ALGORITHMIQUE ET COMPLEXITE

Master Informatique, première année, janvier 2016

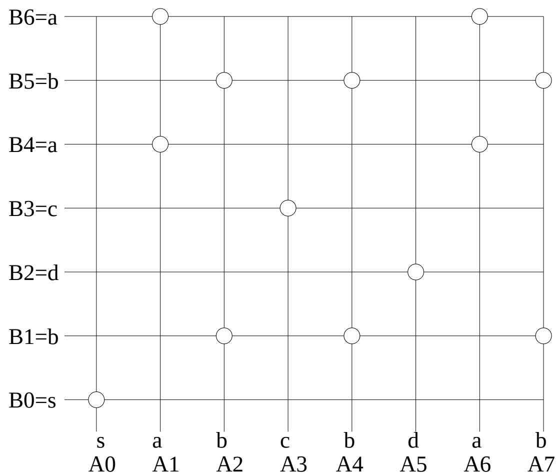
LA CLARTE DE VOS REPONSES EST ESSENTIELLE. AUCUN DOCUMENT N'EST AUTORISE, A PART VOTRE COPIE. REPONDEZ AUX QUESTIONS DANS L'ORDRE. NUMEROTEZ VOS REPONSES.

1 Quizz (8 points)

1. Citez deux problèmes résolus en TP par programmation dynamique.
2. Un algorithme en temps polynomial pour factoriser un grand entier est-il connu ? Répondre soit oui, soit non.
3. Citez deux problèmes indécidables en informatique.
4. Citez une classe de problèmes solubles mais difficiles en informatique.
5. Comment calculeriez-vous le second chemin le plus court dans un graphe ?
6. Qu'est-ce qu'un couplage dans un graphe biparti ? Qu'est-ce qu'un couplage parfait dans un graphe biparti ?
7. A quel autre problème peut-on réduire la recherche d'un couplage de cardinalité maximale dans un graphe biparti ?
8. Soit un graphe biparti complet, où les deux ensembles de sommets ont le même nombre de sommets. Les arêtes sont étiquetées par des coûts (non négatifs). A quel autre problème peut-on réduire la recherche d'un couplage parfait de coût minimal dans ce graphe biparti complet ?

2 Plus longue séquence commune (4 points)

Le problème de la sous séquence la plus longue et commune à deux séquences données est réductible au problème du chemin le plus long dans un graphe sans circuit. Pour les séquences $A = sabcbdab$ et $B = sbdcaba$, dessinez (sur votre copie) le graphe correspondant ; les arcs de transitivité peuvent ne pas être tracés pour simplifier ; vous pouvez ajouter un sommet terminal commun si vous le souhaitez ; marquez sur votre dessin la ou les séquences communes les plus longues ; écrivez-les.



3 Problème SAT (4 points)

On considère le problème 2-SAT :

$$(x \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$$

1. Exprimez ce problème comme un problème de programmation linéaire. Vous utiliserez la convention : vrai vaut 0, faux vaut 1. Vous utiliserez les variables x, $\bar{x}$, y, $\bar{y}$, et les variables d'écart c_1, c_2, c_3, c_4 pour les 4 clauses $x \vee y$, $x \vee \bar{y}$, $\bar{x} \vee y$, $\bar{x} \vee \bar{y}$.

2. Dessinez le polytope admissible dans le plan x, y ; pour cela vous ignorerez les contraintes d'intégrité ($x \in \mathbb{N}$, etc). Est-il vide ? Quelle est son aire ?

3. Pourquoi la programmation linéaire ne peut-elle pas résoudre les problèmes SAT ?

4 Programmation linéaire (4 points)

Rappel. Soit un problème de programmation linéaire, dont le polytope admissible (*feasible*) est : $\max C.X$ sous conditions : $AX \leq B$, et $X \geq 0$; pour trouver un point (non optimal) dans le polytope admissible, on considère le problème auxiliaire : $AX - z \leq B, \min z$ où z est une nouvelle variable. Pour résoudre ce problème auxiliaire, il suffit de partir du simplexe initial $X = 0$ et $z = \max -B_i$. Si le z optimal est nul, alors un point (non optimal) dans le polytope admissible a été trouvé ; sinon le polytope admissible est vide.

Soit le problème : $x \leq 3, y \leq 3, -x-y \leq -5$. L'objectif n'a pas d'importance.

1. Dessinez le polytope admissible, dans le plan x, y.

2. Trouvez un point du polytope admissible en résolvant le problème auxiliaire par la méthode du simplexe. A part les variables x, y, z, vous utiliserez comme variables d'écart x' pour $x \leq 3$, y' pour $y \leq 3$, w pour $-x - y \leq -5$.

SOLUTION

5 Quizz

1. Citez 2 problèmes résolus en TP par programmation dynamique.

Le parenthésage optimal pour le produit de matrices. Le sac à dos avec des poids entiers. La séquence commune la plus longue entre deux séquences donnés (problème suivant). Le chemin critique (le chemin le plus long dans un graphe sans circuit, autrement dit le calcul des dates au plus tôt et au plus tard dans un diagramme PERT). La sous séquence monotone croissante la plus longue dans une séquence donnée.

2. Un algorithme en temps polynomial pour factoriser un grand entier est-il connu ? Répondre soit oui, soit non.

Non.

Précision : pas sur les ordinateurs classiques. C'est peut être un problème "intermédiaire" : à coup sûr, il est dans NP et CO-NP (il y a un certificat polynomial si le nombre est premier, et un certificat polynomial (un diviseur non trivial) si le nombre est composé) ; mais peut-être n'est-il pas dans P. On ignore si la classe des problèmes intermédiaires : (NP $\cap$ CO-NP) - P est vide ou non aujourd'hui.

Le cryptage RSA utilise cette difficulté de factoriser de grands entiers : la clef publique est un grand entier produit de deux très grands premiers ; ces diviseurs sont la clef privée.

Shor a proposé un algorithme polynomial sur les machines quantiques. Nous ne savons pas construire de telles machines aujourd'hui. La possibilité de l'existence de telles machines est très discutée.

3. Citez deux problèmes indécidables en informatique.

L'égalité de deux fonctions, par exemple de $\mathbb{N} \to \mathbb{N}$. L'égalité de deux nombres réels. La résolution des équations diophantienne. L'arrêt d'une machine de Turing (en fait, d'un programme). L'égalité de deux groupes infinis non commutatifs. L'équivalence entre deux mots (problème de Post). La preuve qu'une formule mathématique est exacte.

4. Citez une classe de problèmes solubles mais difficiles en informatique.

La classe NP. (Précision non demandée : en font partie les problèmes NP-complets (3-SAT, coloriage, chemin/circuit hamiltonien), les NP-durs (MAX-SAT, voyageur de commerce).

5. Comment calculeriez-vous le second chemin le plus court dans un graphe ?

En le réduisant à un problème de programmation linéaire, et en ajoutant une contrainte pour éliminer le chemin le plus court. C'est la solution proposée en cours.

Autre méthode : calculer le plus court chemin dans le graphe G (par Dijkstra ou un autre algorithme). Pour chaque arc a du plus court chemin, définir G_a comme le graphe G où l'arc a a été détruit, et calculer le plus court chemin dans G_a ; soit C_a le résultat. Le plus court des C_a est le second chemin le plus court dans le graphe G.

6. Qu'est-ce qu'un couplage dans un graphe biparti ? Qu'est-ce qu'un couplage parfait dans un graphe biparti ?

Un couplage est un sous-ensemble des arêtes, tel qu'en tout sommet du graphe, au plus une des arêtes incidentes appartienne au couplage. Le couplage est parfait si tout sommet a une (et donc une seule) de ses arêtes incidentes dans le couplage.

7. A quel autre problème peut-on réduire la recherche d'un couplage de cardinalité maximale dans un graphe biparti ?

A un problème de flot maximal, et donc à un problème de programmation linéaire. La matrice sous jacente est totalement unimodulaire ; il suffit donc de résoudre sans tenir compte de la contrainte d'intégrité.

8. Soit un graphe biparti complet, où les deux ensembles de sommets ont le même nombre de sommets. Les arêtes sont étiquetées par des coûts (non négatifs). A quel autre problème peut-on réduire la recherche d'un couplage parfait de coût minimal dans ce graphe biparti complet ?

A un problème de flot de débit donné (maximal), et de coût minimal, donc à un problème de programmation linéaire. La matrice sous jacente est totalement unimodulaire ; il suffit donc de résoudre sans tenir compte de la contrainte d'intégrité.

6 Plus longue séquence commune et chemin le plus long

La plus longue séquence commune à deux séquences est un chemin critique dans le graphe acyclique associé. Tous les arcs ont la même durée, 1. Tous les arcs vont de gauche à droite et de bas en haut. En effet, les lettres de la séquence commune doivent apparaître dans le même ordre dans la séquence A et la séquence B.

Sur la figure, les sommets sont étiquetés avec leur date au plus tôt et au plus tard. Les sommets critiques ont leur date au plus tôt égal à leur date au plus tard. Les sommets critiques donnent les séquences communes les plus longues.

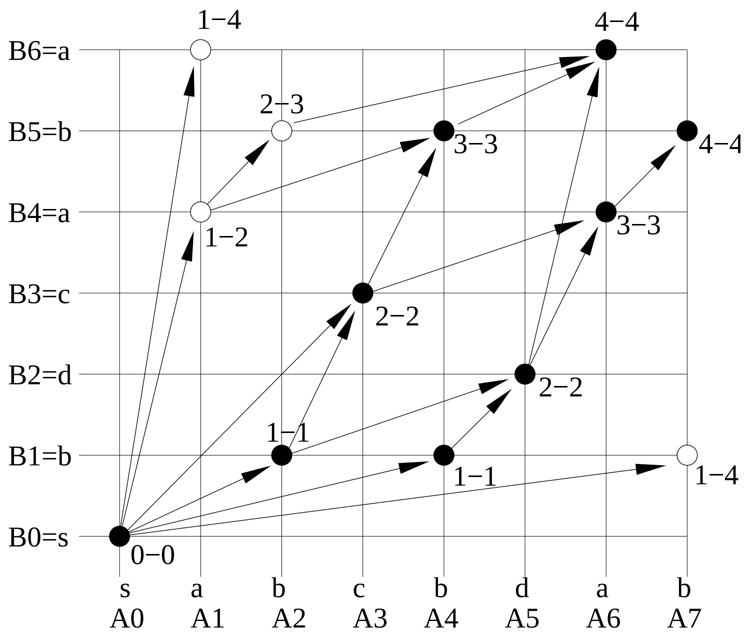
Entre $A = sabcbdab$ et $B = sbdcaba$, les séquences communes les plus longues sont : *sbdab*, *sbcab* et *sbcba* ; dans toutes, il y a 5 lettres communes.

7 Problème SAT et programmation linéaire

On considère le problème 2-SAT :

$$(x \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$$

1. Exprimez ce problème comme un problème de programmation linéaire. Vous utiliserez la convention : vrai vaut 0, faux vaut 1. Vous utiliserez les variables x, $\bar{x}$, y, $\bar{y}$, et les variables d'écart c_1, c_2, c_3, c_4 pour les 4 clauses $x \vee y$, $x \vee \bar{y}$, $\bar{x} \vee y$, $\bar{x} \vee \bar{y}$.



$x \in \mathbb{N}, y \in \mathbb{N}$

Les contraintes linéaires sont :

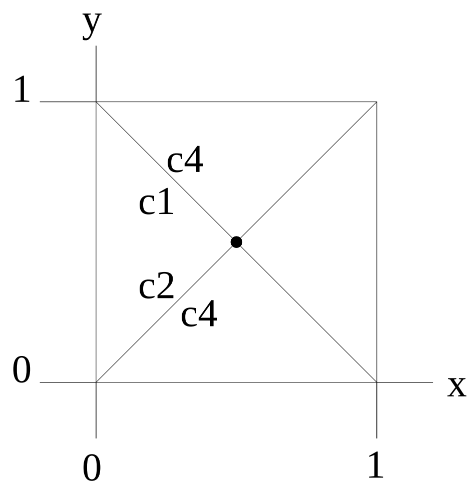
$$x + \bar{x} = 1$$
$$y + \bar{y} = 1$$

et

$$\begin{array}{llll}
x \vee y & \Rightarrow x + y \geq 1 & \Rightarrow -x - y \leq -1 & \Rightarrow -x - y + c_1 = -1 \\
x \vee \bar{y} & \Rightarrow x + \bar{y} \geq 1 & \Rightarrow -x - \bar{y} \leq -1 & \Rightarrow -x - \bar{y} + c_2 = -1 \\
\bar{x} \vee y & \Rightarrow \bar{x} + y \geq 1 & \Rightarrow -\bar{x} - y \leq -1 & \Rightarrow -\bar{x} - y + c_3 = -1 \\
\bar{x} \vee \bar{y} & \Rightarrow \bar{x} + \bar{y} \geq 1 & \Rightarrow -\bar{x} - \bar{y} \leq -1 & \Rightarrow -\bar{x} - \bar{y} + c_4 = -1
\end{array}$$

L'objectif n'a pas d'importance ; tout point entier admissible (*feasible*) donne une solution.

2. Dessinez le polytope admissible dans le plan x, y ; pour cela vous ignorerez les contraintes d'intégrité ($x \in \mathbb{N}$, etc). Est-il vide ? Quelle est son aire ?



Le polytope admissible (en ignorant les contraintes $x \in \mathbb{N}, y \in \mathbb{N}$) ne contient qu'un seul point : $(x, y) = (1/2, 1/2)$ (donc son aire est nulle). Comme ce point n'a pas des coordonnées entières, le problème 2SAT n'a pas de solution.

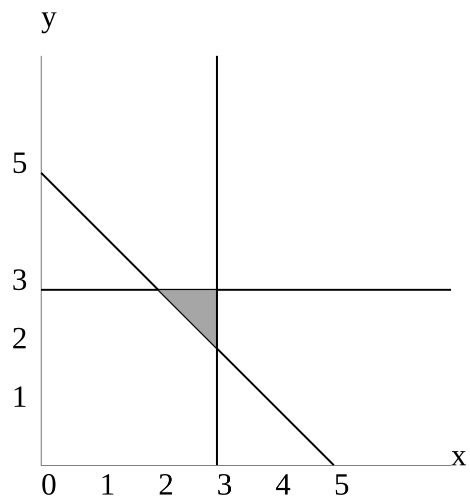
3. Pourquoi la programmation linéaire ne peut-elle pas résoudre les problèmes SAT ?

La matrice du problème de programmation linéaire n'est pas totalement unimodulaire. Dans les cas où cette matrice est totalement unimodulaire (problème de flot de débit maximal, ou de coût minimal et de débit donné par exemple maximal, problème de plus court chemin), il suffit de résoudre le problème de programmation linéaire sans tenir compte des contraintes d'intégrité, et la solution sera entière (sous conditions que B est à coordonnées entières).

8 Programmation linéaire

Soit le problème : $x \leq 3, y \leq 3, -x-y \leq -5$. L'objectif n'a pas d'importance.

1. Dessinez le polytope admissible, dans le plan x, y.



Le polytope admissible est un triangle de sommets : $(3,2),(3,3),(2,3)$.

2. Trouvez un point du polytope admissible en résolvant le problème auxiliaire par la méthode du simplexe. A part les variables x, y, z, vous utiliserez comme variables d'écart x' pour $x \leq 3$, y' pour $y \leq 3$, w pour $-x-y \leq -5$.

$$\begin{aligned}
&\min z \Rightarrow \max -z \\
&x + x' - z = 3 \\
&y + y' - z = 3 \\
&-x - y + w - z = -5
\end{aligned}$$

Le simplexe initial est :

$$\begin{aligned}
&\max -5 + x + y - w \\
&x' = 3 - x + z = 8 - 2x - y + w \\
&y' = 3 - y + z = 8 - x - 2y + w \\
&z = 5 - x - y + w
\end{aligned}$$

Autrement dit, dans le simplexe initial, $x = y = 0$ et z est le plus grand possible, conformément au rappel de l'énoncé du problème.

On pivote sur x et x' :

$$\begin{aligned}
&\max -1 + y/2 - w/2 - x'/2 \\
&x = 4 - y/2 + w/2 - x'/2 \\
&y' = 4 - 3/2y + w/2 + x'/2 \\
&z = 1 - y/2 + w/2 + x'/2
\end{aligned}$$

On pivote sur y et z :

$$\begin{aligned}
&\max -z \\
&x = 3 - x' + z \\
&y' = 1 - w - x' \\
&y = 2 + w + x' - 2z
\end{aligned}$$

C'est terminé : le point $(x, y) = (3, 2)$ appartient au polytope admissible. Ce n'est pas la seule solution ; l'autre est $(x = 2, y = 3)$. Elle est obtenue en pivotant d'abord sur y, y'.