
Branch and bound

Ce document reprend les algorithmes vus au cours. Par convention, ils sont implémentés sous la forme de fonctions avec un nom utilisant CETTE POLICE. Si vous la voyez apparaître dans un algorithme, cela signifie donc qu'on fait appel à un autre algorithme déjà vu.

Algorithme 1 : COUPLAGEMAXIMAL(G)

Entrées : un graphe $G = (V, E)$ non orienté, non pondéré.

Sortie : un couplage maximal pour G .

```
1  $M \leftarrow \emptyset$ ;  
2  $G' \leftarrow G$ ;  
3 tant que  $G.\text{nombre\_arêtes}() \neq 0$  faire  
4    $M \leftarrow M \cup$  une arête arbitraire  $\{u, v\}$  de  $G$ ;  
5    $G.\text{retirer\_sommets}(\{u, v\})$ ;  
6  $G \leftarrow G'$ ;  
7 renvoyer  $M$ ;
```

Algorithme 2 : VCEXHAUSTIF(G)

Entrées : un graphe $G = (V, E)$ non orienté, non pondéré.

Sortie : un ensemble $U \subseteq V$ de taille minimale couvrant toutes les arêtes de G .

```
1 si  $G.\text{nombre\_arêtes}() = 0$  alors renvoyer  $\emptyset$ ;  
2  $U \leftarrow V$ ;  
3 pour chaque  $v$  dans  $G.\text{sommets}()$  de degré  $> 0$  faire  
4   // sélectionner  $v$  ...  
5    $\text{arêtes\_ôtées} \leftarrow G.\text{arêtes\_incidentes}(v)$ ;  
6    $G.\text{supprimer\_arêtes}(\text{arêtes\_ôtées})$ ;  
7    $A \leftarrow \{v\} \cup \text{VCEXHAUSTIF}(G)$ ;  
8   si  $|A| < |U|$  alors  $U \leftarrow A$ ;  
9    $G.\text{ajouter\_arêtes}(\text{arêtes\_ôtées})$ ;  
10  // ... ou sélectionner  $N(v)$   
11   $\text{voisins} \leftarrow G.\text{voisins}(v)$ ;  
12   $\text{arêtes\_ôtées} \leftarrow G.\text{arêtes\_incidentes}(\text{voisins})$ ;  
13   $G.\text{supprimer\_arêtes}(\text{arêtes\_ôtées})$ ;  
14   $B \leftarrow \text{voisins} \cup \text{VCEXHAUSTIF}(G)$ ;  
15  si  $|B| < |U|$  alors  $U \leftarrow B$ ;  
16   $G.\text{ajouter\_arêtes}(\text{arêtes\_ôtées})$ ;  
17 renvoyer  $U$ ;
```

Algorithme 3 : VCbranchandbound(G)

Entrées : un graphe $G = (V, E)$ non orienté, non pondéré.

Sortie : un ensemble $U \subseteq V$ de taille minimale couvrant toutes les arêtes de G .

```

1 si  $G.nombre\_arêtes() = 0$  alors renvoyer  $\emptyset$ ;
2  $M \leftarrow \text{COUPLAGEMAXIMAL}(G)$ ;
3  $U \leftarrow$  les extrémités des arêtes de  $M$ ;
4 pour chaque  $v$  dans  $G.sommets()$  de degré  $> 0$  faire
    // sélectionner  $v$  ...
5 arêtes_ôtées  $\leftarrow G.arêtes\_incidentes(v)$ ;
6  $G.supprimer\_arêtes(arêtes\_ôtées)$ ;
7  $M' \leftarrow \text{COUPLAGEMAXIMAL}(G)$ ;
8 si  $|M'| + 1 < |U|$  alors // solution prometteuse
9      $A \leftarrow \{v\} \cup \text{VCbranchandbound}(G')$ ;
10     si  $|A| < |U|$  alors  $U \leftarrow A$ ;
11  $G.ajouter\_arêtes(arêtes\_ôtées)$ ;
12 si  $|U| = |M|$  alors renvoyer  $U$ ; // solution optimale
    // ... ou sélectionner  $N(v)$ 
13 voisins  $\leftarrow G.voisins(v)$ ;
14 arêtes_ôtées  $\leftarrow G.arêtes\_incidentes(voisins)$ ;
15  $G.supprimer\_arêtes(arêtes\_ôtées)$ ;
16  $M' \leftarrow \text{COUPLAGEMAXIMAL}(G)$ ;
17 si  $|M'| + |voisins| < |U|$  alors // solution prometteuse
18      $B \leftarrow voisins \cup \text{VCbranchandbound}(G)$ ;
19     si  $|B| < |U|$  alors  $U \leftarrow B$ ;
20  $G.ajouter\_arêtes(arêtes\_ôtées)$ ;
21 si  $|U| = |M|$  alors renvoyer  $U$ ; // solution optimale
22 renvoyer  $U$ ;
```