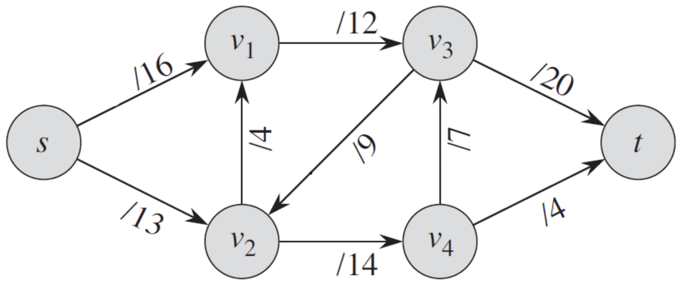
TD 5 : Flots

Jeudi 03 octobre 2024

Page web du cours : `https://people.irisa.fr/David.Baelde/algo/index.html`
Coordonnées des chargés de TD : `malo.revel@ens-rennes.fr` `isseinie.calviac@irisa.fr`
Les TDs ont lieu les jeudis de 9h45 à 11h15, se fier à ADE pour la salle.

1 Algorithme de Ford-Fulkerson



Question 1 – Appliquer l'algorithme de Ford-Fulkerson sur l'exemple ci-dessus.
Question 2 – Identifier une coupure minimale associée.

2 Extensions équivalentes du formalisme de flot

Capacité bornée des sommets

Une variante du problème de flots consiste à considérer un graphe où non seulement les arcs, mais aussi les sommets ont une capacité limitée.

Question 3 – Formaliser le problème.

Le réseau est toujours défini sur un graphe orienté $G = (V, E)$, mais on a maintenant deux fonctions de capacité $c_S : S \to \mathbb{N}$ et $c_A : A \to \mathbb{N}$. En plus de ne pas dépasser la capacité c_A sur les arcs, pour tout sommet s, la somme des flots sur les arcs entrants (ou de façon équivalente sur les arcs sortants) ne doit pas dépasser la capacité du sommet $c_S(s)$.

Question 4 – Montrer que le problème de flot maximal sur de tels graphes peut se réduire au problème de flots standard, et ce sur un graphe convenablement choisi.

On définit un nouveau graphe, où pour chaque sommet s original on crée un sommet s_{in} et s_{out}, un arc $(s_{\text{in}}, s_{\text{out}})$. Tous les arcs incidents à s sont transformés en arcs incidents à s_{in}, et de façons symétrique tous les arcs partant de s sont transformés en arcs partants de s_{out}. Enfin, la capacité de tous les arcs existants est conservée, et la capacité des arcs créés $(s_{\text{in}}, s_{\text{out}})$ est définie comme la capacité de s.

Plusieurs sources et puits

Une autre variante du problème de flots consiste à considérer un graphe avec un ensemble S de sources et un ensemble T de puits.

Question 5 – Montrer que le problème de flot maximal sur de tels graphes peut se réduire au problème de flots standard, et ce sur un graphe convenablement choisi.

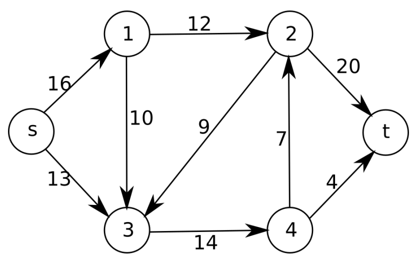
3 Algorithme d'Edmonds-Karp

L'algorithme d'Edmonds-Karp est une spécialisation de l'algorithme de Ford-Fulkerson dans lequel le chemin γ choisi pour l'exécution (reliant s à t dans le graphe résiduel R_f) est un chemin de taille minimale en nombre d'arcs.

Question 6 – Expliquer comment implémenter l'algorithme d'Edmonds-Karp.

On doit chercher les plus courts chemins d'origine s dans le graphe résiduel ; à chaque itération on va effectuer un parcours en largeur depuis s dans le graphe résiduel pour trouver le chemin de longueur minimale.

Question 7 – Appliquer l'algorithme d'Edmonds-Karp au graphe suivant :



Dans l'ordre, on trouve les chemins améliorants suivants :
- $s \to 1 \to 2 \to t$
- $s \to 3 \to 4 \to t$
- $s \to 3 \to 4 \to 2 \to t$

Étudions la complexité de cet algorithme.

Nous allons dans un premier temps montrer que si l'algorithme d'Edmonds-Karp est exécuté sur un graphe $G = (V, E)$ de source s et de cible t, alors pour tous les sommets $v \in V \setminus \{s, t\}$, la distance de plus court chemin $\delta_f(s, v)$ dans le graphe résiduel R_f croît avec chaque augmentation de flot. Nous allons pour cela procéder par l'absurde, en supposant qu'une diminution de distance est possible.

Question 8 – Soit f le flot juste avant une itération qui diminue une certaine distance de plus court chemin, et soit f' le flot juste après. Soit v un sommet tel que $\delta_{f'}(s, v) < \delta_f(s, v)$, et tel que $\delta_{f'}(s, v)$ est minimal, et soit u son prédécesseur sur un plus court chemin de s à v. Montrer que $(u, v) \notin R_f$ (où R_f est le graphe résiduel du flot f).

(pour référence, voir la démonstration du lemme 26.8 dans le Cormen)

Comme $\delta_{f'}(s, v)$ est minimal et $\delta_{f'}(s, u) < \delta_{f'}(s, v)$, on sait que la distance de s à u n'a pas diminué :

$$\delta_{f'}(s, u) \geq \delta_f(s, u).$$

On montre facilement que l'arc (u, v) n'était pas présent dans le graphe résiduel R_f ; en effet sinon on aurait eu

$$\begin{aligned} \delta_f(s, v) &\leq \delta_f(s, u) + 1 && (1) \\ &\leq \delta_{f'}(s, u) + 1 && (2) \\ &= \delta_{f'}(s, v) && (3) \end{aligned}$$

ce qui contredit l'hypothèse que $\delta_{f'}(s, v) < \delta_f(s, v)$.

Question 9 – Montrer que $\delta_f(s, v) \leq \delta_{f'}(s, v)$ et conclure.

Si on reprend, on a montré que (u, v) était dans le graphe résiduel $R_{f'}$, mais pas dans R_f, donc le chemin augmentant p choisi passe par l'arc (u, v) ou (v, u). De plus, le flot a été augmenté entre v et u ; or l'algorithme augmente le flot le long de p, donc c'est l'arc (v, u) qui est dans p. D'où

$$\begin{aligned} \delta_f(s, v) &= \delta_f(s, u) - 1 && (4) \\ &\leq \delta_{f'}(s, u) - 1 && (5) \\ &\leq \delta_{f'}(s, v) - 2 && (6) \end{aligned}$$

ce qui contredit l'hypothèse $\delta_{f'}(s, v) < \delta_f(s, v)$. Il n'existe donc pas de tel sommet v.

Donc $\delta_{f'}(s, v) \geq \delta_f(s, v)$.

On dit qu'un arc (u, v) d'un graphe résiduel R_f est critique sur un chemin améliorant p si la capacité résiduelle de p est la capacité résiduelle de (u, v).

Question 10 – Supposons que (u, v) devient critique à deux reprises durant l'exécution de l'algorithme. Montrer que, entre ces deux moments, la distance entre la source et u augmente au moins de 2.

Pour cette question et les deux suivantes, voir le théorème 26.9 dans le Cormen.

Soit u et v deux sommets de S reliés par un arc de A. Puisque les chemins améliorants sont des plus courts chemins, quand (u, v) est critique pour la première fois, on a

$$\delta_f(s, v) = \delta_f(s, u) + 1 .$$

Après augmentation du flot, l'arc (u, v) disparaît du réseau résiduel. Il ne pourra pas réapparaître sur un autre chemin améliorant tant que le flux de u à v n'aura pas diminué, ce qui n'arrive que si (v, u) apparaît sur un chemin améliorant. Si f' est le flot de G au moment de cet événement, on a

$$\delta_{f'}(s, u) = \delta_{f'}(s, v) + 1 .$$

Comme $\delta_f(s, v) \leqslant \delta_{f'}(s, v)$ d'après le lemme 26.8, on a

$$\begin{aligned} \delta_{f'}(s, u) &= \delta_{f'}(s, v) + 1 \\ &\geqslant \delta_f(s, v) + 1 \\ &= \delta_f(s, u) + 2 . \end{aligned}$$

En conséquence, entre le moment où (u, v) devient critique et celui où il devient à nouveau critique, la distance entre u et la source augmente au moins de 2. La distance

QUESTION 11 – Montrer que chacun des $|E|$ arcs peut devenir critique au plus $|V|/2 - 1$ fois.

nouveau critique, la distance entre u et la source augmente au moins de 2. La distance
entre u et la source est initialement au moins égale à 0. Les sommets intermédiaires d'un plus court chemin de s à u ne peuvent pas contenir s, u ou t (car (u, v) sur le chemin critique entraîne que $u \neq t$). Donc, jusqu'à ce que u devienne éventuellement inaccessible à partir de la source, sa distance est au plus égale à $|S| - 2$. Donc, (u, v) peut devenir critique au plus $(|S| - 2)/2 = |S|/2 - 1$ fois. Puisqu'il existe $O(A)$ paires de sommets pouvant être reliés par un arc dans un graphe résiduel, le nombre total d'arcs critiques pendant toute l'exécution de l'algorithme d'Edmonds-Karp est $O(SA)$. Chaque chemin améliorant comporte au moins un arc critique, et le théorème en découle. ❑

QUESTION 12 – Montrer que le nombre total d'augmentations de flot effectuées par l'algorithme d'Edmonds-Karp est en $O(|V| \cdot |E|)$. En déduire la complexité de l'algorithme.

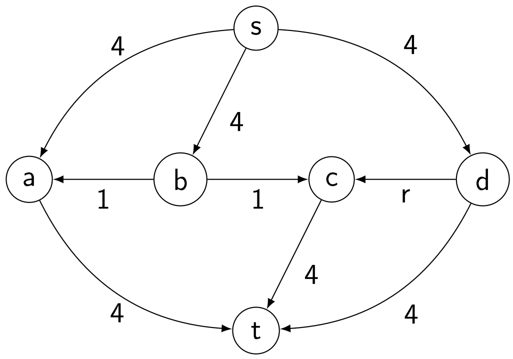
QUESTION 13 – On a vu que la complexité temporelle au pire cas de l'algorithme de Ford-Fulkerson était $O(|f^*|(|V| + |E|))$. Montrer que cette borne est atteinte.

4 Non-terminaison de Ford-Fulkerson

La précondition qui assure la terminaison de l'algorithme de Ford-Fulkerson est le fait que les capacité sont des entiers naturels.

QUESTION 14 – Est-ce que l'algorithme termine aussi avec des poids rationnels positifs ?

On pose $r = \frac{\sqrt{5}-1}{2}$. On remarque que $r + r^2 = 1$. On considère le réseau suivant :



QUESTION 15 – Appliquer l'algorithme de Ford-Fulkerson en choisissant les chemins suivants :

- D'abord le chemin (s, b, c, t) ;
- Puis successivement les chemins (s, d, c, b, a, t), (s, b, c, d, t), (s, d, c, b, a, t), puis (s, a, b, c, t) (autant de fois qu'on veut).

QUESTION 16 – Qu'en pensez-vous ? Quel est le flot maximal dans le graphe ci-dessus ?