

Mini-projet d'algorithmique avancée

Kévin VYTHELINGUM, Jean-Michel NOKAYA

30 décembre 2013

Table des matières

1	Introduction	2
2	Préliminaires	2
3	Méthodes des essais successifs	2
3.1	Analyse	2
3.2	Condition d'élagage	3
3.3	Algorithme	3
3.3.1	Sans élagage	3
3.3.2	Avec élagage	3
3.4	Complexité temporelle	3
3.5	Expérimentations	3
4	Programmation dynamique	3
4.1	Formule de récurrence	3
4.2	Structure tabulaire	3
4.3	Algorithme	3
4.4	Complexité temporelle	3
4.5	Complexité spatiale	3
5	Question complémentaires	3
6	Conclusion	4

1 Introduction

2 Préliminaires

1.

$$\begin{aligned} d_{max} &\Rightarrow c_{min} \\ \sum_{i=1}^k d_{max} &\Rightarrow \sum_{i=1}^n c_{min} \\ D &\Rightarrow C_{opt} \end{aligned}$$

2. complexité = k^n (On peut faire un arbre pour le démontrer)

3 Méthodes des essais successifs

3.1 Analyse

Solution : un candidat est un vecteur de taille n où chaque coefficient est une durée choisie parmi l'ensemble $\{1, \dots, k\}$ (à chaque tâche on associe une durée). On choisit d'enregistrer les choix réalisés dans un tableau T de taille n .

S_i : l'ensemble des durées possibles de 1 à k

$$\text{satisfaisant}(x_i) = \sum_{j=1}^i x_j \leq D$$

(la somme partielle des durées choisies est inférieure à la durée maximale autorisée)

$$\text{enregistrer}(x_i) = T[i] \leftarrow x_i$$

$$\text{soltrouvee} : i = n$$

$$\text{defaire}(x_i) = T[i] \leftarrow 0$$

Pour simplifier les vérifications au niveau de satisfaisant et des conditions d'élagage, on utilisera les variables entières *cout* et *duree* initialisée à 0, qui représenteront le cout courant et la durée courante due à nos choix de durées. Elles seront mises à jour dans *enregistrer* et dans *defaire*. En effet, en notant CD le tableau à deux dimensions ayant les coûts en ligne et les durées en colonne, on effectuera dans *enregistrer* :

$$\begin{aligned}\text{coût} &\leftarrow \text{coût} + CD[i][x_i] \\ \text{durée} &\leftarrow \text{durée} + x_i\end{aligned}$$

Ensuite on effectura dans *défaire* :

$$\begin{aligned}\text{coût} &\leftarrow \text{coût} - CD[i][x_i] \\ \text{durée} &\leftarrow \text{durée} - x_i\end{aligned}$$

3.2 Condition d'élagage

3.3 Algorithme

3.3.1 Sans élagage

Procédure *ordonnancement_simple*(*enti*);

var ent k, x_i , D;

début

$S_i = [1..k]$;

pour x_i de 1 à k faire si $\sum_1^i x_l \leq D$ alors $T[i] < -x_i$; $cout < -cout + cd[i][x_i]$; $duree < -duree +$

Appel : *ordonnancement_simple*(1);

3.3.2 Avec élagage

3.4 Complexité temporelle

3.5 Expérimentations

4 Programmation dynamique

4.1 Formule de récurrence

4.2 Structure tabulaire

4.3 Algorithme

4.4 Complexité temporelle

4.5 Complexité spatiale

5 Question complémentaires

- 1.
- 2.

3.

4.

5.

6 Conclusion