Liens entre problèmes de plus courts chemins, de fermeture transitive et de multiplication de matrices

Vors Guillaume

13 avril 2009

Table des matières

1 Introduction

En algorithmique, il est intéressant de montrer que deux problèmes sont reliés, pour cela une approche usuelle consiste à exhiber des réductions d'un problème à l'autre. Une réduction d'un problème A à un problème B consiste à montrer que, via des transformations et des calculs, pour résoudre A je peux me ramener à résoudre B. Autrement dit, si je sais résoudre B, je peux l'utiliser pour résoudre A. Cela est le plus souvent utilisé afin de montrer la NP-complétude d'un problème, mais on peut aussi s'en servir afin de relier des complexités. Mais pour cela, il faut faire attention lors de la réduction, car si pour montrer la NP-complétude il suffit de savoir réduire en temps polynômial, si on souhaite relier des complexités, il faut connaître la complexité exacte de la réduction.

Afin d'utiliser cette notion de réduction, je vais étudier, et essayer de relier, des problèmes polynômiaux classique, pour lesquels on connaît des méthodes de résolution plus ou moins efficace. Ainsi, il se peut que l'on réussisse à obtenir de meilleures complexités en réduisant à partir d'un autre problème. Mais pour cela il faut avoir une réduction efficace, on se rend compte que maintenant la réduction utilisée est elle aussi importante.

En théorie des graphes, les problèmes concernant les chemins, tel que la recherche de chemin optimal, la fermeture transitive, sont des problèmes polynômiaux très intéressants, et surtout très importants. De même, sachant que l'on peut représenter un graphe sous forme de matrice, il peut être intéressant de regarder les problèmes matriciels tels que la multiplication de matrice.

Dans la suite, je vais commencer par exposer quelques problèmes, avec quelques unes des meilleures méthodes actuellement connues. Ensuite, dans un deuxième temps, je vais proposer des réductions entre certains de ces problèmes et voir si l'on obtient des résultats significatifs. On supposera que toute donnée est un entier, ainsi les poids, les éléments de matrices,... seront entiers.

2 Présentation des problèmes

Dans la suite, toutes les complexités sont données en fonction d'opérations de base, telle que l'addition, la comparaison, l'accès à un élément d'un tableau,... De plus on suppose que chaque graphe possède m arêtes et n sommets, et que les matrices sont des matrices carrés de taille n.

2.1 Problèmes de cheminement

Un des problèmes de cheminement le plus connu est celui du calcul de plus courts chemins à partir d'une source unique. On considère un graphe orienté dont les arêtes ont des poids positifs, on cherche l'ensemble des chemins de poids minimaux entre la source et les autres sommets du graphe. Le poids d'un chemin étant la somme des poids des arêtes qui le composent. Pour résoudre ce problème, le meilleur algorithme est l'algorithme de Dijkstra, mais celui-ci n'est valable que si les poids sont tous positifs. En représentant le graphe par listes d'adjacences (et en se servant de tas de Fibonacci dans l'algorithme), on obtient une complexité en $O(m + n*ln(n))$. Si on suppose tous les sommets accessibles depuis la source, avec Dijkstra on a une complexité en $O(m*ln(n))$, dans ce même cas l'algorithme de Gabow donne une complexité en $O(m*ln(max_{arretes}\{poids\}))$.

Une variante du problème précédent est la recherche de plus courts chemins entre tous les sommets. On se place dans le cas précédent, si ce n'est que l'on souhaite obtenir les chemins de poids minimal pour tous les couples de sommets. Une solution simple est de réduire à partir du problème précédent, il suffit d'appliquer n fois Dijkstra, ce qui donne une complexité de $O(mn + n^2 * ln(n))$ ou $O(mn*ln(n))$ si tous les sommets sont mutuellements atteignables. Il ne s'agit pas de la seule méthode possible, on verra par la suite que l'on peut réduire à partir d'autres problèmes.

On considère maintenant que les poids peuvent être négatifs, l'algorithme de Dijkstra n'est donc plus applicable. Une première remarque est le fait qu'il faille faire attention aux chemins de poids strictement négatifs (à cause des cycles de poids négatifs qui vont faire boucler à l'infini notre algorithme). Il faut alors utiliser l'algorithme de Bellman-Ford, celui nous renvoi soit le plus court chemin, soit une indication indiquant que l'on a un chemin de poids strictement négatif. Cet algorithme s'exécute en temps $O(n * m)$. Pour la seconde variante avec les poids négatifs, on a alors une complexité en $O(n^2 * m)$. D'autres algorithmes ont été crée, avec de meilleures complexités, ainsi si l'on suppose qu'il n'y a aucun circuit de poids négatifs, l'algorithme de Floyd-Warshall donne une complexité en $O(n^3)$.

Etudions maintenant le cas où il n'y a aucun poids, le plus courts chemins est alors celui contenant le moins d'arête. L'algorithme le plus simple et le plus évident pour résoudre le problème du plus courts chemins à partir d'une source unique est le parcours en largeur. Ce dernier s'exécute en temps $O(m + n)$. Lorsque l'on prend l'ensemble des sommets comme source, on a alors une complexité de $O(n * m + n * n)$.

2.2 Fermeture transitive

On considère un graphe G=(X,A), la fermeture transitive de G est le graphe G^*=(X,*), où * est définie par : (x,y)∈* si et seulement s'il existe un chemin de x à y. Dans le cas général, le calcul de la fermeture transitive d'un graphe s'effectue en temps $O(n^3)$. De plus, on peut ne travailler que sur des valeurs booléennes, ce qui se révèle plus efficace sur certains ordinateurs.

Le problème de la fermeture transitive peut être relié à de nombreux autres problèmes, mais on peut aussi en étudié certains cas particuliers. On peut citer en particulier l'existence de chemin de longueur k, le calcul de tous les chemins entre 2 sommets, mais aussi la fermeture transitive de graphe orienté acyclique, de graphe non orienté (cela revient au calcul des composantes connexes), etc...

2.3 Problèmes matriciels

Le problème matriciel le plus évident est celui de la multiplication de 2 matrices à coefficients entiers. C'est un problème très étudié et pour lequel on trouve de nombreuses méthodes de résolutions. Actuellement, la meilleure méthode permet de calculer le produit en un temps $\theta(n^{2,376})$.

Mais on peut aussi redéfinir le produit matriciel, par exemple, en remplaçant la multiplication, et/ou l'addition, par d'autres opérations. Et si on les remplace par des opérations effectuables en temps constant, alors on peut conserver les complexités que l'on a dans le cas d'un produit matriciel usuel.

Si à la place de considérer des entiers, on considère des booléens (en remplaçant la multiplication par le ET et l'addition par le OU), on obtient un problème très similaire, pour lequel, toutes les méthodes de résolutions du cas précédent sont valables. On a donc une complexité similaire (si on considère que le ET et le OU sont des opérations en temps constant).

Une première variante du cas précédent est l'élévation à la puissance d'une matrice booléenne. Ce problème se réduit facilement à partir de la multiplication de matrice booléenne. En résonnant un peu, on peut obtenir une complexité en $O(M(n) * log(k))$, où M(n) est la complexité de la multiplication, et k la puissance recherchée.

Le dernier problème est celui de la fermeture transitive d'une matrice booléenne. La fermeture transitive d'une matrice A est la matrice A* définie comme étant la plus grande puissance de $(I \vee A)$, c'est à dire $(I \vee A)^k$ tel que $(I \vee A)^k = (I \vee A)^{k+1}$. Il a été montré que si on résout la multiplication de matrice booléenne en temps M(n), alors on peut résoudre la fermeture transitive d'une matrice en temps $O(M(n) * log(n))$.

3 Réduction

3.1 Plus court chemin

3.1.1 Poids

Dans cette section, on va montrer qu'il est possible de réduire le problème de plus courts chemins sans fonction de poids à partir de celui avec fonction de poids positives. On considère une graphe quelconque, auquel on associe la fonction de poids f' définie par f'(a) = 1 pour toute arête du graphe, cela nous donne une fonction f' positive. Montrons qu'un plus courts chemins pour f' est un plus courts chemins dans le graphe. Soit $(a_1, \ldots, a_r)$ un plus courts chemins entre 2 sommets pour f', de poids r ; supposons qu'il ne soit pas optimal dans le graphe sans poids, il existe donc un chemin contenant r'¡ r arêtes. Alors ce chemin est de poids r' ¡ r pour la fonction f', donc le chemin $(a_1, \ldots, a_r)$ n'est pas optimal pour f, ce qui est une contradiction.

Cette réduction permet donc de résoudre les problèmes de plus courts chemins à partir d'une source unique, dans un graphe orienté sans poids si on sait résoudre celui du courts chemins à partir d'une source unique, dans un graphe orienté avec poids positif. Ceci est aussi valable si on souhaite trouver les plus courts chemins à partir de tous les sommets. La réduction consiste simple à définir une fonction de poids, ce qui se fait en temps constant vu que la fonction de poids est constante égale à 1. Ainsi si on sait résoudre un problème de chemin optimal avec poids positif en temps P(n,m), on sait le résoudre pour un graphe sans poids en temps P(m,n). Comme on peut s'y attendre, vu qu'avoir une fonction de poids positive quelconque est plus contraignant, cela ne permet pas d'obtenir de meilleurs résultats, car le parcours en largeur a une meilleure complexité que Dijkstra ou Gabow.

3.1.2 Produit matriciel

On commence tout d'abord par définir un nouveau produit matriciel sur les matrices à coefficient dans $N \cup \{\infty\}$, où $\infty + x = \infty$ et min(x,∞)=x. Sait A et B deux matrices, on défini la matrice produit C par $\forall i, j \in [1;n], c_{ij} = min_{k\in[1;n]}(a_{ik} + b_{kj})$. Maintenant, montrons que l'on peut réduire le problème de plus courts chemins entre tous les sommets dans un graphe orienté dont les arêtes ont des poids positifs à partir de celui du produit matriciel. Pour cela on pose P comme étant la matrice de poids, c'est à dire que P_{ij} est le poids de l'arête ij si elle existe, sinon P_{ij} vaut ∞, ou 0 si i=j.

Montrons que P^{n-1} est la matrice contenant les poids des plus courts chemins entre tous les sommets. En effet, regardons ce que vaut la matrice P^{n-1}, on a $P^{n-1}_{ij} = min_{k\in[1;n]}(P^{n-2}_{ik} + P_{kj})$, ensuite en simplifiant P^{n-2} puis P^{n-3},... on obtient $P^{n-1}_{ij} = min_{a_1,\ldots,a_{n-1}\in[1;n]}(P_{ia_1} + P_{a_1a_2} + \ldots + P_{a_{n-1}j})$. Ce qui veut dire que P^{n-1}_{ij} est le poids minimal parmi tous les chemins possibles entre i et j.

Cette méthode est constructive, mais pour cela il faut prendre soin de noter quels sont les arêtes prisent en compte lors du calcul (ce qui ne rajoute rien à la complexité). Ainsi, si on sait calculer le produit matriciel ci-dessus en temps M(n), on peut calculer les plus courts chemins en temps $O(n * M(n))$. Un premier constat, cette nouvelle complexité ne dépend pas du nombre d'arête. De plus on peut améliorer le facteur n devant M(n) si on a $P^i * P = P * P^i$ pour tout i, en effet dans ce cas là on a une complexité en $O(log(n) * M(n)$. De plus, s'il n'existe aucun cycle de longueur strictement négative, alors cette méthode est valable pour des graphes avec une fonction de poids quelconque.

3.2 Multiplication et existence de chemin

3.2.1 Multiplication simple de 2 matrices booléenne

L'objectif est de montrer une réduction entre la multiplication de matrice booléenne est la recherche de chemin de longueur exactement égale à 2. On considère 2 matrice booléenne A et B, qui peuvent être considérées comme les matrice d'adjacence de deux graphes orientés G_a et G_b, dont les sommets sont numérotés de 1 à n. Construisons le graphe G_{ab} de la manière suivante, il s'agit d'un graphe possédant 2n sommets, numérotés $1_a, \ldots, n_a, 1_b, \ldots, n_b$. Si (x,y) est une arête de G_a, alors (x_a, y_b) est une arête de G_{ab}, et si (x,y) est une arête de G_b, alors (x_b, y_a) est une arête de G_{ab}, le graphe G_{ab} est donc un graphe biparti (avec d'un côté les indices a, de l'autre les indices b). Montrons maintenant que, si on note C = $A * B$, C_{ij} vaut 1 si et seulement s'il existe un chemin de longueur exactement égale à 2 entre i_a et j_a dans G_{ab}.

D'après la définition de la multiplication de matrice booléenne, on a $C_{ij} = \vee_{k \in [1;n]}(A_{ik} \wedge B_{kj})$. C'est à dire que C_{ij} vaut 1 si et seulement s'il existe un k tel que (i,k) soit une arête de G_a et (k,j) une arête de G_b. Si un tel k existe, alors (i_a, k_b, j_a) est un chemin de longueur 2 dans G_{ab}. S'il existe un chemin de longueur 2 dans G_{ab} alors, vu que G_{ab} est un graphe biparti, il existe un sommet k_b tel que (i_a, k_b, j_a) soit un chemin de longueur 2, et par définition on a (i,k) est une arête de G_a et (k,j) est une arête de G_b.

Supposons que l'on sait calculer l'existence d'un chemin de longueur exactement 2 entre 2 sommet en temps M(n,m). La construction du graphe G_{ab} se fait en temps $O(m_a + m_b + n)$, sachant que m_a et m_b sont bornés par n^2. On peut donc calculer la multiplication de matrice booléenne en temps $O(n^2 * M(n,m))$.

3.2.2 Puissance d'une matrice booléenne

Si on étudie le cas précédent avec A=B, on se rend compte que l'on a pas besoin de créer un nouveau graphe car le graphe dont A est la matrice d'adjacence est suffisant. Cela laisse penser que l'on a à faire à un cas plus simple, ou du moins possédant des caractéristiques plus simples, et donc proposant sûrement de meilleurs résultats. Commençons par montrer que A^2_{ij} vaut 1 si et seulement s'il existe un chemin de taille exactement 2 entre i et j dans le graphe dont A est la matrice d'adjacence. Ceci est une simple utilisation de la définition de la multiplication de matrices booléennes, A^2_{ij} vaut 1 si et seulement s'il existe un k tel que (i,k) et (k,j) soit des arêtes du graphe, ce qui est la définition d'un chemin de longueur 2.

Montrons maintenant que $\forall k \in N$, A^k_{ij} vaut 1 si et seulement s'il existe un chemin de taille exactement k entre i et j dans le graphe dont A est la matrice d'adjacence. On va démontrer le résultat par récurrence, car on sait déjà qu'il est vrai pour k égal à 1 et à 2. On suppose qu'il est vrai pour k = r-1, montrons qu'il est alors vrai pour k =r. On sait que $A^r = A^{r-1} * A$ donc $A^r_{ij} = \vee_{k\in[1;n]}(A^{r-1}_{ik} \wedge A_{kj})$, donc A^k_{ij} vaut 1 si et seulement s'il existe un k tel que A^{r-1}_{ik} et A_{kj} valent 1, or A^{r-1}_{ik} vaut 1 si et seulement s'il existe un chemin de taille r-1 entre i et k dans le graphe, ce qui nous donne le résultat souhaité.

Supposons que l'on sait calculer s'il existe un chemin de taille k entre deux sommets en un temps M(k,n,m), alors on peut effectuer l'élévation d'une matrice à la puissance k en un temps $O(n^2 * M(k,n,m))$. Cela est un résultat intéressant dans le sens où on peut espérer que la fonction M(k,n,m) varie faiblement en fonction de k, ainsi pour un k assez élevé, on aura un résultat significatif en terme de complexité.

3.3 Fermetures transitives

La dernière réduction de problème est une réduction significative, dans le sens où elle permet une amélioration de complexité. Voici le théorème :

Soit G un graphe, et A sa matrice d'adjacence, la fermeture transitive de A est la matrice d'adjacence de la fermeture transitive de G.

Si ce théorème est vrai, alors si on sait calculer la fermeture transitive d'un graphe en temps M(n,m), on sait calculer la fermeture transitive d'une matrice avec la même éfficacité, et vice-versa.

La fermeture transitive de A est $(I \vee A)^k$ tel que $(I \vee A)^k = (I \vee A)^{k+1}$, on pose G' le graphe dont $(I \vee A)$ est la matrice d'adjacence. Il est nécessaire de remarquer la présence de I dans la formule, et de comprendre son utilité, en effet cela veut dire qu'il existe dans le graphe G' une arête reliant chaque sommet à lui même. Donc s'il existe un chemin de longueur k entre i et j dans G', il en existe un de longueur k', pour tout k' supérieur ou égal à k. Donc on se rend compte que k est inférieur ou égal à n, car s'il existe un chemin de taille r ¿ n, on peut en suprimer les cycles, pour avoir un chemin de taille inférieure ou égale à n, et donc un chemin de taille n. Ainsi la fermeture transitive de A vaut $(I \vee A)^n$ (bien que n ne soit pas forcément le plus petit facteur possible).

Remarquons maintenant que si 2 sommets sont reliés dans G, ils le sont dans G', à l'exeception d'un sommet avec lui même. Donc les fermetures transitive de G et de G' sont égales. Et la fermeture transitive de A est la matrice d'adjacence de la fermeture transitive de G'. Ceci conclu la preuve du théorème, et nous permet de pouvoir "unifier" les complexités des 2 problèmes.

4 Conclusion

La réduction de problème est un outil très puissants, et pas seulement pour la NP-complétude. Car elle permet de relier entre eux des problèmes n'étant pas forcement considérer comme étant semblables. De plus on a vu qu'il était extrêmement simple d'obtenir des réductions si les problèmes s'y prêtent.

Mais dans ce document, il n'y a que des réductions assez simples, entre des problèmes plutôt "facile" à relier. Il peut être intéressant de chercher d'autres réductions, peut être plus efficace. Il faut aussi voir le fait que chercher des réductions peut amener à trouver des méthodes, n'étant pas des réductions, mais qui permettent de trouver de meilleures complexités. Ainsi le problème des réductions est un problème qu'il sera très intéressant d'explorer, et qui pourra avoir de grandes répercussions.