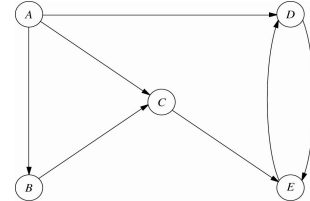


3. Optimisation de réseaux

- a. Graphes et réseaux
- b. Plus courts chemins
- c. Arbres de poids minimum
- d. Flot maximum

Graphe orienté

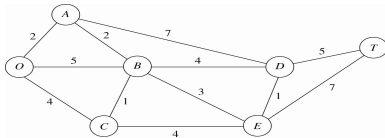
- Exemple : réseau de distribution



- Sommets : A, B, C, D, E
- Arcs : (A,B), (A,C), (A,D), (B,C), (C,E), (D,E), (E,D)

Graphe [non orienté]

- Exemple : Parc Seervada (H&L sec. 9.1)



- Sommets : O, A, B, C, D, E, T
- Arêtes : {O,A}, {O,B}, {O,C}, {A,B}, {A,D}, {B,C}, {B,D}, {B,E}, {D,E}, {D,T}, {E,T}
- Le nombre sur chaque arête représente la distance entre les deux sommets reliés par cette arête

Transformations

- *Graphe orienté dérivé* d'un graphe non orienté : obtenu en introduisant deux arcs pour chaque arête, un dans chaque direction
- *Graphe sous-jacent* à un graphe orienté : obtenu en enlevant l'orientation des arcs
- Si G est un graphe non orienté, le graphe sous-jacent au graphe orienté dérivé de G n'est pas G !
- De même, si G est un graphe orienté, le graphe dérivé du graphe sous-jacent à G n'est pas G !

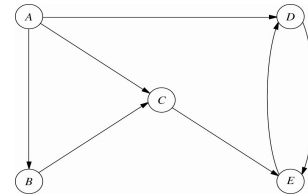
Chemins et circuits

- **Chemin** [chaîne] : suite d'arcs [d'arêtes] distinct[e]s reliant deux sommets
- **Chemin non orienté** : suite d'arcs distincts qui relient deux sommets lorsqu'on ne considère pas l'orientation des arcs
- En d'autres mots, un chemin non orienté est une chaîne dans le graphe sous-jacent
- **Circuit** [cycle] : chemin [chaîne] qui commence et finit au même sommet
- **Circuit non orienté** : cycle dans le graphe sous-jacent

3. Optimisation de réseaux

5

Chemins et circuits (exemples)



- A->C->E->D = chemin (= chemin non orienté)
- A->D->E->C->B = chemin non orienté (\neq chemin)
- D->E->D = circuit (= circuit non orienté)
- A->B->C->A = circuit non orienté (\neq circuit)

3. Optimisation de réseaux

6

Connexité

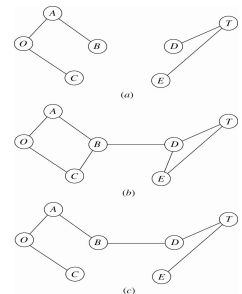
- Deux sommets sont **connexes** s'il existe au moins un chemin non orienté les reliant
- Un graphe est **connexe** si toute paire de sommets est connexe
- Le plus petit graphe connexe à n sommets possède $n-1$ arêtes
- On l'appelle un **arbre**
- Définition alternative : un **arbre** est un graphe connexe sans cycle
- **Arbre partiel** : arbre obtenu à partir d'un graphe connexe en incluant tous les sommets

3. Optimisation de réseaux

7

Arbre partiel

- Pas un arbre partiel : non connexe
- Pas un arbre partiel : possède des cycles
- Arbre partiel



3. Optimisation de réseaux

8

Flot dans un réseau

- Réseau : graphe orienté ayant
 - des capacités sur les arcs
 - des sommets d'offre (ou sources)
 - des sommets de demande (ou puits)
 - des sommets de transfert
- Flot dans un réseau : nombre d'unités circulant sur les arcs du réseau de façon à respecter les capacités et les contraintes de *conservation de flot*
- En chaque sommet, flot sortant – flot entrant =
 - Offre (si le sommet est une source)
 - -Demande (si le sommet est un puits)
 - 0 (en un sommet de transfert)

3. Optimisation de réseaux

9

Flot en langage mathématique

- x_{ij} = quantité de flot qui passe sur l'arc (i,j)

$$\sum_{(i,j) \in A^+(i)} x_{ij} - \sum_{(j,i) \in A^-(i)} x_{ji} = b_i \quad i \in V$$
$$0 \leq x_{ij} \leq u_{ij} \quad (i,j) \in A$$

- $b_i = 0$ (transfert), offre (source), -demande (puits)
- V = ensemble des sommets
- A = ensemble des arcs du réseau
- $A^+(i)$ = ensemble des arcs sortants du sommet i
- $A^-(i)$ = ensemble des arcs entrants au sommet i

3. Optimisation de réseaux

10

Problème du chemin le plus court

- On a un graphe non orienté et connexe
- Il y a deux sommets spéciaux
 - Source (ou origine) O
 - Puits (ou destination) T
- A chaque arête $\{i,j\}$, on associe une distance $c_{ij} \geq 0$
- On cherche le chemin non orienté (ou chaîne) le plus court reliant O à T
- Chemin le plus court : celui dont la distance totale (somme des distances des arêtes du chemin) est minimale parmi tous les chemins de O à T

3. Optimisation de réseaux

11

Algorithme de Dijkstra

- Méthode itérative
- À chaque itération, on choisit le sommet j le plus près de O et on fixe $d(j)$, la variable calculant la distance entre O et j (on dit que le sommet j est *marqué*)
- Au départ, O est marqué et $d(O) = 0$
- Le sommet le plus près est choisi parmi les sommets non marqués reliés à au moins un sommet marqué
- Le sommet choisi j est celui qui atteint
 - $\min_{\text{sommets } k \text{ non marqués}} \{ \min_{\text{sommets } i \text{ marqués}} d(i) + c_{ik} \}$
- $d(j)$ est fixée à cette valeur
- On arrête lorsque T est marqué

3. Optimisation de réseaux

12

Dijkstra (1930 – 2002)

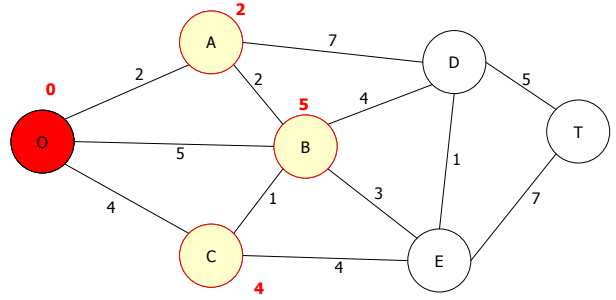


- Un membre très influent de la première génération des informaticiens
- Expertises:
 - Conception d'algorithmes
 - Langages de programmation
 - Conception de programmes
 - Systèmes d'opérations
 - Calcul distribué
 - Spécification formelle et vérification de programmes
 - Conception d'arguments mathématiques
- Pour plus d'information:
 - <http://www.cs.utexas.edu/users/EWD/>

3. Optimisation de réseaux

13

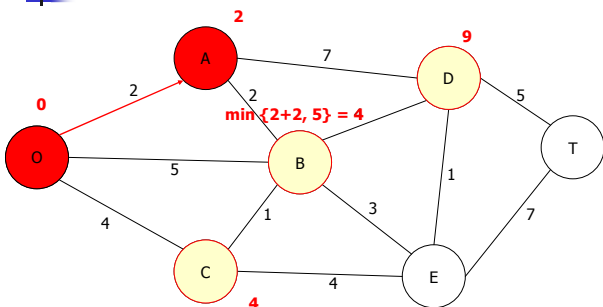
Algorithme de Dijkstra (étape 1)



3. Optimisation de réseaux

14

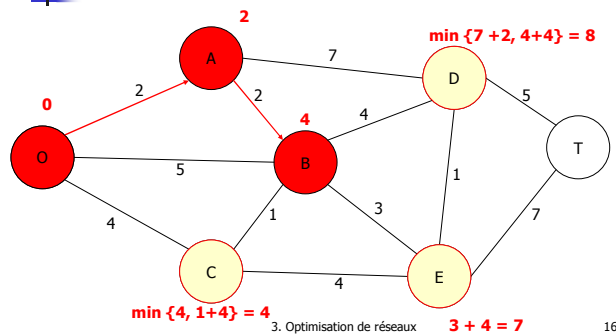
Algorithme de Dijkstra (étape 2)



3. Optimisation de réseaux

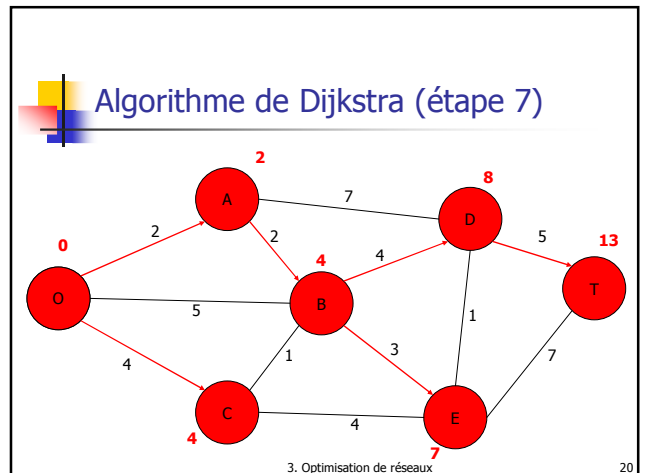
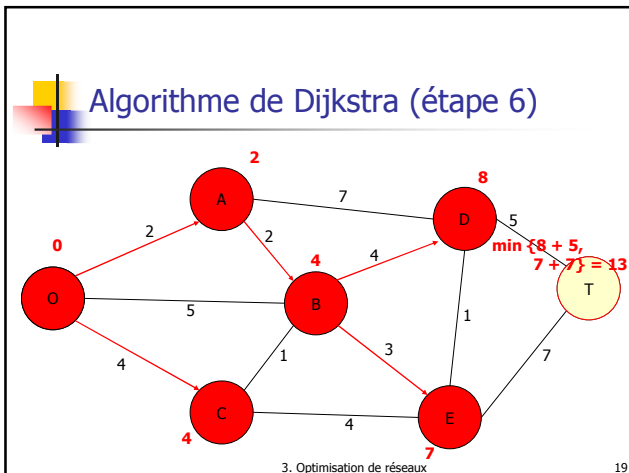
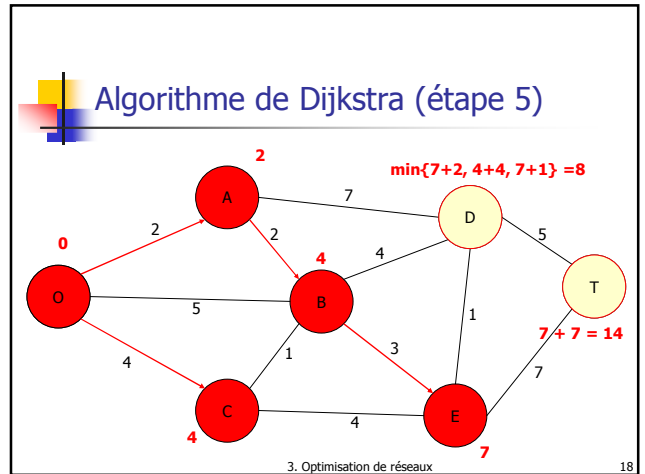
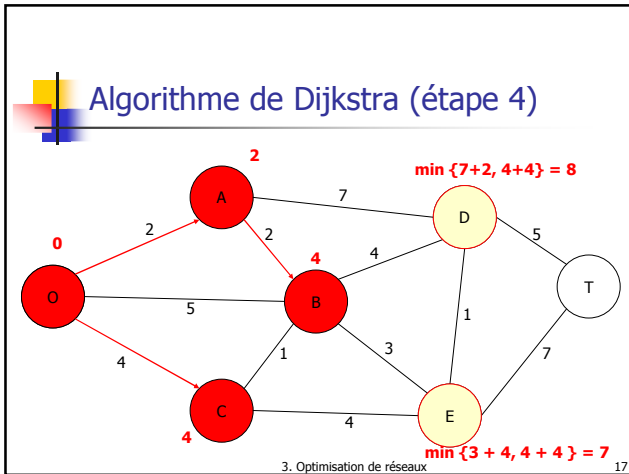
15

Algorithme de Dijkstra (étape 3)

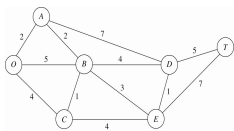


3. Optimisation de réseaux

16



Exemple sous forme tableau



i marqués reliés à j non marqué	j non marqué le plus près	Dist	Dist min	j le plus près	$p(j)$
O	A	2	2	A	O
O	C	4	4	C	O
A	B	4	4	B	A
A	D	9			
B	E	7	7	E	B
C	E	8			
A	D	9			
B	D	8	8	D	B
E	D	8	8	D	E
D	T	13	13	T	D
E	T	14			

3. Optimisation de réseaux

21

Modèle de flot

- Chemin le plus court = flot dans un réseau
- Réseau = graphe orienté dérivé
- On enlève les arcs entrant à O et les arcs émanant de T
- O est la seule source, avec une offre = 1
- T est le seul puits, avec une demande = 1
- Le flot sur chaque arc (i,j) est soit 1, si l'arc appartient au chemin le plus court, soit 0, sinon
- Voir l'exemple modélisé avec [Excel Solver](#)

3. Optimisation de réseaux

22

Autres variantes

- Graphe orienté : l'algorithme de Dijkstra s'applique
- Trouver les chemins les plus courts entre la source et tous les autres sommets : l'algorithme de Dijkstra s'applique
- Trouver les chemins les plus courts entre toutes les paires de sommets : n applications de l'algorithme de Dijkstra (mais il est possible de faire mieux)
- Si certaines « distances » sont négatives : l'algorithme de Dijkstra ne s'applique pas (essayez de voir pourquoi sur un exemple à 3 sommets)

3. Optimisation de réseaux

23

Problème de l'arbre partiel minimum

- On a un graphe non orienté et connexe
- A chaque arête $\{i,j\}$, on associe une distance $c_{ij} \geq 0$
- On cherche à construire un arbre partiel (plus petit graphe connexe contenant tous les sommets) dont la somme des distances soit minimum parmi tous les arbres partiels du graphe
- Exemple simple de problème de **conception de réseaux** (network design) : choisir une configuration de réseau (sous-ensemble d'arcs) qui optimise un certain critère

3. Optimisation de réseaux

24

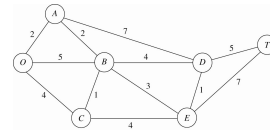
Algorithme de Prim (1957)

- Initialisation: choisir un sommet i (arbitrairement) et le relier au sommet j le plus près: ajouter $\{i,j\}$
- À chaque itération: choisir le sommet non relié j le plus près d'un des sommets déjà reliés i et ajouter $\{i,j\}$
- Arrêter lorsque tous les sommets ont été reliés
- En cas d'égalité, on peut choisir arbitrairement
- De telles égalités indiquent qu'il *pourrait* y avoir plusieurs solutions optimales

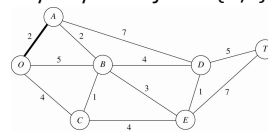
3. Optimisation de réseaux

25

Algorithme de Prim : exemple



- Initialisation: choisir le sommet O et le relier au sommet le plus près: ajouter $\{O,A\}$

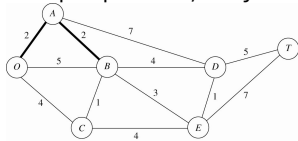


3. Optimisation de réseaux

26

Exemple (suite)

- Le sommet non relié le plus près de O ou de A est B; comme il est plus près de A, on ajoute $\{A,B\}$



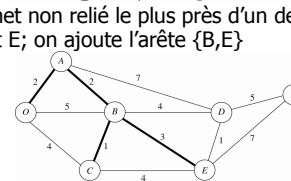
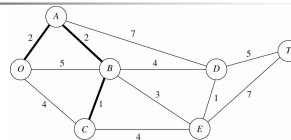
- Le sommet non relié le plus près de O, de A ou de B est C; puisqu'il est plus près de B, on ajoute $\{B,C\}$

3. Optimisation de réseaux

27

Exemple (suite)

- Le sommet non relié le plus près d'un des sommets reliés est E; on ajoute l'arête $\{B,E\}$

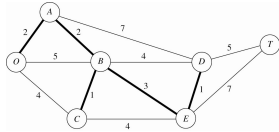


3. Optimisation de réseaux

28

Exemple (suite)

- Le sommet non relié le plus près d'un des sommets reliés (E) est D; on ajoute l'arête {E,D}

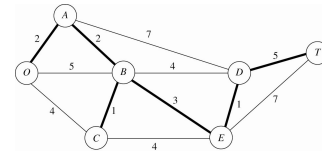


- Le sommet non relié le plus près d'un des sommets reliés (D) est T; on ajoute l'arête {D,T}

3. Optimisation de réseaux

29

Exemple (suite et fin)



- On arrête car tous les sommets ont été reliés
- La valeur optimale correspond à la somme des distances des arêtes ajoutées, soit 14

3. Optimisation de réseaux

30

Problème du flot maximum

- On a un graphe orienté et connexe
- A chaque arc (i,j) , on associe une capacité $u_{ij} > 0$
- Il y a deux sommets spéciaux
 - Source (ou origine) O
 - Puits (ou destination) T
- Tous les autres sont des sommets de transfert
- L'offre en O et la demande en T sont variables
- Offre en O = demande en T = valeur du flot entre O et T
- On cherche à maximiser la valeur du flot entre O et T

3. Optimisation de réseaux

31

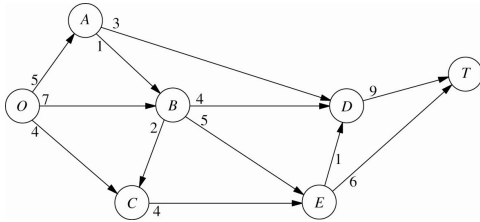
Exemple : parc Seervada

- En période de grande affluence, on dispose d'une flotte d'autobus pour faire visiter les différents postes d'observation du parc
- La réglementation limite le nombre d'autobus pouvant circuler sur chaque tronçon de route
- Comment faire circuler les autobus dans le parc de façon à maximiser le nombre total d'autobus allant de l'origine (O) à la destination (T)?

3. Optimisation de réseaux

32

Exemple: parc Seervada (suite)



3. Optimisation de réseaux

33

Graphe résiduel

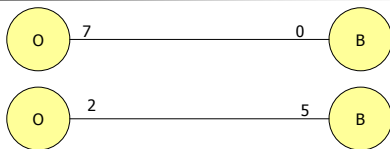
- Supposons qu'on a déjà affecté un flot sur les arcs
- Capacité résiduelle d'un arc (i,j) : $u_{ij} - x_{ij}$
- Graphe résiduel :
 - Graph non orienté sous-jacent
 - Sur chaque arête, on associe deux valeurs :
 - Capacité résiduelle
 - Flot déjà affecté
- Exemple : on a affecté 5 unités de flot sur l'arc (O,B)



3. Optimisation de réseaux

34

Interprétation du graphe résiduel



- On a affecté 5 unités de flot sur l'arc (O,B)
 - Si on traverse $O \rightarrow B$
 - 2 = capacité résiduelle
 - 5 = flot sur (O, B)
 - Si on traverse $B \rightarrow O$
 - 5 = capacité résiduelle
 - 2 = flot sur (B, O)

3. Optimisation de réseaux

35

Chemin d'augmentation

- Chemin allant de la source au puits dans le *graphe orienté dérivé du graphe résiduel*
- Pour chaque arête $\{i,j\}$
 - L'arc (i,j) possède une capacité résiduelle = $u_{ij} - x_{ij}$
 - L'arc (j,i) possède une capacité résiduelle = x_{ij}
- Chaque arc du chemin possède une capacité résiduelle > 0
- Capacité résiduelle d'un chemin d'augmentation : minimum des capacités résiduelles de tous les arcs du chemin

3. Optimisation de réseaux

36

Algorithme de Ford-Fulkerson

1. Initialiser le flot: 0 unité sur chaque arc
2. Si aucun chemin d'augmentation ne peut être identifié, arrêter: le flot est maximum
3. Identifier un chemin d'augmentation P ; soit c sa capacité résiduelle
4. Sur chaque arc de P
 - a. Augmenter le flot de c
 - b. Diminuer la capacité résiduelle de c
5. Retourner à l'étape 2

3. Optimisation de réseaux

37

Identifier un chemin d'augmentation

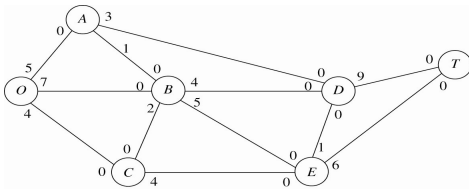
1. Marquer la source O (aucun autre sommet n'est marqué); tous les sommets sont non visités
2. S'il n'y a aucun sommet marqué non visité, arrêter: il n'existe aucun chemin d'augmentation
3. Choisir un sommet marqué non visité i
4. Visiter i : pour chaque (i,j) de capacité résiduelle >0 dans le graphe orienté dérivé du graphe résiduel, marquer j
5. Si T est marqué, arrêter: un chemin d'augmentation a été identifié
6. Retourner à l'étape 2

3. Optimisation de réseaux

38

Exemple : parc Seervada

- Graphe résiduel initial



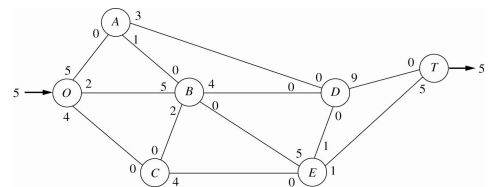
- Identifier un chemin d'augmentation : $O \rightarrow B \rightarrow E \rightarrow T$
- Capacité résiduelle = $\min\{7, 5, 6\} = 5$

3. Optimisation de réseaux

39

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 5 unités sur tous les arcs de $O \rightarrow B \rightarrow E \rightarrow T$



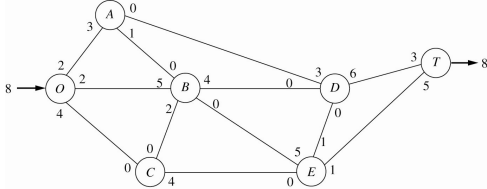
- Identifier un chemin d'augmentation : $O \rightarrow A \rightarrow D \rightarrow T$
- Capacité résiduelle = $\min\{5, 3, 9\} = 3$

3. Optimisation de réseaux

40

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 3 unités sur tous les arcs de $O \rightarrow A \rightarrow D \rightarrow T$



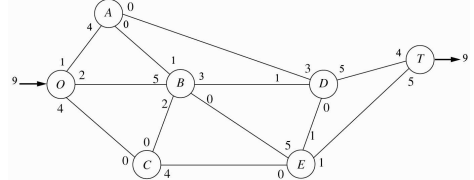
- Chemin d'augmentation : $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$
- Capacité résiduelle = $\min\{2, 1, 4, 6\} = 1$

3. Optimisation de réseaux

41

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 1 unité sur tous les arcs de $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$



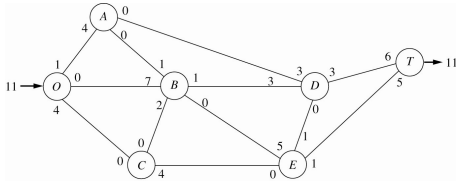
- Chemin d'augmentation : $O \rightarrow B \rightarrow D \rightarrow T$
- Capacité résiduelle = $\min\{2, 3, 5\} = 2$

3. Optimisation de réseaux

42

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 2 unités sur tous les arcs de $O \rightarrow B \rightarrow D \rightarrow T$



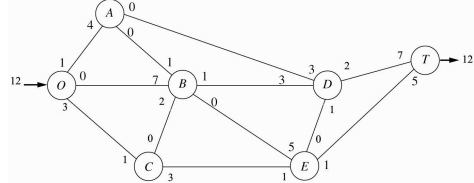
- Chemin d'augmentation : $O \rightarrow C \rightarrow E \rightarrow D \rightarrow T$
- Capacité résiduelle = $\min\{4, 4, 1, 3\} = 1$

3. Optimisation de réseaux

43

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 1 unité sur tous les arcs de $O \rightarrow C \rightarrow E \rightarrow D \rightarrow T$



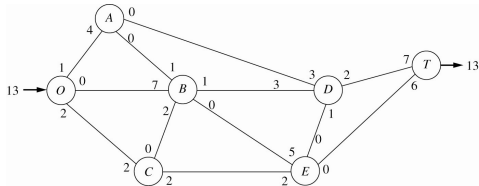
- Chemin d'augmentation : $O \rightarrow C \rightarrow E \rightarrow T$
- Capacité résiduelle = $\min\{3, 3, 1\} = 1$

3. Optimisation de réseaux

44

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 1 unité sur tous les arcs de $O \rightarrow C \rightarrow E \rightarrow T$



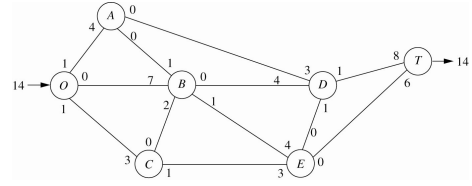
- Chemin d'augmentation : $O \rightarrow C \rightarrow E \rightarrow B \rightarrow D \rightarrow T$
- Capacité résiduelle = $\min\{2, 2, 5, 1, 2\} = 1$

3. Optimisation de réseaux

45

Exemple (suite)

- Augmenter le flot et diminuer la capacité résiduelle de 1 unité sur tous les arcs de $O \rightarrow C \rightarrow E \rightarrow B \rightarrow D \rightarrow T$

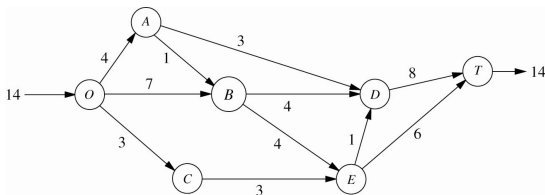


- Aucun chemin d'augmentation possible
- Flot maximum

3. Optimisation de réseaux

46

Exemple (suite et fin)



3. Optimisation de réseaux

47

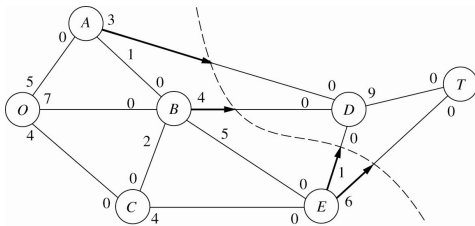
Flot maximum - Coupe minimum

- Supposons qu'on partitionne l'ensemble des sommets en deux sous-ensembles X, Y
- Coupe : sous-ensemble d'arcs allant d'un sommet de X vers un sommet de Y
- Capacité d'une coupe : somme des capacités des arcs de la coupe
- Coupe minimum : coupe dont la capacité est minimum parmi toutes les coupes possibles
- Théorème flot max - coupe min* : la valeur du flot maximum est égale à la capacité d'une coupe minimum

3. Optimisation de réseaux

48

Flot max – Coupe min : exemple



Les sommets marqués lors de la dernière itération de l'algorithme de Ford-Fulkerson définissent la coupe min

3. Optimisation de réseaux

49

Flot maximum avec Excel Solver

- Flot maximum = modèle de flot
- Tous les sommets, sauf O et T, sont des sommets de transfert
- Il y a une capacité sur chaque arc
- L'objectif est de maximiser la valeur du flot, soit le flot net en O
- Voir l'exemple modélisé avec [Excel Solver](#)

3. Optimisation de réseaux

50