**Sarah Ghiri – 000334719 8 decembre 2017**

**Evguéniy Starygin – 000443325**

**INFO-F203 – Algorithmique 2**

**Projet : "Cycles et hypergraphes"**

****

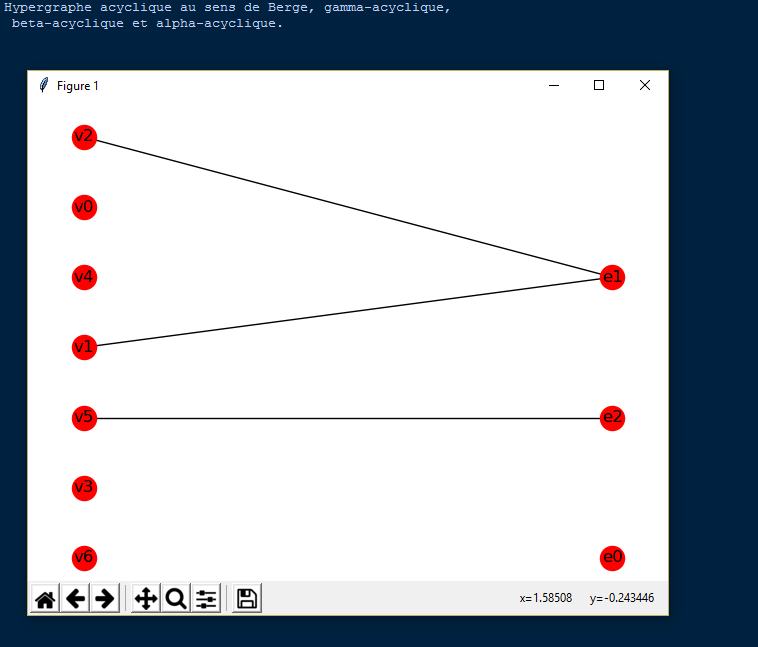
**Gestion des hypergraphes**

L’hypergraphe est généré aléatoirement via la fonction graph\_generator() avec entre 7 et 15 sommets et entre 2 et 5 hyperarêtes. Elle génère le graphe d’incidence de l’hypergraphe via la bibliothèque Networkx[[1]](#endnote-1) et l’affiche avec matplotlib[[2]](#endnote-2) qui est gérée par la fonction bipartite\_draw(g).

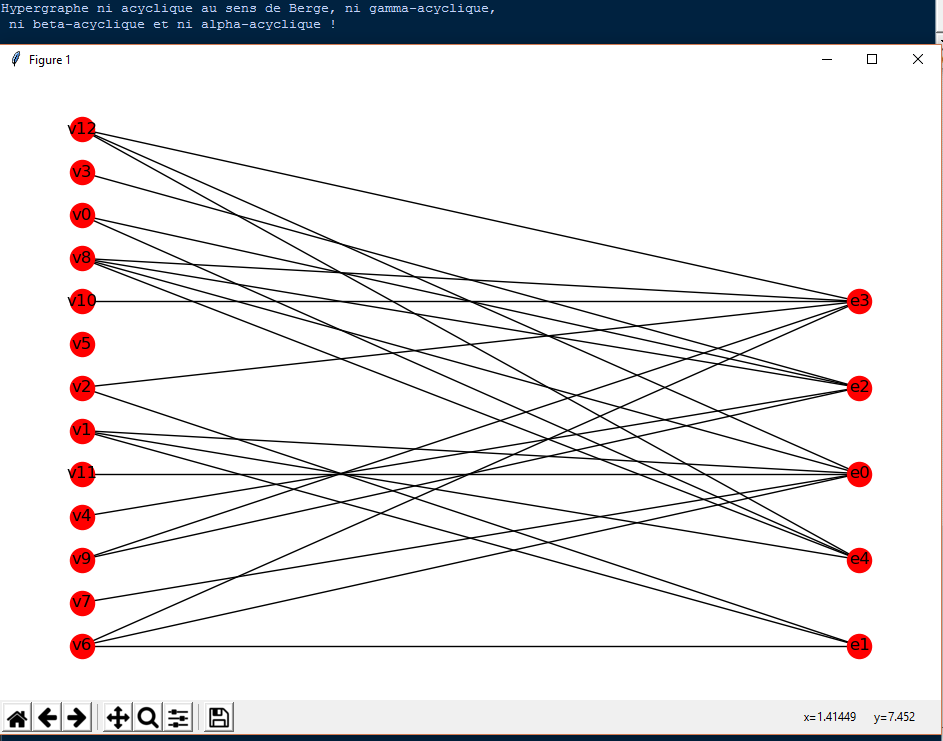
Le choix de Networkx est motivé par les vastes fonction entourant les graphes permettant un gain de temps et certaines fonction permettant de vérifier nos propres implémentations comme la is\_chordal(G) ou [chordal\_graph\_cliques](https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.chordal.chordal_alg.chordal_graph_cliques.html#networkx.algorithms.chordal.chordal_alg.chordal_graph_cliques)(G)[[3]](#endnote-3).

On choisit de représenter l’hypergraphe via son graphe d’incidence car on en a besoin pour trouver ou non un cycle dans le sens de Berge, si ce dernier est Berge-acyclique on peut alors ne pas construire de graphe primale pour l’alpha acyclicité. Mais pas seulement ce dernier convient très bien pour les autres acyclicité comme beta et gamma car il suffit de voir les neighbors des sommets e qui sont les hyperarêtes pour avoir les sommets contenu dans une hyperarêtes, ect

**Exemples d’exécutions**



Le script affiche dans le terminal si le script est berge, gamma, beta, alpha acyclique et puis affiche la représentation du graphe d’incidence de l’hypergraphe via mathplotlib. Dans l’exemple ci-dessus est berge-acyclique du coup il est également gamma, beta et alpha (berge ⇒ γ-acyclique ⇒ β-acyclique ⇒ α-acyclique)



Celui-ci possède un cycle au sens de berge, gamma, beta et alpha.

**β-acyclicité**

Un hyper graphe H = (V, E) est β-acyclique si en appliquant successivement les 2 règles suivantes, on obtient un hypergraphe vide[[4]](#endnote-4) :

1. Si un sommet est un nest point (Un sommet v dans un hypergraphe H est un nest point si l’ensemble des hyperarêtes de H le contenant forme une chaîne pour l’inclusion, c.-à-d pour toutes hyperarêtes E et F contenant v, E ⊂ F ou F ⊂ E), alors on le retire de H.
2. Si une hyperarête est vide, alors on la retire de E.

Dans le cas où l’on arrive à un point où l’application des ces règles n’est plus possible et que l’hypergraphe n’est pas vide alors on détecte un β cycle. Un hypergraphe est également β-acyclique s’il est γ-acyclique (γ-acyclique ⇒ β-acyclique) et (β-acyclique ⇒ α-acyclique). Cette règle est utilisée dans la fonction beta\_acyclic qui prend en paramètre le graphe d’incidence de l’hypergraphe.

La complexité de l’algorithme est en O(n²).

1. <https://networkx.github.io/documentation/networkx-1.10/index.html> [↑](#endnote-ref-1)
2. <https://matplotlib.org/2.1.0/index.html> [↑](#endnote-ref-2)
3. <https://networkx.github.io/documentation/networkx-1.10/reference/algorithms.chordal.html> [↑](#endnote-ref-3)
4. <https://www.imj-prg.fr/theses/pdf/2009/david_duris.pdf> p42,43 [↑](#endnote-ref-4)