

TD3 - Algorithmes d'approximation

OPTIMISATION ET RECHERCHE OPERATIONNELLE

M1 Info - semestre d'automne 2021-2022

UNIVERSITÉ CLAUDE BERNARD LYON 1

Christophe Crespelle

christophe.crespelle@univ-lyon1.fr

Florian Ingels

florian.ingels@ens-lyon.fr

Exercice 1.

Au suivant!

a. Appliquer l'algo *List Scheduling* vu en cours pour l'équilibrage de charge sur les listes suivantes :

— $L_1 = (5, 9, 4, 12, 4, 7, 5)$ et 3 machines

— $L_2 = (4, 4, 5, 5, 7, 9, 12)$ et 3 machines

— $L_3 = (3, 5, 2, 4)$ et 2 machines

— $L_4 = (1, 7, 8, 7, 2, 8)$ et 3 machines

b. Pour chacune des solutions obtenues, dites si elle est optimale ou non et prouvez le. Donnez le ratio d'approximation des solutions non optimales.

On note Π_1 le probleme donne par la liste de duree des taches $T = \{5, 9, 4, 12, 4, 7, 5\}$ et 3 machines ; on note Π_2 le probleme donne par $T = \{3, 5, 2, 4\}$ et 2 machines.

c. Pour chacun des deux problemes Π_1 et Π_2 donnez deux listes qui produisent respectivement le meilleur et le pire resultat pour l'algo *List Scheduling*. Donnez le ratio d'approximation de chacune de ces solutions.

d. Prouvez que vos reponses a la question precedente sont bien les meilleurs et les pires resultats pour l'algo *List Scheduling*.

e. Pour chacun des deux problemes Π_1 et Π_2 quelle est la solution obtenue en appliquant l'algo *List Scheduling* sur la liste trieée par ordre decroissant de duree des taches ? Cette solution est elle optimale ?

f. Donnez un exemple de probleme sur deux machines tel que l'algo *List Scheduling* qui recoit en entree la liste trieée par ordre decroissant de duree des taches ne fournit pas la solution optimale au probleme.

Exercice 2.

Le but de cet exercice est de montrer qu'il existe toujours une liste qui, fournie en entree a l'algo *List Scheduling*, resulte en une affectation des taches realisant le makespan

minimum. On note \mathcal{T} l'ensemble des n tâches et \mathcal{M} l'ensemble des m machines d'une instance du problème d'équilibrage de charges. Soit A une affectation des tâches de \mathcal{T} aux machines de \mathcal{M} , c'est à dire une application de \mathcal{T} dans \mathcal{M} , qui à chaque tâche associe une et une seule machine. Pour une machine $M \in \mathcal{M}$, on note $\mathcal{T}_A(M)$ l'ensemble des tâches affectées à M dans A .

De plus, on définit le profil d'une affectation A , note $p(A)$, comme la séquence décroissante des charges des m machines dans A . Un profil p est donc une séquence de m réels que l'on note $p = (p_1, p_2, \dots, p_m)$ avec $p_1 \geq p_2 \geq \dots \geq p_m$. L'ordre lexicographique définit un ordre total sur les profils. C'est à dire que pour deux profils $p = (p_1, p_2, \dots, p_m)$ et $p' = (p'_1, p'_2, \dots, p'_m)$, avec $p \neq p'$, en notant i le plus petit indice dans $\llbracket 1, m \rrbracket$ tel que $p_i \neq p'_i$, on a $p < p'$ ssi $p_i < p'_i$.

a. Soit A une affectation dont le profil est minimum pour l'ordre lexicographique. Montrez que A réalise le makespan minimum.

b. Soit A une affectation dont le profil est minimum pour l'ordre lexicographique. Soit $S \subsetneq \mathcal{T}$ un sous-ensemble strict des tâches et soit $A[S]$ l'affectation obtenue en restreignant A aux tâches de S . Montrez que l'ensemble $R_{min} \subseteq \mathcal{M}$ des machines dont la charge est minimum dans $A[S]$ contient au moins une machine M telle que $\mathcal{T}_{A[S]}(M) \neq \mathcal{T}_A(M)$.

c. Montrez que pour toute instance du problème d'équilibrage de charges, il existe toujours une liste qui, fournie en entrée à l'algorithme *List Scheduling*, donne la solution optimale au problème sur cette instance.

Indication. Considérez une affectation A dont le profil est minimum pour l'ordre lexicographique et construisez, tâche par tâche, une liste L telle que lorsque L est donnée en entrée à l'algorithme *List Scheduling*, il existe une exécution de l'algorithme qui donne exactement l'affectation A .

d. Si les n tâches ont des durées deux à deux distinctes, combien y a-t-il de listes distinctes sur les durées des tâches ? L'algorithme qui consiste à essayer toutes les listes possibles en entrée de l'algorithme de liste est-il un algorithme efficace ? Comparez sa complexité à celle de l'algorithme *brute force* vu en cours.

Exercice 3.

k-center selection

Dans ce problème, on donne un ensemble S de n sites dans le plan euclidien (mais en toute généralité, n'importe quelle distance symétrique et qui vérifie l'inégalité triangulaire permettrait d'obtenir les mêmes résultats) et on veut sélectionner au plus k centres (pas nécessairement sur les sites données en entrée) ou placer des casernes de pompiers pour protéger au mieux les n sites. Le critère que l'on veut minimiser pour l'ensemble des k centres $C = \{c_1, c_2, \dots, c_k\}$ sélectionnés est le rayon de couverture $r(C) = \max_{s \in S} \{d(s, C)\}$, ou $d(s, C) = \min_{c \in C} \{d(s, c)\}$. Le but de l'exercice est de concevoir un algorithme d'approximation du rayon de couverture minimum avec k centres, qui ait un ratio d'approximation de 2 sur le rayon de couverture et qui fournisse un ensemble de k centres réalisant cet objectif.

Supposons pour commencer qu'on connaisse le rayon de couverture minimum r . On propose alors l'algorithme suivant.

Algorithme 1 : Algorithme de 2-approximation en connaissant le rayon de couverture minimum r .

```
1  $S' \leftarrow S$ ;  
2  $C \leftarrow \emptyset$ ;  
3 tant que  $S' \neq \emptyset$  faire  
4   | Selectionner un  $s \in S'$  quelconque;  
5   |  $C \leftarrow C \cup \{s\}$ ;  
6   | Retirer de  $S'$  tous les sites qui sont a distance au plus  $2r$  de  $s$ ;  
7 si  $|C| \leq k$  alors  
8   | retourner  $C$ ;  
9 sinon  
10  | retourner "le rayon de couverture minimum est  $> r$ ";
```

a. Que retourne l'algorithme? Qu'est-ce qui est approxime ici? Que retourne l'algorithme si le rayon r qui lui est fourni n'est pas le rayon de couverture minimum (contrairement a ce que nous supposons ici)?

b. Quel est la complexite de cet algorithme?

Le but des questions suivantes est de montrer que l'algorithme 1 est correct et realise un rapport d'approximation de 2 sur le rayon de couverture qui lui est fourni en entree. C'est a dire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r , alors l'algorithme 1 termine avec $|S| \leq k$ et S realise un rayon de couverture d'au plus $2r$. Soit C^* une solution de rayon de couverture r . Soit C la solution retournee par l'algorithme.

c. Montrez que pour tout $c \in C$, il existe $c^* \in C^*$ tel que $d(c, c^*) \leq r$.

d. Soit $c^* \in C^*$ tel qu'il existe $c \in C$ tel que $d(c, c^*) \leq r$. Montrez que quel que soit $c' \in C \setminus \{c\}$, $d(c', c^*) > r$.

e. Montrez par l'absurde que si l'algorithme termine avec $|C| > k$ alors il n'existe pas d'ensemble d'au plus k centres qui ait un rayon de couverture r .

f. En deduire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r , alors l'algorithme trouve un ensemble d'au plus k centres qui a un rayon de couverture au plus $2r$.

On veut maintenant faire un algorithme de 2-approximation sur le rayon de couverture sans connaitre le rayon de couverture optimal. Pour cela, on remarque qu'a la ligne 4 du precedent algo, on peut choisir n'importe quel site s qui ne soit pas encore couvert par les centres selectionnes auparavant avec un rayon de $2r$.

g. Lorsqu'il existe un site non couvert par les centres selectionnes auparavant dans l'algorithme avec un rayon $2r$, comment peut-on choisir un site s pour etre sur qu'il soit non couvert sans meme connaitre r ?

h. En utilisant le resultat de la question precedente, ecrivez un algorithme pour le probleme k -center selection qui retourne toujours un ensemble C d'exactly k centres (on supposera $n > k$ sinon le rayon de couverture minimum est 0) et qui garantisse un facteur d'approximation de 2 sur le rayon de couverture minimum, sans supposer cette fois aucune connaissance sur le rayon de couverture minimum.

- i.** Quel est la complexite de l'algorithme que vous avez propose a la question precedente ?
- j.** Prouvez que le rayon de couverture \tilde{r} des k -centres retournes par votre algorithme est une 2-approximation du rayon de couverture minimum.