Feuille exercices 4 : Ordonnancement

eric.duchene@univ-lyon1.fr, aline.parreau@univ-lyon1.fr

Exercice 1 : **Ordonnancement de processus**

On souhaite exécuter des programmes sur un ordinateur à architecture parallèle. Voici les caractéristiques du problème :

- chaque programme a une durée donnée d_i
- chaque programme peut être interrompu à tout moment, stocké en mémoire et relancé sur un autre processeur, éventuellement le même ;
- chaque programme peut démarrer à tout moment ;
- on ne peut pas exécuter deux morceaux d'un même programme en même temps.

Question 1 – Si on a quatre processeurs à notre disposition, calculer à la main la date minimale de fin d'exécution pour les deux ensembles de tâches suivants.

Tâche	Durée
A	12
B	7
C	4
D	4
E	6

Tâche	Durée
A	6
B	7
C	4
D	7
E	5
F	3
G	4

Question 2 – En déduire une formule générale pour calculer par avance la date de fin d'éxécution en fonction des durées et du nombre de processeurs ainsi qu'un algorithme qui donne un ordonnancement optimal.

Exercice 2 : **Minimiser les retards**

La fin d'année approchant, un étudiant a pris un peu de retard dans son planning de projets à rendre. Voici ses devoirs à rendre avec leur date limite (nombre de jours restants pour rendre le travail) et leur durée estimée (en jours).

Devoir	Durée	Date dûe
A	8	35
B	6	20
C	6	11
D	1	8
E	6	6
F	8	25
G	7	28
H	3	9

Aucun retard n'est toléré : l'étudiant a automatiquement 0. Il cherche donc à minimiser le nombre de retards.

Question 1 – L'étudiant choisit de parer au plus pressé : il fait toujours le devoir dû le plus tôt qu'il a encore le temps de faire. Appliquer cette méthode sur les données précédentes.

Son ami plutôt organisé lui fait remarquer que cette méthode n'est pas optimale avec l'exemple suivant :

Devoir	Durée	Dâte dûe
A	10	10
B	2	11
C	2	11

Question 2 – Donner l'ordonnancement prévu avec la première méthode ainsi qu'un ordonnancement optimal.

Le même ami lui propose plutôt la méthode suivante [1] :
- Classer tous les devoirs par ordre croissant de date butoir.
- Considérer le premier devoir de la liste à traiter.
- S'il peut être réalisé à la suite de ce qui a été prévu, l'ajouter à ce qui est prévu.
- Sinon, supprimer le plus grand devoir parmi ceux déjà prévu et celui-ci et décaler les devoirs prévus.
- Recommencer tant qu'il y a des devoirs à traiter.

Question 3 – Appliquer cet algorithme à la liste des données du début. Expliquer brièvement pourquoi cet algorithme est optimal.

Exercice 3 : **Gestion d'une file d'attente**

Une entreprise souhaite optimiser le passage de ses n clients à son guichet. Elle connait à l'avance, pour chaque client son heure d'arrivée t_i, et la durée de traitement de ce client d_i (temps à passer au guichet, qui n'inclut pas l'attente). Il y a un unique guichet, qui ne peut prendre qu'un seul client à la fois et qui ne change pas de client en cours de traitement. Le but est de minimiser l'attente totale des clients.

Question 1 – On note f_i l'heure où le client i a fini d'être traité. Expliquer pourquoi minimiser l'attente totale des clients revient à minimiser la somme des f_i.

Dans la suite de l'exercice, on cherche donc à minimiser la somme des fins de traitement. Ce problème est en fait NP-complet mais nous allons maintenant voir un algorithme qui est une 2-approximation.

La première étape pour cet algorithme est de calculer une solution dans le cas où l'on peut interrompre un client pour en prendre un autre. Dans ce cas, l'algorithme suivant, noté $\mathcal{A}_P$ est optimal : à la date t, traiter le client avec qui on terminera le plus vite, parmi ceux présents (quitte à interrompre un client en court).

Question 2 – Appliquer $\mathcal{A}_P$ avec les données suivantes :

Client	Durée	Date arrivée
A	10	0
B	4	2
C	1	4
D	1	6
E	2	8

Quelle est la somme finale des temps de fin ?

On va adapter l'algorithme $\mathcal{A}_P$ à notre problème initial avec l'algorithme $\mathcal{A}_{approx}$ suivant :
- Appliquer $\mathcal{A}_P$ sur les données, en supposant que l'on peut couper les tâches.
- Ordonner les clients suivant l'ordre de fin de traitement obtenu.
- Traiter les clients dans cet ordre, en respectant les dates d'arrivées.

Question 3 – Appliquer $\mathcal{A}_{approx}$ sur les données précédentes et donner la somme finale des temps de fin.

On va montrer que $\mathcal{A}_{approx}$ est une 2-approximation. Pour simplifier les notations, nous supposons, quitte à renommer les clients, que l'ordre obtenu avec $\mathcal{A}_P$ lorsque l'on peut interrompre les tâches est l'ordre naturel : 1,2,3 ... Soit f_i^P la date de fin du traitement du client i obtenue avec $\mathcal{A}_P$

1. Autrement connue sous le nom d'algorithme de Hodgson et Moore - il est optimal !

lorsque l'on peut interrompre les tâches et f_i la date de fin obtenue avec $\mathcal{A}_{approx}$. Soit F_{opt} le résultat optimal que l'on peut atteindre.

Question 4 – Montrer les inégalités suivantes :

1. $f_i \leq f_i^P + \sum_{k=1}^{i} d_k$
2. $\sum_{k=1}^{i} d_k \leq f_i^P$
3. $\sum_{i=1}^{n} f_i^p \leq F_{opt}$

En déduire que $\mathcal{A}_{approx}$ est une 2-approximation.

Question 5 – Donner un exemple où cet algorithme est loin d'être optimal.

Question 6 – (Pour ceux qui ont fini le reste du TD) Que donne l'algorithme de liste avec les clients ordonnés par durée croissante ? Est-ce un algorithme d'approximation avec un facteur constant ?

Exercice 4 : **Anomalie de Graham**

On considère un problème d'ordonnancement sur plusieurs machines avec règle de précédence données dans le tableau suivant. On cherche à minimiser le temps final (les jobs ne peuvent pas être découpés).

Job	Durée	Précédences
A	3	
B	2	
C	2	
D	2	
E	4	D
F	4	D
G	4	D
H	4	D
I	9	A

Question 1 – Nous disposons tout d'abord de trois machines. Donner l'ordonnancement obtenu avec un algorithme de liste sur la liste $\{A, B, C, ...\}$ (c'est-à dire : dès qu'une machine est libre, lui donner le premier job possible). Est-ce optimal ?

Question 2 – Même question sur 4 machines. Qu'observez-vous ?

Question 3 – Finalement, tous les jobs prennent une unité de temps en moins. Recalculer l'ordonnancement obtenu sur 3 machines. Qu'observez-vous ?

Exercice 5 : **Ordonnancement de tâches avec précédence**

Un architecte a décomposé le chantier qu'il doit faire pour son prochain projet suivant les postes ci-dessous. Pour chaque poste, sa durée est indiquée (en jours) ainsi que les postes qui doivent être finis avant de le commencer.

Numero	Poste	Durée	Postes antérieurs
1	Démolition	2	
2	Fenêtres	3	1
3	Construction escalier	3	
4	Pose escalier	1	1,3
5	Isolation	5	1,2
6	Electricite	4	5
7	Plomberie	5	1,2,5
8	Peinture	2	6,7
9	Sols	3	1,4

Question 1 – Dessiner le graphe potentiel-tâches correspondant.

Question 2 – Déterminer les dates au plus tôt et au plus tard de chaque poste ainsi qu'un chemin critique. Combien de temps faut-il prévoir pour le projet ?