La Théorie des graphes

Fabio Daussy

21 février 2023

Dans ce cours texte, nous allons voir brievement des éléments essentiels de la théorie des graphes. Inspiré du cours de Philippe Langevin. Notamment sa définition et jusqu'où le sujet peut s'étendre. Vous devez probablement connaître ce fameux jeu où l'on vous demande de déssiner une maison sans lever le crayon. Ce même problême est aussi connu sous la forme du problême des ponts et des îles. C'est un problême type de la théorie des graphes qui a été résolu par Euler [1].

Table des matières

1 Introduction

"La théorie des graphes s'est développée au cours du XX^e siècle, Las Vergnas nous rappelle que terminologie de graphe a été introduite par Sylvester en 1877, et que le premier livre sur la théorie des graphes a été écrit par D.König en 1936. La génèse de la théorie des graphes semble être une étude de Léonard Euler, un très célèbre mathématicien du XVIII^e siècle. Dans un article publié en 1736, il traite un problème devenu classique,

1. Leonhard Euler (1707-1783) est un mathématicien et physicien suisse qui a fait de nombreuses découvertes en théorie des graphes et en calcul infinitésimal. Source : Wikipédia

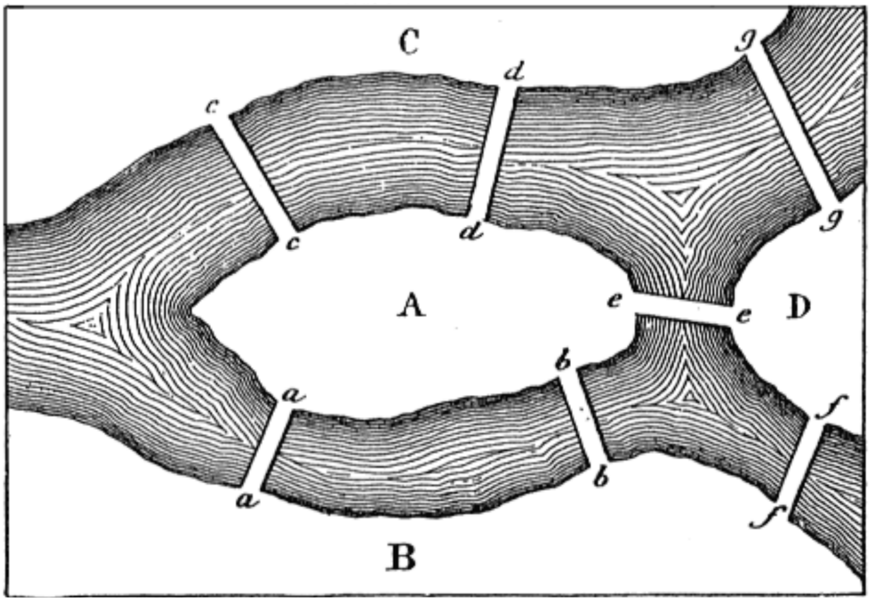

FIGURE 1 – Les ponts de Koenigsberg en 1759

illustré par la devinette : peut-on faire une promenade passant une fois par chacun des sept ponts de la ville de Koenigsberg (1) ? Il suffit de faire quelques essais pour se convaincre de l'impossibilité de réaliser une telle promenade. L'objectif de cette section est de dégager un résultat général." [2]

2 Notion de bases

Définition 2.1. *Un graphe G est un couple (X, U), où X est un ensemble de* ***sommets*** *et U un sous ensemble tel que $U \subseteq \mathcal{P}_2(X)$, ce sont les* ***arêtes*** *du graphe*

Exemple. Ainsi, pour le graphe maison G en figure 2 nous avons :

$$X := \{1, 2, 3, 4, 5\} \quad U := \{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{2, 4\}, \{2, 5\}, \{4, 5\}, \{2, 3\}, \{1, 4\}\}$$

Définition 2.2. *On dit que deux sommets $x, y \in X$ sont* ***adjacents*** *si et seulement si $\{x, y\} \in U$.*

Définition 2.3. *Soit un sommet s. On appelle* ***degré*** *de s le nombre d'arêtes qui sont* ***incidentes*** *en s. Le degré d'un sommet $s \in X$ se note aussi $deg(s)$.*

Exemple. L'ensemble des degrés des sommets de G (1).

Proposition 2.4. *Soit m le nombre d'arêtes du graphe,*

$$\sum_{s \in X} deg(s) \quad = \quad 2 \times m \tag{1}$$

2. Introduction du cours sur la théorie des graphes de Monsieur Philippe Langevin [1]

sommet	1	2	3	4	5
degré	3	4	3	4	2

Table 1 – L'ensemble des degrés des sommets du graphe G (2)

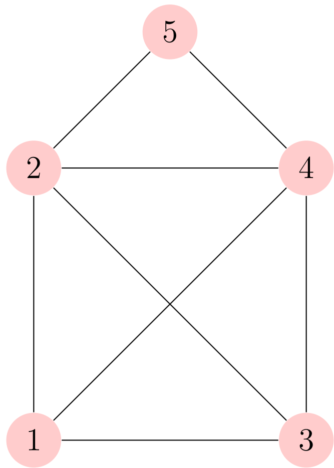

Figure 2 – Le graphe maison $G(X,U)$

Définition 2.5. *On appelle l'**ordre** d'un graphe G le nombre de sommets qui le composent (i.e. le cardinal de X).*

Définition 2.6. *Un sommet s est dit **isolé** si et seulement si son degré vaut 0.*

Définition 2.7. *Soit n l'ordre du graphe. Un sommet s est dit **dominant** si et seulement si son degré vaut $n-1$.*

Définition 2.8. *Si tous les sommets d'un graphe d'ordre n sont dominants alors on appelera ce graphe, le **graphe complet** d'ordre n. On le note $\mathbb{K}_n$.*

3 Connexité

Définition 3.1. *Soit $G(X,U)$ un graphe d'ordre n, on appelle **chemin** toute suite de sommets*

$$x_1, x_2, \ldots, x_l \quad \forall x_i \in X,\ i \in [1,n] \qquad \textit{tel que deux sommets consécutifs sont adjacents}$$

Remarque. Un chemin dont les extrémités est le même sommet est un **cycle**.

Notons que :

- l est la **longueur** du chemin.
- x_1 et x_l sont les **extrémités** du chemin.

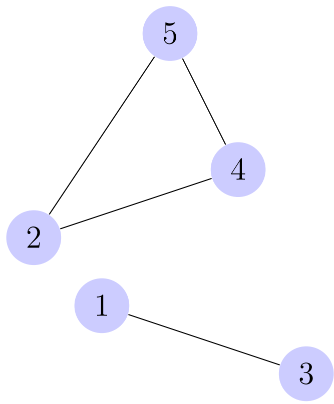

FIGURE 3 – Un graphe d'ordre 5

Définition 3.2. *On dit que deux sommets x, y sont **liés** si et seulement si, il existe un chemin d'extrémités x et y. On note la liaison :*

$$x \rightsquigarrow y$$

Proposition 3.3. *La relation de liaison entre deux sommets est une **relation d'équivalence**. Les classes d'équivalences de cette relation sont appelées **composantes connexes**.*

Exemple. Le graphe (3) possède 2 composantes connexes

Remarque. On appelle **graphes connexes** les graphes à une composante connexe.

4 Graphe Eulerien

Définition 4.1. *Un chemin **eulérien** dans un graphe est un chemin qui passe par toutes les arêtes du graphe une et une seule fois*

Le théorème d'Euler permet de vérifier s'il existe ou non un cycle Eulérien dans le graphe maison (2)

Théorème 4.2. *Soit $G(X, U)$, un graphe d'ordre n, sans point isolé, possède un **cycle eulérien** si et seulement si :*

1. *Le graphe est **connexe***
2. *Tous les degrés des sommets sont **pairs***

Le graphe maison (2) est bien connexe. En revanche, le sommet 1 est de degré impair donc il n'est pas eulérien. Démontrons ce théorême :

Démonstration. Soit $G(X, U)$, un graphe d'ordre n, dont tous les sommets sont de degré pair. On choisit un sommet x_0. On construit un chemin de proche en proche en s'interdisant de repasser deux fois pas la même arête tant que possible. Cette promenade s'arrête en x_0 (point de départ). On considère le graphe partiel composé des arêtes

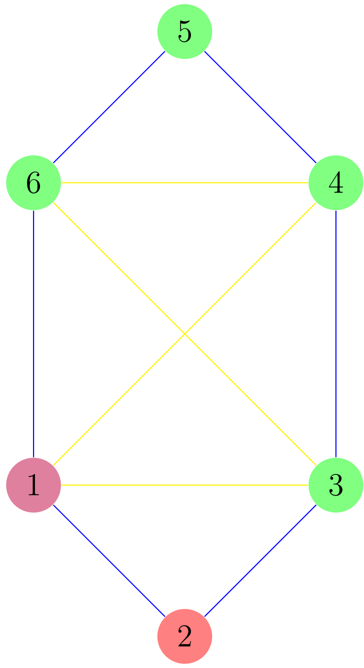

Figure 4 – Graphe eulérien d'ordre 6

restantes, il n'est pas forcément connexe. Chaque composante connexe est un graphe d'ordre $< n$ car x_0 est isolé. Les parités des degrés restent inchangées donc chaque composante connexe admet un **cycle eulérien**. On dit que chaque composante connexe du graphe partiel possède un représentant sur la promenade.

Ainsi pour faire le cycle, il suffit de passer sur la promenade initial, et à chaque représentant, on trace son cycle eulérien jusqu'à x_0 et le cycle eulérien sur le graphe initial est effectué. □

Exemple. Dans le graphe 4, le sommet 2 joue le rôle de x_0. La promenade est dessinée en bleu et le cycle eulérien du graphe partiel sans la promenade est dessiné en jaune . Ainsi, pour tracer sans "lever le stylo" ce cycle eulérien, on part de 2, on suit la promenade bleue . Une fois arrivé au sommet 1 , on réalise le cycle jaune, puis on boucle le cycle en sur 2.

1 est représentant de la promenade

5 Implantation en langage C

Nous allons implanter les graphes en langage C. Pour cela nous allons utiliser la structure de données suivante (5).

```
typedef struct{
  int nbs;
  char ** mat;
  char * clr;
}t_graphe;
```

nbs correspond au nombre de sommet du graphe de le structure. *mat* est une matrice de booléen de taille nbs $\times$ nbs. Chaque sommet est représenté par un nombre $<$ nbs. Ainsi, si vous souhaitez accéder à l'arête entre le sommet 1 et 3 ; il suffit de chercher dans la matrice soit **mat**[**1**][**3**], soit **mat**[**3**][**1**]. S'il y a une arête entre le sommet 1 et 3 alors **mat**[**1**][**3**] vaut 1. 0 sinon. *clr* est un tableau de taille nbs qui va simplement permettre de se repérer lors d'un parcours du graphe. On colorie les sommets qu'on a parcouru pour ne plus retourner dessus plus tard dans l'algorithme. Il sert de condition d'arrêt. On peut alors en déduire un algorithme de décision pour savoir si un graphe est eulérien ou non. Le code se trouve ici (5)

```
int deg(t_graphe G, int s){
  int degre = 0;
  for (int t = 0; t < G.nbs; t++){
    degre += G.mat[s][t];
  }
  return degre;
}

char eulerien(t_graphe G){
  for (int s = 0; s < G.nbs; s++){
    if (deg(G,s) % 2){
      return 0;
    }
  }
  return 1;
}
```

La complexité de cet algorithme est quadratique, il est en $O(n^2)$ si on considère que n est la taille l'ordre du graphe passé en paramètre.

6 Conclusion

Nous n'avons qu'effleurer le sujet immense que sont les graphes, ceux-ci sont partout autour de nous. Mais l'intêret de cette courte initiation était de vous donner une petite soif de savoir sur les graphes. En espérant que j'ai, au moins partiellement, réussi à vous donner cette envie.

Références

[1] Philippe Langevin. *Théorie et Algorithmique des Graphes, Licence Informatique.* PhD thesis, 2022.