Complexité paramétrée

Christophe Paul

Examen (rattrapge) Mars 2014

Notes de cours autorisées. **2h00**

Note: La clarté, la concision et ainsi que la complétude des preuves seront prises en compte dans la notation. Le sujet est beaucoup trop long, n'essayer pas de tout faire, mais essayez d'attaquer au moins une question difficile, marquée par ($\star$) ou ($\star\star$).

1 Noyaux

On s'intéresse au jeu de billes colorées suivant :

- *Entrée :* Soit B un ensemble de n billes valuées $\omega : B \to \mathbb{N}$ et colorées $\chi : B \to \mathcal{C}$ avec $\mathcal{C}$ l'ensemble des couleurs. Soit σ une séquence – permutation – des billes de B.
- *Question :* Peut-on supprimer un ensemble $S \subseteq B$ de billes de poids total au plus $k \in \mathbb{N}$ (i.e. $\omega(S) = \sum_{b \in S} \omega(b) \leqslant k$) tel que les billes d'une même couleur dans la séquence résultante soient consécutives ?

1. **Règle 1 :** Montrer que toute paire de billes consécutives b_1 et b_2 de même couleur – $\chi(b_1) = \chi(b_2) = c \in \mathcal{C}$ – peuvent être remplacées par une nouvelle bille de c et poids $\omega(b_1) + \omega(b_2)$.
2. Une couleur est dite *saine* si elle ne colore qu'une seule bille, *malsaine* sinon.
   (a) Est-il possible de supprimer une bille ayant une couleur saine ? Si oui justifiez votre réponse par une preuve, sinon expliquez pourquoi (contre-exemple).
   (b) **Règle 2 :** Montrer que si b_1 et b_2 sont deux billes consécutives ayant chacune une couleur saine, il est possible de recolorer b_1 avec $\chi(b_2)$ (sans changer le poids) ? Si oui justifiez votre réponse par une preuve, sinon expliquez pourquoi (contre-exemple).
3. **Règle 3 :** Montrer que s'il existe au moins $2k + 1$ couleurs malsaines, alors l'instance est négative.
4. Déterminer le nombre maximum de billes d'une instance réduite par les règles précédentes.
5. Une instance réduite par les règles précédentes est-elle un noyau ? (*indication :* quelle est la taille d'un codage d'une instance de n billes dont le poids maximum est M ?) Si oui, justifiez votre réponse. Si non, ajoutez une règle permettant d'obtenir un noyau que vous démontrerez.

2 Ensemble Dominant

Paramétré par la taille de la solution le problème Ensemble Dominant est connu pour être $W[2]$-complet sur les graphes quelconques.

2.1 Graphes d-dégénérés

Un graphe est *d-dégénéré* si tout sous-graphe contient un sommet de degré au plus d. Par exemple, les graphes planaires sont 5-dégénérés.

1. On veut montrer que sur les graphes de degré borné d, le problème Ensemble Dominant, paramétré par la taille k de la solution, est FPT.

   (a) Soit x un sommet du graphe donné G. On observe que pour tout ensemble dominant D, soit D contient x, soit D intersecte le voisinage $N(x)$ de x. Quel est le nombre maximum d'ensembles obtenus par l'intersection de D avec le voisinage $N(x)$ d'un sommet x (en fonction de d)?

   (b) En déduire un algorithme d'arbre de recherche dont vous analyserez la complexité. Justifiez la correction de votre algorithme.

2. L'algorithme précédent peut-il se généraliser aux grahes planaires. Expliquez la modification nécessaire et analyser la complexité.

2.2 Décomposition arborescente et programmation dynamique

Intéressons nous maintenant au problème Ensemble Dominant paramétré par la largeur arborescente $tw(G) = k$ du graphe donné G. Nous souhaitons proposer un algorithme par programmation dynamique sur un arbre de décomposition *simple*. Nous reprendrons les notations vues en cours par rapport à chaque nœud t de l'arbre de décomposition: X_t l'ensemble de sommets associé à t; V_t l'ensemble des sommets associés à un descendant de t; G_t le graphe induit par V_t;... Rappelons que les nœuds de l'arbre de décomposition simple sont de quatre types: *feuille, ajout, suppression, fusion.*

L'idée de l'algorithme de programmation dynamique repose sur le calcul de solutions partielles en chaque nœud t de l'arbre de décomposition. Il s'agira de composer entres elles les solutions partielles des fils $t_1, \dots t_l$ d'un nœud t pour obtenir les solutions partielles associées à t (et donc à G_t). Pour cela, nous proposons de colorier les sommets d'un nœud t de l'arbre de décomposition à l'aide de trois couleurs afin de décrire leurs statuts vis à vis d'une solution partielle de G_t:

- *Rouge :* les sommets de X_t sélectionnés dans l'ensemble dominant
- *Jaune :* les sommets de X_t déja dominés
- *Bleu :* les sommets de X_t pas encore dominés

Nous pouvons supposer que les colorations des sommets de $V_t \setminus X_t$ sont héritées des solutions partielles associées aux fils de t. Mais nous remarquons ainsi que pour qu'une solution partielle de G_t soit valide, les sommets bleus sont néssairement des sommets de X_t.

Ainsi nous noterons $DS(t, R, J, B)$ un ensemble S de taille minimum dans G_t tel que S domine exactement $V_t \setminus B$ (i.e. aucun sommet de B n'est dominé par S) et $S \cap X_t = R$

1. Quelle est la taille des tables à stocker pour chaque nœud t d'un arbre de décomposition enraciné de largeur k ?
2. Comment déduire la taille du plus petit ensemble dominant de G de la table associée à la racine de l'arbre de décomposition ?
3. Que dire des ensembles $DS(t, R, J, B)$ où B contient des sommets non présents dans $X_{t'}$ avec t' le père de t ?
4. **(*)** Formule permettant de calculer $DS(t, R, V, B)$ pour chaque 3-coloration des sommets de X_t. Pour cela nous distinguerons le type du nœud t:
   - lorsque t est une feuille.
   - lorsque t est un nœud *ajout* (en fonction des valeurs de la table de l'unique fils t')
   - lorsque t est un nœud *suppression* (en fonction des valeurs de la table de l'unique fils t')
   - lorsque t est un nœud *fusion* (en fonction des valeurs des tables des deux fils t_1 et t_2)
5. **(*)** Quelle est la complexité de l'algorithme de programmation dynamique résultant ?