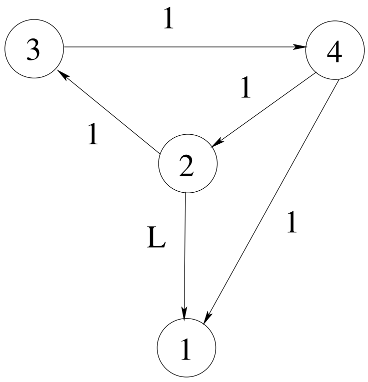
TD : Modèles et algorithmes des réseaux

18 septembre 2018

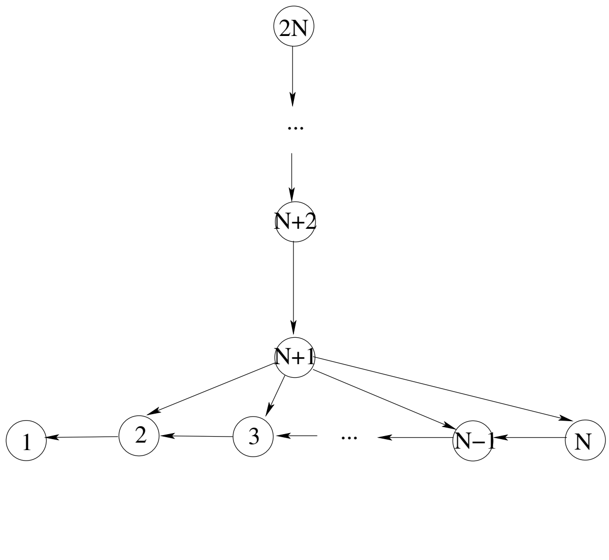
L'algorithme Bellman-Ford

Exercice 1 (Les mauvaises nouvelles se propagent lentement)**.** Supposons que les conditions initiales de l'algorithme Bellman-Ford (synchrone) sont les distances les plus courtes depuis les nœuds 2, 3, et 4 à la destination 1 avant la panne du lien $(4, 1)$. Quel est le nombre d'itérations nécessaire (en fonction de L) avant que le nœud 2 réalise que son plus court chemin vers la destination 1 passe par le lien $(2, 1)$?



Exercice 2 (Nombre de messages dans la version asynchrone)**.** On suppose que les estimations initiales de distances vers la destination 1 sont toutes à $+\infty$.

Calculer le nombre de messages envoyés dans la version asynchrone dans le pire des cas et comparer ce résultat au nombre de messages envoyés dans la version synchrone. On considère que les messages sont envoyés vers les voisins uniquement quand l'estimateur de la distance est modifié. (Remarque : deux messages envoyés simultanément depuis un sommet i vers deux sommets j et k peuvent arriver à des instants différents à j et k !)



Exercice 3 (Terminaison de l'algorithme Bellman-Ford distribué)**.** Dans cet exercice, nous cherchons à modifier l'algorithme Bellman-Ford distribué, pour que l'algorithme décide en temps réel que le calcul a terminé. Pour cela considérons la modification suivante : chaque noeud j maintient, en plus de l'estimation D_j qui lui correspond, l'identité de son plus proche parent π_j depuis la destination i. On suppose que tous les liens sont bidirectionnels (i.e. s'il existe un lien (i,j), alors le lien (j,i) existe aussi, mais on peut avoir $w_{ij} \neq w_{ji}$), et ont une valeur positive.

Initialement, $D_j = +\infty$ pour $j \neq i$ et $D_i = 0$. Le nœud i commence l'algorithme en envoyant l'estimateur D_i à tous les voisins de i.

Pour les nœuds $j \neq i$, nous effectuons les opérations suivantes :

— Quand j reçoit un message de k avec estimation de sa distance vers la destination D_k :
  - — Si $D_j \leq D_k + w_{jk}$, il envoie un accusé de réception à k.
  - — Si $D_j > D_k + w_{jk}$, il envoie un accusé de réception à son parent actuel π_j (s'il existe), actualise la valeur de π_j à k, puis il modifie son estimation D_j, et l'annonce à l'ensemble de ses voisins.

— De plus, chaque noeud envoie un accusé de réception à son plus proche parent dès qu'il a reçu un accusé de la part de tous ceux à qui il a envoyé un estimateur.

On suppose que la destination i envoie un accusé de réception à chaque estimation reçue.

1. Montrer que la destination i recevra un accusé de chacun de ses voisins, et qu'à cet instant, l'ensemble des estimateurs seront égaux à la solution du problème de plus court chemin.
2. Discuter des avantages et des inconvénients de cette technique en la comparant avec une relaxation asynchrone simple.

Semi-anneaux

Exercice 4.

1. Montrer que $([0,1], \max, \times)$ est un semi-anneau. Est-ce que c'est un dioïde ? Est-il idempotent ? Sélectif ?
2. Montrer que les seuls semi-anneaux à deux éléments sont le semi-anneau booléen et $\mathbb{Z}/2\mathbb{Z}$.
3. Montrer que le semi-anneau booléen est isomorphe au sous-semi-anneau $(\{0, +\infty\}, \oplus, \otimes)$ de $\mathbb{R}_{\min}$.

Exercice 5 (Matrices irréductibles)**.** Une matrice $M \in \mathbb{R}_{\min}^{n \times n}$ est irréductible si pour tous $i, j \in \mathbb{N}_n$, il existe $k \in \mathbb{N}^*$ tel que $M_{ij}^k < +\infty$.

1. Montrer qu'une matrice est irréductible si et seulement si le graphe associé à cette matrice est fortement connexe.
2. Montrer que si une matrice est irréductible alors il est impossible d'appliquer une même permutation à ses lignes et ses colonnes afin d'obtenir une matrice triangulaire par blocs.

Exercice 6 (Matrices inversibles)**.** Soit A une matrice de $\mathbb{R}_{\min}^{n \times n}$. On dit que A est inversible s'il existe une matrice B de $\mathbb{R}_{\min}^{n \times n}$ telle que $A \otimes B = B \otimes A = Id$ où Id est la matrice identité de $\mathbb{R}_{\min}^{n \times n}$. Une matrice diagonale D est une matrice telle que $\forall i \neq j,\ D_{ij} = +\infty$. La matrice d'une permutation σ est définie par

$$\forall i,\ P_{i\sigma(i)} = 0 \qquad \text{et} \qquad \forall j \neq \sigma(i),\ P_{ij} = +\infty.$$

Quelles sont les matrices inversibles de $\mathbb{R}_{\min}^{n \times n}$?

Exercice 7 (Unicité de Bellman-Ford)**.**

1. Traduire le théorème d'unicité de solution de l'équation de Bellman-Ford dans le cas d'un dioïde $(S, \oplus, \otimes)$ avec élément nul ε et neutre e.
2. Y trouver un contre-exemple, i.e. un dioïde (idempotent) S et un graphe pondéré G avec poids dans S tels que l'énoncé ci-dessus est faux.
3. Quelle propriété de $(\mathbb{R}_{\min}, \oplus, \otimes)$ garantit cette unicité ? Cette propriété est-elle une condition nécessaire (si S est idempotent) ?

Exercice 8. Dans cet exercice on s'intéresse à déterminer les k meilleurs chemins entre deux sommets donnés. Nous considérons l'ensemble suivant :

$$S = \{(x_1, x_2, \ldots, x_k) \in \mathbb{R}^k_{\min} \mid x_1 \leq \ldots \leq x_k\}$$

On définit l'opération $\oplus$ de manière suivante : si $x \in (x_1, \ldots, x_k)$ et $y \in (y_1, \ldots, y_k)$, alors $x \oplus y = (z_1, \ldots, z_k)$ où z_i est le i-ème plus petit élément de l'ensemble $\{x_1, y_1, x_2, y_2, \ldots\}$. L'opération $\otimes$ est définie par $x \otimes y = (z_1, \ldots, z_k)$ où z_i est le i-ème plus petits termes de la forme $x_i + y_j$ avec $i, j \in \{1, \ldots, k\}$.

1. Quelle est la complexité de calcul des opérations $\oplus$ et $\otimes$ exprimée en nombre d'opérations élémentaires (addition, comparaison) ?
2. Montrer que $(S, \oplus, \otimes)$ est un semi-anneau et donner ses éléments nul et neutre. Est-ce que c'est un dioïde ? idempotent ? sélectif ?
3. Décrire un algorithme de calcul des poids des k meilleurs chemins en utilisant le semi-anneau $(S, \oplus, \otimes)$. Préciser les conditions sous lesquelles votre algorithme se termine en un temps fini et donner une borne supérieure du nombre d'étapes nécessaires. En déduire la complexité de votre algorithme.
4. (Bonus) Soit w^* le poids optimal, i.e., le poids du chemin ayant le poids le plus faible. On fixe $\eta > 0$ et l'on dit qu'un chemin est quasi-optimal si son poids est inférieur ou égal à $w^* + \eta$. En s'inspirant du problème de k meilleurs chemins, proposer un modèle algébrique pour déterminer les poids de tous les chemins quasi-optimaux. Notons que l'on ne connaît pas *a priori* la valeur de w^*.
5. (Bonus) Peut-t-on en déduire un algorithme pour calculer le poids de tous les chemins quasi-optimaux en temps fini ? Combien d'itérations cet algorithme nécessite-t-il ?