Couplage dans un graphe biparti équilibré (d'après Mines 2012)

Durée : libre

Un *graphe* est défini par deux ensembles S et A. L'ensemble S est un ensemble fini d'éléments appelés *sommets*. L'ensemble A est un ensemble de paires de sommets ; un élément $\{x, y\}$ de A est appelé *arête* de G et x et y sont les *extrémités* de l'arête. L'*ordre* d'un graphe G est le nombre de sommets de G. Si $\{x, y\}$ est une arête de G on dit que x et y sont *voisins*. Le *degré* d'un sommet x est le nombre de voisins de x.

On dit que deux arêtes d'un graphe G sont *incidentes* si elles ont une extrémité en commun. On appelle *couplage* dans G un ensemble d'arêtes de G deux à deux non incidentes.

Un graphe G est dit *biparti* si on peut partitionner son ensemble de sommets S en deux sous-ensembles A et B non vides de sorte que toute arête ait une extrémité dans A et une extrémité dans B. Si les ensembles A et B ont même cardinal on dit qu'il s'agit d'un *graphe biparti équilibré*. Dans toute le problème on ne considère que des graphes bipartis équilibrés. On note n le cardinal commun aux ensembles A et B ; l'ordre du graphe est donc égal à $2n$. On suppose qu'on a toujours $n \geqslant 1$. Les sommets de A sont numérotés de 0 à $n-1$ et nommés $0_A, 1_A, 2_A, \dots, (n-1)_A$; les sommets de B sont numérotés de 0 à $n-1$ et nommés $0_B, 1_B, 2_B, \dots, (n-1)_B$. Une arête de G est toujours écrite en mettant d'abord l'extrémité qui est dans A puis celle qui est dans B.

On représente toujours les graphes bipartis équilibrés par des schémas comme on peut le voir dans la figure 1 avec le graphe G_0 en représentant les sommets de A à gauche et les sommets de B à droite.

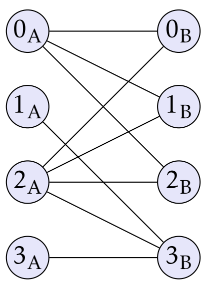

Figure 1 – Le graphe G_0.

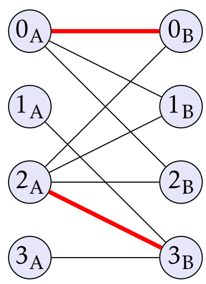

Figure 2 – Le graphe G_0 et le couplage C_0.

Dans le graphe G_0 les arêtes $\{0_A, 0_B\}$ et $\{2_A, 3_B\}$ étant non incidentes elles forment un couplage nommé C_0 dont les arêtes sont dessinées en gras figure 2. On dit que dans ce couplage :

- le sommet 0_A est couplé au sommet 0_B et réciproquement ;
- le sommet 2_A est couplé au sommet 3_B et réciproquement ;
- les sommets 1_A, 3_A, 1_B et 2_B sont non couplés.

Le cardinal d'un couplage est le nombre d'arêtes de celui-ci ; par exemple le cardinal de C_0 vaut 2.

Partie I. Généralités

Question 1. Exhiber un couplage de cardinal 3 de G_0, puis indiquer s'il existe dans G_0 un couplage de cardinal 4. Justifier la réponse.

Un graphe biparti équilibré d'ordre $2n$ est représenté par une matrice de taille $n \times n$ dont les lignes correspondent aux éléments de A et les colonnes aux éléments de B. Les cases de cette matrice sont indicées par (i, j) avec $0 \leqslant i \leqslant n-1$ et $0 \leqslant j \leqslant n-1$ et contiennent des valeurs booléennes : la case d'indice (i, j) contient la valeur « vrai » (`True` en Python) si $\{i_A, j_B\}$ est une arête du graphe ; elle contient la valeur « faux » (`False` en Python) dans le cas contraire. Le graphe G_0 ci-dessus est donc représenté par la matrice et codé en Python par : :

$$\begin{array}{cc|cccc|}
 & & & j & & \\
 & & 0 & 1 & 2 & 3 \\
\hline
 & 0 & V & V & V & F \\
 & 1 & F & F & F & V \\
i & 2 & V & V & V & V \\
 & 3 & F & F & F & V \\
\hline
\end{array}$$

```
g0 = [ [True,  True,  True,  False],
       [False, False, False, True],
       [True,  True,  True,  True],
       [False, False, False, True] ]
```

Un couplage est codé par une liste uni-dimensionnelle indicée de 0 à $n-1$. Pour tout $i \in [\![0, n-1]\!]$, si le sommet i_A est couplé avec le sommet j_B alors la case d'indice i contient la valeur j ; si le sommet i_A n'est pas couplé, la case d'indice i contient la valeur -1. Ainsi, le couplage C_0 est codé par :

```
c0 = [0, -1, 3, -1]
```

Enfin, une arête est codée par un couple `a` de deux entiers avec `a[0]` dans A et `a[1]` dans B.

Question 2. Soit G un graphe biparti équilibré d'ordre $2n$. On considère une liste C d'entiers de longueur n et contenant dans ses cases indicées de 0 à $n-1$ soit la valeur -1, soit une valeur comprise entre 0 et $n-1$.
Écrire en Python une fonction `verifie` telle que si `g` est une matrice représentant le graphe G et `c` une liste représentant C alors `verifie(g, c)` renvoie `True` si la liste C représente un couplage dans G et `False` sinon.

Indiquer la complexité de la fonction `verifie`.

Question 3. On considère une liste C de longueur n codant un couplage d'un graphe G.
Écrire en Python une fonction `cardinal` telle que si `c` est une liste codant un couplage alors `cardinal(c)` renvoie le cardinal de ce couplage.

Indiquer la complexité de la fonction `cardinal`.

Partie II. Un algorithme pour déterminer un couplage maximal

On dit qu'un couplage C dans un graphe G est *maximal* si toute arête de G n'appartenant pas à C est incidente à au moins une arête de C. Par exemple, le couplage C_0 de G_0 est maximal. Un couplage maximal de G n'est pas forcément de cardinal maximum parmi les couplages de G. On cherche à concevoir un algorithme qui détermine un couplage maximal dans un graphe biparti équilibré G.

L'algorithme, nommé *algo_approche*, est le suivant :

- on commence avec un couplage vide C ;
- tant que G possède au moins une arête :
  - on choisit une arête a de G dont la somme des degrés des extrémités soit minimale ;
  - on ajoute l'arête a au couplage C ;
  - on retire de G l'arête a et toutes les arêtes incidentes à a.

On admettra que le résultat est, par construction, un couplage maximal.

Question 4. Appliquer *algo_approche* au graphe G_0 (représenté figure 1).

On considère par la suite le graphe biparti équilibré G_1 d'ordre 12 représenté sur la figure 3.

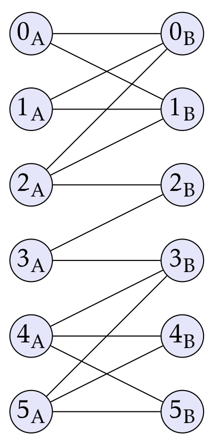

Figure 3 – Le graphe G_1.

Question 5. On applique *algo_approche* au graphe G_1. Déterminer la première arête a_1 choisie par *algo_approche* ; tracer le graphe obtenu après suppression de a_1 et des arêtes incidentes à a_1. Montrer que le couplage obtenu par *algo_approche* est de cardinal au plus 5 et indiquer s'il est de cardinal maximal parmi les couplages de G_1.

Question 6. Soit G un graphe biparti équilibré. Il s'agit d'écrire une fonction *arete_min* qui détermine une arête de G dont la somme des degrés des extrémités soit minimum.
Écrire en Python une fonction `arete_min` qui prend pour argument un graphe biparti équilibré et renvoie un couple `(True, a)` où `a` est un couple représentant une arête qui atteint ce minimum lorsque G possède au moins une arête. Dans ce cas contraire cette fonction renvoie la valeur `(False, None)`.

Indiquer la complexité de la fonction `arete_min`.

Question 7. Écrire en Python une fonction `supprimer` telle que si `g` est une matrice codant un graphe biparti équilibré G eta un couple de deux entiers codant une arête a de G alors `supprimer(g, a)` modifie `g` pour que, après modifications, `g` code le graphe obtenu à partir de G en supprimant a ainsi que toutes les arêtes incidentes à a.

Indiquer la complexité de la fonction `supprimer`.

Question 8. Il s'agit enfin de définir l'algorithme *algo_approche* décrit au début de la deuxième partie.
Écrire en Python une fonction `algo_approche` telle que si `g` est une matrice qui code un graphe biparti équilibré G, `algo_approche` effectue *algo_approche* à partir d'une copie de G et renvoie la liste codant le couplage obtenu.
Indiquer la complexité de la fonction `algo_approche`.

Partie III. Recherche exhaustive d'un couplage de cardinal maximum

Question 9. Écrire une fonction Python `une_arete` qui prend pour argument une matrice représentant un graphe biparti équilibré G et renvoie un couple `(True, a)` où a est un arête quelconque de G si G s'il en possède au moins une, et `(False, None)` dans le cas contraire.

Question 10. On cherche à établir un algorithme récursif nommé *meilleur_couplage* qui permette de déterminer un couplage de cardinal maximal dans un graphe biparti équilibré. Le principe est le suivant.

Si le graphe courant ne contient aucune arête, le cardinal maximum d'un couplage est 0 et aucun sommet n'est couplé. Dans le cas contraire, l'algorithme considère une arête quelconque a du graphe courant et recherche successivement :

- un couplage de cardinal maximal parmi les couplages du graphe courant ne contenant pas a ;
- un couplage de cardinal maximal parmi les couplages du graphes contenant a.

L'algorithme déduit alors un couplage de cardinal maximal.

Écrire en Python une fonction récursive `meilleur_couplage` telle que, si `g` est une matrice codant un graphe biparti équilibré G, `meilleur_couplage(g)` renvoie une liste codant un couplage de cardinal maximal dans G. La fonction utilisera le principe décrit plus haut.