CentraleSupélec

Algorithmique et complexité

TD 7/7 – Résolution problèmes NP-difficiles

Exercices complémentaires

Remarque: Les éléments de correction fournis dans le sujet ne sont pas forcément complets. Ils sont là pour vous guider dans votre travail personnel. Nous vous invitons à rédiger une solution comme vous le feriez à l'examen et, si vous avez des questions, à vous tourner vers votre chargé de TD.

Exercice 1 : Glouton

Question 1

Proposez un algorithme glouton pour le problème du rendu de monnaie.

Élements de correction :

Sélectionner la pièce de valeur la plus grande et inférieure à la somme restante, mettre à jour la somme restante et recommencer.

Question 2

Est ce que cet algorithme donne la solution optimale ? Prouvez le.

Élements de correction :

Non dans le cas général. Un contre exemple : l'ensemble de pièces $1, 3, 4$ et la valeur à rendre 6. L'algorithme glouton utilise 3 pièces alors que la solution optimale n'en utilise que 2.

Il existe des ensembles de pièces pour lesquels l'algorithme glouton est optimal, ces ensembles sont dits canoniques.

Exercice 2 : Approximation

On considère le problème de couverture par sommets :

> **VERTEX COVER**
> **Entrée** :
> — $G = (V, E)$ un graphe non-orienté avec $E = V \times V$
> — $k \in \mathbb{N}$
>
> **Question** : Existe-t-il un sous-ensemble $S \subseteq V$ de taille k tel que chaque arête $(u, v) \in E$ est connectée à au moins un sommet de S (i.e. $u \in S \vee v \in S$) ?

Le problème d'optimisation correspondant (minimisant la taille de S) a des applications pratiques telles que l'installation dans le métro d'un système de vidéosurveillance couvrant toutes les allées avec un budget minimum.

Le problème de couverture par sommets est NP-difficile.

On considère l'algorithme glouton qui répète ces étapes tant qu'il reste des arêtes non couvertes : on considère les arrêtes non couvertes par la solution courante, on en prend une au hasard et on ajoute ses **2 extrémités** à la solution courante.

Plus formellement :

Fonction `glouton_arêtes`(*graphe* $G = (V, E)$)

- $S \leftarrow \emptyset$;
- non_couvertes $\leftarrow E$;
- **while** *non_couvertes* $\neq \emptyset$ **do**
  - $e = \{u, v\} \leftarrow$ une arête quelconque dans non_couvertes;
  - $S \leftarrow S \cup \{u, v\}$;
  - v_couvre $\leftarrow \{e : e \in E \wedge e = \{v, w\}\}$;
  - u_couvre $\leftarrow \{e : e \in E \wedge e = \{u, w\}\}$;
  - non_couvertes $\leftarrow$ non_couvertes $-$ v_couvre $-$ u_couvre
- **end**
- **return** S

Question 1

Montrez que cet algorithme possède un ratio d'approximation de 2.

> Élements de correction :
>
> Un ratio d'approximation de 2 signifie que le rapport entre la solution donnée par l'algorithme et la solution optimale ne peut jamais être supérieur à 2.
>
> Soient S une solution produite par l'algorithme glouton et S^* une solution optimale du *Minimum Vertex Cover*.
>
> Soit A l'ensemble des arêtes sélectionnées par l'algorithme glouton.
>
> $|S| = 2|A|$
>
> $\bigwedge_{a,b \in A} a \cap b = \emptyset$
>
> En conséquence $|A| \leq |S^*|$
>
> Après avoir pris tout en compte $|A| \leq |S^*| \leq |S| = 2|A| \leq 2|S^*|$
>
> Et finalement $1 \leq \frac{|S|}{|S^*|} \leq 2$
>
> L'algorithme glouton est bien une **2-approximation**.