



HAL
open science

Théorie des graphes

Mohammed Tamali

► **To cite this version:**

| Mohammed Tamali. Théorie des graphes. Doctoral. Algeria. 2013. cel-01385792

HAL Id: cel-01385792

<https://hal.science/cel-01385792>

Submitted on 22 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Béchar

Laboratoire des Études Énergétiques en Zones Arides
Équipe Modélisation & Simulation des Systèmes



CHAPITRE IV :

Théorie des graphes

Concepts de base & fondements

Modélisation graphique.

Les sept ponts de Königsberg

Cours réalisé par : Prof. TAMALI Mohammed,

<http://www.univ-bechar.dz/mtamali>

Université de Béchar | FS&T
(ENERGARID Lab./SimulIA)



Simulia Introduction



The University of Bechar was born in 1986 as the National Institutes of Higher Education (INES) in 1992 it becomes University Center and on January 07, 2007, it was officially declared as a University. Since then, many Research Teams have seen the day. In 2011, The Laboratory for Energy Systems Studies Applied to Arid Zones was run by a group of young and well motivated researchers (7 Research teams) to solve real problems affecting arid zones, Simulia Team is one of them in the same laboratory. The Workload of Simulia concern studies and applications of modeling and simulation of systems in Arid Areas.

Research areas:

- Energy & Environment (Modeling & Simulation)
- Application of heat in arid zones
- Energy economy.
- Mapping and development of resources in arid zones.
- SIMULIA for the task in the short term, to develop the computer code for modeling and simulation which can be accessed online.

Website of the laboratory team: <http://energarid.wordpress.com/>



Plan

Généralités & Présentations

Définitions & Bases

Le point de vue étymologique

Cadre d'utilisation

Modélisation

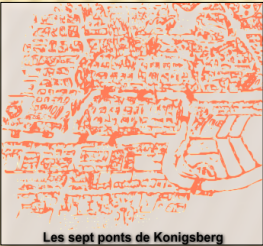
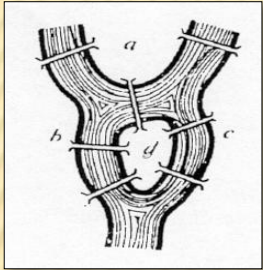
Conclusions

Références



Généralités & Présentations

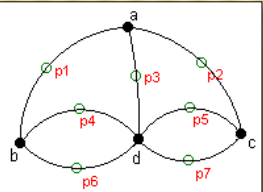
On fait généralement remonter la naissance de la Théorie des Graphes au célèbre problème des ponts de Königsberg (aujourd'hui Kaliningrad) qui passionnait la bourgeoisie prussienne du XVIIIème siècle : La Ville de Königsberg, sur la Pregel, était pourvue de 7 ponts et la question était de savoir si l'on pouvait imaginer une promenade dans la ville qui emprunterait chacun des 7 ponts une fois et une seule pour revenir à son point de départ.



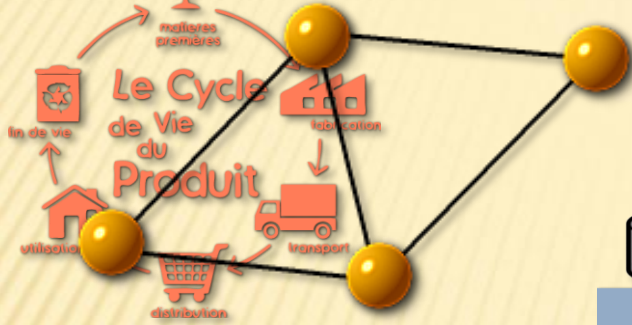
Les sept ponts de Königsberg

La théorie des graphes s'est alors développée et intégrée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales et sans oublier les réseaux d'ordinateurs et de télécommunication. Depuis le début du XXe siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de König, Menger, Cayley puis de Berge et d'Erdős [1].

De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe dit 'système' (S) en exprimant les relations entre ses éléments tel que les réseaux de communication, les réseaux routiers, interaction de diverses espèces animales, circuits électriques, en programmation et le plus intéressant son application aux sciences de l'Internet.



Définitions & Bases (1)



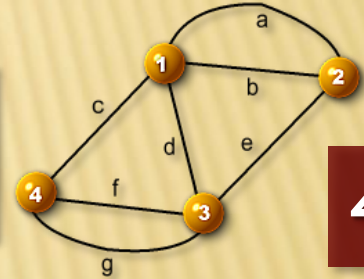
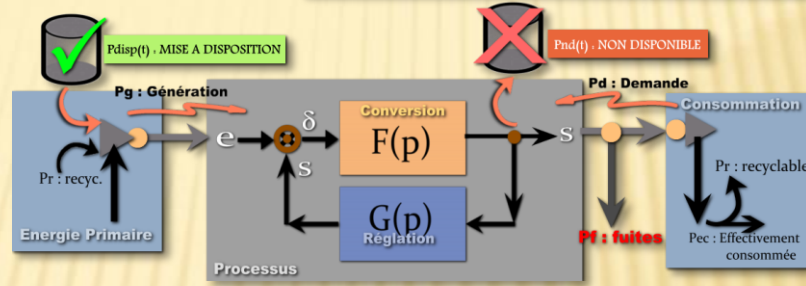
Un graphe simple noté G est un couple formé de deux ensembles liés par une application mathématique. L'un d'eux est l'ensemble $X = \{x_1, x_2, \dots, x_n\}$ dont les éléments sont appelés '**sommets**', l'autre est l'ensemble $A = \{a_1, a_2, \dots, a_m\}$, partie de l'ensemble $P^2(X)$ des parties à deux éléments (couple de sommets) de X , dont les éléments sont appelés '**les arêtes**'. On notera cette relation $G = (X, A)$.

Lorsque $a = \{x, y\} \in A$, on dit que a est l'arête de G d'extrémités x et y , ou que a joint x et y , ou que a passe par x et y . Les sommets x et y sont dits **adjacents** dans G .

Un **multi-graphe** $G = (X, A, f)$ est déterminé par:

- L'ensemble X des sommets
- L'ensemble A , cette fois abstrait
- L'application $f : A \rightarrow P^2(X)$

un multi-graphe avec boucles peut comprendre des arêtes multiples entre deux sommets donnés ainsi que des boucles multiples en un sommet.



4

Exemple de cas où le modèle graphe est recommandé

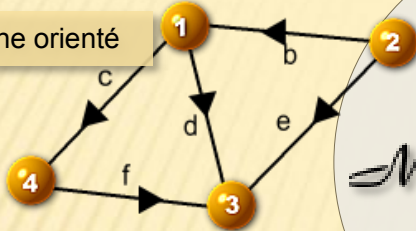
Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs. Les derniers travaux en théorie des graphes sont souvent effectués par des informaticiens, du fait de l'importance que revêt l'aspect algorithmique.

Dans le cadre général, un graphe est dit non orienté si le ne contient pas de sens sur les arêtes.

Le nombre de sommets présents dans un graphe est l'ordre du graphe. **Le degré** d'un sommet x , noté $d(x)$ est le nombre d'arêtes dont ce sommet est une extrémité.

Définitions & Bases

Exemple graphe orienté



$$\mathcal{M} = \begin{bmatrix} 4 & 3 \\ 4 & 3 \end{bmatrix}$$

Un **graphe orienté** G_n est formé de deux ensembles: un ensemble $X=\{x_1, x_2, \dots, x_n\}$ dont les éléments sont appelés sommets, et un ensemble $A=\{a_1, a_2, \dots, a_n\}$, partie du produit cartésien $X \times X$, dont les éléments sont appelés arcs. On notera $G_n=(X, A)$. Si $a=(x, y)$ est un arc du graphe G , x est l'**extrémité initiale** de a et y l'**extrémité finale** de a .

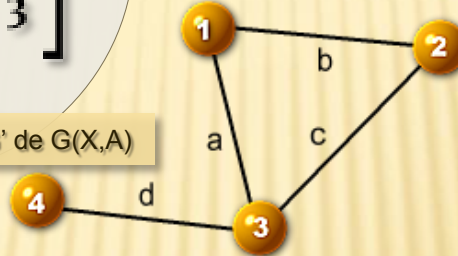
À tout graphe orienté $G_n(X, A)$, on associe le graphe simple $G(X, B)$ où :

$$\{x, y\} \in B \neq ((x, y) \in A \text{ ou } (y, x) \in A).$$

On appelle **graphe complet** un graphe dont tous les sommets sont adjacents.

Un graphe dont les arêtes sont caractérisées par une quantité est dit **valué**.

Sous-graphe G' de $G(X, A)$

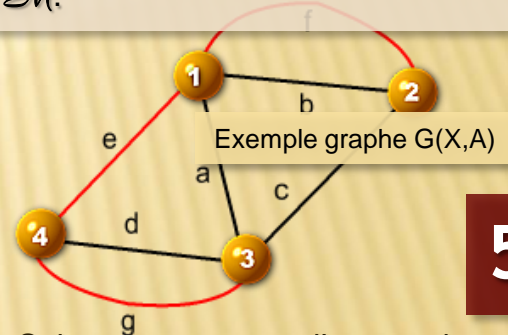


Une **chaîne** est une liste ordonnée de sommets telle que chaque sommet de la liste soit adjacent au suivant. La longueur d'une chaîne est le nombre d'arêtes qui la composent.

La **distance** entre deux sommets est la plus courte longueur des chaînes qui les relient. Le **diamètre** d'un graphe est la plus grande distance entre deux sommets.

Un **sous-graphe** d'un graphe G est un graphe G' composé de certains sommets de G , ainsi que toutes les arêtes qui relient ces sommets.

La **matrice associée** à un graphe d'ordre n dont les sommets sont numérotés de 1 à n est une matrice symétrique, de dimension $n \times n$, où le terme à l'intersection de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne vaut k , nombre d'arêtes reliant i et j . C'est encore la matrice de liaison \mathcal{M} .



Soit x un sommet d'un graphe orienté. On note $d^+(x)$ le nombre d'arcs ayant x comme extrémité initiale, et $d^-(x)$ le nombre d'arcs ayant x comme extrémité finale. Ainsi, on a :

$$d(x) = d^+(x) + d^-(x)$$

La matrice des distances du graphe G est la matrice

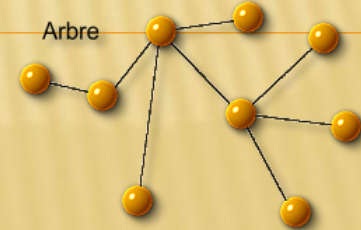
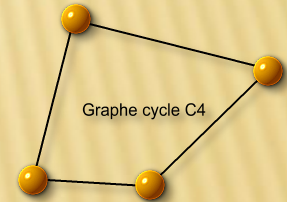
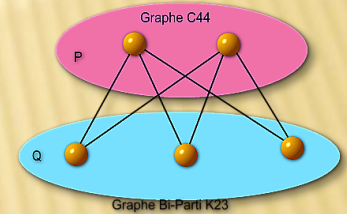
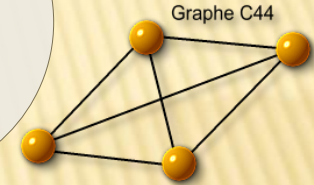
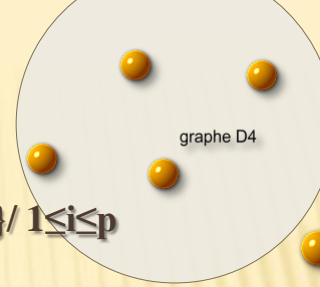
$$D = \{d(i, j) = (d(x_i, x_j))\}.$$

Le point de vue étymologique

- Le graphe **discret** d'ordre n , $D_n = (X, \emptyset)$.
- Le graphe **complet** d'ordre n , K_{nn} où $X = \{1, 2, \dots, n\}$ et $A = P^2(X)$
- Le graphe **biparti-complet** K_{pq} où $X = \{x_1, x_2, \dots, x_p, y_1, y_2, \dots, y_q\}$ et $A = \{f(x_i, y_j) / 1 \leq i \leq p \text{ et } 1 \leq j \leq q\}$.
- Le **cycle** C_{nn} où $X = \{1, 2, \dots, n\}$ et $A = \{\{1, 2\}, \{2, 3\}, \dots, \{n-1, n\}, \{n, 1\}\}$.
- Un graphe **valué** est un graphe orienté $G_n(X, A)$, muni d'une fonction C appelée fonction de coût.
- Le **coût d'un chemin** est la somme des coûts des arcs de ce chemin. On peut définir la matrice des coûts du graphe, c'est la matrice $C = \{c_{i,j}\}$ où : $c_{i,j}$ est le coût de l'arête (i, j) .
- Un graphe est dit **simplement connexe** si pour tout couple $(i, j) \in A$, il y a toujours un chemin de i vers j .
- Un **arbre** est un sous-graphe simple connexe ne possédant pas de cycle simple et construit à base de tous les nœuds du graphe d'origine.
- La **matrice d'incidence** A d'un graphe orienté G_n est définie par :

$$a_{i,j} = \begin{cases} 1 & \text{si l'arête } j \text{ est incidente sur le nœud } i \\ -1 & \text{si l'arête } j \text{ est sortante du nœud } i \\ 0 & \text{sinon} \end{cases}$$

- **Valeur d'une chaîne**: la somme des valeurs des arêtes (arcs) d'une chaîne d'un graphe **valué**.



Modélisation par la topologie des graphes



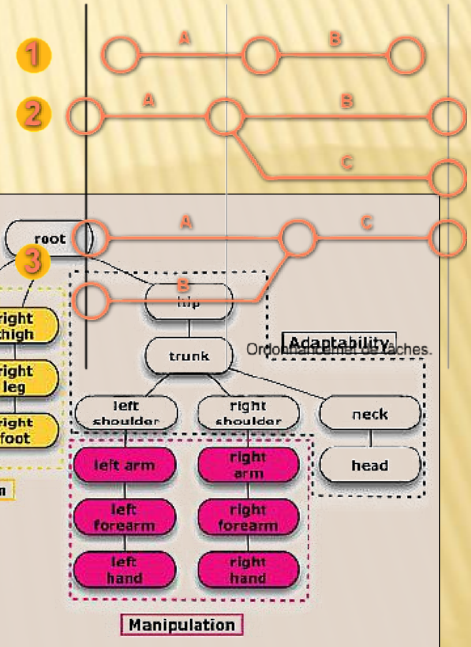
Coordination du mouvement et des prises

Le qualificatif '**un robot marcheur**' veut dire quoi? Ensemble matériel pouvant se déplacer d'un point A vers un autre B. *Marcher* c'est la capacité du déplacement. La marche nécessite l'existence de **membres articulés** dont la fonction est d'aider leur support à marcher.

Les articulations désignent les membres, les arêtes. L'adaptation aux sommets est modélisable par un graphe. Les caractéristiques relationnelles sont mises en évidence.

La planification d'un projet intervient après établissement des plans pour se mettre à l'exécution de l'objet du projet. C'est en ce moment que les termes se clarifient en fixant les limites des contrats, des budgets, des plannings, du staff, des notes d'organisation, des structures logistiques et d'infrastructures. Les cahiers des charges, moraux, financiers et techniques sont édités.

Pour se faire, la théorie des graphes est recommandée. Les diagrammes GANT et PERT sont les outils par excellence, utilisés. Les gains d'une telle modélisation à stratégie graphique est le gain en temps et coût.



Modélisation par la topologie des graphes

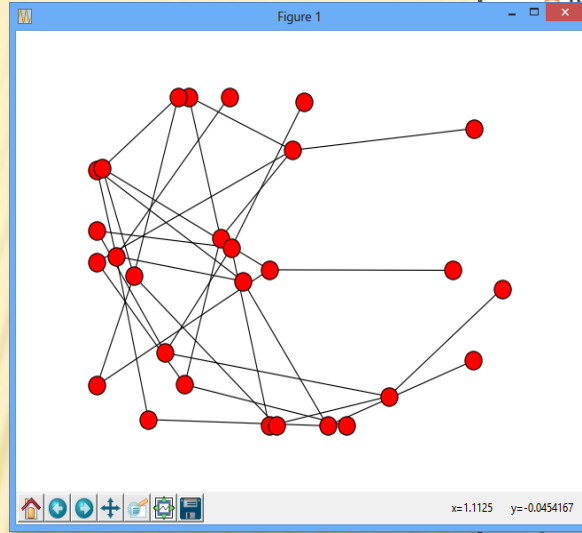
En environnement Python, le module **NETWORKX**, présente des facultés surprenantes pour la manipulation des graphes.

```
import networkx as nx
import matplotlib.pyplot as plt
G=nx.Graph()
G.add_node("a")
G.add_nodes_from(["b","c"])

G.add_edge(1,2)
edge = ("d", "e")
G.add_edge(*edge)
edge = ("a", "b")
G.add_edge(*edge)

print("Nodes of graph: ")
print(G.nodes())
print("Edges of graph: ")
print(G.edges())
# adding a list of edges:
G.add_edges_from([("a","c"), ("c","d"), ("a",1), (1,"d"), ("a",2)])

nx.draw(G)
plt.savefig("simple_path.png") # save as png
plt.show() # display
```



```
[*]: import networkx as nx
import matplotlib.pyplot as plt
G=nx.Graph()
G.add_node("a")
G.add_nodes_from(["b","c"])

G.add_edge(1,2)
edge = ("d", "e")
G.add_edge(*edge)
edge = ("a", "b")
G.add_edge(*edge)

print("Nodes of graph: ")
print(G.nodes())
print("Edges of graph: ")
print(G.edges())
# adding a list of edges:
G.add_edges_from([("a","c"), ("c","d"), ("a",1), (1,"d"), ("a",2)])

nx.draw(G)
plt.savefig("simple_path.png") # save as png
plt.show() # display

Nodes of graph:
['a', 1, 'c', 'b', 'e', 'd', 2]
Edges of graph:
[('a', 'b'), (1, 2), ('e', 'd')]
```

Conclusions

Pour entreprendre des actions sûres et avec impact réel, la méthodologie est d'une grande importance. C'est, en d'autres termes, ce qui justifie le prix payé avant d'atteindre son but.

Nous en tant que créations, les systèmes qui nous entourent, recèlent de beaucoup de surprises. L'adaptation d'une stratégie d'observation nous permet de délimiter la zone appropriée pour entamer son étude et la région équivalente au domaine de définition de notre système.

Juger c'est la dernière action mais appréhender en est la première. Une amélioration, une évolution d'un système donné, ne sont acceptable que si l'on a, à priori, bien collecté toutes les informations relatives à la composition et constitution, au fonctionnement et à la dépendance vis-à-vis des autres systèmes adjacents.

Le coût encourut si l'erreur est commise pourrait être fatale, pas seulement pour le système en question mais aussi pour tout les systèmes en relation directe ou indirecte.

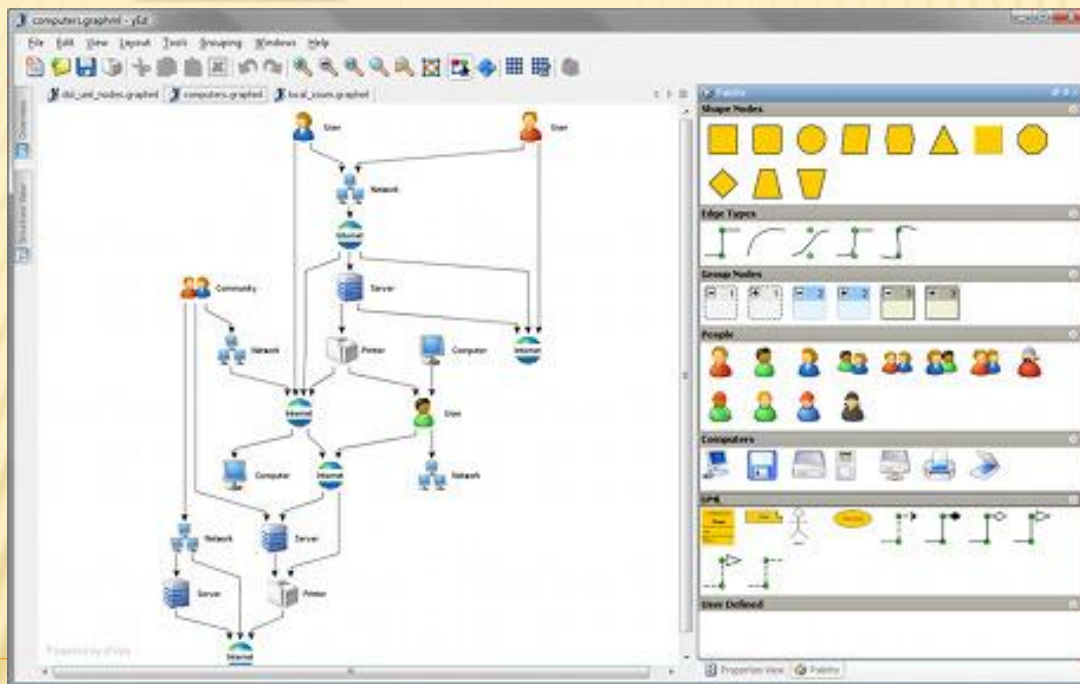
Garder l'équilibre universelle est une affaire primordiale. L'observation scientifique, la modélisation et la simulation sont des outils de manœuvres très importants.

Leur optimisation reste pour toujours une question de possibilités offertes à l'opérateur pour améliorer selon son besoin sans enfreindre à l'équilibre des compositions et relations totales.

Les libertés à l'introduction d'une certaine mise à jour est toujours garantie, sauf nécessité de garantir la non interférence avec la sûreté des ensemble voisins.

Nous sommes interpellés par le besoin : ne cherchez jamais à en inventer !

Faites LA bonne Observation, décomposer, recomposer et valider une **Conception**



MERCI POUR VOTRE ATTENTION
Fin du quatrième chapitre

Références

- [1]- E.SIGWARD; 'Introduction à la théorie des graphes', e.sigward@ac-nancy-metz.fr
- [2]- L.-V. Bertalanfy, 'General System Theory', Edition MASSON, 1972.
- [3]- <http://intelligence-artificielle-robotique.weebly.com/planification-du-mouvement-pour-les-systegravemes-anthropique.html.09/11/2014> (0h10min).
- [4]- <http://www.yworks.com/>, 09/11/2014 (0h42).