ISTY - 2ième année
CONTEXTE :
Epreuve de *Outils pour la conception d'algorithmes*
Date : 10/10/2019
Horaire : 8h45 – 9h30

Controle Continu
Durée du sujet : 45minutes
Nom du rédacteur : J. Cohen
Aucun document autorisé

**Indications :** Il faudra justifier toutes vos réponses. A chaque affirmation, il faudra faire un commentaire pertinent. Un soin tout particulier devra être apporté à la présentation de la copie.

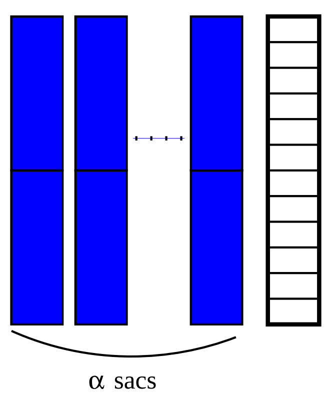
**Question 1**. Problème Bin Packing.

Nous considérons le problème Bin Packing que l'on a vu en cours. Il consiste à ranger un ensemble de $n$ objets de capacité fixé $w_1, w_2, \ldots w_n$ dans des sacs de même capacité que l'on notera $c$ en utilisant le moins possible de sacs.

Considérons l'instance suivante. Soit $\alpha$ un entier positif. Nous considérer qu'il faut ranger $4\alpha(= n)$ objects dont les capacités sont les suivantes $w_i = \begin{cases} 1 & si \quad i \ est \ paire \\ \alpha & si \quad i \ est \ impaire \end{cases}$

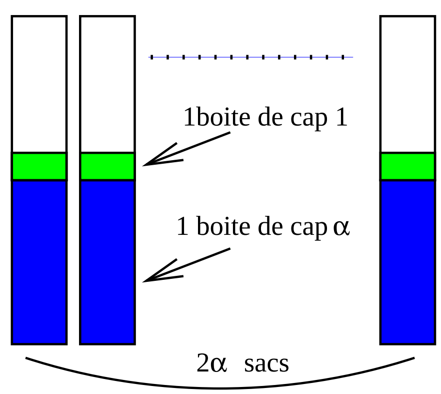
La capacité des sacs est égale à $c = 2\alpha$.

**Question 1.1** Donner la solution optimale



**Question 1.2** Donner la solution de l'algorithme NEXT FIT vue en cours :
***Entrée :*** un ensemble de $n$ objets $\mathcal{B} = \{b_1, \ldots, b_n\}$ de capacité fixé $w_1, w_2, \ldots w_n$, un entier $c$ correspondant à la capacité des sacs.
***Sortie :*** un entier $k$, un rangement $\mathcal{S}$
***Algorithme :***

1. $j = 1$ ; $\mathcal{S} \leftarrow \emptyset$
2. Pour $i$ allant de 1 à $n$ faire
   (a) Si $b_i$ peut être mis dans $S_j$, $S_j \leftarrow S_j \cup \{b_i\}$
   (b) Sinon
      i. $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_j\}$ ; $j \leftarrow j + 1$
      ii. $S_j \leftarrow \{b_i\}$
3. retourner $j$, $\{S_1, \ldots, S_j\}$



**Question 1.3** Qu'en déduisez vous ?

### Partition en cliques et Couverture de sommets

Une ***clique*** est un graphe dans lequel chaque paire de sommets est liée par une arête. Une clique $C$ est dite ***maximale*** dans le graphe $G$ (au sens de l'inclusion) s'il n'existe pas un sommet $u$ de $G$ tel que $u$ est voisin de tous les sommets de $C$ dans $G$.

**Question 2**. Donner un algorithme polynomial qui étant donnés un graphe $G$ et un sommet $u$, retourne une clique maximale contenant $u$ dans $G$. Donner la complexité de l'algorithme.

**Clique_Max(G,u) ;**
**Entrée** : un graphe $G$, un sommet $u$
**Sortie** : Un ensemble de sommets $C_u$
**Variables locales** : $b$ un booléen

1. $C_u \leftarrow \{u\}$ ; ***// O(1) opérations***
2. Pour tout sommet $v$ faire ***// la boucle est exécutée*** $O(|V|)$ ***fois***
   (a) $b \leftarrow True$ ; ***// O(1) opérations***
   (b) Pour chaque sommet $x$ de $C_u$ ***// la boucle est exécutée*** $O(|V|)$ ***fois au pire***
      si $x$ n'est pas voisin de $v$ alors $b \leftarrow False$ ; ***// O(1) opérations***
   (c) Si $b == False$ alors $C_u \leftarrow C_u \cup \{v\}$ ; ***// O(1) opérations***
3. Retourner $C_u$ ; ***// O(1) opérations***

Complexité : $O(|V|^2)$ opérations

Une ***couverture de sommets*** $S$ d'un graphe $G$ est un sous-ensemble de sommets tel que toute arête de $G$ est incidente à au moins un sommet de $S$. Dans la suite, nous allons nous concentrer sur la couverture de sommets de plus petite cardinalité (en terme du nombre de sommets).

**Question 3**. Dans une clique $C$ de $n$ sommets, donner une couverture de sommets de plus petite cardinalité.

Toute couverture de sommets $S$ de $C$ contient $n - 1$ sommets de cette clique.

Par la suite, nous noterons $S^*$ une couverture de sommets de plus petite cardinalité. Nous allons noter $OPT$ le nombre de sommets de la solution $S^*$ optimale de la couverture de sommets de $G$.

Nous appelons une ***partition en cliques maximales*** du graphe $G = (V, E)$, une partition des sommets $V$ en sous-ensembles disjoints $\mathcal{C} = \{C_1, \ldots, C_k\}$ tels que

1. $\bigcup_{i=1}^{k} C_i = V$ ;
2. pour tout couple $(i, j)$ tel que $1 \leq i < j \leq k$, nous avons $C_i \cap C_j = \emptyset$ ;
3. chaque sous-graphe induit par $C_i$ dans $G$ est une clique maximale ;

***Nous supposerons qu'il existe un algorithme polynomial qui retourne une partition en cliques maximales de $G$.***

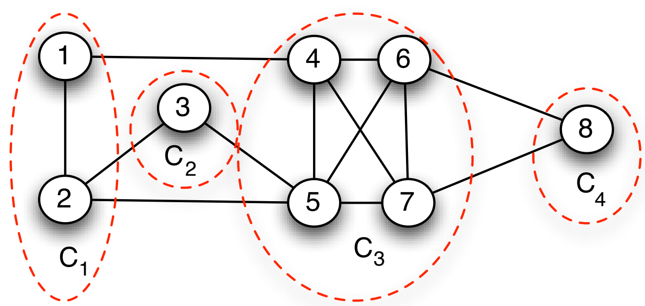



FIGURE 1 – Partition en cliques : les cliques $\{1, 2\}$ et $\{2, 3, 5\}$ sont des cliques maximales. L'ensemble $C = \{C_1, C_2, C_3, C_4\}$ est une partition en cliques maximales.

Une clique qui ne contient qu'un seul sommet est appelée une ***clique triviale***. Dans la figure 1, les cliques $C_4$ et $C_2$ sont triviales.

Soit $\mathcal{C} = \{C_1, \ldots, C_k\}$ une partition en cliques maximales du graphe $G$. Considérons $C'$ l'ensemble de sommets qui sont dans une clique non triviale de la partition $C$. Dans la figure 1, $C'$ vaut $\{1, 2, 4, 5, 6, 7\}$.

**Question 4**. Montrer que $C'$ est une couverture de sommets.

Soit $e$ une arête de $G$. Deux cas se posent :

1. Si $e$ est une arête dont au moins une de ces extrémités sont dans des cliques non-triviales, alors $e$ est couvert par un sommet de $C'$.
2. Si $e$ est une arête dont les deux de ces extrémités sont dans des cliques triviales, alors, $e$ formerait une clique non-trivial. Ceci amène à une contradiction avec la partition en cliques maximales du graphe $G$.

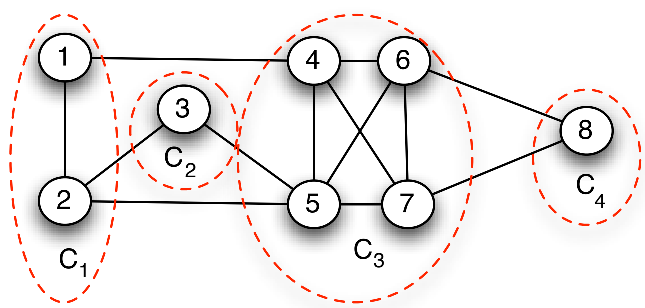



FIGURE 2 – Partition en cliques : les cliques $\{1, 2\}$ et $\{2, 3, 5\}$ sont des cliques maximales. L'ensemble $C = \{C_1, C_2, C_3, C_4\}$ est une partition en cliques maximales.

**Question 5**. Trouver un graphe tel que $|C'| = 2OPT$ ?

Il suffit de considérer le graphe ayant uniquement une arête $(u, v)$. Sa partition en cliques maximales se réduit en une clique composée de l'arête $(u, v)$. Alors $C' = \{u, v\}$ et une solution optimale contient simplement un seul noeud. Donc nous avons $|C'| = 2OPT$ dans ce cas.

**Question 6**. Soit $c$ le nombre de cliques non-triviales de $\mathcal{C}$. Montrer que $OPT \geq c$.

Comme chaque clique non-triviale possède au moins deux sommets, une solution optimale à au moins un sommet dans une clique non triviale. Pour couvrir cette arête, il faut au moins un sommet dans $S^*$. Ainsi, nous avons $OPT \geq c$.

**Question 7**. Montrer que $|C'| \leq 2OPT$ si $G$ possède moins une arête. *Indication : Utiliser les questions 3 et 6.*

Rappellons que pour chaque clique $C$ non triviale de $\mathcal{C}$, toutes couvertures de sommets contient au moins $|C| - 1$ sommets.

C'est-à-dire qu'au moins $|C| - 1$ sommets sont aussi dans une solution optimale.

Soit $c$ le nombre de cliques non-triviales.

Nous avons l'inéquation suivante ;

$$OPT \geq |C'| - c \tag{1}$$

Nous pouvons réécrire l'équation (précédente) (1) comme suit $OPT + c \geq |C'|$.

Ce qui nous permet de déduire que $2OPT \geq |C'|$.

**Question 8**. Qu'en déduisez-vous des quatre questions précédentes ?

Comme il existe un algorithme polynomial qui retourne une partition en cliques maximales de $G$, cet algorithme est un algorithme approché avec un facteur 2. On a la bonne borne.

**Question 9**. Donner un algorithme polynomial qui retourne une partition en cliques maximales de $G$. *Indication : utiliser la question 2.*

**Entrée** : un graphe $G$
**Sortie** : Un ensemble de partition en cliques $\mathcal{C}$
**Variables locales** : $Q$, $C$ deux ensembles de sommets

1. $Q \leftarrow V$ // $O(|V|)$ ***opérations au pire***
2. $\boldsymbol{C} \leftarrow \emptyset$ ; // ***O(1) opérations***
3. Tant que $Q \neq \emptyset$ faire // ***la boucle est exécutée*** $O(|V|)$ ***fois au pire***
   (a) Choisir un sommet $u$ de $Q$ ; // ***O(1) opérations***
   (b) $C \leftarrow Clique\_Max(G, u)$; // $O(|V|^2)$ ***opérations au pire***
   (c) $Q \leftarrow Q \backslash C$; // $O(|V|^2)$ ***opérations au pire***
   (d) $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$ // ***O(1) opérations***
4. Retourner $\mathcal{C}$ ; // ***O(1) opérations***

Complexité : $O(|V|^3)$ opérations