

- Question 1 : Donner le pseudo-code de chaque algorithme approché.

Algorithme 4 Procedure-MC(C, P, \max)

```

1: while  $P \neq \emptyset$  do
2:   if  $|C| + |P| < \max$  : return
3:   choisir  $v$  le dernier élément de  $P$ 
4:    $C \leftarrow C + v$ 
5:   if  $|P \cap \Gamma(v)| == 0 \ \&\& \ |C| > \max$ 
6:     Sauvegarder  $C$  comme plus la plus grande clique
7:      $\max \leftarrow |C|$ 
8:   end if
9:   if  $|P \cap \Gamma(v)| \neq 0$ 
10:    Procedure-MC( $C, P \cap \Gamma(v), \max$ )
11:   end if
12:    $C \leftarrow C - v$ 
13:    $P \leftarrow P - v$ 
14: end while

```

- Question 2 : Expliquer l'utilité des différents paramètre de l'algorithme.

Le premier algorithme, MC, utilise 2 ensembles :

- Un ensemble C de nœuds, initialement vide, qui contient la clique partielle construite à un instant t et qui contient la clique maximum à la fin de l'algorithme.
- Un ensemble P , initialisé avec tous les nœuds du graphe, qui contient les nœuds candidats pour agrandir la clique partielle.

L'algorithme utilise en plus une variable \max qui contient la taille de la clique maximum trouvée. La procédure détaillée est présentée dans Algorithme 4.

- Question 3 : Comment peut-on appliquer chaque algorithme approché pour résoudre votre problème d'optimisation?

Une première modification permettant de gagner un peu de temps est la procédure MC0, on ne prend en compte que les voisins dont l'identifiant est inférieur au nœud pivot v sélectionné, ce qui évite des doublons

présente ensuite différentes versions de l'algorithme MC : MCQ, MCS et BBMC. MCS est lui-même décliné en 2 versions MCSa et MCSb. Tous ces algorithmes ont pour point commun de colorer le graphe, en fonction du tri préalablement effectué par une procédure de coloration choisie. Ces colorations, qui permettent de grouper les nœuds, sont ensuite

utilisées par chacun des algorithmes pour résoudre le problème de clique maximale (voir[4] pour une description exhaustive de ces algorithmes)