OPTIMISATION ET RECHERCHE OPÉRATIONELLE *Janvier 2019*

Examen Final

Numéro copie :

- Cet examen comporte deux exercices indépendants. Le deuxième exercice est lui-même constitué de trois parties indépendantes. Vous pouvez utiliser des questions précédentes même si vous n'y avez pas répondu.
- Durée de l'examen : 1h30
- Document autorisé : une feuille de notes A4 recto-verso
- Toutes les réponses doivent être justifiées.
- Le barème (sur 22 points) est donné à titre indicatif.
- Le sujet est à rendre avec votre copie, n'oubliez pas d'y indiquer le numéro de la copie.

Exercice 1 : (7pts) Vous allez diriger une colonie de vacances et vous devez organiser le transport de tous les enfants vers le lieu de la colonie qui est situé tout près de la ville de Poitiers. Les enfants partiront des villes d'Orléans (20 enfants), Rennes (30 enfants) et Nantes (15 enfants). Vous choisissez de faire voyager les enfants en train, malheureusement il est déjà tard et les seules places restantes sont indiquées dans le tableau suivant. Ce tableau donne pour chaque trajet qui est possible en train le nombre de places sur ce trajet (il y a par exemple 15 places sur le train Orléans → Le Mans mais pas de trajet Nantes → Tours possible).

Départ \ Arrivée	Le Mans	Nantes	Angers	Tours	Poitiers
Orléans	15	-	-	8	-
Rennes	15	10	10	-	-
Le Mans	-	-	10	5	-
Nantes	-	-	10	-	15
Angers	-	-	-	15	28
Tours	-	-	-	-	25

Par chance, les correspondances se font bien et il n'y a pas de contraintes à ce niveau là. Vous vous demandez s'il est possible d'acheminer tout le monde à Poitiers.

Question 1. (1pt) Modéliser ce problème sous forme d'un problème de flot (sans le résoudre !). Vous dessinerez le graphe associé.

Question 2. (3pts) Est-il possible d'acheminer tous les enfants jusqu'à Poitiers ? Si cela n'est pas possible, combien d'enfants au maximum peut-on acheminer ? Justifier en expliquant en détail votre méthode.

Constatant cela, vous décidez de louer un (et un seul) minibus pour faire un bout du trajet. Les possibilités raisonnables en terme de prix sont sur les trajets Le Mans → Nantes, Rennes → Tours ou Nantes → Tours.

Question 3. (2pts) Pour chacune de ces possibilités, expliquez si elle permet d'acheminer tout le monde ou pas, et si elle le permet, donnez un plan de transport possible.

Vous pouvez maintenant amener tout le monde à bon port. Ceci dit, vous remarquez qu'il y a plusieurs solutions possibles, et vous voudriez prendre la solution la moins chère. Vous connaissez le prix par personne sur chaque trajet.

Question 4. (1pt) Modéliser ce problème sous forme d'un programme linéaire. On pourra utiliser les notations suivantes pour les données du problème :

- c_{ij} le nombre de places sur le trajet $i \to j$,
- p_{ij} le prix par personne sur le trajet $i \to j$.

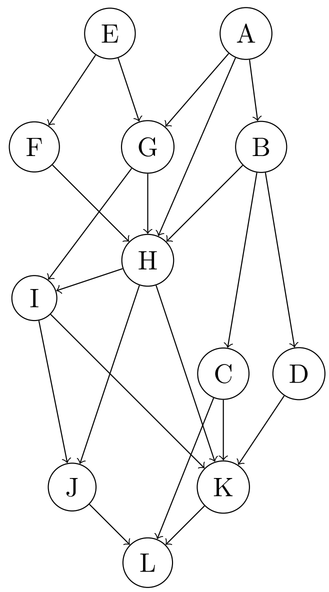
Si vous introduisez d'autres notations, veillez à bien les définir. **On ne vous demande pas de résoudre ce problème.**

Exercice 2 :

Valentin a décidé d'inviter sa famille à diner ce soir et son sens aigü du détail le pousse à optimiser l'intendance de cette réception. Dans un premier temps (Partie 1), il va se poser des questions sur la préparation d'une sauce à steak légendaire qui nécessite une bonne gestion de ses plaques électriques. Dans un second temps (Parties 2 et 3), c'est l'optimisation de la vaisselle qui le préoccupera. Les trois parties sont indépendantes.

Partie 1 : Cuisine (5,5pts)

Pour son repas, Valentin a opté pour des steaks vegans accompagnés d'une sauce au poivre très difficile à préparer. Pour celle-ci, il doit faire chauffer 12 ingrédients, notés de A à L, dans un certain ordre. L'ordre de chauffage des ingrédients est donné par le graphe de précédences suivant :



Par exemple, ce graphe indique que l'ingrédient A doit avoir été chauffé avant les ingrédients B, G et H.

Dans sa recette, la durée de chauffage est estimée identique pour chaque ingrédient (qu'on supposera égale à 1 unité de temps). Enfin, pour préparer cette sauce, il n'a à sa disposition que 2 plaques de cuisson, et il cherche à minimiser le temps de chauffe total pour la préparer.

Valentin s'aperçoit que son problème revient à un problème d'ordonnancement sur 2 machines avec des tâches unitaires et des précédences entre les tâches. Pour le résoudre, il décide d'appliquer un algorithme de liste, c'est-à-dire qu'étant donnée une liste ordonnée des ingrédients, il chauffe les ingrédients en choisissant à chaque fois le premier ingrédient de la liste qui est disponible en fonction des contraintes de précédence.

Question 1. (1.5pt) Valentin a choisi d'appliquer un tel algorithme avec la liste

$$L = (A, B, C, D, E, F, G, H, I, J, K, L)$$

Calculez l'ordonnancement obtenu par cet algorithme pour cette liste, en donnant l'ordre de chauffage pour chaque plaque, ainsi que la durée totale obtenue.

Il s'avère qu'un algorithme de liste optimal pour ce problème existe en calculant une liste L_{opt} de la façon suivante : on va attribuer un numéro (de 1 à 12) à chaque ingrédient correspondant à son ordre dans la liste. Pour un ingrédient s, on notera $num(s)$ son numéro.

1. Numéroter tous les ingrédients qui n'ont pas de successeur dans le graphe de précédence avec les valeurs $1, 2 \ldots$.
2. Pour chaque ingrédient dont **tous** les successeurs $s_1, \ldots, s_k$ ont déjà un numéro, lui affecter le tuple $(num(s_1), \ldots, num(s_k))$ trié par ordre décroissant. On continue ensuite la numérotation des ingrédients en parcourant les tuples ainsi calculés dans l'ordre lexicographique.

3. On itère le (2) tant que tous les ingrédients n'ont pas été numérotés.
4. La liste L_{opt} est calculée en prenant les ingrédients dans l'ordre décroissant de leurs numéros, soit $12, 11, \ldots, 2, 1$.

Pour que cela soit bien clair, on vous donne le début de la numérotation :

— L'ingrédient L est numéroté 1 car c'est le seul sommet sans successeur dans le graphe.

— Les ingrédients J et K ont désormais tous leurs successeurs numérotés (ils n'ont qu'un seul successeur, à savoir L). On leur attribue à chacun le même tuple $(num(L)) = (1)$. J et K sont donc numérotés respectivement avec les numéros 2 et 3 (ici, on aurait pu indifféremment inverser les numéros vu que J et K possèdent le même tuple).

— Les ingrédients I, C et D ont désormais tous leurs successeurs numérotés. On leur associe respectivement les tuples $(3, 2)$, $(3, 1)$ et (3). En parcourant ces tuples dans l'ordre lexicographique, on obtient $num(D) = 4$, $num(C) = 5$ et $num(I) = 6$.

Question 2. (1.5pt) Donnez le numéro de chaque ingrédient à la fin de cet algorithme et en déduire la liste L_{opt}. Vous pouvez calculer le numéro sur le graphe directement.

Question 3. (1pt) Appliquez l'algorithme de liste pour L_{opt} et écrivez l'ordonnancement obtenu sur chaque plaque de cuisson. Quelle est sa durée totale ?

Question 4. (1.5pt) Montrer que sur cet exemple précis, l'algorithme de liste pour L_{opt} est bien optimal.

Partie 2 : Plonge (6,5pts)

Après avoir réussi d'excellents steaks au poivre vegans, Valentin doit à présent laver tous ses ustensiles de cuisine. Hélas il n'y a pas assez de place dans l'égouttoir (de capacité $W = 15$) pour pouvoir y mettre toute la vaisselle. Valentin va donc devoir sécher certaines choses à la main. En bon informaticien, il décide de chercher à optimiser cette tâche, il crée donc le tableau ci-dessous qui récapitule les informations, à savoir, pour chaque type d'ustensile $U_1 \ldots U_5$, sa quantité q_i, la place w_i qu'il prend dans l'égouttoir, le temps ℓ_i (en sec) mis pour le laver, et le temps s_i (en sec) de séchage à la main.

ustensile	quantité q_i	temps lavage ℓ_i	place dans l'égouttoir w_i	temps séchage à la main s_i
U1	4	10	1	10
U2	1	20	2	20
U3	3	20	4	24
U4	2	30	5	40
U5	1	40	5	100

Valentin va laver toute la vaisselle, sécher certaines choses à la main et en laisser d'autres à sécher dans l'égouttoir. Son objectif est de **minimiser son temps de vaisselle**, c'est-à-dire lavage + séchage à la main. Il a déjà calculé les informations suivantes à partir du tableau ci dessus :

— Le temps total de lavage pour tous les ustensiles est de 220 secondes.

— S'il devait sécher à la main tous les ustensiles, il y passerait 312 secondes.

On précise que pour un même type d'ustensile présent plusieurs fois, il peut choisir d'en sécher certains à la main et laisser les autres dans l'égouttoir.

Question 5. (2pts) Modélisez ce problème par un programme linéaire en nombres entiers. Précisez bien quelles à quoi correspondent vos variables ainsi que leur domaine de définition. 0.5 pour les variables - on mettra 0 si c'est pas clair que c'est dans N, 0.5 fonction obj, 0.5 par contrainte.

De nature maligne, Valentin s'aperçoit que son problème revient à mettre dans l'égouttoir l'ensemble d'ustensiles qui prendra le plus de temps à sécher à la main. Il cherche donc à maximiser le temps de séchage à la main des ustensiles qu'il va mettre dans l'égouttoir -tout en respectant la capacité de celui-ci.

On va chercher à résoudre ce problème à l'aide de la programmation dynamique. On introduit la fonction $f(i, w)$ qui correspond au temps maximum de séchage à la main pour les ustensiles de type $U_1, \ldots, U_i$, que l'on peut mettre dans un égouttoir de taille w. Valentin cherche donc à calculer la valeur $f(5, 15)$. Il a commencé à remplir les valeurs de $f(i, w)$ pour $i \leq 3$ et $w \leq 15$, elles se trouvent dans le tableau suivant. Par exemple, $f(1, 15)$ vaut $4 \times 10 = 40$ car il y a 4 ustensiles du type U_1.

$i\backslash w$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	10	20	30	40	40	40	40	40	40	40	40	40	40	40	40
2	0	10	20	30	40	50	60	60	60	60	60	60	60	60	60	60
3	0	10	20	30	40	50	60	60	64	74	84	84	88	98	108	108
4																
5																

Question 6. (1.5pt) Ecrivez la relation de récurrence satisfaite par $f(4,w)$ et $f(5,w)$. On fera attention au fait que l' ustensile U_4 est présent en deux exemplaires.

Question 7. (2pts) En utilisant la relation de récurrence de la question précédente, complétez les deux dernières lignes du tableau.

Question 8. (1pt) En déduire quelle est la stratégie optimale pour Valentin, c'est-à-dire combien d'ustensiles de chaque type il doit mettre dans l'égouttoir pour minimiser son temps de vaisselle total. Expliquez.

Partie 3 : Lave-vaisselle (6pts)

Après quelques mois passés à faire la vaisselle, Valentin s'est décidé à acheter un lave vaisselle. Mais celui-ci a une capacité limitée que l'on notera W. Face à toute sa vaisselle accumulée, il se demande combien de lave-vaisselles il va devoir faire tourner.

Question 9. (0,5pt) De quel problème classique de recherche opérationnelle s'agit-il ?

Pour résoudre ce problème, Valentin étant quelque peu fatigué décide de faire au plus simple :il prend les ustensiles dans l'ordre où ils se présentent. Tant que l'ustensile rentre dans le lave-vaisselle, il le met. S'il ne rentre pas, il lance le lave-vaisselle et le mettera dans le prochain lave-vaisselle et ainsi de suite.

Question 10. (1pt) Donner le nombre de lave-vaisselles et la composition de ceux-ci obtenus avec l'algorithme sur les données suivantes : un volume estimé à $W = 20$ et les ustensiles qui arrivent dans l'ordre suivant (chaque ustensile est présent une seule fois) :

Ustensile	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9
Place	3	4	10	5	7	9	2	10	3

On veut montrer que cet algorithme est une 2-approximation. Pour cela, considérons une instance quelconque du problème : on dispose de n ustensiles, et la place que prend l'ustensile i est notée w_i. On note N_{opt} le nombe optimal de lave-vaisselles pour cette instance et N_{algo} le nombre donné par l'algorithme précédent.

Question 11. (1pt) Donner un minorant de N_{opt} en fonction des w_i et de W.

Soit L_k, $k \in \{1, ..., N_{algo}\}$, l'ensemble des ustensiles qui sont dans le lave-vaisselle k.

Question 12. (0,5pt) On considère deux ensembles consécutifs L_k et L_{k+1}. Expliquer pourquoi

$$\sum_{i \in L_k \cup L_{k+1}} w_i > W$$

Question 13. (1 pt) Pour simplifier la preuve, on va supposer que N_{algo} est pair (le raisonnement est similaire dans le cas impair). Montrer que l'on a $\sum_{i=1}^{n} w_i > \frac{N_{algo}}{2} W$.

Question 14. (0,5pt) Conclure que l'on a bien une 2-approximation.

Question 15. (1,5pt) Donner une instance très mauvaise pour cet algorithme. On pourra par exemple trouver un ensemble de $4m$ ustensiles qui pourraient se ranger dans $m + 1$ lave-vaisselles en théorie mais qui donne $2m$ lave-vaisselles pour un mauvais ordre.