Problème Sac à Dos.

Le problème du Sac à Dos est un problème classique en informatique. Il modélise une situation analogue au remplissage d'un sac. Une personne veut remplir un sac à dos ne pouvant pas supporter plus d'un certain poids $C \in \mathbb{N}$, et elle dispose de n objets (On note l'ensemble des objects par $\mathcal{O} = \{1, \ldots, n\}$). Chaque objet i a une valeur $v_i \in \mathbb{N}\backslash\{0\}$ et un poids $p_i \in \mathbb{N}\backslash\{0\}$. Le problème est de trouver un ensemble d'objets tels que

- tous les objets de cet ensemble puissent être mis dans le sac.
- la somme des valeurs de ces objets soit maximale.

Exercice 1 Complexité du problème Sac à Dos

Montrer que le problème Sac à Dos est faiblement NP-complet sachant que le problème SubsetSum est faiblement NP-complet.

Exercice 2 Programmation dynamique

Nous allons construire un tableau T dans lequel les lignes seront indexées par les objets et les colonnes par les valeurs. L'élément $T[i, j]$ représentera la valeur maximale pour un sac à dos de capacité j à l'aide des i premiers objets.

Question 2.1 Donner la formule de récurrence.

Correction

Soit OPT une solution optimale avec un sac à dos de capacité C et un ensemble $\mathcal{O}$ de n éléments. Notons $T[i, j]$ représentera la valeur maximale pour un sac à dos de capacité j à l'aide des i premiers objets.

1. Si le $n$ième élément appartient à la solution optimale, cela signifie, que la solution optimale privée du $n$ième élément est aussi une solution optimale avec un sac à dos de capacité $C - p_n$ et un ensemble $\mathcal{O}$ privé du $n$ième élément.
$$T[n, C] = v_n + T[n-1, C - p_n]$$
2. Si le $n$ième élément n'appartient pas à la solution optimale, cela signifie, que la solution optimale OPT est aussi une solution optimale avec un sac à dos de capacité C et un ensemble $\mathcal{O}$ privé du $n$ième élément.
$$T[n, C] = T[n-1, C]$$

Donc par récurence on peut en déduire pour tout $i \in \{1, \ldots, n\}$

$$\text{Si } j < v_i, \text{ alors } T[i, j] = 0 \text{ sinon } T[i, j] = max(T[i-1, j], T[i-1, j-p_i] + v_i)$$

On supposera que $\forall j \in \{1, \ldots, C\}$, on a $T[0, j] = 0$

□

Question 2.2 Donner l'algorithme utilisant la programmation dynamique.

Correction

Entrée : un ensemble d'objects $\mathcal{O} = \{1, \ldots, n\}$. L'objet i a une valeur v_i et un poids p_i.

Sortie : un entier

1. Intialiser tous les éléments du tableau T à zéro.
2. Pour tout i allant de 1 à n

(a) Pour tout j allant de 1 à C

2 (a).1 Si $j < v_i$, alors $T[i,j] = 0$
sinon $T[i,j] = max(T[i-1,j], T[i-1, j-p_i] + v_i)$

3. Retourner $T[n, C]$

□

Question 2.3 Donner la complexité de cet algorithme.

Correction

La complexité de cet algorithme est $\mathcal{O}(nC)$ car

1. l'initialisation du tableau se fait en $\mathcal{O}(nC)$ opérations ;
2. L'instruction 2 (a).1 coûte $\mathcal{O}(1)$ opérations. Et elle est exécutée nC fois.

□

Le problème du k-centre

Dans cette partie, nous allons nous concentrer sur le problème du k-centre : étant donné un ensemble de villes dont les distances sont spécifiées, choisir k villes afin d'installer des entrepôts de façon à minimiser la distance maximale d'une ville à l'entrepôt le plus proche. Un tel ensemble de k villes est appelé k-centre.

Le problème associé de décision est le suivant k-CENTRE
Données : un graphe complet $K = (V, E)$ muni d'une fonction de poids w sur les arêtes, et des entiers strictement positifs k et b.
Question : Existe-il un ensemble S de sommets tel que $|S| = k$ et tel que tout sommet v de V satisfait la condition suivante

$$min\{w(v,u) : u \in S\} \leq b$$

Rappellons les définitions suivantes :
- une couverture de sommets S du graphe G est un sous-ensemble de sommets tel que toutes les arêtes de G sont incidentes à au moins un sommet de S.
- un ensemble dominant C du graphe G est un sous-ensemble de sommets tel que tout sommet est soit dans C soit voisin d'un sommet de C.

Approximation et in-approximation

Exercice 3 Approximation du problème

Nous allons nous focaliser sur le problème de k-CENTRE MÉTRIQUE. Ici, la fonction de poids respecte l'inégalité triangulaire.

Question 3.1 Le *carré* d'un graphe G, noté G^2, est défini comme le graphe qui contient une arête (u, v) pour chaque couple de sommets u et v reliés par un chemin de longueur au plus 2 dans G.

Montrer que si le sous-ensemble de sommets I est un stable de G^2 et alors pour tout ensemble dominant D de G, on a $|I| \leq |D|$.

Correction

Soit D un dominant **minimum** de G. Par conséquent, G se décompose en $|D|$ étoiles

– **couvrant tous les sommets** de G
– dont les sommets de D sont les centre

Dans le graphe G^2, une étoile de G est une clique (puisqu'ils sont tous à distance au plus 2 dans G). Ainsi, G^2 se décompose en au plus $|D|$ cliques **couvrant tous de sommets** G^2.

Par conséquent, tout stable I de G^2 ne pourra choisir qu"un unique sommet dans chacune de ces D cliques.

□

Question 3.2 Dans la suite de l'exercice, nous supposerons que les arêtes de K sont triés par poids croissant, c'est-à-dire $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$ et nous notons $K_i = (V, E_i)$, où $E_i = \{e_1, e_2, \ldots, e_i\}$.

Montrer que trouver un k-centre est équivalent à trouver le plus petit index i tel que G_i admet un dominant de taille au plus k.

Correction

Si G_i possède un k-centre C, alors cet ensemble C est un ensemble de sommets de cardinalité k tel que chaque sommet de G_i (et donc de G), est soit voisin de C ou soit dans C. Et réciproquement.

□

Question 3.3 Voici l'algorithme $\mathcal{A}$ du k-centre :

1. Construire $(G_1)^2, (G_2)^2, \ldots, (G_m)^2$.
2. Calculer un stable maximal, M_i, pour chaque $(G_i)^2$.
3. Calculer le plus petit indice i tel que $|M_i| \leq k$. Notons-le j.
4. Renvoyer M_j.

Donner la complexité de cet algorithme.

Notons OPT le coût d'un k-centre optimal. Notons aussi par i^* le plus petit indice, $w(e_i*) = OPT$ et tel que G_i* contient un k-centre optimal.

– Montrer que si j est l'indice calculé par l'algorithme, alors $w(e_j) \leq OPT$.

Correction

Lors de l'exécution de l'algorithme (voir particulièrement instruction 3), pour tout entier i allant de 1 à $j-1$, on a $|M_i| > k$. Cela signifie aussi que d'après la question précédente (3.1) que l'ensemble dominant minimun de G_i est $> k$, et donc que $i^* > j$.

De plus, comme $i^* > j$, pour les définitions des G_i, on a $w(e_j) \leq w(e_{i^*})$. □

– Montrer que l'algorithme $\mathcal{A}$ est une 2-approximation.

Correction

Soit $\mathcal{A}(I)$ la valeur de la solution retournée par $\mathcal{A}$. Il faut montrer que $\mathcal{A}(I) \leq 2 \cdot OPT$.

Tout d'abord remarquons qu'un stable maximal I d'un graphe est aussi un dominant. En effet, si un sommet v n'était pas dominé par I, l'ensemble $I \cup \{v\}$ serait un stable, contredisant ainsi la propriété que i soit un stable maximal de I.

Il existe donc dans G_j^2, Il existe dans des étoiles centrées sur les sommets de M_j. Ces étoiles couvrent tous les sommets. Or, l'inégalité triangulaire donne que toutes les arêtes de G_j utilisées par ces étoiles ont un poids au plus $2 \cdot w(e_j)$.

D'après la question précédente, comme $w(e_j) \leq OPT$, on a $\mathcal{A}(I) \leq 2 \cdot OPT$.

□

Question 3.4 Considérons le graphe complet de $n+1$ sommets possédant un sommet singulier u. Chaque arête incidente à u est de poids 1 et les autres sont de poids 2.

Appliquer l'algorithme sur ce graphe en considérant la pire solution. Qu'en déduisez-vous ?

Correction

En prenant $k = 1$,

- la solution optimale $= \{u\}$: $OPT = 1$.
- en appliquant l'algorithme, on a $j = n$. Remarquons que G_n^2 est une clique, il suffit de prendre un stable correspondant à un sommet qui n'est pas u. Donc $\mathcal{A}(I) = 2$

C'est une instance critique. En effet, cet exemple montre que l'analyse d'un algorithme d'approximation est au plus près. □

Exercice 4 In-approximation du problème

Question 4.1 Montrer que si $P \neq NP$, quel que soit $\epsilon > 0$, il n'existe pas de $(2-\epsilon)$-approximation pour le problème du k-CENTRE MÉTRIQUE.

Correction

Il faut réduire le problème du dominant minimum à celui de l'approximation du problème du k-CENTRE MÉTRIQUE. Soit I une instance du problème du dominant minimum : $G = (V, E)$ et un entier k.

Nous construisons l'instance du problème du k-CENTRE MÉTRIQUE : le graphe complet $K = (V, E')$ muni de la fonction de poids sur les arêtes suivantes :

$$w(u,v) = \begin{cases} 1 & si \quad (u,v) \in E \\ 2 & si \quad (u,v) \notin E \end{cases}$$

Remarquons que la fonction de poids vérifie l'inégalité triangulaire.

Cette réduction se réalise en temps polynomiale et elle vérifie les propriétés suivantes :

- $dom(G) \leq k$ alors K admet un k-centre de coût 1 ;
- $dom(G) > k$ alors K admet un k-centre de coût 2 ;

Si $dom(G) \leq k$, la solution optimale est de coût 1. Une $(2-\epsilon)$approximation pour le k-centre donne nécessairement une solution de coût $1 * (2-\epsilon)$. Puisqu'elle ne peut pas utiliser d'arête de coût 2, elle retourne une solution de coût 1.

Utiliser un tel algorithme permettrait de distinguer entre ces deux cas, et de résoudre le problème du dominant minimum en temps polynomial. Et cela supposerait que $P = NP$. □

Question 4.2 Montrer que quel que soit $\alpha(n)$ calculable en temps polynomial, il n'existe pas de $\alpha(n)$-approximation pour k-CENTRE, à moins que $P = NP$.

Correction

Par l'absurde, soit A une $\alpha(n)$-approximation (en temps polynomial) pour k-CENTRE. Nous allons démontrer que A permet de résoudre le problème NP-difficile du dominant minimum en temps polynomial, et donc que $P = NP$. Soit I une instance du problème du dominant minimum : $G = (V, E)$ et un entier k.

Nous construisons l'instance du problème du k-CENTRE : le graphe complet $K = (V, E')$ muni de la fonction de poids sur les arêtes suivantes :

$$w(u,v) = \left\{ \begin{array}{ccc} 1 & si & (u,v) \in E \\ \alpha(n)+1 & si & (u,v) \notin E \end{array} \right.$$

Cette réduction se réalise en temps polynomiale et elle vérifie les propriétés suivantes :

- $dom(G) \leq k$ alors K admet un k-centre de coût 1 ;
- $dom(G) > k$ alors K admet un k-centre de coût $\alpha(n)+1$;

Si $dom(G) \leq k$, la solution optimale est de coût 1. Une $\alpha(n)$-approximation pour le k-centre donne nécessairement une solution de coût $1 * \alpha(n)$ qui ne peut pas utiliser d'arête de coût $\alpha(n)+1$, elle retourne une solution de coût 1.

Utiliser un tel algorithme permettrait de distinguer entre ces deux cas, et de résoudre le problème du dominant minimum en temps polynomial. Et cela supposerait que $P = NP$.

□

Problème du voyageur de commerce métrique

Nous allons étudier le problème suivant : étant donné un graphe complet muni d'une fonction de coût positive sur les arêtes, respectant l'inégalité triangulaire, trouver un cycle de coût minimum passant par chaque sommet exactement une fois.

Exercice 5 Algorithme polynomial ayant un facteur d'approximation 2.

Considérons le problème suivant :

1. Calculer un arbre couvrant de poids minimal T de G.
2. Dupliquer toutes les arêtes de T pour obtenir un graphe eulérien.
3. Calculer un cycle eulérien $\mathcal{X}$ de ce graphe.
4. Renvoyer le cycle C qui passe par tous les sommets de G dans l'ordre de leur première occurrence dans $\mathcal{X}$.

Soit OPT la valeur de la solution optimale.

Question 5.1 Montrer que $cout(T) \leq OPT$.

Correction

Soit S une solution optimale.

Soit T' un arbre couvrant de G en otant une arête d'une solution optimale du TSP. Par définition, on a $cout(T') \leq OPT$ puisque les coût sont positifs ou nuls.

Comme T est un arbre couvrant de poids minimun, on a

$$cout(T) \leq cout(T') \leq OPT$$

□

Question 5.2 Montrer que l'algorithme approche la solution optimale à un facteur 2.

Correction

D'après la question précédente, nous avons $cout(T) \leq OPT$.

Puisque le cycle eulerien $\mathcal{X}$ contient chaque arête de T en double, $cout(\mathcal{X}) = 2cout(T)$ et donc $cout(\mathcal{X}) \leq 2OPT$.

L'instruction 4 modifie le cycle eulerien pour former un cycle correspont à une tournée du voyageur de commerce. Pour chaque arête e du cycle C, 2 cas sont possibles

1. e est dans T

2. e n'est pas dans T. Elle a été obtenue par la suppréssion d'un chemin p connectant les deux extrémités de e dans $\mathcal{X}$. Par l'inégalité triangulaire on a $cout(e) \leq cout_{e' \in P} cout(e')$

Donc on a $cout(C) \leq cout(\mathcal{X})$.

Comme $cout(\mathcal{X}) \leq 2OPT$, on peut déduire que $cout(C) \leq 2OPT$.

□

Question 5.3 Donner une instance critique.

Correction

Fixons un entier n. Considérons l'instance suivante : un graphe complet à n sommets dont les arêtes sont de coût 1 ou 2. Pour un n quelconque, le graphe a $2n-2$ arêtes de coût 1 :

- $n-1$ arêtes forment une étoile centrée en u
- $n-1$ arêtes forment un cycle de longueur $n-1$ ne passant pas par u ;

les autres arêtes sont de coût 2.

- la solution optimale = le cycle hamiltonien utilisant que les arêtes de coût 1. Donc $OPT = n$.
- en appliquant l'algorithme, on a
  - l'arbre couvrant de poids minimal trouvé par l'algorithme est l'étoile centrée en u (utilisant que les arêtes de coût 1.
  - Ensuite à partir de cet arbre, de cet arbre transformé en graphe eulerien, il faut construire un cycle eulerien de telle façon que pour la prochaine étape, le cycle de voyageur de commerce utilise uniquement que 2 arêtes de coût 1.
  - Donc la valeur de la solution obtenue est $2n-2$. C'est **asymptotiquement** le double de OPT

  $j = n$. Remarquons que G_n^2 est une clique, il suffit de prendre un stable correspondant à un sommet qui n'est pas u. Donc $\mathcal{A}(I) = 2$

C'est une instance critique. En effet, cet exemple montre que l'analyse d'un algorithme d'approximation est au plus près. □

Problèmes de logique

k-SAT NAE

Données : un ensemble U de variables $\{u_1, u_2, \ldots, u_n\}$ et une formule logique $L = C_1 \wedge \cdots \wedge C_\ell$ avec $C_i = (y_{i,1} \vee y_{i,2} \vee \cdots \vee y_{i,k})$ où $y_{i,j}$ est égal soit à l'un des u_k ou soit à l'un des $\overline{u_k}$

Question : Existe-t-il une fonction $t : U \to \{0,1\}$ telle que t satisfait L et telle que les littéraux de chaque clause ne sont pas toutes de la même valeur ?

Exercice 6 Montrer que 4-SAT NAE est NP-complet sachant que 3-SAT est NP-complet.
Indication : Introduire une nouvelle variable z et l'insérer dans toutes les clauses

Correction

Le problème est dans NP car étant donné une affectation des variables, on peut vérifier en temps polynomial que cette affectation satisfait ϕ. On va réduire 3-SAT à 4-SAT NAE. Soit *phi* une formule de 3-SAT sur les variables U. On ajoute une unique variable distincte z et on forme les clauses pour 4-SAT NAE en remplaçant chaque clause $C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$ de l'instance de 3-SAT par $C'_i = y_{i,1} \vee y_{i,2} \vee y_{i,3} \vee z$.

Cette transformation se fait bien en temps polynomial en la taille de l'instance 3-SAT.

Si l'instance donnée de 3-SAT est satisfiable, la même affectation des variables tout en fixant pour z la valeur 0 fournit une affectation valide pour 4-SAT NAE.

Réciproquement, supposons que l'instance construite de 4-SAT NAE soit satisfaisable.

Si la valeur de vérité de z dans l'affectation correspondante est 0, alors les valeurs des variables u_i dans l'affectation donnent une affectation valide pour la formule ϕ pour l'instance de 3-SAT . Si au contraire z vaut 1, on change toutes les valeurs de toutes les variables dans l'affectation. L'affectation reste valide pour 4-SAT NAE car au moins un littéral par clause dans l'affectation initiale valait 0, et vaut donc maintenant 1, tandis que z vaut 0. Et on se retrouve dans le cas précédent.

On a donc bien prouvé que $3 - SAT \leq 4 - SATNAE$.

□

Problèmes liés aux cycles hamiltoniennes

Exercice 7 Chaine hamiltonienne et cycle hamiltonien.
Considérez les trois problèmes suivants :

CYCLE HAMILTONIEN
Données : un graphe non-orienté G.
Question : G contient-il un cycle hamiltonien ?

CHAINE HAMILTONIENNE
Données : un graphe non-orienté G, deux sommets u et v distincts de G.
Question : G contient-il une chaine hamiltonienne entre u et v ?

CHAINE
Données : un graphe non-orienté G de n sommets, deux sommets u et v distincts de G.
Question : G contient-il une chaine de longueur $n/2$ entre u et v ?

Sachant que le problème CYCLE HAMILTONIENNE est NP-complet, montrer que les problèmes CHAINE HAMILTONIEN et CHAINE sont NP-complets.

Correction

Le problème CHAINE HAMILTONIENNE est dans NP car étant donnée une chaîne, on peut vérifier en temps polynomial si elle passe une fois et une seule par chaque sommet du graphe et qu'elle a u et v comme extrémité.

Nous allons faire la réduction à partir du problème CYCLE HAMILTONIEN. Soit $\mathcal{I} =< G = (V, E) >$ une instance du problème CYCLE HAMILTONIEN. Maintenant, nous allons transformer cette instance en une instance $\mathcal{I}'$ du problème CHAINE HAMILTONIENNE de la façon suivante : Nous allons construire un graphe $G' = (V', E')$ tel que

- Soit u un sommet arbitraire de V
- $V' := V \cup \{v\}$ tel que v est un sommet n'appartenant pas dans V
- $E' := E \cup \{(v, \ell) : \ell \text{ est un voisin de } u \text{ dans } G\}$

Cette transformation peut se faire en temps polynomial (nous avons juste copier le graphe G en rajoutant un sommet et des arêtes).

Il est facile de prouver que :

– Si il existe un cycle hamiltonien dans G, alors il existe une chaine hamiltonnienne dans G'.
Soit $C = (u, \ell_1, \ldots, \ell_{n-1}, u)$ un cycle hamiltonien dans G. Nous construisons la chaîne $\mathcal{P} = (u, \ell_1, \ldots, \ell_{n-1}, v)$ dans le graphe G'. Cette chaîne est hamiltonienne : elle passe une fois et une seule par v et par chaque sommet de G puisque C est un cycle hamiltonien.
– Si il existe une chaine hamiltonnienne dans G', alors il existe un cycle hamiltonien dans G.
Soit $\mathcal{P} = (u, \ell_1, \ldots, \ell_{n-1}, v)$ une chaîne dans un graphe G'. Le cycle $C = (u, \ell_1, \ldots, \ell_{n-1}, u)$ est hamiltonnien pour le graphe G.

Donc CYCLE HAMILTONIEN $\leq$ CHAINE HAMILTONIENNE

Donc le problème CHAINE HAMILTONIENNE est NP-complet.

□

Correction

Le problème CHAINE est dans NP car étant donnée une chaîne, on peut vérifier en temps polynomial si sa taille est de longueur $n/2$ et qu'elle a u et v comme extrémité.

Nous allons faire la réduction à partir du problème CHAINE HAMILTONIENNE. Soit $\mathcal{I} =< G = (V, E), u, v >$ une instance du problème CHAINE HAMILTONIENNE.

Nous transformons cette instance en une instance $\mathcal{I}'$ du problème CHAINE de la façon suivante. Nous construisons un graphe $G' = (V', E')$ tel que

G' est une copie du graphe G plus une chaîne de $|V|$ sommets dont un seul sommet de cette chaîne est voisin de u.

Cette transformation se fait en temps polynomial (nous avons juste copier le graphe G en rajoutant un sommet et des arêtes).

Il est facile de prouver que

il existe une chaine hamiltonnienne dans G si et seulement si il existe une chaine hamiltonnienne dans G' de longueur $\frac{|V'|}{2}$.

Donc le problème CHAINE HAMILTONIENNE se réduit au problème CHAINE ; et ce dernier est donc NP-complet.

□

Exercice 8 Chevaliers de la table ronde

Etant donnés n chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

Correction

– Le problème est dans NP car étant donné un plan de table, on peut vérifier en temps polynomial si pour chaque chevalier, il n'est pas à coté d'un ennemi.
– Nous allons faire la réduction à partir du problème du cycle hamiltonien
Soit $\mathcal{I} =< G = (V, E) >$ une instance du problème du cycle hamiltonien.
Maintenant, nous allons transformer cette instance en une instance $\mathcal{I}_2$ du problème des CHEVALIERS DE LA TABLE RONDE de la façon suivante :
  – chaque sommet du graphe est un chevalier.
  – Deux chevaliers sont des ennemis si et seulement si il n'existe pas une arête dans G impliquant les deux sommets représentés par ces deux chevaliers
Cette transformation peut se faire en temps polynomial (nous avons construit le

complémentaire du graphe G).

Il est facile de prouver que :

- Si il existe un cycle hamiltonien dans G, alors il existe un plan de table. Il suffit de voir qu'une arête dans G correspond au fait que les deux chevaliers ne sont pas ennemis. Donc le plan de table correspond au cycle hamiltonien.
- Si il existe un plan de table alors il existe un cycle hamiltonien dans G.

Donc le problème des Chevaliers de la table ronde est plus difficile que le problème du cycle hamiltonien.

Donc le problème des Chevaliers de la table ronde est NP-complet.

□