

Exercice 3 (Sous-Graphe sans Circuit Maximum - 7 points)

Un graphe orienté $G = (V, A)$ est un graphe dans lequel chaque arc de A possède une orientation (représentée par une flèche). Un arc (i, j) dans A est donc orienté *de i vers j* . Un *circuit* dans un graphe orienté est un ensemble d'arcs qui forme un cycle lorsqu'on suit l'orientation des flèches. Par exemple, dans le graphe orienté fourni ci-dessous, un circuit existe entre les sommets 1, 7 et 6.

Le problème de maximisation SOUS-GRAPHE SANS CIRCUIT MAXIMUM (ou MAX-SGC) se définit comme suit:

SOUS-GRAPHE SANS CIRCUIT MAXIMUM (MAX-SGC)

Instance : Un graphe orienté $G = (V, A)$

Solution : Un sous-graphe $G' = (V, A')$ de G , $A' \subseteq A$, tel que G' ne contient pas de circuit

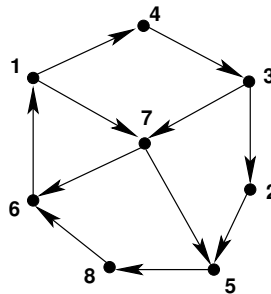
Mesure : $|A'|$, le nombre d'arcs dans A'

Dans la suite de cet exercice, on supposera que le graphe orienté G que l'on étudie possède toujours n sommets et m arcs.

1. Le problème MAX-SGC est un problème d'optimisation. Écrire, sous la forme NOM/Instance/Question le problème DEC-SGC, qui est le problème de décision associé à MAX-SGC.
2. Démontrer que DEC-SGC est dans NP.

En réalité, MAX-SGC est un problème NP-complet. Dans le reste de cet exercice, on se propose d'étudier un algorithme d'approximation pour MAX-SGC. Cet algorithme s'appelle Approx-SGC, et il est décrit ci-dessous en français:

- (a) on attribue au hasard à chaque sommet de G un unique entier $1 \leq p \leq n$
- (b) on sépare l'ensemble des arcs A du graphe orienté G en A_1 et A_2 de la manière suivante:
 - A_1 est l'ensemble des arcs (i, j) tels que $i < j$
 - A_2 est l'ensemble des arcs de A qui ne sont pas dans A_1 (donc c'est l'ensemble des arcs (i, j) tels que $i > j$)
- (c) L'ensemble A' renvoyé sera l'ensemble (A_1 ou A_2) qui contient le plus d'arcs



3. Appliquer l'algorithme Approx-SGC sur le graphe orienté ci-dessus, en utilisant la numérotation des sommets qui y est indiquée. En particulier, indiquer les contenus de A_1 , A_2 et A' .
4. Montrer que, quel que soit le graphe orienté G donné en entrée, les arcs de l'ensemble A' calculé par Approx-SGC ne forment jamais de circuit.
5. Approx-SGC est-il polynomial ? Justifier.
6. Proposer un exemple de graphe orienté G à $n = 5$ sommets pour lequel Approx-SGC ne fournit pas une solution optimale au problème MAX-SGC. Justifier.

Pour tout graphe orienté G , on appelle $opt(G)$ le nombre d'arcs d'une solution optimale au problème MAX-SGC pour G , et $sol(G)$ le nombre d'arcs d'une solution fournie par l'algorithme Approx-SGC pour G .

7. Pour tout graphe orienté G , donner une borne supérieure pour $opt(G)$, qui dépend de m (le nombre d'arcs du graphe). Justifier.
8. Pour tout graphe orienté G , donner une borne inférieure pour $sol(G)$, qui dépend aussi de m . Justifier.
9. En déduire un ratio d'approximation pour l'algorithme Approx-SGC.
10. Quel ratio d'approximation est obtenu par Approx-SGC sur le graphe orienté de la figure ? Justifier.

Distanciel

Algorithmes Approchés pour deux Problèmes d'Optimisation: MAX-SCG et MIN MAKESPAN

Automne 2021

Ce “projet distanciel” se compose de deux parties : un exercice (de type TD), et un projet de programmation. Ce projet est à réaliser en **binômes** (un seul monôme autorisé par groupe de TD, si celui-ci possède un nombre impair d'étudiants). Ne pas hésiter à utiliser le forum Madoc “Échanges et questions autour du projet Distanciel” pour échanger entre vous, et aussi poser vos questions à l'enseignant.

Un compte-rendu est à rédiger, qui contiendra donc deux parties. La Partie 1 contient vos réponses aux questions de l'exercice ; la Partie 2 contient un rapport de type “rapport de projet de programmation”. La Partie 2 ne doit pas dépasser 10 pages. Le format de fichier attendu pour ce compte-rendu (Parties 1 et 2) est le PDF.

L'ensemble des éléments de ce “projet distanciel” est à rendre sous la forme d'une archive NOM1-NOM2.zip (ou NOM.zip si le projet a été effectué seul.e). La décompression de l'archive doit produire un répertoire NOM1-NOM2 (ou NOM) contenant tous les éléments de votre travail (réponses aux questions de l'exercice, rapport de projet, sources du projet, exécutable du projet, etc.) et *un fichier texte contenant les instructions détaillées de compilation et d'exécution du projet.*

Cette archive est à **déposer sur Madoc**, au plus tard le **Vendredi 3 décembre 2021 à 20h21**. Un malus sera appliqué pour chaque consigne donnée ci-dessus qui n'aura pas été respectée. Notamment, en cas de retard, le malus sera le suivant : -1 point par heure de retard (toute heure entamée étant comptée comme complète).

1 Le Problème MAX-SCG – Exercice

Faire l'Exercice 3 (“Sous-Graphe sans Circuit Maximum”) de l'Examen 2020-2021, dont vous trouverez le sujet sur Madoc.

2 Le Problème MIN MAKESPAN – Projet de Programmation

Dans ce projet, la programmation se fera dans le langage de votre choix. *Assurez-vous que vos programmes compilent et fonctionnent sous Linux sur les machines du CIE.*

Le but de ce projet est d'étudier le problème d'optimisation MIN MAKESPAN, dans lequel les durées des tâches sont toujours des valeurs entières. La définition du problème MIN-MAKESPAN est donnée ci-dessous (voir aussi l'Exercice 3.2 de la feuille TD3, disponible sur Madoc, ainsi que les transparents disponibles dans la partie “Distanciel” de Madoc) :

MIN-MAKESPAN

Instance : n tâches de durées $d_1, d_2 \dots d_n$; m machines identiques $M_1, M_2 \dots M_m$, chaque machine ne pouvant réaliser qu'une tâche à la fois

Solution : Une affectation des n tâches aux m machines

Mesure : le temps de terminaison T de l'ensemble des tâches (aussi appelé *makespan*)

On s'intéressera en particulier à un type d'instance, que l'on appellera I_p , p étant un paramètre entier. Dans une instance I_p , le nombre de machines m est égal à $2p$, et on a $n = 2p^2 + 2p + 1$ tâches. Les durées des tâches sont, dans cet ordre, les suivantes :

- $4p$ tâches de durée 1, suivies de
- $2p(p - 1)$ tâches de durée 2, suivies de
- 1 tâche de durée $2p$

Il a été démontré que le problème MIN-MAKESPAN est NP-complet. Dans ce projet on vous demande d'implémenter et de tester plusieurs algorithmes répondant, de manière approchée, à ce problème.

Les trois algorithmes à implémenter sont :

1. l'algorithme List Scheduling Algorithm (LSA), qui prend les tâches dans l'ordre initialement fourni, et affecte chaque tâche à la première machine disponible¹ ;
2. l'algorithme Largest Processing Time (LPT) du, qui trie d'abord les tâches par ordre décroissant. Ensuite, LPT affecte chaque tâche, dans l'ordre du tri, à la première machine disponible² ;
3. l'algorithme Random Machine Assignment (RMA), qui, pour chaque tâche dans l'ordre initialement fourni, détermine au hasard quelle machine va l'exécuter.

Remarque : pour ce projet, on ne s'intéresse qu'au *temps total de réalisation* calculé par les algorithmes. La réalisation précise (c'est-à-dire, quelles tâches affecter à quelle machine et dans quel ordre) n'est pas demandée.

Ainsi, pour implémenter ces algorithmes, on peut juste manipuler deux tableaux d'entiers :

- un tableau $D[]$ de taille n (n =nombre de tâches) qui va contenir les durées des tâches à affecter aux machines ;
- un tableau $M[]$ de taille m (m =nombre de machines), dont toutes les valeurs sont initialisées à 0. Chaque case i de M contiendra, à tout moment de l'algorithme considéré, la *durée cumulée* des tâches affectées à la machine i .

Par exemple, si à un moment dans l'algorithme la machine 2 doit exécuter une tâche de longueur 7, alors on l'écrira simplement de la manière suivante : $M[2] \leftarrow M[2] + 7$.

2.1 Travail demandé

Proposer un programme facile d'utilisation qui, par l'intermédiaire d'un menu, permet à l'utilisateur (1) de renseigner l'instance de son choix et (2) de lire à l'écran les résultats de cette instance sur les trois algorithmes évoqués ci-dessus.

Instances d'entrée. L'utilisateur doit pouvoir choisir entre deux modes de saisie des instances d'entrée :

1. **(Génération d'une instance de type I_p)** L'utilisateur entre un entier p au clavier, et l'instance I_p , décrite auparavant, est générée.
2. **(Génération aléatoire de plusieurs instances)** L'utilisateur fournit 5 entiers m, n, k, d_{min} et d_{max} . Il faut alors générer k instances différentes : pour chacune de ces k instances, on a m machines, n tâches, et les durées des tâches sont générées de manière *aléatoire*, à la condition que la durée de chaque tâche soit comprise entre d_{min} et d_{max} .

On appellera ces deux modes de création d'instance I_p et I_R (pour *Random*).

Production des résultats. Pour chaque mode de création d'instances (I_p et I_R), les trois algorithmes LSA, LPT et RMA seront exécutés. Selon le mode choisi, les résultats seront fournis de façons différentes.

1. **(Mode I_p)** Les résultats seront affichés à l'écran de la façon suivante.

```
Borne inférieure ``maximum`` =
Borne inférieure ``moyenne`` =
```

```
Résultat LSA =
ratio LSA =
```

-
1. voir les transparents disponibles dans la partie "Distanciel" de Madoc pour un exemple
 2. voir les transparents disponibles dans la partie "Distanciel" de Madoc pour un exemple

Résultat LPT =
ratio LPT =

Résultat RMA =
ratio RMA =

Pour davantage d'informations sur les notions de Borne inférieure ``maximum`` et de Borne inférieure ``moyenne``, voir les transparents disponibles dans la partie "Distanciel" de Madoc (et également les Questions 2. et 3. de l'exercice de TD 3.2).

La valeur Résultat LSA est le temps T_{LSA} calculé sur l'instance considérée.

La valeur ratio LSA se calcule de la façon suivante :

- (a) on prend la valeur la plus grande entre Borne inférieure ``maximum`` et Borne inférieure ``moyenne`` : appelons cette valeur B ;
- (b) ratio LSA est le résultat de la division de Résultat LSA par B (donc ce n'est pas forcément un entier).

La valeur Résultat LPT est le temps T_{LPT} calculé sur l'instance considérée.

La valeur ratio LPT se calcule de façon similaire à ratio LSA, à ceci près que dans le (b), c'est Résultat LPT (au lieu de Résultat LSA) que l'on divise par B .

La valeur Résultat RMA est le temps T_{RMA} calculé sur l'instance considérée.

Enfin, la valeur ratio RMA se calcule de façon similaire à ratio LSA, à ceci près que dans le (b), c'est Résultat RMA (au lieu de Résultat LSA) que l'on divise par B .

- 2. (Mode I_R) Dans ce mode, on n'affichera à l'écran que les *valeurs moyennes* des ratios. Plus précisément, seules les trois valeurs suivantes seront affichées :

ratio moyen LSA =
ratio moyen LPT =
ratio moyen RMA =

La valeur de ratio moyen LSA est la moyenne des ratios LSA (tels que définis ci-dessus) sur les k exemples générés. La valeur de ratio moyen LPT est la moyenne des ratios LPT (tels que définis ci-dessus) sur les k exemples générés. La valeur de ratio moyen RMA est la moyenne des ratios RMA (tels que définis ci-dessus) sur les k exemples générés.

2.2 Questions

- 1. Discuter de la complexité en temps et au pire de la génération des instances (mode I_p et I_R);
- 2. Discuter de la complexité en temps et au pire de chacun des trois algorithmes implémentés;
- 3. Réaliser une campagne de tests pour l'algorithme LSA avec les instances de type I_p , en testant au moins les valeurs listées dans le tableau ci-dessous (il n'est donc pas interdit d'en tester plus).³.

p	ratio LSA	p	ratio LSA	p	ratio LSA
1		15		70	
2		20		80	
3		25		90	
4		30		100	
5		35		120	
6		40		140	
7		45		160	
8		50		180	
9		55		200	
10		60		300	

- 3. Si le temps d'exécution est "trop long" ou qu'il y a dépassement de mémoire, indiquer "PR" (Pas de Réponse)

4. Que remarquez-vous en ce qui concerne le ratio LSA sur les instances de type I_p quand p grandit ? Pouvez-vous fournir une explication à cette observation ?
5. Réaliser une campagne de tests pour l'algorithme LPT avec les instances de type I_p , en reprenant les valeurs testées pour LSA.
6. Que remarquez-vous en ce qui concerne le ratio LPT sur les instances de type I_p quand p grandit ? Pouvez-vous fournir une explication à cette observation ?
7. Réaliser une campagne de tests pour l'algorithme RMA avec les instances de type I_p , en reprenant les valeurs testées pour LSA. Discuter et expliquer les résultats obtenus.
8. Exécuter le mode I_R sur au moins une trentaine d'exemples différents, en faisant varier les paramètres d'entrée, et remplir le tableau suivant. Discuter et expliquer les résultats obtenus.

n	m	k	d_{min}	d_{max}	ratio moyen LSA	ratio moyen LPT	ratio moyen RMA
...	