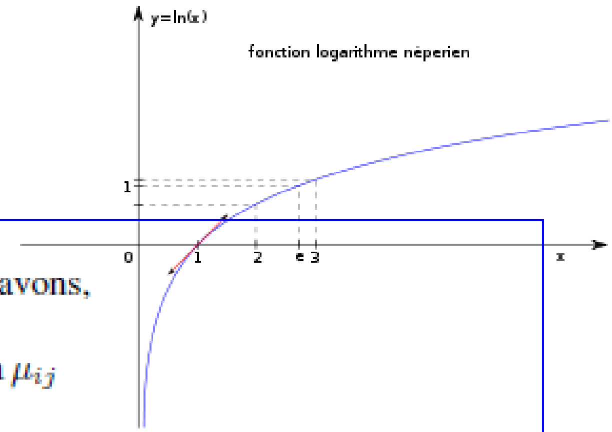
(a) Montrer que l'utilisation des logarithmes permet de réduire la recherche des chemins les plus probables à la recherche de plus courts chemins.

(b) Supposons que les logarithmes soient interdits. Modifier l'algorithme de Dijkstra pour résoudre le problème.

(c) Justifier cet algorithme.

(d) Vérifier si l'algorithme de Dijkstra peut résoudre le problème sachant que certaines probabilités sont remplacées par des coefficients $\mu_{ij} > 1$.

Plus Courts Chemins



Exercice : **Chemin de fiabilité maximale (12)**

Exercise 12(a) Par défintion, $\ln ab = \ln a + \ln b$. Par conséquent nous avons,

$$\ln \mu(P) = \ln\Big(\prod_{(i,j)\in P} \mu_{ij}\Big) = \sum_{(i,j)\in P} \ln \mu_{ij}$$

La fonction $\ln$ étant négative sur $]0;1[$, nous utilisons le résultat suivant pour résoudre le problème : $\max f(x) = -\min -f(x)$. De la sorte, nous nous ramenons à un problème de plus courts chemins avec des longueurs positives. □

Exercise 12(b) La longueur du chemin étant le produit des longueurs des arcs, l'opérateur $+$ est remplacé par l'opérateur $\times$. De plus, nous recherchons un plus long chemin au lieu d'un plus court chemin, donc dans l'algorithme de Dijkstra, le $\min_{j\in S} d(j)$ devient un $\max_{j\in S} d(j)$ et le test Si $d(j) > d(i) + \mu_{ij}$ devient Si $d(j) < d(i) \times \mu_{ij}$. □

Exercise 12(d) Non, car avec des coefficients $\mu_{ij} > 1$, il sera possible de trouver un chemin plus long que celui vérifiant le $\max_{j\in S} d(j)$. □

Plus Courts Chemins

Exercice : Chemin de fiabilité maximale (12)

Exercise 12(c) Par souci de clareté, nous supposons le calcul des chemins de fiabilité maximale à partir du sommet s.
Raisonnons par récurrence pour montrer que

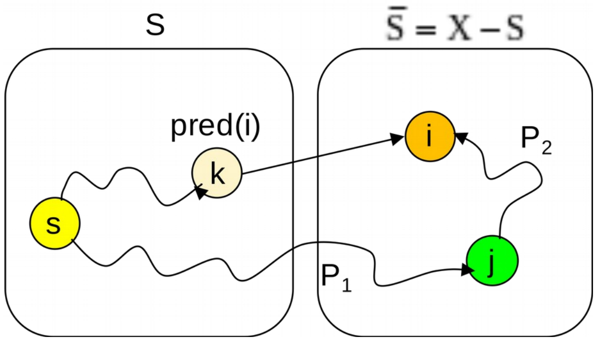
1. les distances associées aux sommets de S sont optimales,
2. les distances associées aux sommets de $\bar{S}$ correspondent aux longueurs des chemins de fiabilité maximale empruntant uniquement les sommets de S.

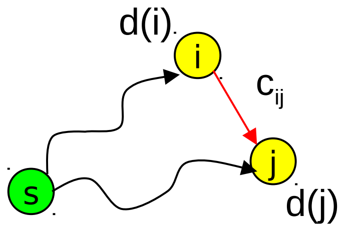
Montrons la première relation.
Soit i le sommet de $\bar{S}$ inséré dans S.
D'après l'algorithme de Disjkstra modifié, $d(i) = \max_{j \in \bar{S}} d(j)$. Nous allons montrer que la longueur de tout chemin allant de s à i et empruntant des sommets de $\bar{S}$ est inférieure ou égale à $d(i)$.
Soit P un chemin allant de s à i et empruntant au moins un sommet de $\bar{S}$.
Ce chemin peut se décomposer en deux segments P_1 et P_2 où P_1 se termine en k un sommet de $\bar{S}$, seul sommet d P_1 appartenant à $\bar{S}$.

D'après l'hypothèse de récurrence, la longueur de P_1 est inférieure à $d(k)$, et puisque le sommet i vérifie $d(i) = \max_{j \in \bar{S}} d(j)$, $d(k) \leq d(i)$.
Donc le longueur de P_1 est inférieure ou égale à $d(i)$. De plus, les arcs étant tous de longueurs comprises entre 0 et 1, la longueur de P_2 est comprise entre 0 et 1. En conséquence, la longueur du chemin P est inférieure ou égale à $d(i)$. Nous en déduisons que $d(i)$ est la longueur chemin de fiabilité maximale allant de s à i.

Montrons que l'algorithme conserve la seconde hypothèse de récurrence.
A chaque itération, après avoir introduit le sommet i dans S, l'algorithme examine les arcs $(i, j) \in U$ et si $d(j) < d(i) \times c_{ij}$ alors il impose $d(j) = d(i) \times c_{ij}$.
En conséquence, d'après l'hypothèse de récurrence, le chemin allant de s à j vérifie la proposition ?? et donc $d(j)$ correspond à la longueur du chemin de fiabilité maximale allant de s à j tout en n'empruntant que les sommets de $S \cup \{i\}$.

□





Plus Courts Chemins

Exercice : **Propriétés des coûts réduits (14)**

Soit d un vecteur de distances entre le sommet s et les autres sommets. Le coût réduit c_{ij}^d d'un arc (i, j) est défini comme suit :

$$c_{ij}^d = c_{ij} + d(i) - d(j)$$

(a) Montrer que pour tout circuit μ, $\sum_{(i,j)\in\mu} c_{ij}^d = \sum_{(i,j)\in\mu} c_{ij}$.

(b) Montrer que pour tout chemin P allant du sommet k au sommet l,

$$\sum_{(i,j)\in P} c_{ij}^d = \sum_{(i,j)\in P} c_{ij} + d(k) - d(l)$$

(c) Si d représente le vecteur des plus courts chemins, montrer que

$$\forall (i, j) \in U, c_{ij}^d \geq 0$$

Plus Courts Chemins

Exercice : **Propriétés des coûts réduits (14)**

a

$$\mu = (i_o, i_1, \dots, i_q = i_o) \Rightarrow \bar{c}(\mu) = \sum_{(i,j)\in\mu} \bar{c}_{ij}$$

$$\bar{c}(\mu) = (c_{i_o,i_1} + d_{i_o} - d_{i_1}) + (c_{i_1,i_2} + d_{i_1} - d_{i_2}) + \dots + (c_{i_{q-1},i_q} + d_{i_{q-1}} - d_{i_q})$$

$$\bar{c}(\mu) = c_{i_o,i_1} + c_{i_1,i_2} + \dots + c_{i_{q-1},i_q} = c(\mu)$$

b

$$P = (i_o = k, i_1, \dots, i_q = l) \Rightarrow \bar{c}(\mu) = \sum_{(i,j)\in P} \bar{c}_{ij}$$

$$\bar{c}(P) = (c_{i_o,i_1} + d_{i_o} - d_{i_1}) + (c_{i_1,i_2} + d_{i_1} - d_{i_2}) + \dots + (c_{i_{q-1},i_q} + d_{i_{q-1}} - d_{i_q})$$

$$\bar{c}(P) = d_k + c_{i_o,i_1} + c_{i_1,i_2} + \dots + c_{i_{q-1},i_q} - d_l = c(P) + d_k - d_l$$

b

On a alors : $\forall (i,j) \in U,\ d_j \quad d_i \quad c_{ij} \quad \bar{c}_{ij} \quad c_{ij} \quad d_i \quad d_j \quad 0$

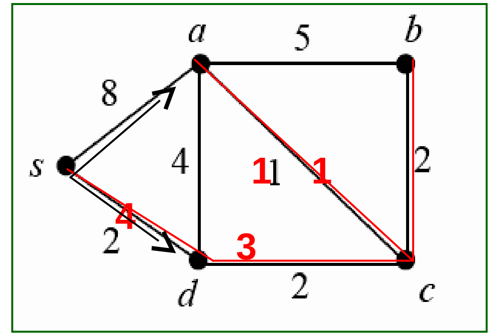
Plus Courts Chemins

Exercice : **Propriétés des coûts réduits (14)**

$$\min \sum_{(i,j)\in U} c_{ij}x_{ij}$$

$$Sc\begin{cases} \displaystyle\sum_{j/(i,j)\in U} x_{ij} - \sum_{j/(j,i)\in U} x_{ij} = \begin{cases} n-1, & i=s \\ -1, & i\in X-\{s\} \end{cases} \\ \forall (i,j)\in U,\ x_{ij}\in\{0,1\} \end{cases}$$

Arbre des PCCs



	(s,a)	(s,d)	(a,b)	(a,c)	(a,d)	(b,a)	(b,c)	(c,a)	(c,b)	(c,d)	(d,a)	(d,c)	b	
s	-1	-1											4	d_s
a	1		-1	-1	-1	1		1			1		-1	d_a
b			1			-1	-1		1				-1	d_b
c				1			1	-1	-1	-1		1	-1	d_c
d		1			1					1	-1	-1	-1	d_d

$$\boxed{\overline{c}_{ij} = c_{ij} + d_i - d_j}$$

Plus Courts Chemins

Exercice : **Propriétés des coûts réduits (14)**

$$[P] : \min \sum_{(i,j)\in U} c_{ij} x_{ij}$$

$$Sc \begin{cases} \displaystyle\sum_{j/(i,j)\in U} x_{ij} - \sum_{j/(j,i)\in U} x_{ij} = \begin{cases} -1, i = s \\ 1, i \in X - \{s\} \end{cases} \\ \forall (i,j) \in U, x_{ij} \in \{0,1\} \end{cases}$$

$$[D] : \max \sum_{i \in X} b_i d_i$$

$$Sc \begin{cases} \forall (i,j) \in U, d_j - d_i \leq c_{ij} \\ \forall i \in X, d_i \ qcq \end{cases}$$

PRIMAL (Max)	DUAL (Min)
Second Membre	**Fonction Economique**
Matrice des Contraintes A	Matrice des Contraintes ${}^tA$
Contrainte n°i $\leq$	Variable $u_i \geq 0$
Contrainte n°i $\geq$	Variable u_i
Contrainte n°i =	Variable u_i non astreinte
Variable $x_i \geq 0$	Contrainte n°i $\geq$
Variable $x_i \leq$	Contrainte n°i $\leq$
Variable x_i non astreinte	Contrainte n°i =

Plus Courts Chemins

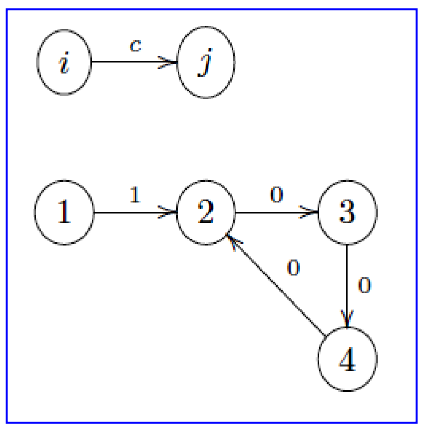
Exercice : **Equations de Bellman (15)**

(a) Montrer que les distances $d()$ des plus courts chemins doivent satisfaire les équations suivantes :

$$\forall j \in X, d(j) = min_{(i,j)\in U(i)}\{d(i) + c_{ij}\}$$

(b) Montrer la proposition suivante :
si un ensemble de distances $d()$ vérifie les équations de Bellman et si le graphe ne contient aucun circuit de longueur zéro, alors $d()$ représente les distances des plus courts chemins.

(c) Montrer sur l'exemple de la figure 4 que les distances $d = (0, 0, 0, 0)$ vérifient les conditions de Bellman mais ne représentent pas les distances des plus courts chemins.



$\forall (i, j) \Sigma U, d_j \quad d_i + c_{ij}$

Plus Courts Chemins

Exercice : **Analyse de sensibilité (16)**

Soit un graphe $G = (X, U)$ sans circuits négatifs. Pour chaque sommet j de G, d_{ij} représente la longueur d'un plus court chemin allant de i à j.

(a) Montrer qu'un arc (i, j) appartient à un plus court chemin allant de s à t si et seulement si $d_{st} = d_{si} + c_{ij} + d_{tj}$.

Algorithme 2 Analyse de sensibilité

si $d_{qp} + c'_{pq} < 0$ **alors**
  le graphe possède un circuit négatif ;
sinon
  pour chaque paire [i,j] de sommets **faire**
    $d_{ij} = min\{d_{ij}, d_{ip} + c'_{pq} + d_{qj}\}$;
  fin pour
fin si

(b) Montrer que $d_{st} = min_{(i,j)\in U}\{d_{si} + c_{ij} + d_{jt}\}$.

(c) Supposons que la longueur d'un arc (p, q) décroît de c_{pq} à c'_{pq}. Montrer que l'algorithme 2 permet de calculer les nouvelles distances des plus courts chemins entre chaque paire de sommets.

Plus Courts Chemins

Exercice : **Algorithme de Sedgewick et Vitter (18)**

Cet algorithme a été conçu pour des graphes $G = (X, U)$ euclidiens, c'est-à-dire des graphes non orientés dont les sommets sont des points d'un espace euclidien et les arêtes, des segments entre points valués par la longueur euclidienne du segment (distance entre points). Il est conçu pour calculer le plus court chemin entre deux sommets s et t.

l_{ij} désigne la distance euclidienne du sommet i au sommet j. Pour la calculer, il faut connaître les coordonnées x_i et y_i de tout point i :

$$l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

La valeur l_{ij} représente la longueur de l'arête (i, j) si celle-ci existe. Dans l'algorithme suivant, $d()$ et $b()$ sont des tableaux. L'ensemble S est l'ensemble des sommets restant à traiter.

Plus Courts Chemins

Exercice : **Algorithme de Sedgewick et Vitter (18)**

Algorithme 3 Sedgewick et Vitter

$\forall i \in X, d(i) = +\infty$
$d(s) = 0$
$b(s) = l_{st}$
$S = \{s\}$
répeter
  Soit $i \in S$ tel que $b(i) = \min_{k \in S} b(k)$
  $S = S \setminus \{i\}$
  pour $j \in \Gamma_i^+$ **faire**
    si $d(i) + l_{ij} < d(j)$ **alors**
      $d(j) = d(i) + l_{ij}$
      $b(j) = d(j) + l_{jt}$
      si $j \notin S$ **alors**
        $S = S \cup \{j\}$
      fin si
    fin si
  fin pour
jusqu'à $(S = \emptyset)$ ou $(i = t)$

Plus Courts Chemins

Exercice : **Algorithme de Sedgewick et Vitter (18)**

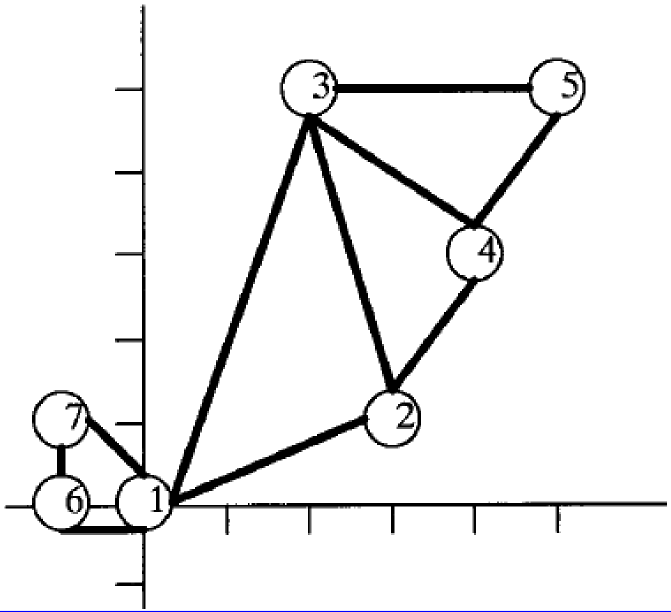
	1	2	3	4	5	6	7
1	0	3.16	5.38	5	7.07	1	1.41
2		0	4.12	2.23	4.47	4.12	4
3			0	2.82	3	5.83	5
4				0	2.23	5.83	5.38
5					0	7.81	6.92
6						0	1
7							0

Matrice des distances

Plus Courts Chemins

Exercice : Algorithme de Sedgewick et Vitter (18)

Pour les trois premières questions, on se propose de travailler sur le graphe $G = (X, U)$ (figure 5) à 7 sommets suivant : (Les arêtes sont en gras sur le dessin, alors que le repère du plan est dessiné en traits fins)



Plus Courts Chemins

Exercice : **Algorithme de Sedgewick et Vitter (18)**

(a) Calculer le plus court chemin allant de 1 à 5 avec l'algorithme de Dijkstra en détaillant une étape.

(b) Calculer le plus court chemin allant de 1 à 5 avec l'algorithme de Sedgewick et Vitter en détaillant une étape.

(c) Quelle est la principale différence entre les deux algorithmes ?

(d) Donner un graphe à n sommets pour lequel l'algorithme de Sedgewick et Vitter trouve le plus court chemin allant du sommet 1 au sommet 2 en deux étapes alors que l'algorithme de Dijkstra trouve le plus court chemin allant du sommet 1 au sommet 2 en n étapes. (une étape est l'exécution de toutes les instructions à l'intérieur de la boucle répéter).

(e) Montrer que $b(j) \geq b(i) \Rightarrow d(i) \leq d(j) + l_{ji}$.

(f) Soit i tel que $b(i) = min_{k \in S} b(k)$. Montrer que $d(i)$ est la longueur du plus court chemin allant du sommet s au sommet i.

(g) Quelle est la complexité dans le pire des cas de cet algorithme ?

Plus Courts Chemins