

Projet pratique Algorithme / Complexité / Calculabilité

Jean-Marc Lagniez, Viktor Lesnyak, Pierre-Alexandre
Cimbe, Ahmed Rafik

Master Informatique - Université Montpellier II

2013

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisée
 - Temps d'exécution en fonction du nombre de sommets comptant le temps de construction des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisée
 - Temps d'exécution en fonction du nombre de sommets comptant le temps de construction des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisitine
 - Temps d'execution en fonction du nombre de sommets comptant le temps de crion des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

Plan

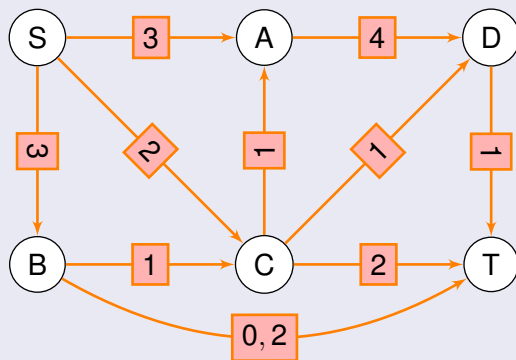
- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisitine
 - Temps d'exécution en fonction du nombre de sommets comptant le temps de crion des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisitine
 - Temps d'execution en fonction du nombre de sommets comptant le temps de crion des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

AlgoFF

Graphe initiale

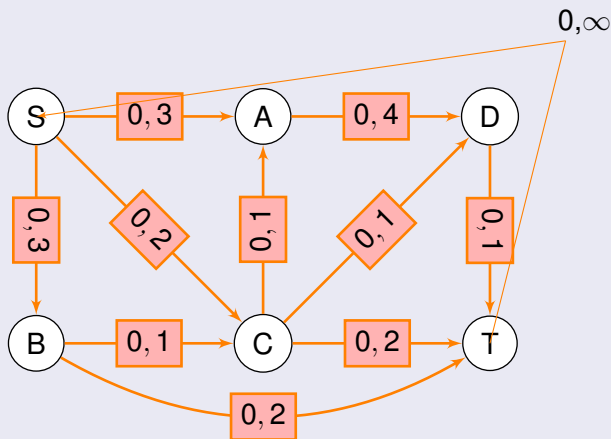


Graphe initiale

Soit $G = (V, E)$ un graphe, avec V -ensemble des arcs et E -ensemble des sommets.

AlgoFF

Graphe d'ecart

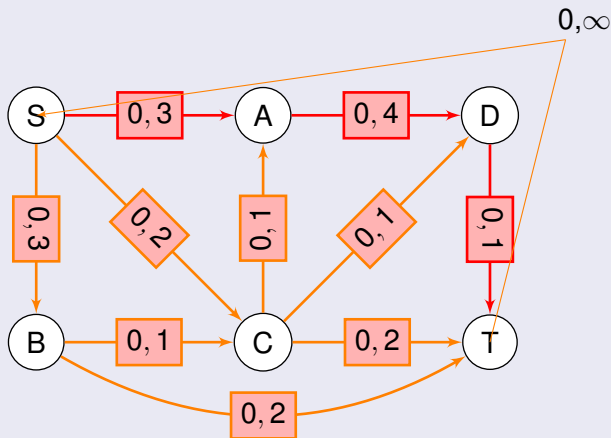


Graphe d'ecart

Pour passer de notre graphe G au Graphe d'ecart G_e on applique un flot null sur toutes les arcs et on ajout un arc qui va de la source(S) vers le puit(T).

AlgoFF

Chemin ameliorant

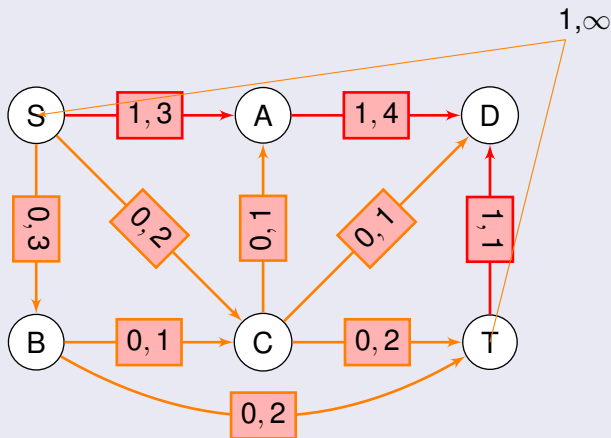


Chemin ameliorant

Ensuite on choisit un chemin améliorant sur le graphe d'écart obtenue grâce à un parcours en largeur.

AlgoFF

Chemin ameliorant

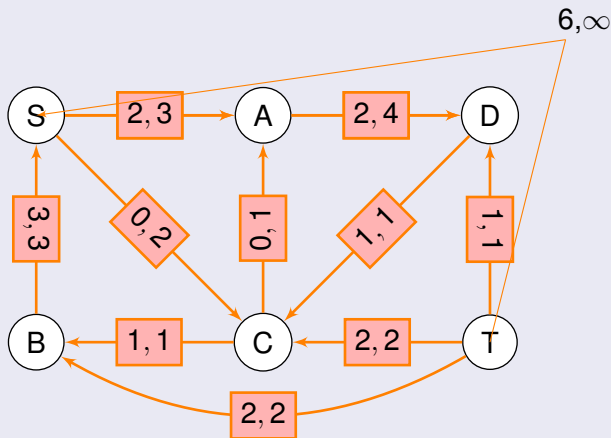


Chemin ameliorant

En utilisant le flot le plus petit de ce chemin on met à jour le graphe d'écart.

AlgoFF

Chemin ameliorant

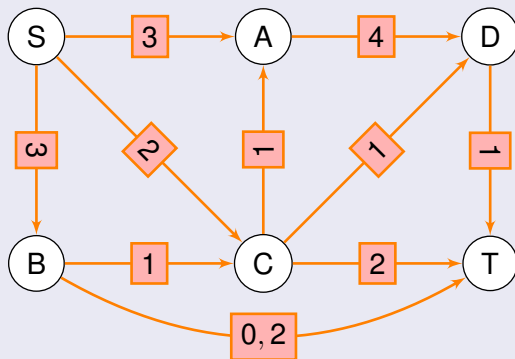


Chemin ameliorant

Une fois tous les chemins améliorants sont parcourus, on obtient un graphe d'écart complet avec le flot maximal (dans notre cas c'est 6).

AlgoEK

Graphe initiale



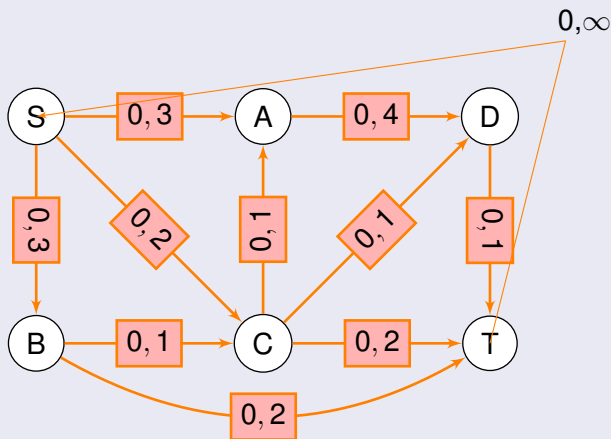
Graphe initiale

On reprends le
meme graphe
initiale.

Soit $G = (V, E)$ un
graphe, avec
 V -ensemble des
arcs et E -
ensemble des
sommets.

AlgoEK

Graphe d'ecart

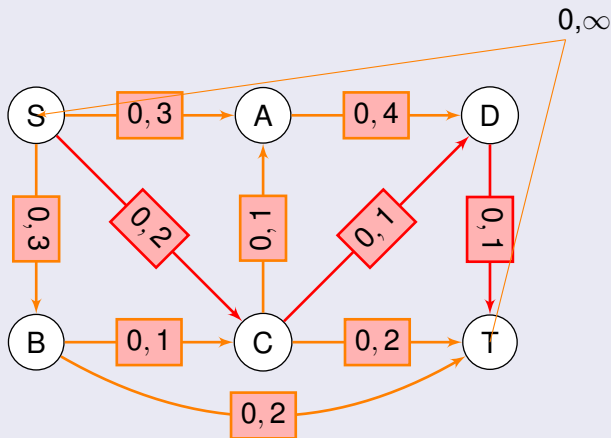


Graphe d'ecart

On refait a nouveau a partir de graphe G le Graphe d'ecart G_e on applique un flot null sur toutes les arcs et on ajout un arc qui va de la source(S) vers le puit(T).

AlgoEK

Chemin ameliorant

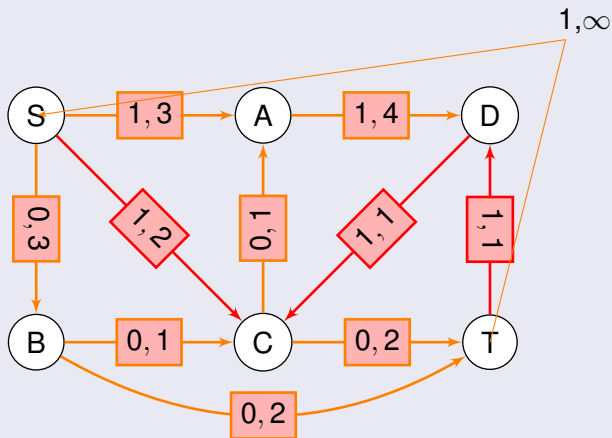


Chemin ameliorant

On choisi ici un chemin ameliorant en fonction de plus court chemin, qui dans notre cas est calculec l'algo de Dijkstra

AlgoFF

Chemin ameliorant



Chemin ameliorant

En utilisant le flot le plus petit de ce chemin on met à jour le graphe d'écart.

AlgoD

AlgoCS

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisitine
 - Temps d'execution en fonction du nombre de sommets comptant le temps de crion des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 **Experimentation et Performance**
 - Performance de l'ordinateur utilisitine
 - Temps d'execution en fonction du nombre de sommets comptant le temps de crion des graphes
- 4 Demonstration du fonctionnement sous TIKZ
 - Demonstration du fonctionnement sous TIKZ

Version Ubuntu : 10.04 Intel(R) Pentium(R) Dual CPU T3200 @
2.00GHZ

Ford-Fulkerson

Capacity-Scaling

Edmons-Karp

Dinic

Comparaison des algorithmes

Plan

- 1 Les algorithmes étudiés
 - Ford-Fulkerson
 - Edmonds-Karp
 - Dinic
 - Capacity Scaling
- 2 Experimentation et Performance
- 3 Experimentation et Performance
 - Performance de l'ordinateur utilisitine
 - Temps d'execution en fonction du nombre de sommets comptant le temps de crion des graphes
- 4 **Demonstration du fonctionnement sous TIKZ**
 - **Demonstration du fonctionnement sous TIKZ**

Démo