Le problème PARTITION consiste, étant donné un ensemble d'entiers $A = \{a_1, a_2, \ldots, a_k\}$, à dire s'il est possible de partitionner cet ensemble en deux sous-ensembles de somme égale (i.e. trouver $I \subset \{1, \ldots, k\}$ tel que $\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i$). $\boxed{\text{Sujet 1}}$

Question 1. Que signifie l'affirmation suivante : le problème PARTITION est $\mathcal{NP}$-complet. On admettra ce résultat dans la suite.

Un problème d'ordonnancement.

On considère des tâches $T_1, T_2, \ldots, T_n$ à réaliser sur p processeurs identiques séquentiels. Chaque tâche T_i a une durée d'exécution d_i. Le problème d'optimisation p-AFFECTE consiste à affecter les tâches aux processeurs de façon que la durée d'utilisation de ceux-ci soit minimale (plus précisément la durée d'utilisation du processeur le plus utilisé).

Question 2. On dispose de 3 processeurs et on doit exécuter la suite de tâches de durées 4,3,2,7,2,1,6,3,5,6,3. Donner une affectation optimale.

Étant donnés les taches $T_1, \ldots, T_n$, et un entier D, le problème de décision p-AFFECTE consiste à dire s'il est possible de trouver une affectation de durée d'utilisation inférieure à D.

Question 3. Montrer que le problème de décision p-AFFECTE est $\mathcal{NP}$-complet.

Un algo d'approximation.

On considère l'algorithme de type "on-line glouton" qui consiste à considérer toutes les tâches dans un ordre quelconque et à affecter chaque tâche T_i au processeur le moins chargé au moment où l'on considère celle-ci. Ainsi sur l'exemple précédent, en utilisant l'ordre donné ci-dessus, on obtient les affectations suivantes : $\{4, 1, 6, 3\}, \{3, 2, 3, 5\}, \{2, 7, 6\}$.

Question 4. On note u_i la somme des durées des tâches affectées par l'algorithme glouton au processeur i, m un processeur tel que u_m est maximal et j la dernière tâche affectée au processeur m. Donner une relation entre u_i et $u_m - d_j$.

Question 5. Expliquer et justifier l'assertion suivante : l'algorithme glouton précédent est un algorithme approché à un facteur $2 - 1/p$ pour le problème d'optimisation p-AFFECTE. Motiver la terminologie "on-line glouton".

Question 6. Donner un exemple ayant $p(p-1)+1$ tâches pour p processeurs, pour lequel la solution optimale est de durée maximale p, et tel que l'algorithme glouton donne une durée de $2p - 1$ pour l'un des processeurs.

Question 7. L'algorithme qui effectue la même itération que l'algorithme glouton mais qui commence par trier les tâches dans l'ordre de durée décroissant est-il meilleur ?

Complexité pour une variante du problème

Chaque tâche a maintenant en plus une date limite d'exécution et on considère l'algorithme d'ordonnancement sur 1 processeur suivant : on construit une liste de tâches exécutables dans les temps impartis en partant de la liste vide et en essayant d'ajouter les tâches dans l'ordre croissant des dates limites ; si l'ajout d'une tâche à l'ensemble courant produit un dépassement de sa date limite, on retire de l'ensemble la tâche de durée maximale.

Question 8. Discuter de la complexité de cet algorithme. En particulier peut on l'implanter en temps $O(n \log n)$?

Arbre couvrant minimal.

SUJET 2

Un graphe non-orienté G est la donnée d'un couple (V, E) d'un ensemble fini V (les sommets) et d'un ensemble fini E de paires $\{v_0, v_1\}$ de sommets distincts (les arêtes). Un chemin d'extrémités v_0 et v_n est une suite alternée $v_0 e_1 v_1 e_2 \ldots v_{n-1} e_n v_n$ de sommets et d'arêtes telles que $e_i = \{v_{i-1}, v_i\}$ pour tout i, et les arêtes e_i sont deux à deux distinctes. Un cycle est un chemin tel que $v_0 = v_n$. Un graphe est connexe s'il existe un chemin entre toute paire de sommets. Un arbre couvrant de G est un ensemble A d'arêtes tel que le graphe (V, A) soit connexe et sans cycle.

Question 9. Montrer que les conditions suivantes sont équivalentes à dire que A est un arbre :
- A contient au plus $n-1$ arêtes et (V, A) est connexe,
- A contient au moins $n-1$ arêtes et (V, A) est sans cycle.

Question 10. Soient T et T' deux arbres couvrants de G, et e une arête de T qui n'est pas dans T'. Montrer qu'il existe une arête $e' \neq e$ sur le cycle créé par e avec T' telle que $T \setminus \{e\} \cup \{e'\}$ soit aussi un arbre couvrant de G.

À chaque arête $e = (x, y)$ de G on associe un poids $p(e)$. Le poids $p(A)$ d'un ensemble d'arête A est la somme des poids de ses éléments : $p(A) = \sum_{e \in A} p(e)$.

Question 11. Montrer que l'algorithme glouton suivant produit l'arbre A^* de poids minimum parmi tous les arbres couvrants de G.
- Classer les m arêtes de G par poids croissants : $p(e_1) \leq p(e_2) \leq p(e_3) \leq \ldots \leq p(e_m)$,
- Initialiser $A^* = \emptyset$ puis, pour $i = 1$ à m, ajouter l'arête e_i à A^* si celle-ci ne crée pas de cycle avec les arêtes déjà dans A^*.

Question 12. Quelle est la complexité de l'algorithme glouton ? En quoi est il "glouton" ?

Voyageur de commerce avec inégalité triangulaire.

On suppose maintenant que $V = \{1, \ldots, n\}$ et que $G = (V, E)$ est le graphe complet (comprenant les $n(n-1)/2$ arêtes possibles). Un voyageur de commerce désire partir du sommet 1 et faire un tour passant une fois et une seule par chacun des autres sommets du graphe avant de revenir en 1. Le problème du voyageur de commerce est de déterminer un tour optimal, c'est-à-dire un cycle H^* passant par tous les sommets et de poids $p(H^*)$ minimal.

Question 13. Le problème de décision associé au problème du voyageur de commerce est de répondre à la question : existe-t-il un tour de longueur inférieure à ℓ ?

Ce problème est dit *NP-complet*. Qu'est ce que ceci signifie ?

On se place dans le cas où les poids des arêtes satisfont l'inégalité triangulaire : pour tous sommets u, v, w, $d(u, w) \leq d(u, v) + d(v, w)$. On considère l'algorithme suivant, qui fabrique un tour non nécessairement optimal :
- calculer un arbre couvrant minimal T^*.
- renvoyer le tour H obtenu en regardant l'ordre dans lequel les sommets apparaissent lors d'un parcours en profondeur de l'arbre T^* à partir du sommet 1.

Question 14. Montrer que le poids $p(T^*)$ de T^* est inférieur ou égal à la longueur $p(H^*)$ d'un tour optimal H^*.

Question 15. En utilisant l'inégalité triangulaire, montrer que la longueur $p(H)$ de H est inférieure ou égale à $2p(T^*)$.

Question 16. Justifier l'affirmation : l'algorithme précédent est un algorithme d'approximation à un facteur 2 pour le problème du voyageur de commerce.