Projet pratique Algorithme / Complexité / Calculabilité

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

Master Informatique - Université Montpellier II

2013

Plan

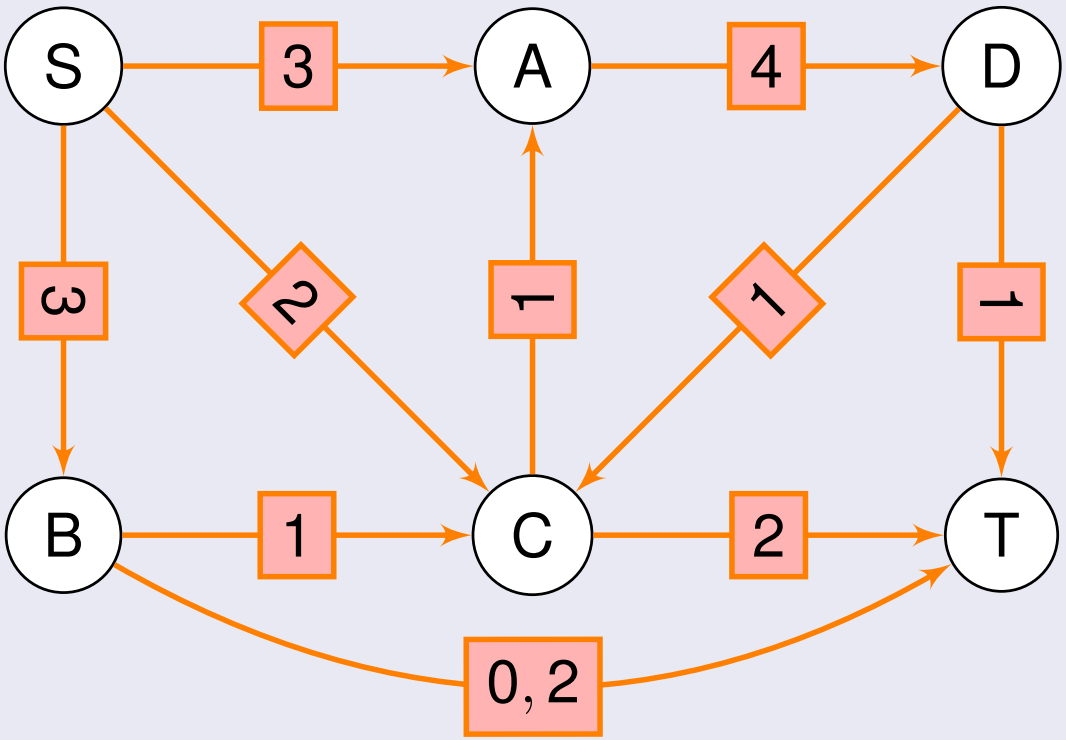
Plan

Plan

Plan

Algo Ford-Fulkerson

Graphe initiale

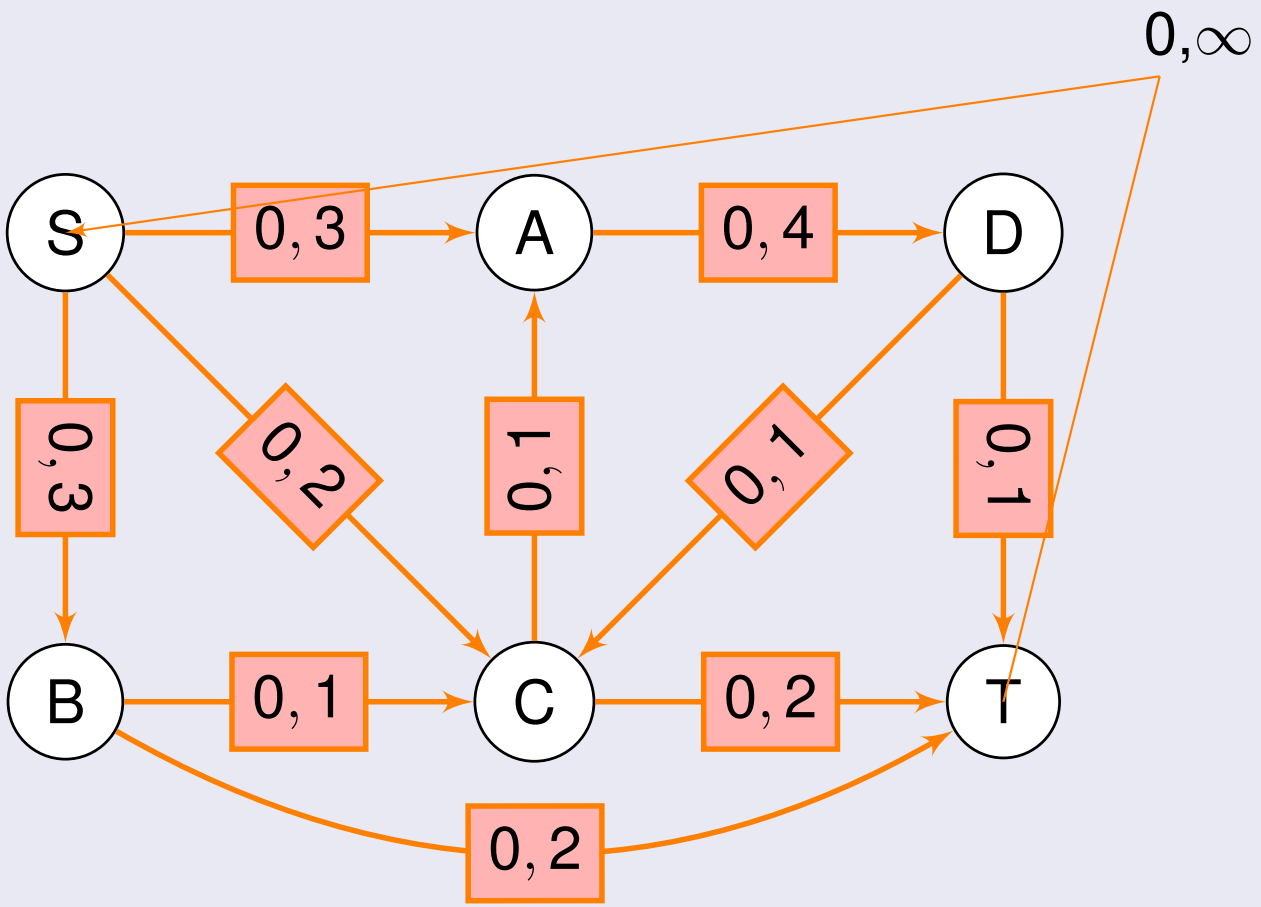


Graphe initiale

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

Algo Ford-Fulkerson

Graphe d'ecart

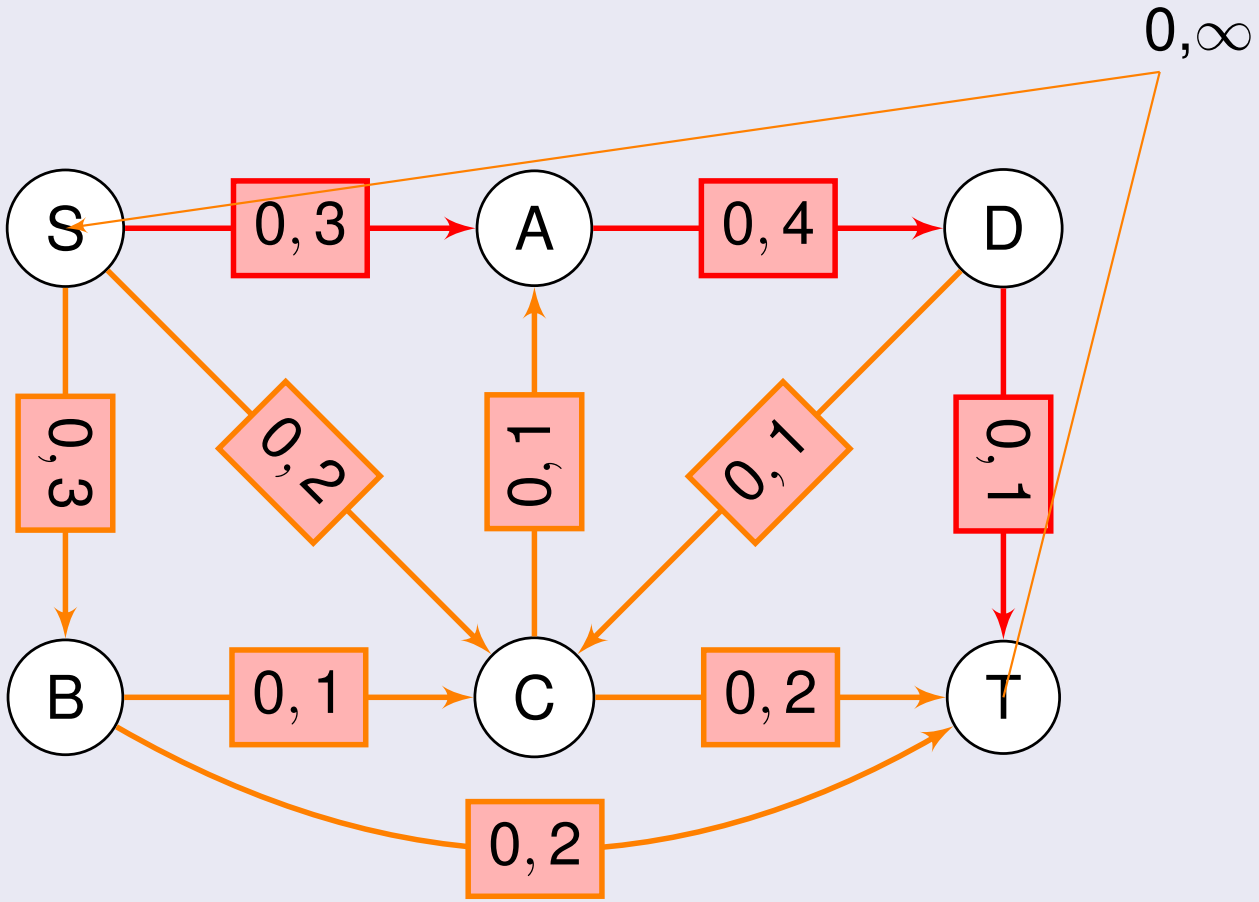


Graphe d'ecart

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).

Algo Ford-Fulkerson

Chemin ameliorant

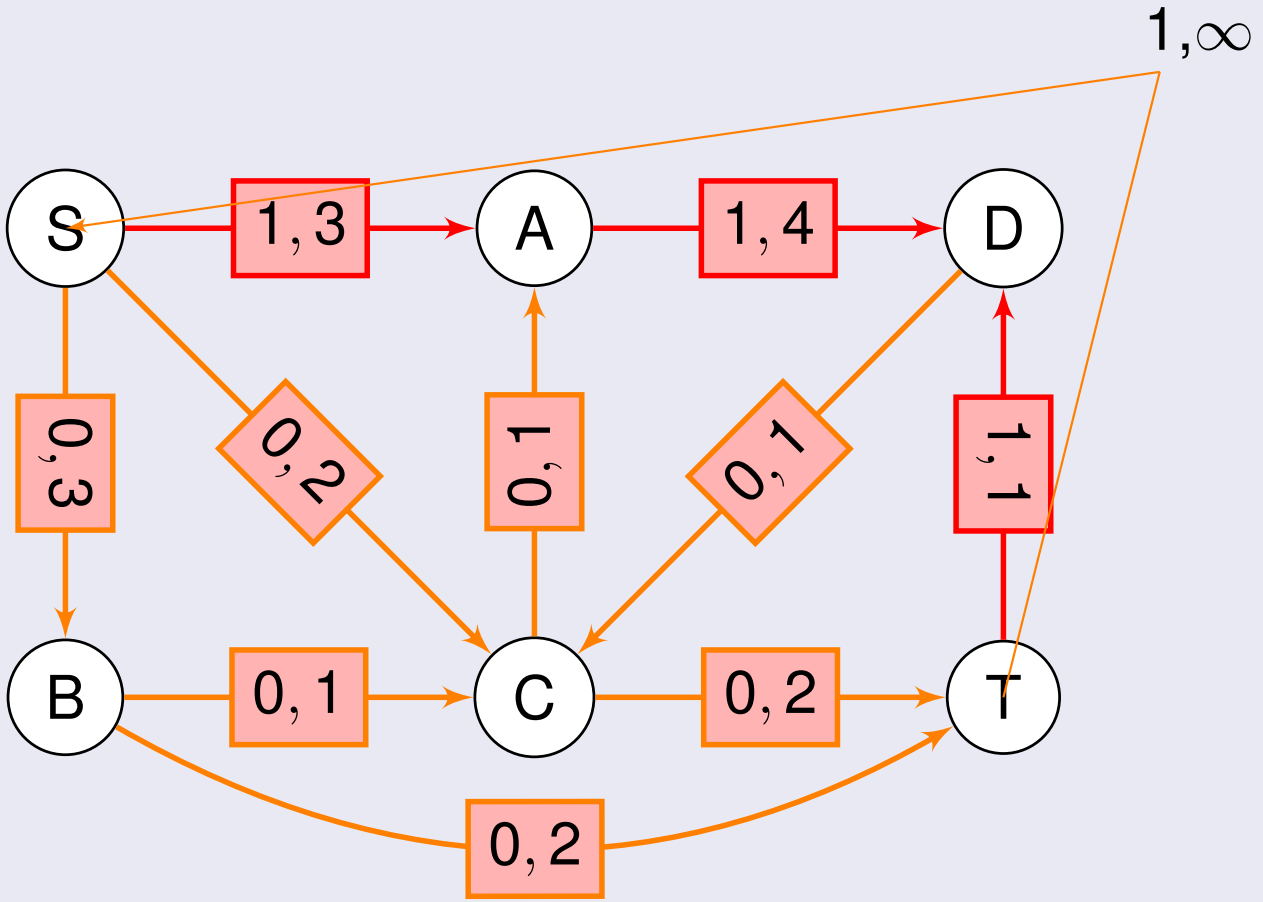


Chemin ameliorant

Ensuite on choisi un chemin ammeliorant sur le graphe d'ecart obtenue grace a un parcour en profondeur.

Algo Ford-Fulkerson

Chemin ameliorant

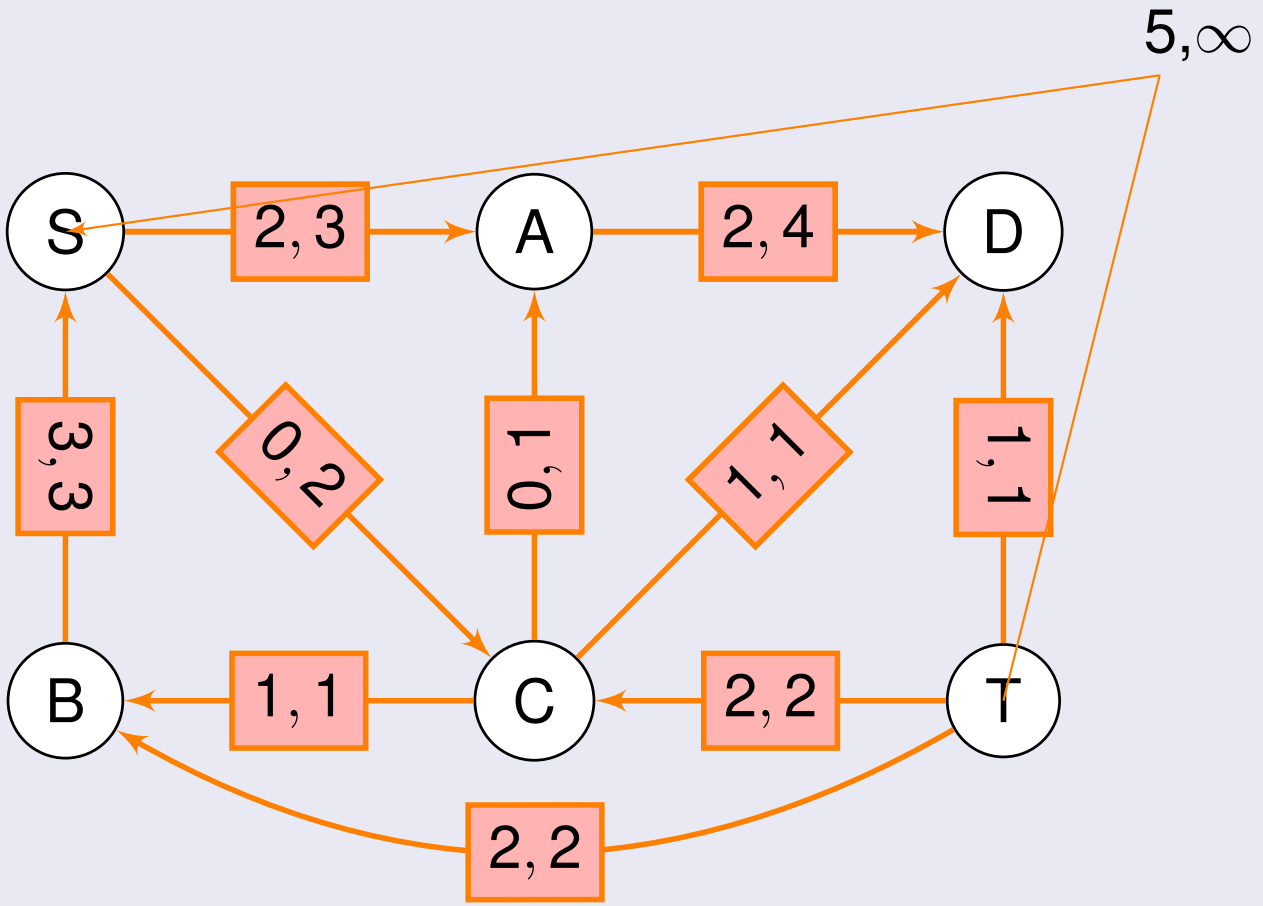


Chemin ameliorant

En utilisant la capacite la plus petite de ce chemin on met a jour le graphe d'ecart.

Algo Ford-Fulkerson

Graphe finale

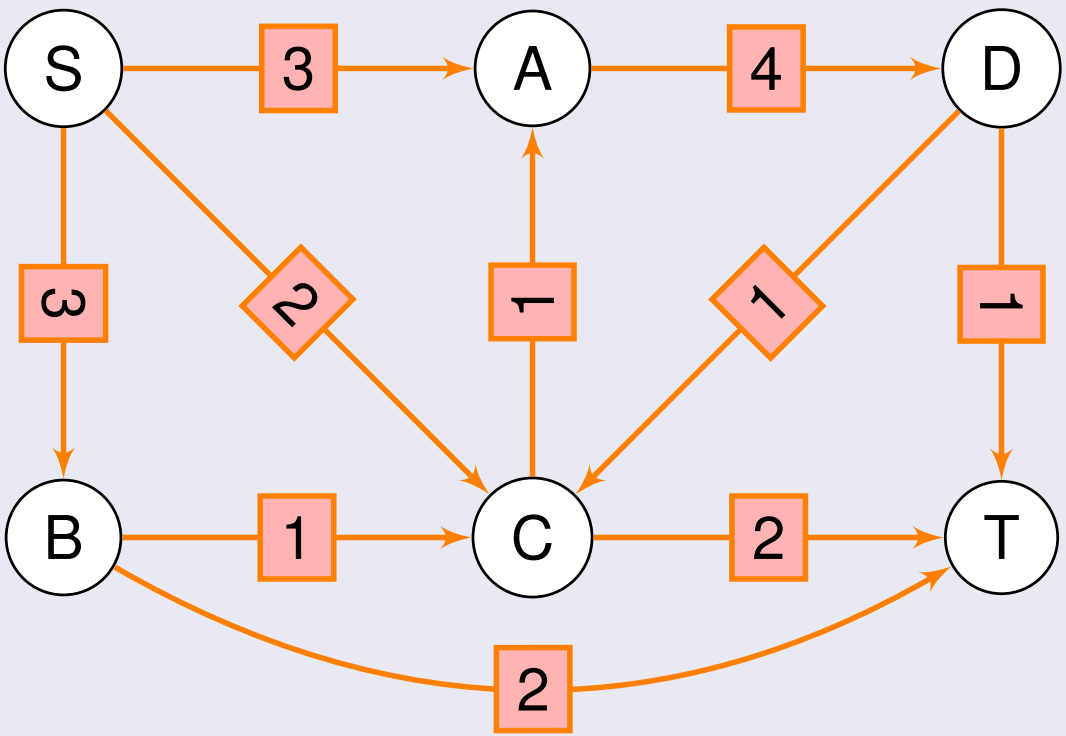


Graphe finale

Une foi tout les chemin ameliorants sont parcouru, on obtien un graphe d'ecart complet avec le flot maximal (das notre cas c'est 5).

Algo Edmonds-Karp

Graphe initiale

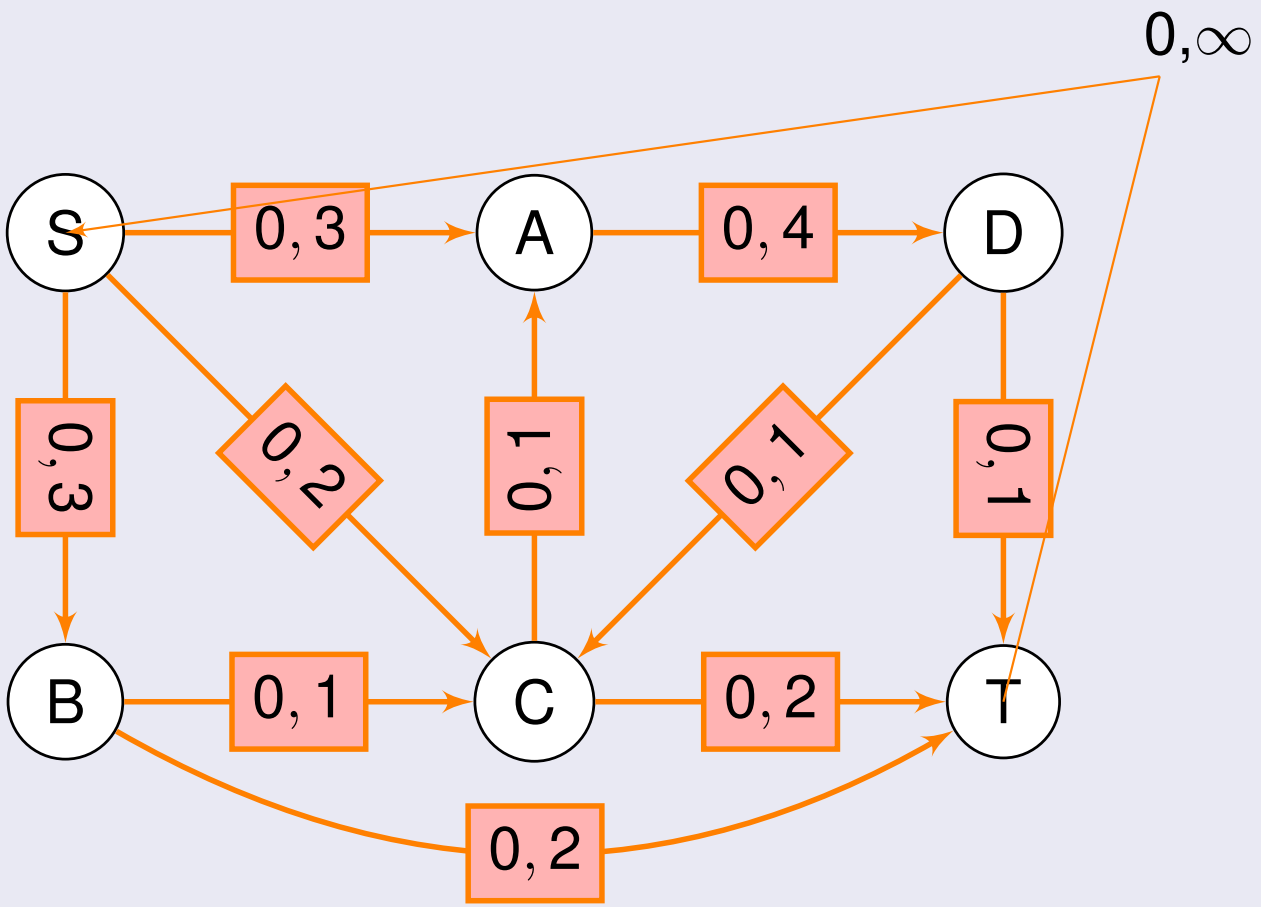


Graphe initiale

On reprends le meme graphe initiale.
Soit G = (V,E) un graphe,avec V-ensemble des sommets et E-ensemble des arcs.

Algo Edmonds-Karp

Graphe d'ecart

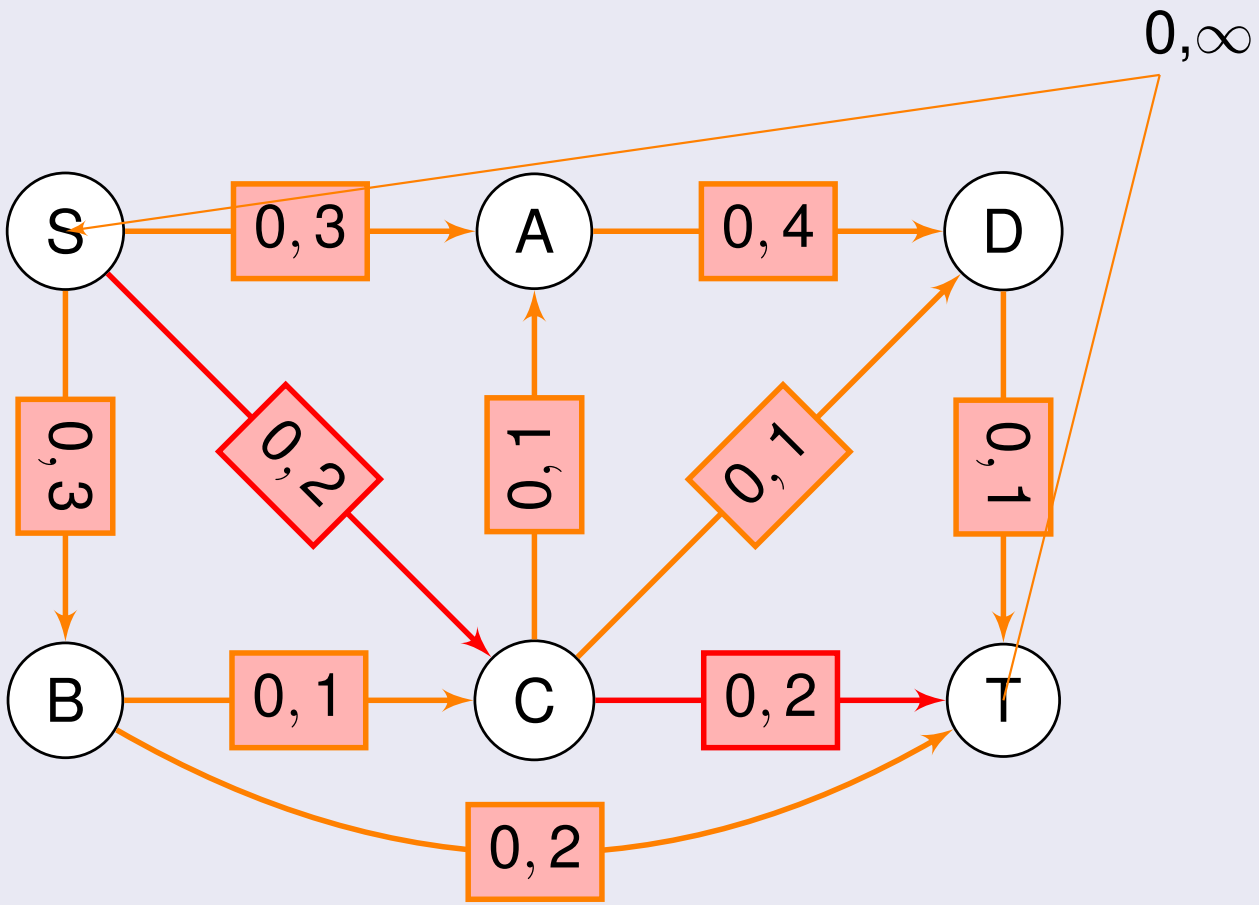


Graphe d'ecart

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).

Algo Edmonds-Karp

Chemin ameliorant

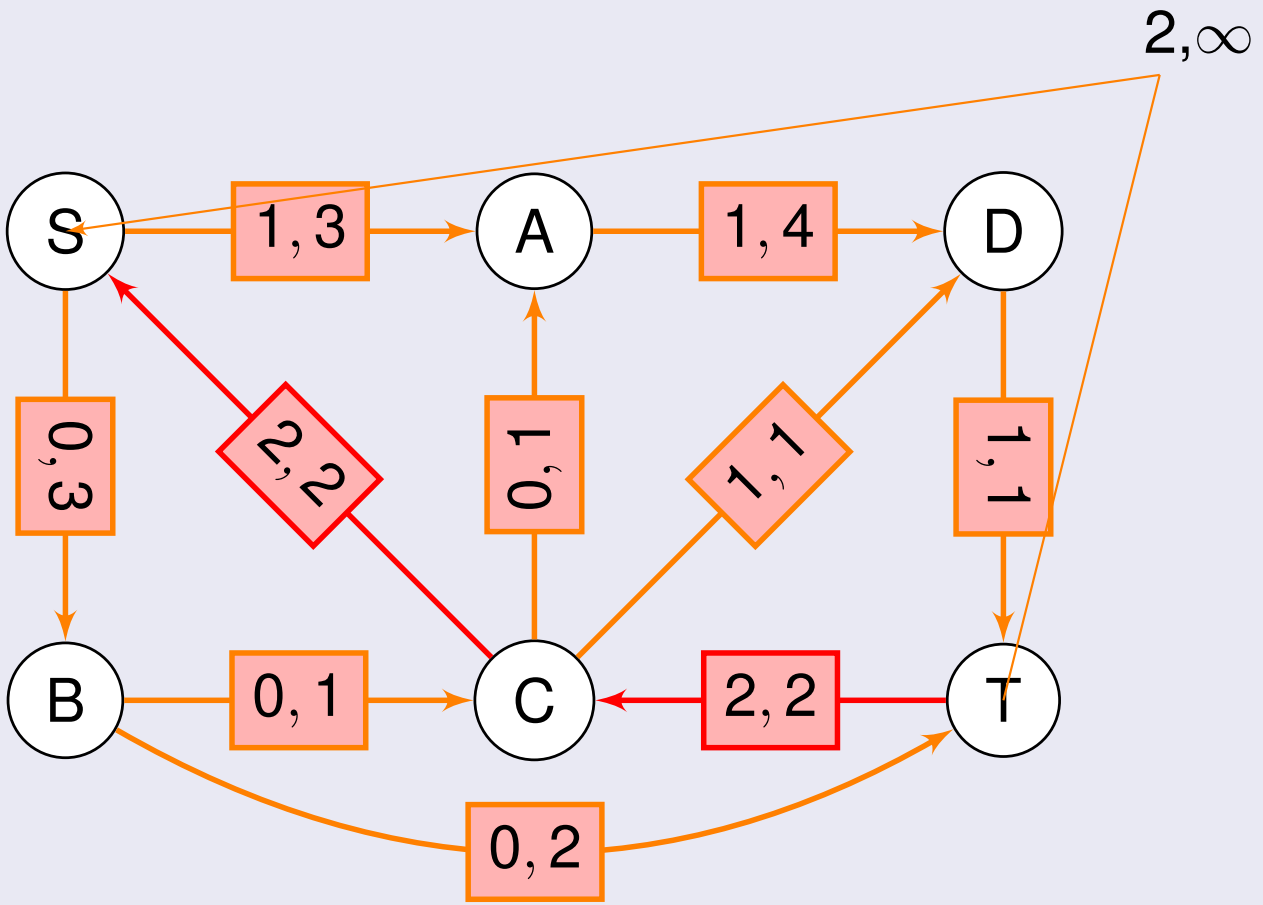


Chemin ameliorant

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

Algo Edmonds-Karp

Chemin ameliorant

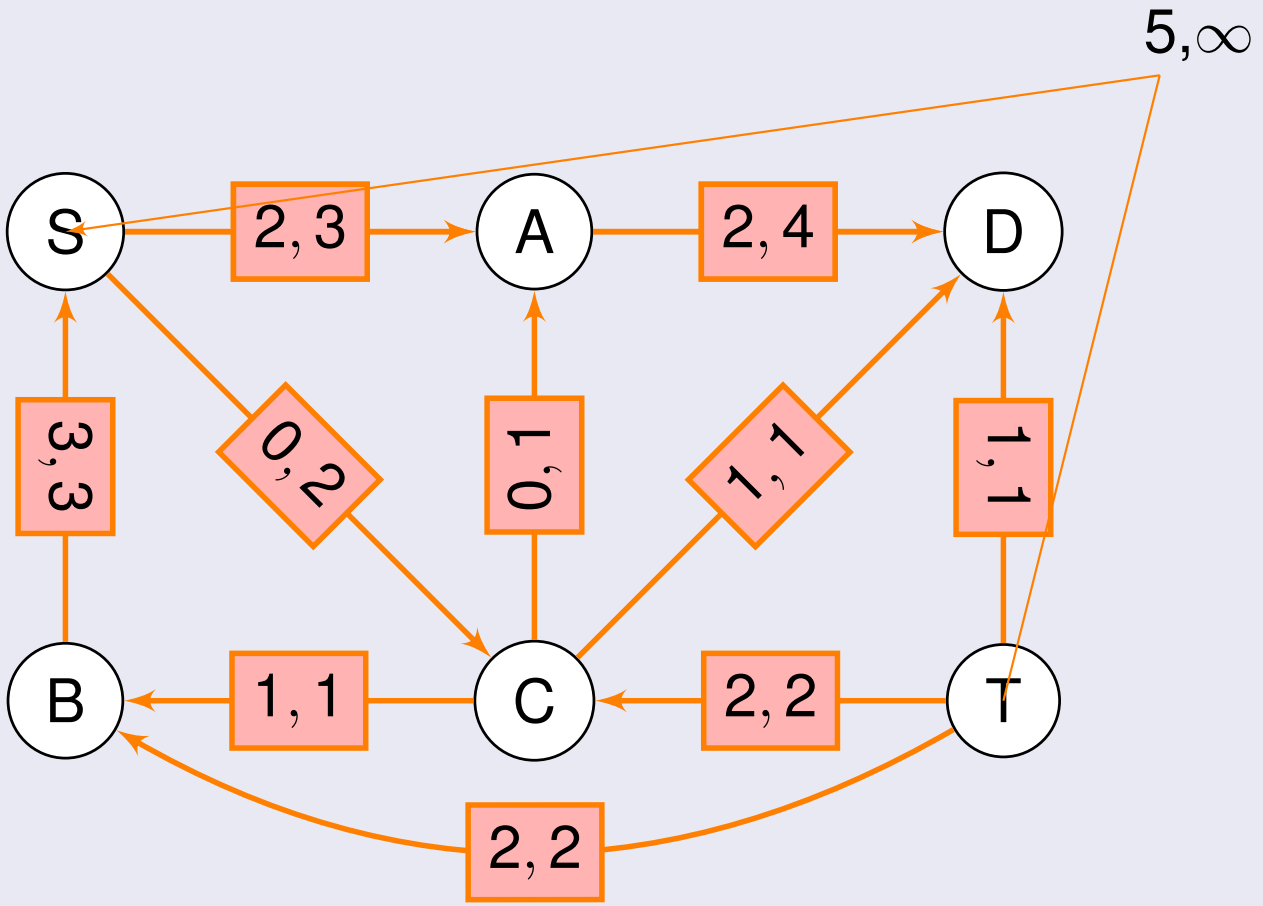


Chemin ameliorant

En utilisant le flotle plus petit de ce chemin on met a jour le graphe d'ecart.

Algo Edmonds-Karp

Graphe finale



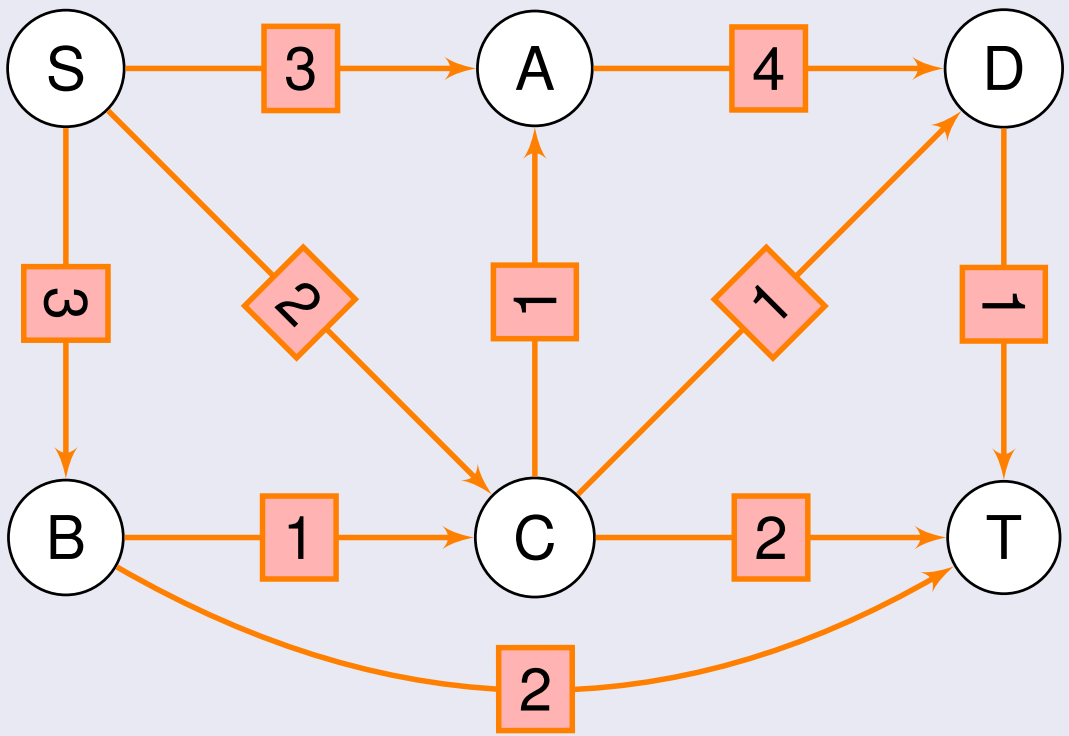
Graphe finale

Une foi tout les chemin ameliorants sont parcouru, on obtien un graphe d'ecart complet avec le flot maximal (das notre cas c'est 5).

AlgoD

Algo Capacity Scaling

Graphe initial

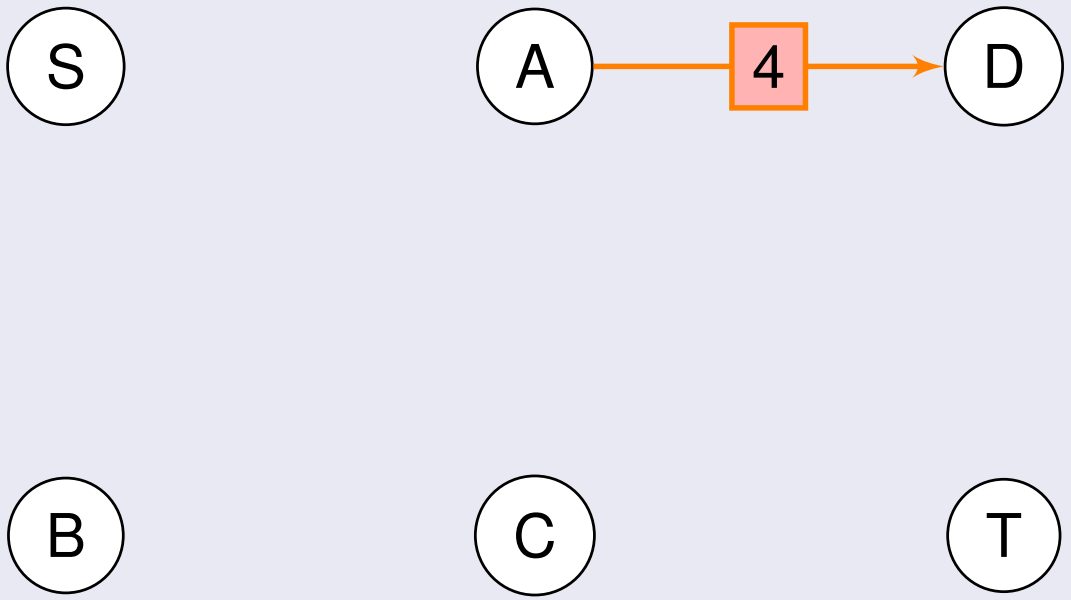


Initialisation

C=4 ;
$\Delta = 4$;
F le flot max = 0 ;

Algo Capacity Scaling

G(f,4)



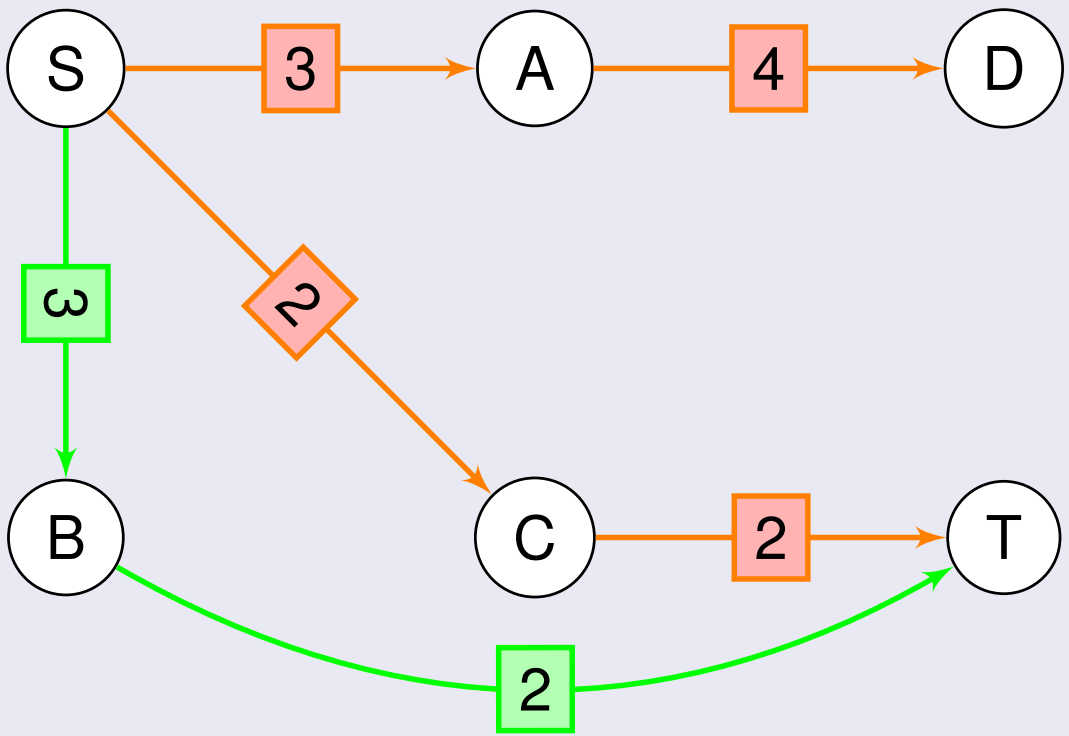
Augmentation

Pas de chemin de s

$\Delta = \Delta/2 = 2$

Algo Capacity Scaling

G(f,2)



Augmentation

Premier chemin :

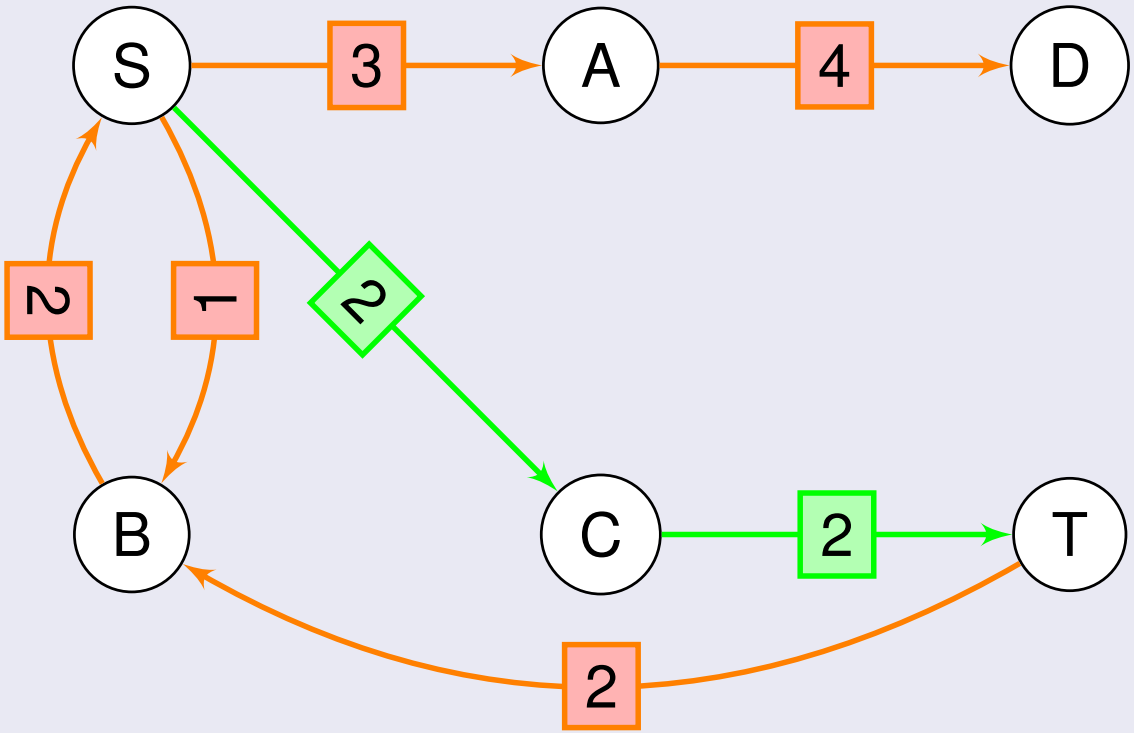
$s \rightarrow b \rightarrow t$

$\delta = 2$

$F = 2$

Algo Capacity Scaling

G(f,2)



Mise ur du graphe d'rt et Augmentation

Second chemin :

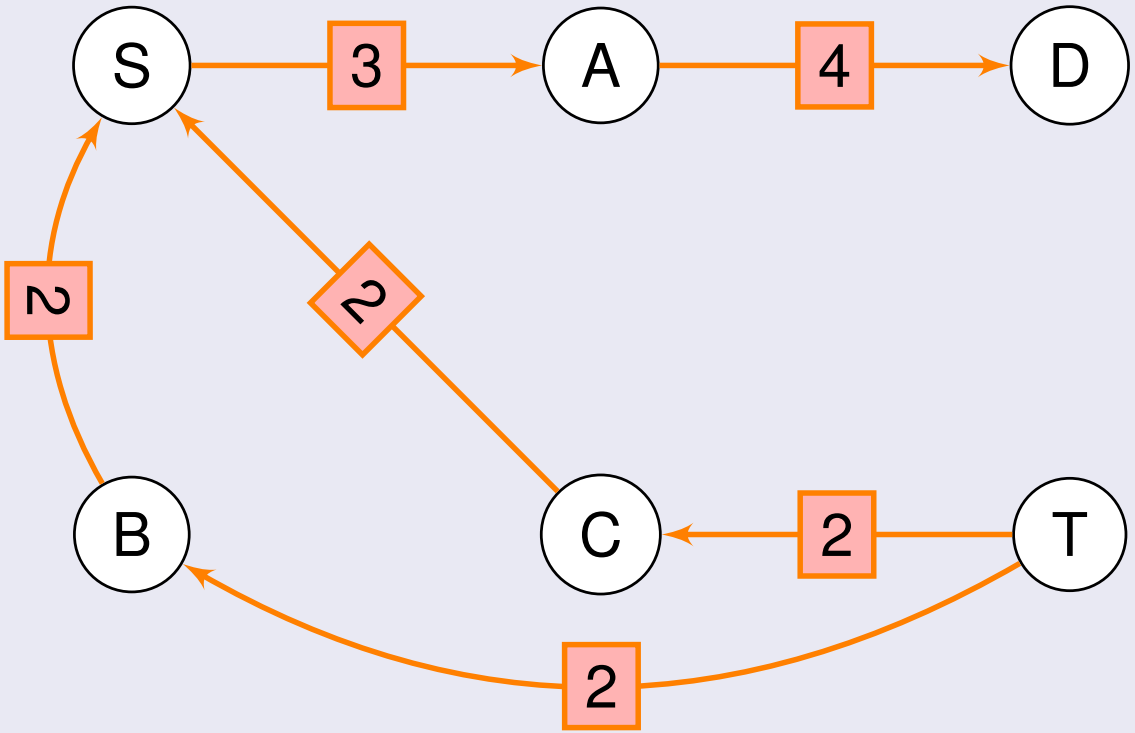
$s \rightarrow c \rightarrow t$

$\delta = 2$

$F = 4$

Algo Capacity Scaling

G(f,2)

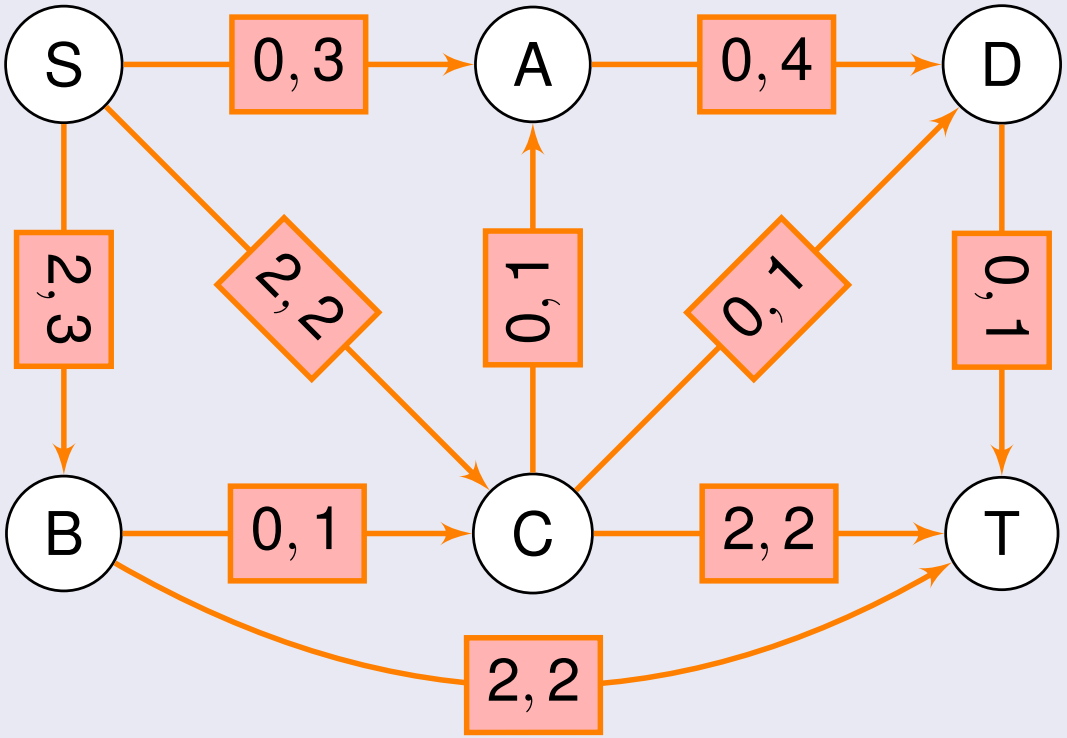


Mise ur du graphe d'rt et Augmentation

Pas d'autre chemin dans le graphe.
$\Delta = \Delta/2 = 1$

Algo Capacity Scaling

Flot actuel dans G



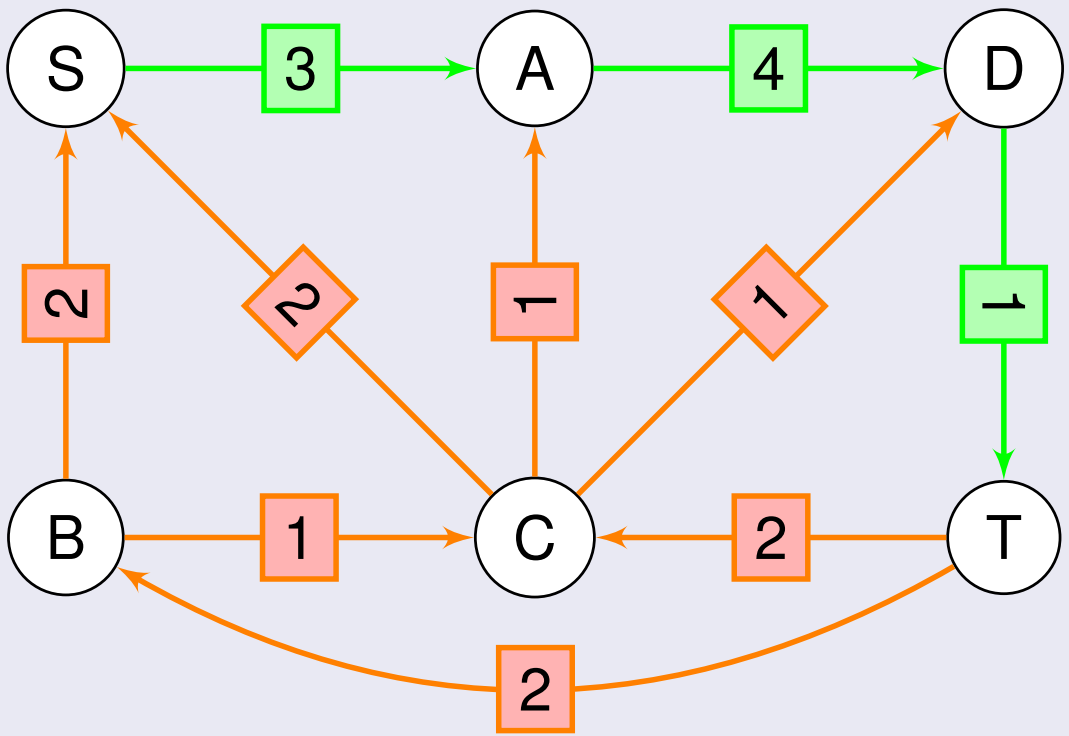
Rappel :

$\Delta = 1$

$F = 4$

Algo Capacity Scaling

G(f,1)



Mise ur du graphe d'rt et Augmentation

Premier chemin :

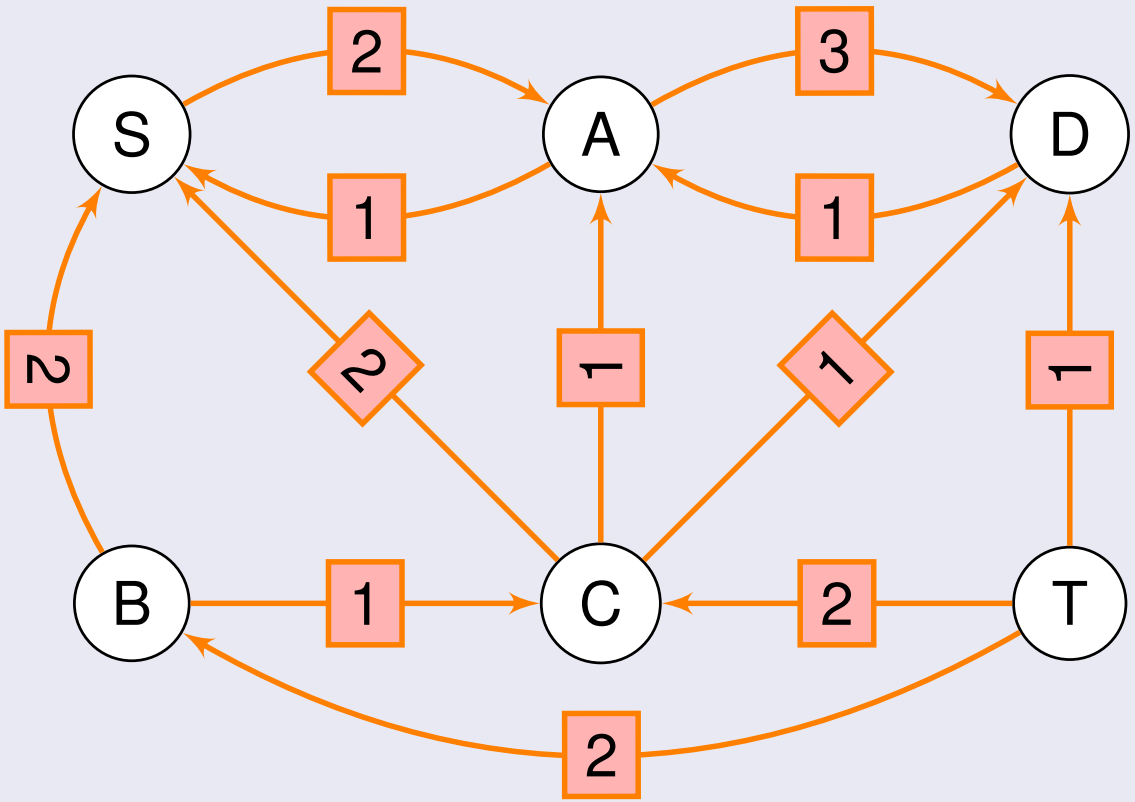
$s \to a \to d \to t$

$\delta = 1$

$F = 5$

Algo Capacity Scaling

G(f,1)

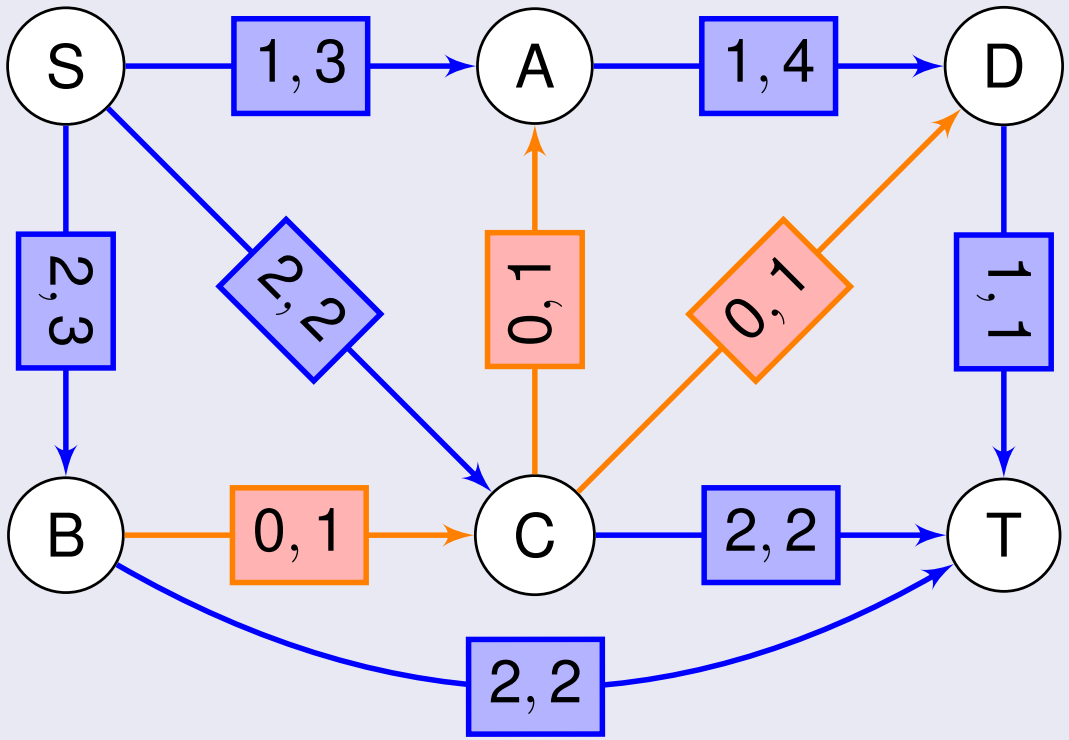


Mise ur du graphe d'rt et Augmentation

Pas d'autre chemin dans le graphe. L'algorithme est termin

Algo Capacity Scaling

Flot maximum)



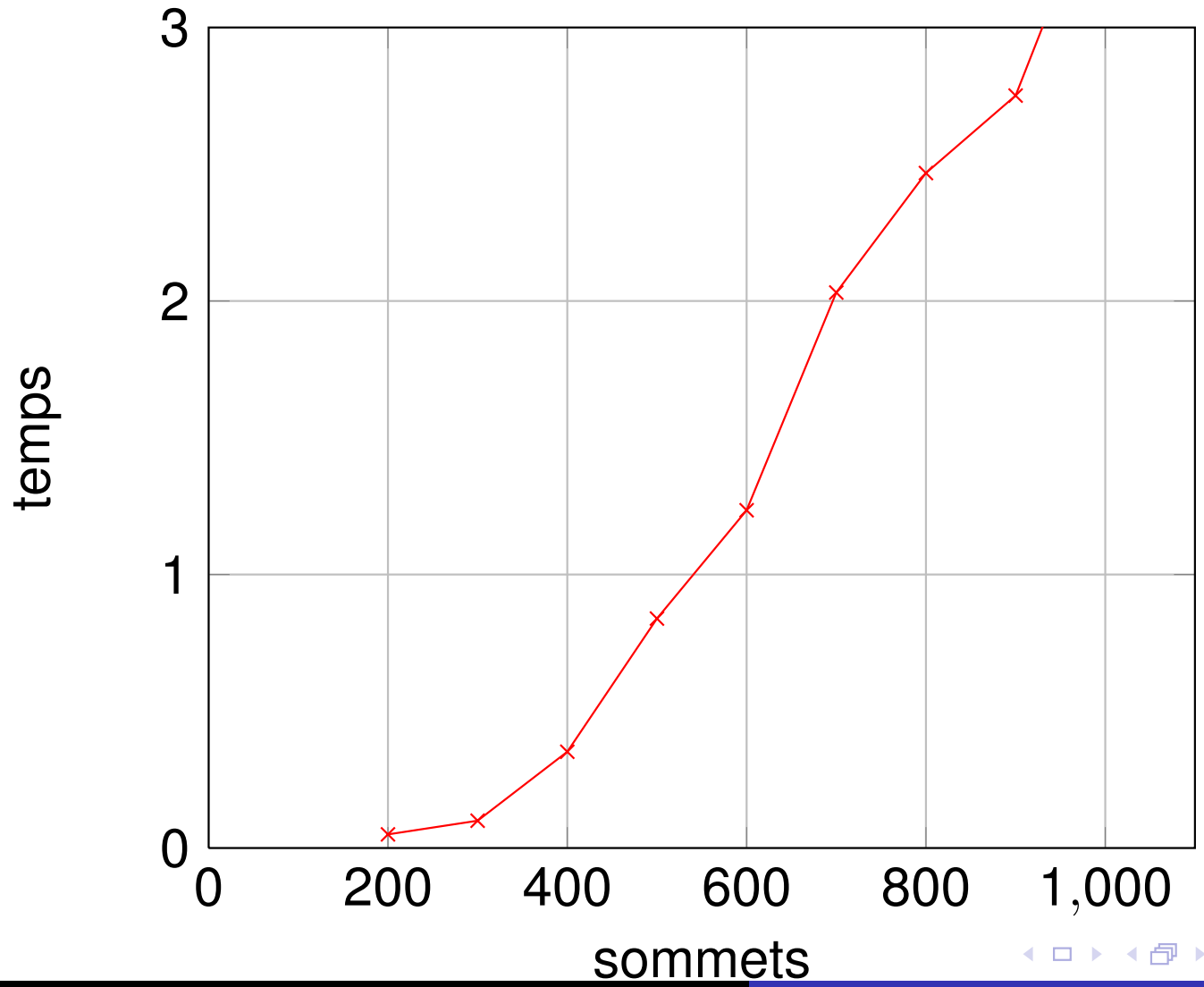
Valeur du flot

F=5 ;

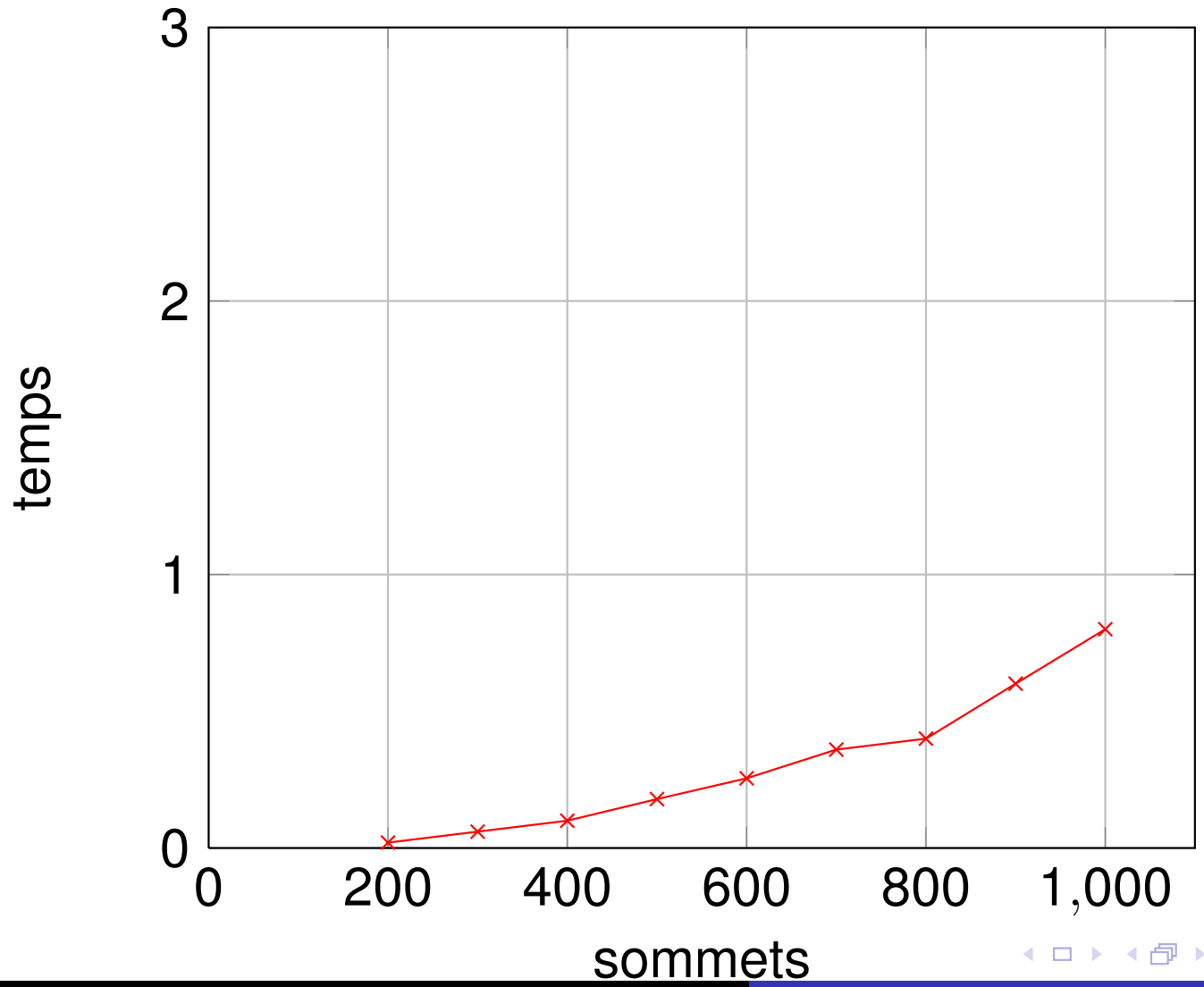
Plan

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

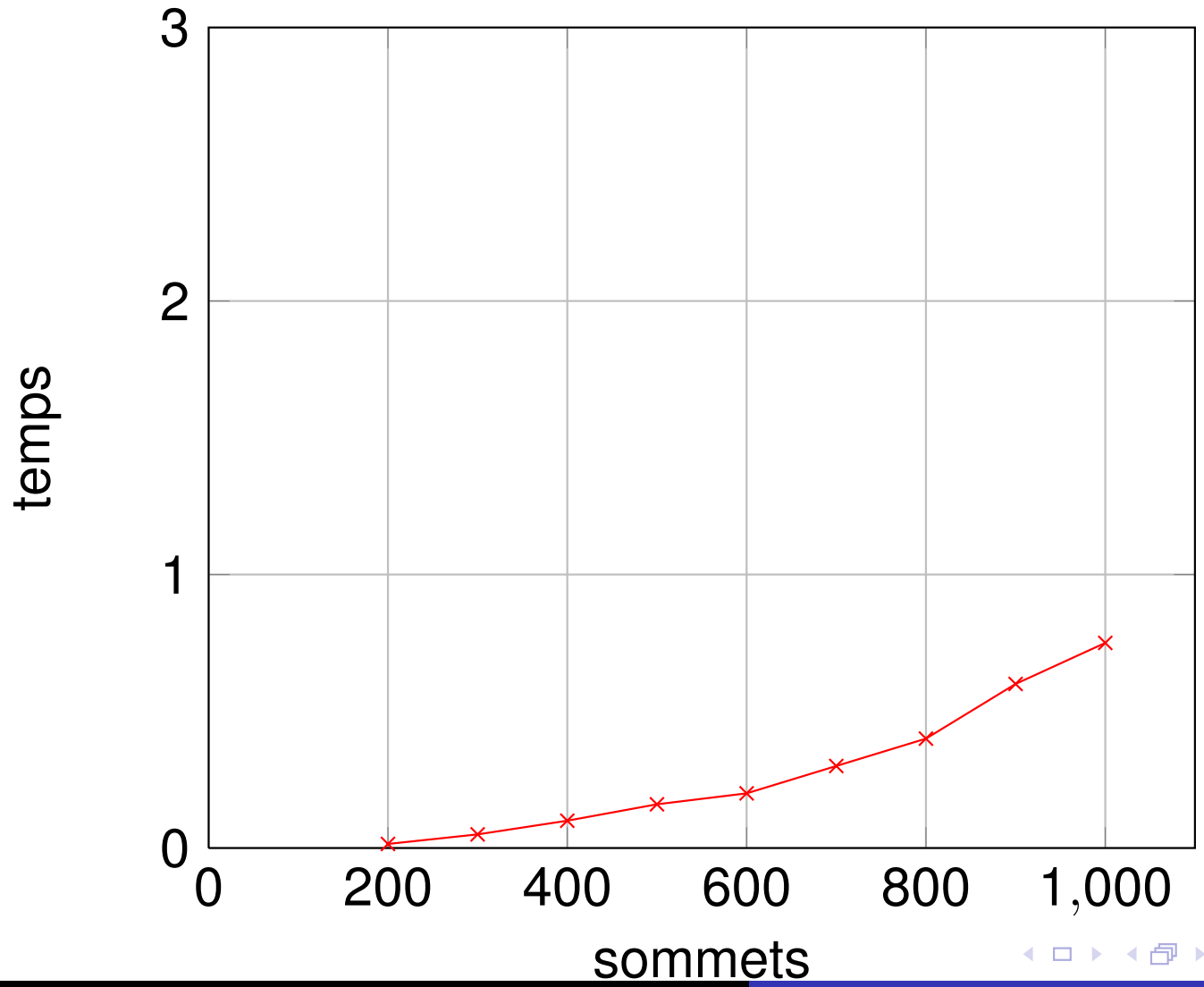
Ford-Fulkerson



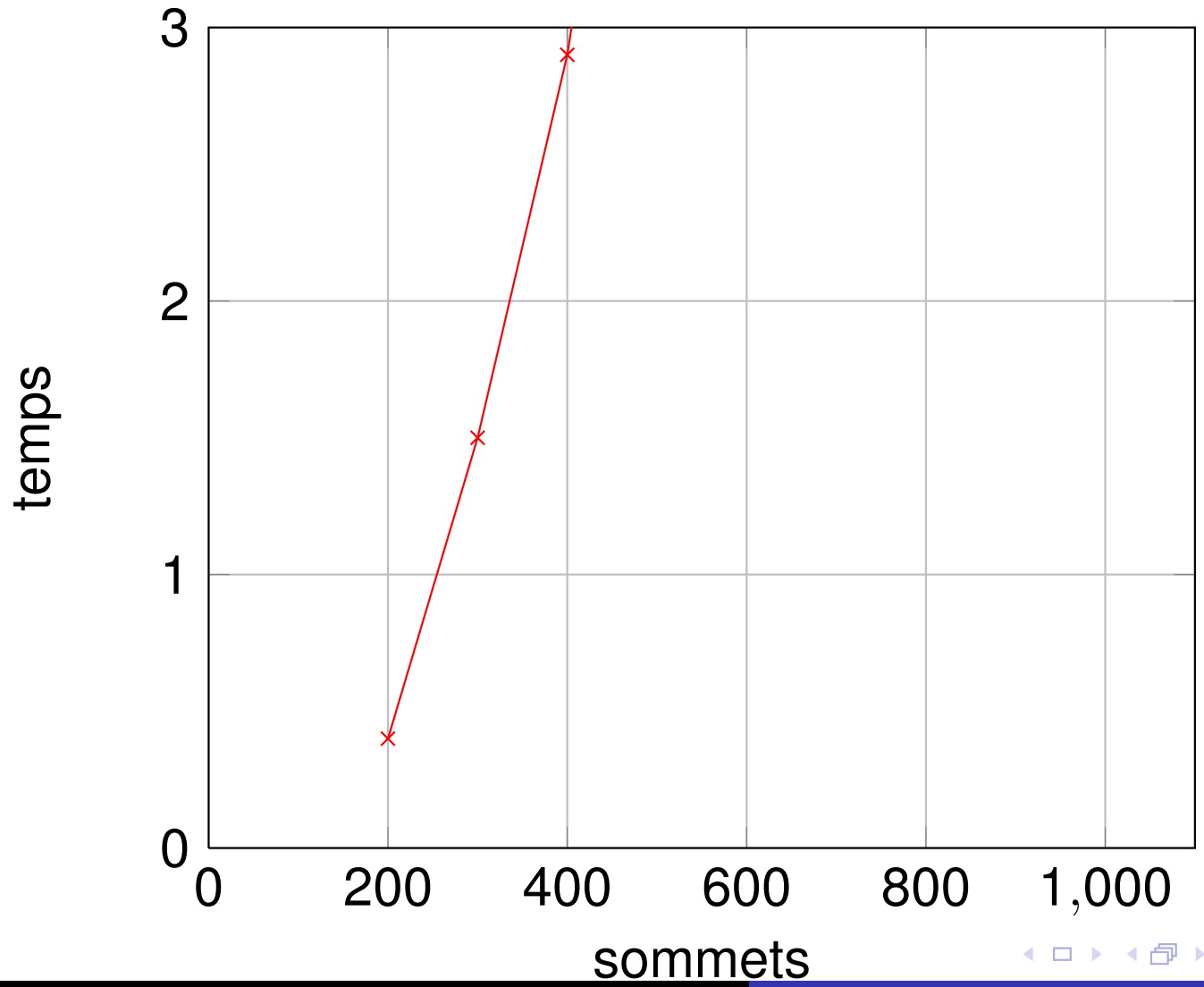
Capacity-Scaling



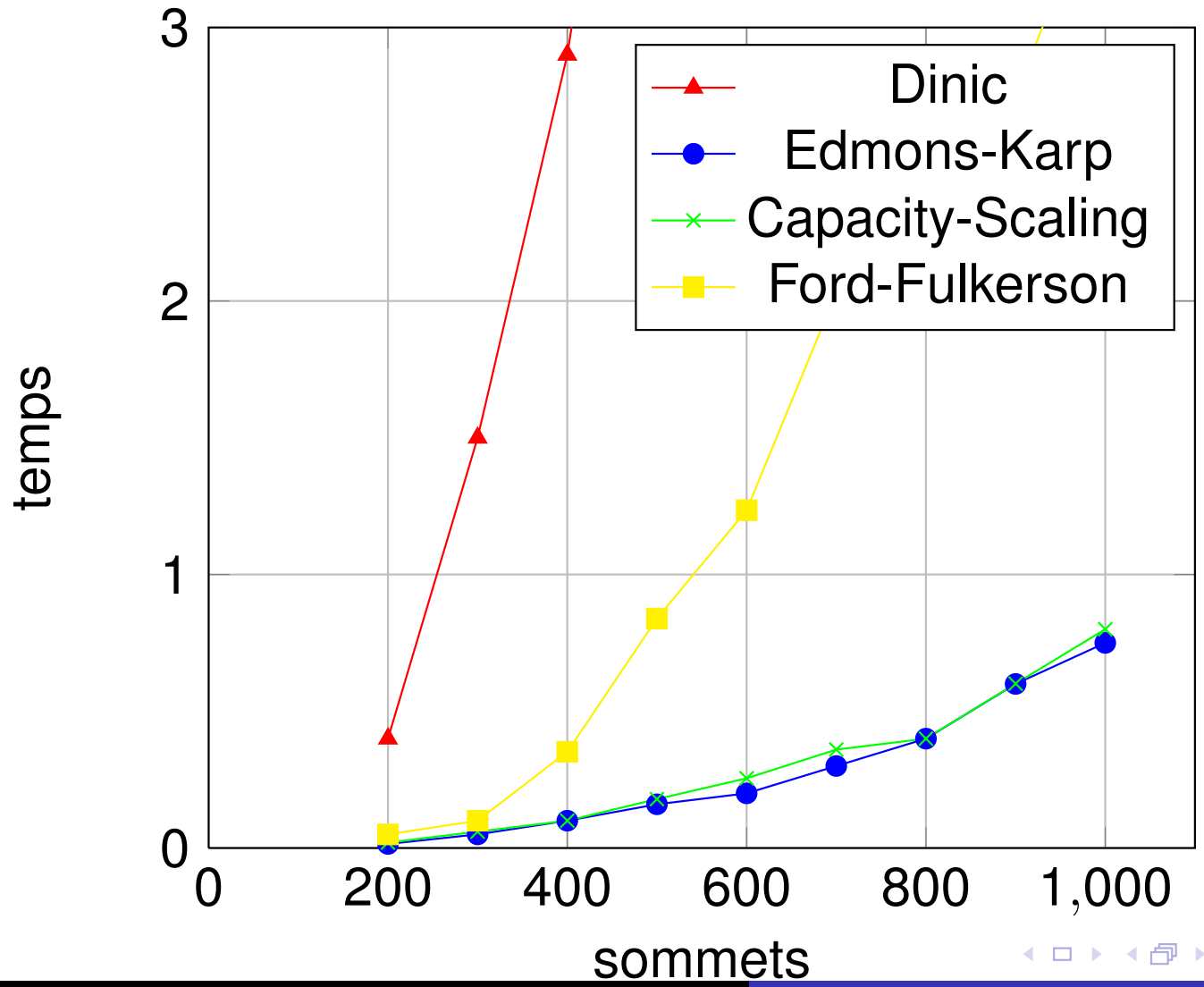
Edmons-Karp



Dinic



Comparaison des algorithmes



Evolutions

Reprntation du graphe

La reprntation sous forme de matrice nous a parue interrante dans un premier temps, de par sa simplicit'utilisation mais nous avons ris'une liste de voisins aurait certainement plus reprntative pour les parcours de chemin.

Graphe de couche

Le graphe de couche est peut-etre mal reprnter car nous l'avons reprnter comme un graphe normal, dans une matrice. Ceci gene un peu l'utilisation de ce graphe

Evolutions

Algorithme de Dinic

L'algorithme de Dinic, comme les autres, fonctionne mais il manque certainement d'optimisation

Plan

Démo