MA351 : Introduction à la théorie des graphes

Plus courts chemins dans les graphes orientés

Yann Kieffer

ESISAR

Les problèmes de plus courts chemins

Trois formulations

1. Etant donnés $\overrightarrow{G} = (V, A)$ un graphe orienté, $l : A \rightarrow \mathcal{R}$ une fonction longueur sur les arcs, et deux sommets s et t de $\overrightarrow{G}$, trouver (la longueur d')un plus court chemin de s à t dans $\overrightarrow{G}$.
2. Etant donnés $\overrightarrow{G} = (V, A)$, $l : A \rightarrow \mathcal{R}$, et s sommet de $\overrightarrow{G}$, trouver (la longueur) des plus courts chemins de s à chaque sommet du graphe.
3. Etant donnés $\overrightarrow{G} = (V, A)$, et $l : A \rightarrow \mathcal{R}$, trouver (la longueur) des plus courts chemins de chaque sommet du graphe à chaque sommet du graphe.

La *longueur* d'un chemin est la somme de la longueur de ses arcs.

Difficultés avec la définition des plus courts chemins

Quelques observations :

- si le graphe comporte un circuit de longueur totale strictement négative, alors il peut très bien ne pas y avoir de plus court chemin de s à t, même si t est accessible depuis s !
- une parade serait d'interdire les circuits, et de travailler dans un graphe sans circuit ;
- une autre parade, celle qui sera adoptée dans la suite, sera de se limiter à des fonctions "longueur" à valeurs positives : $l : A \rightarrow \mathcal{R}_+$
- on suppose dans la suite que tout sommet est accessible depuis s.

Difficultés comparées des 3 formulations

- Le problème de la formulation 2 contient le problème de la formulation 1 ; et le problème de la formulation 3 contient le problème de la formulation 2 ;
- Si on sait résoudre la formulation 1, alors via des appels répétés de cet algorithme, on peut résoudre 2 ;
- Si on sait résoudre la formulation 2, alors via des appels répétés de cet algorithme, on peut résoudre 3 ;
- Il n'existe pas d'algorithme pour résoudre la formulation 1, sans résoudre au passage la formulation 2 ;
- Il existe un algorithme pour la formulation 3 qui soit plus rapide que la répétition n fois de la résolution de la formulation 2. Nous ne le présenterons pas dans le cadre de ce cours.

On s'attaque donc au problème suivant : [formulation 2, poids positifs] : Etant donnés $\overrightarrow{G} = (V, A)$, $l : A \to \mathcal{R}_+$, et s sommet de $\overrightarrow{G}$, trouver (la longueur) des plus courts chemins de s à chaque sommet du graphe.

Algorithme de Dijkstra

L'algorithme principal de ce chapitre est le célèbre algorithme dû à Dijkstra (1959) :

<u>Dijkstra</u>$\{\overrightarrow{G} = (V, A),\ s \in V,\ l : A \to \mathcal{R}_+\}$:
– $M \leftarrow \{s\}$
– $o \leftarrow \emptyset$
– $d \leftarrow \{(s, 0)\}$
– **<u>TantQu</u>**'il existe des arcs $\overrightarrow{uv}$ avec $u \in M,\ v \notin M$, **<u>Faire</u>** :
  ∗ soit $\overrightarrow{xy}$ un tel arc minimisant $d(u) + l(\overrightarrow{uv})$
  ∗ $M \leftarrow M \cup \{y\}$
  ∗ $o \leftarrow o \cup \{(y, x)\}$
  ∗ $d \leftarrow d \cup \{\big(y, d(x) + l(\overrightarrow{xy})\big)\}$
– **<u>Retourner</u>** (M, o, d)

Déroulons cet algorithme

Remarques sur l'algorithme

- A l'issue de l'algorithme, M est l'ensemble des sommets accessibles depuis s ; pouvez-vous justifier pourquoi ?
- cet algorithme a un lien de parenté certain avec Accessibles() ;
- la fonction $d()$, qui est une sortie, est aussi utilisée pour le choix de l'arc ; la retourner, et choisir les arcs, sont les deux seules raisons de son calcul ;
- l'intention de l'algorithme est qu'à l'issue du calcul, $d(t)$ soit la longueur d'un plus court chemin de s à t, pour tout sommet t de M.

Il nous faut maintenant prouver la correction de cet algorithme, et donc que cette intention est bien atteinte.

Théorème : correction de l'algorithme de Dijkstra()

Théorème

A l'issue de l'appel :
$(M, o, d) = Dijkstra(\overrightarrow{G} = (V, A), s \in V, l : A \rightarrow \mathcal{R}),$
la fonction $d()$ *indique la longueur d'un plus court chemin de* s *au sommet (supposé accessible) passé en argument ; et un tel plus court chemin peut être retrouvé en utilisant la fonction* $o()$ *et la routine Reconstruire_chemin().*

Définition

La *distance* du sommet s au sommet x, dans le graphe $\overrightarrow{G}$ muni des distances l, est la longueur d'un plus court chemin de s à x.

Début de la preuve de correction de Dijkstra()

Exécutabilité

En parcourant ligne à ligne l'algorithme, on constate que les seules instructions problématiques sont celles qui tentent d'accéder à une valeur de la fonction partielle $d()$ (lignes 6 et 9). Or on n'accède à $d(u)$ ou $d(x)$ que pour un u ou un x dans M (lignes 2 et 7). Mais à chaque fois qu'un sommet est mis dans M, on lui affecte une valeur dans $d()$ (lignes 4 et 9).

Arrêt

Si on oublie le choix spécifique de l'arc à l'entrée de la boucle, ne gardant que le fait que c'est un arc qui sort de M ; et si on efface les instructions d'affectation de $d()$, on retrouve la routine Accessibles(). Or on sait qu'Accessibles() s'arrête quelles que soient ses entrées ; donc Dijkstra() s'arrête également pour chacune de ses entrées.

Reste de la preuve

Correction des valeurs retournées

Il nous reste à démontrer que la fonction partielle d() retournée encode bien les distances depuis s, et que la fonction o() encode bien des plus courts chemins depuis s.
Nous allons nous focaliser sur le premier point, le second sera montré en passant.
Nous séparons la preuve en deux temps :

1. pour t dans M, la longueur d'un plus court chemin de s à t est au plus $d(t)$;
2. pour t dans M, la longueur d'un plus court chemin de s à t est au moins $d(t)$.

Cette distinction fait écho à la distinction "certificat du oui"/"certificat du non" dans les chapitres précédents.
Et comme dans les chapitres précédents, un sens est plus facile que l'autre ; nous commençons par le sens le plus facile.

"Il y a un chemin..."

Si t est dans M, alors il y a un chemin de s à t de longueur $d(t)$

Notez que la formulation du titre implique le point (1) de la diapo précédente : s'il y a un chemin de longueur $d(t)$, alors un plus court chemin est de longueur au plus $d(t)$!
Puisque t est dans M, Reconstruire_chemin(o, s, t) retourne un chemin de s à t dans $\overrightarrow{G}$: l'argument de la preuve pour Accessibles() marche à l'identique.
Il nous reste à mesurer la longueur de ce chemin.

La longueur du chemin...

Observez d'abord que les arcs empruntés par le chemin retourné par Reconstruire_chemin() sont uniquement des arcs choisis par l'algorithme Dijkstra() à l'entrée de sa boucle.
Observez maintenant que pour chaque arc $\overrightarrow{xy}$ choisi, le sommet x est stocké dans $o(y)$, et la valeur $d(x) + l(\overrightarrow{xy})$ est stockée dans $d(y)$.
L'interprétation de ces faits est la suivante : pour arriver à y, on passe par x, puis emprunte l'arc $\overrightarrow{xy}$; ce faisant, le chemin pour arriver à y a une longueur $d(x) + l(\overrightarrow{xy})$. Les chemins retournés par Reconstruire_chemin() suivent à la lettre ces principes, ce qui permet de conclure.
(Une preuve complètement rédigée devrait procéder par récurrence sur le numéro de passage dans la boucle, avec comme hypothèse : "au $k^{ème}$ passage dans la boucle, pour le sommet y rajouté à M, la longueur du chemin retourné par Reconstruire_chemin(o, s, y) est $d(y)$"...)

Correction des valeurs retournées : la réciproque

Tous les chemins de s à t sont de longueur au moins $d(t)$

Ceci est la partie la plus élaborée de la preuve. Comme dans les chapitres précédents, cette partie nécessite un mécanisme sophistiqué : ce n'est pas en regardant des chemins qu'on va montrer qu'il n'y a pas de chemin plus cours qu'une certaine longueur !
Nous allons être amenés à énoncer un lemme. . .mais avant, nous aurons besoin d'une définition, illustrée par quelques propriétés. C'est le pendant de la notion de *tri topologique* pour le chapitre sur l'existence de circuits.

La notion de potentiel

Définition

Un *potentiel* pour le graphe $\overrightarrow{G}$ et les distances $l : A \to \mathcal{R}_+$ est une fonction : $\phi : V \to \mathcal{R}$ telle que pour tout arc $\overrightarrow{xy}$ de $\overrightarrow{G}$, on ait : $\phi(y) \leq \phi(x) + l(\overrightarrow{xy})$

Proposition

La fonction $\phi : V \to \mathcal{R}$, définie par $\phi(v) = 0$ pour tout sommet v du graphe, est un potentiel pour tout graphe $\overrightarrow{G} = (V, A)$ muni de n'importe quelle fonction longueur $l : A \to \mathcal{R}_+$.

L'existence de potentiels n'est donc pas une question intéressante.

Proposition

Si ϕ est un potentiel pour $\overrightarrow{G}$ et l, et si $\lambda \in \mathcal{R}$, alors la fonction ϕ' définie par $\phi'(x) = \phi(x) + \lambda$ est également un potentiel pour $\overrightarrow{G}$ et l.

Ici comme en électricité, la valeur d'un potentiel en un "point" donnée ne signifie pas grand-chose, puisqu'on peut fixer arbitrairement le potentiel de n'importe quel point (et d'un seul !). Seules les différences de potentiels sont significatives.

Un lemme reliant potentiels et longueurs de chemins

Et voici maintenant le lemme annoncé :

Lemme

Si ϕ est un potentiel pour $\overrightarrow{G}$ et l, et si C est un chemin de s à t dans $\overrightarrow{G}$, alors nécessairement :

$$l(C) \geq \phi(t) - \phi(s)$$

où $l(C)$ désigne la longueur du chemin C.

(notez que c'est une différence de potentiels qui apparaît dans le lemmme !)
L'intérêt de ce lemme est qu'il est de la forme : "tout chemin allant du sommet s au sommet t est nécessairement d'une longueur plus grande qu'une certaine valeur" : il permet de borner *par en-dessous* des longueurs de chemins.

La preuve du lemme

Démonstration.
Notons $C = (x_0, x_1, \ldots, x_p)$. Par définition de potentiel, on a :

$$\text{pour tout } k \text{ de } 1 \text{ à } p : \phi(x_k) \leq \phi(x_{k-1}) + l(\overrightarrow{x_{k-1}x_k})$$

En sommant toutes ces inégalités, de nombreux termes disparaissent, et on obtient :

$$\phi(x_p) \leq \phi(x_0) + \sum_{k=1}^{p} l(\overrightarrow{x_{k-1}x_k})$$

c'est-à-dire :

$$\phi(t) - \phi(s) \leq l(C)$$

qui était justement ce qu'il fallait démontrer.

$\square$

Fin de la preuve de correction de Dijkstra

Nous en étions donc à...

Tous les chemins de s à t sont de longueur au moins $d(t)$

Voyez-vous le lien avec le lemme ?
Il nous suffit de montrer qu'à l'issue de Dijkstra, la fonction $d()$ est un potentiel pour $\overrightarrow{G}$ et l pour achever la preuve.
Comme la preuve est un peu technique, nous allons nous échauffer en démontrant d'abord une conséquence de ce fait.
Pour alléger les notations, on suppose pour la fin de ce chapitre que tous les sommets de $\overrightarrow{G}$ sont accessibles depuis s : donc à l'issue de Dijkstra, le M retourné est en fait V.

Lemme

La fonction $\delta : V \to \mathcal{R}$ définie par $\delta(x) = dist(s, x)$ où $dist(u, v)$ désigne la distance de u à v, c'est-à-dire la longueur d'un plus court chemin de u à v, est un potentiel.

Notez que ce lemme ne dit rien de l'algorithme de Dijkstra ; il ne parle que de distances, et la définition des distances est indépendante de tout algorithme.

Preuve de ce lemme

Démonstration.

Par définition de potentiel, il faut montrer que pour chaque arc $\overrightarrow{xy}$ de $\overrightarrow{G}$, on a : $\delta(y) \leq \delta(x) + l(\overrightarrow{xy})$.
$\delta(u)$ est la longueur d'un plus court chemin de s à u.
Considérons donc un plus court chemin C de s à x. Sa longueur est $l(C) = \delta(x)$. On lui ajoutant l'arc $\overrightarrow{xy}$, on obtient un chemin C' de s à y, de longueur $l(C') = l(C) + l(\overrightarrow{xy}) = \delta(x) + l(\overrightarrow{xy})$. On reconnaît le membre droit de l'inégalité à démontrer.
Mais comme C' est un chemin de s à y, un plus court chemin de s à y sera forcément plus court que celui-ci, donc plus court que $\delta(x) + l(\overrightarrow{xy})$: et c'est justement ce qui était à démontrer. $\square$

Avant-dernier lemme...

Lemme

La fonction d() retournée par Dijkstra est un potentiel pour $\overrightarrow{G}$ et l.

On doit montrer que : pour tout arc $\overrightarrow{vy}$ de $\overrightarrow{G}$, $d(y) \leq d(v) + l(\overrightarrow{vy})$. (Le choix des notations n'est pas innocent, et nous facilitera la suite de la preuve.)
Nous distinguerons deux cas, suivant que v soit dans M ou hors de M au moment où y est ajouté à M.
Commençons par le cas où v est dans M. y est ajouté à M via le choix de l'arc $\overrightarrow{xy}$. Notez qu'il n'est pas exclu que x puisse être v.
Dans ce cas, au moment du choix de $\overrightarrow{xy}$, l'arc $\overrightarrow{vy}$ est également un arc candidat, puisqu'à ce moment-là, v est dans M, et y n'y est pas. Comme $\overrightarrow{xy}$ a été choisi, on sait que $d(x) + l(\overrightarrow{xy}) \leq d(v) + l(\overrightarrow{vy})$. Or $d(y) = d(x) + l(\overrightarrow{xy})$: nous avons donc fini avec ce cas.
Dans le cas où v n'est pas dans M, cela signifie que y est ajouté à M avant v. Pour démontrer ce cas, nous allons montrer un dernier lemme.

C'est le lemme final !

Lemme

Les valeurs affectées à la foncion d() au cours de l'exécution de Dijkstra() sont affectées par ordre croissant.

Démonstration.

Notons $\overrightarrow{xy}$ l'arc choisi à l'étape k de l'algorithme, et $\overrightarrow{ab}$ l'arc choisi à l'étape $k+1$. Si $y = a$, alors $d(b) = d(a) + l(\overrightarrow{ab}) = d(y) + l(\overrightarrow{ab}) \geq d(y)$. Si $y \neq a$, alors a était déjà dans M à l'étape précédente, donc $\overrightarrow{ab}$ était déjà un arc candidat au moment du choix de $\overrightarrow{xy}$ à l'étape k. Donc $d(x) + l(\overrightarrow{xy}) \leq d(a) + l(\overrightarrow{ab})$, donc $d(y) \leq d(b)$. □

Ce fait étant acquis, et retournant à notre preuve, nous pouvons en déduire que $d(y) \leq d(v)$. Mais de ceci on déduit facilement que $d(y) \leq d(v) + l(\overrightarrow{vy})$, puisque l est à valeurs positives.

On a montré que d() est un potentiel pour $\overrightarrow{G}$ et l. On en déduit, d'après le premier lemme du chapitre, que tout chemin de s à t est forcément de longueur au moins $d(t) - d(s) = d(t)$. C'est la preuve de la deuxième moitié de la correction de l'algorithme de Dijkstra. Cette preuve est donc finie.

Conclusion et moralité

L'algorithme de Dijkstra calcule les distances de s à chaque sommet du graphe. La fonction o() retournée permet de retrouver des plus courts chemins de s à chaque sommet.
Afin d'écrire un algorithme avec certificats, nous allons nous intéresser a la version "décision" (à réponse oui/non) du problème de plus court chemin. Voici l'énoncé de ce problème :

Plus court chemin, version décision

Etant donnés $\overrightarrow{G} = (V, A)$, $l : A \to \mathcal{R}_+$, deux sommets s et t de $\overrightarrow{G}$, et un réel K, existe-t-il un chemin de s à t de longueur inférieure ou égale à K ?

Exercice

Ecrire un algorithme qui résoud ce problème ; chaque réponse (OUI et NON) doit être retournée accompagnée d'un certificat autonome *(dont la correction ne repose pas sur une preuve complète de l'algorithme de Dijkstra) prouvant la validité de cette réponse.*