Génération de graphes connexes aléatoires avec séquence de degrés donnée

Stage au LIAFA (Paris 7) sous la direction de Matthieu Latapy
pour le DEA d'Algorithmique
ENS Paris - Paris 6

Fabien Viger
fabien.viger@normalesup.org

12 novembre 2004

Résumé

La génération de graphes aléatoires dont les sommets ont des degrés préalablement fixés a déjà suscité un travail important, aussi bien de la part des informaticiens et des mathématiciens que des physiciens et des sociologues.

Le problème devient complexe quand le graphe est *orienté*, ou quand on le veut *connexe*. Nous présentons ici une amélioration d'un algorithme [1] de génération aléatoire dans le cas connexe, et une variante permettant de réduire la complexité de quadratique à linéaire.

Table des matières

Chapitre 1

Introduction

1.1 Motivations

L'étude des grands réseaux d'interactions prend de plus en plus d'importance, avec l'apparition des réseaux de télécommunications et le progrès des techniques d'analyse, couplé avec l'augmentation de la puissance de calcul des ordinateurs. Ces réseaux sont de natures variées : biologique (interactions protéiques, rapports prédateur-proie d'un ecosystème), sociale (collaboration, co-authoring), physique (réseau des lignes à haute tension, Internet), virtuelle (World Wide Web) ...

Pour les modéliser, on a recours à la structure formelle de *graphe* : un ensemble de sommets, reliés ensemble par des arêtes. On dispose ainsi d'un objet simple, permettant d'étudier tout une variété de systèmes complexes. On peut enrichir la structure canonique du graphe en assignant une couleur, ou un poids, aux arêtes ou aux sommets du graphe. On pourra ainsi modéliser le réseau routier en tenant compte du nombre de voies de chaque route, ou de la taille des villes. Qu'il soit enrichi ou non, on appelle *graphe réel* tout graphe issu d'un réseau présent dans la réalité.

Il est souvent difficile de recueillir un graphe réel dans sa totalité [20]. On préfère souvent mesurer les propriétés locales d'un réseau, puis on génère un graphe respectant au mieux ces propriétés. On parle alors de graphe *réaliste*. La génération de graphes permet également de faire des simulations, comme pour experimenter l'effet résultant de la variation de la topologie du réseau. Par exemple, beaucoup de travaux sur la robustesse aux pannes de l'Internet se basent sur des graphes générés.

La génération de graphes est un problème crucial dans le développement des réseaux : actuellement, on ne sait pas encore générer de graphe vraiment réaliste, et les modèles les plus proches de la réalité demandent un temps de calcul trop important.

1.2 Propriétés observées

Un graphe généré doit correspondre le mieux possible au graphe réel dont il est inspiré : on mesure certaines grandeurs sur le graphe réel, qu'on impose ensuite lors de la génération.

Définitions

. La **distance** entre deux sommets est la longueur du plus court chemin (en nombre d'arêtes) reliant ces deux sommets. La distance moyenne est la moyenne sur tous les couples de sommets du graphe.

. Le **clustering** du graphe est la probabilité que deux voisins[1] d'un sommet, pris au hasard, soient eux-mêmes voisins

. Le **degré** d'un sommet est le nombre d'arêtes liées à ce sommet. Le degré moyen est appelé *densité* du graphe. La **séquence** de degrés est la donnée des degrés de chaque sommet.

La plupart des graphes présents dans la nature, bien que d'origines différentes, ont des caractéristiques communes :

- une faible densité : le degré moyen est asymptotiquement constant, i.e. ne dépend pas de la taille du graphe
- une faible distance moyenne entre les sommets, en général de l'ordre du logarithme du nombre de sommets
- une distribution de degrés en loi de puissance (on parle de graphe sans-échelle), impliquant notamment la présence de quelques sommets de très grand degré.
- un clustering fort, i.e. asymptotiquement constant en la taille du graphe.

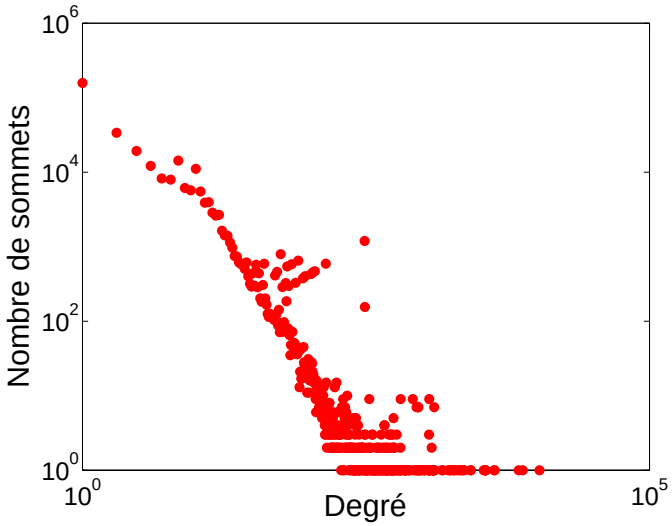

FIG. 1.1 – Distribution de degrés sur un sous-graphe du Web

1.3 Modèles

Différents modèles existent, prenant en compte une ou plusieurs des quantités citées ci-dessus.

[1] un sommet est dit voisin d'un autre quand ils sont reliés par une arête

. Le modèle d'Erdös et Rényi [17], qui place un nombre prédéfini d'arêtes au hasard sur l'ensemble des arêtes possibles, respectant ainsi la densité du graphe pris en modèle, produit des graphes aléatoires avec un faible diamètre. En revanche la distribution des degrés est en loi de Poisson : tous les sommets ont donc un degré similaire, et nous ne retrouvons pas le nombre important de sommets de tres faible degré, ni les sommets de très fort degré caractéristiques des lois de puissance (et donc des graphes réels)
. Molloy et Reed ont élaboré un modèle [3, 4] qui permet de génerer un graphe aléatoire ayant la distribution de degrés voulue. L'avantage de ce modèle est que sa simplicité permet encore de nombreuses analyses formelles. Mais il ne correspond pas encore à la réalité, notamment à cause de son faible clustering.
. Le modèle d'Albert et Barab'asi [18], où les sommets s'ajoutent un à un en s'attachant préférentiellement aux sommets de plus fort degré, engendre des graphes ayant une distribution de degrés en loi de puissance. Certaines variantes permettent d'obtenir en plus un fort clustering [19].

Modèle	Densité	Diamètre	Sans-Échelle	Clustering
Réalité	**faible**	**petit**	**oui**	**fort**
Erdös-Rényi	faible	petit	non	très faible
Molloy-Reed	faible	petit	oui	faible
Albert-Barab'asi	faible	petit	oui	faible

TAB. 1.1 – Caractéristiques des différents modèles

Dans la suite, nous nous attachons particulièrement au modèle de Molloy et Reed, car il a deux avantages importants. Tout d'abord, c'est un modèle de graphe *aléatoire* au sens fort : le graphe généré est tiré aléatoirement sur l'espace des graphes vérifiant la propriété voulue, avec probabilité uniforme. De plus, il respecte fidèlement la séquence de degrés prescrite. Or la distribution de degrés contient beaucoup d'information en elle-même, et implique souvent bon nombre de propriétés inhérentes au graphe considéré [10]. Ce modèle a été largement utilisé par le passé, donnant lieu a de nombreux résultats, et ses applications sont nombreuses [9].

Cependant, la version native de ce modèle a le défaut de produire des graphes qui ne sont pas forcément simples[2] , ni connexes. Or c'est le cas de la plupart des graphes observés dans la réalité. Notons que le modèle ER produit des graphes simples, et que le modèle AB produit des graphes simples et connexes. Des variantes permettant d'imposer la simplicité et la connexité à un graphe MR existent, soit en modifiant la séquence de degrés, soit au prix d'une complexité quadratique en la taille du graphe, et ne permettent donc pas la génération de très grands graphes (typiquement, de taille supérieure à 10^6).

Nous présenterons dans un premier temps un état de l'art en génération de graphe aléatoire, simple, connexe, et à séquence de degrés fixée. Puis nous mettrons en place

[2]Un graphe est dit *simple* quand il n'a pas de multi-arêtes (plusieurs arêtes reliant les deux mêmes sommets) ni boucles (arêtes liant un sommet à lui-même)

une analyse formelle de l'algorithme utilisé, avant de proposer une amélioration significative de celui-ci. Enfin, nous proposerons une dernière optimisation permettant de passer d'une complexité quadratique à une complexité linéaire.

Chapitre 2

État de l'Art

2.1 Le modèle de Molloy et Reed

La génération d'un graphe suivant l'algorithme de Molloy et Reed est un random-matching sur les arêtes du graphe (voir Fig. 2.1)

Algorithme

. Donner à chaque sommet le bon nombre de demi arêtes
. Numéroter toutes les demi arêtes de 1 à m.
. Relier la première demi arête à une autre, prise au hasard parmi les restantes, puis faire de même avec la deuxième arête (sauf si elle a déjà été prise), et ainsi de suite.

La somme des degrés doit être paire, puisqu'elle est égale à $2m$. La génération se fait en temps linéaire ; elle est aléatoire uniforme sur l'espace des graphes ayant la séquence de degrés voulue.

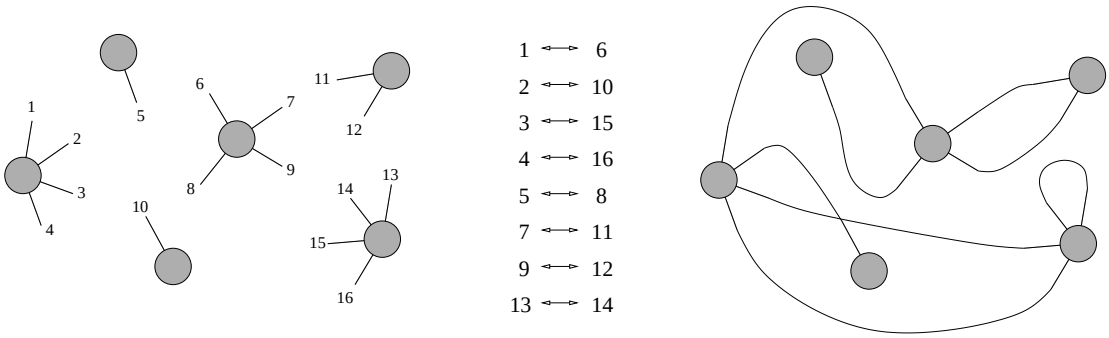

FIG. 2.1 – Algorithme de Molloy et Reed pour la génération d'un graphe 4-1-2-4-1-4

Pour que le graphe généré ainsi soit simple et connexe, la pratique actuelle consiste à éliminer les mutli-arêtes, les boucles, et à ne garder que la composante connexe géante. Cela engendre des déviations :

– Le graphe n'a plus le nombre de sommets voulu, ni la bonne séquence de degrés

– La distribution de degrés est biaisée. En enlevant les multi-arêtes et les boucles, on a surtout diminué les très grands degrés. En ne gardant que la composante connexe géante, on a aussi diminué sensiblement la proportion de sommets de faible degré (plus aptes à rester isolés).
– Le graphe lui-même n'est plus aléatoire au sens strict, i.e. tiré uniformément dans l'espace des graphes vérifiant les propriétés voulues

2.2 Génération aléatoire de graphes simples, connexes et à séquence de degrés prescrite

De nombreuses variantes ont été conçues pour pallier aux défauts décrits ci-dessus. Nous présentons dans cette section la méthode décrite dans un article de Gkantsidis et al. [1], qui nous a paru la plus performante. Une étude récente [2] a elle aussi élu cette méthode de génération, qui se fait en trois étapes :

1. Réaliser un graphe simple vérifiant la bonne séquence de degrés
2. Le rendre connexe, en le gardant simple et sans changer ses degrés.
3. Le mélanger pour le rendre aléatoire, en gardant toutes ses propriétés.

La troisième étape est cruciale pour supprimer le biais du graphe créé par les deux premières étapes.

2.2.1 Réalisation d'une séquence de degrés

La méthode de Molloy et Reed générait des graphes selon n'importe quelle séquence de degrés de somme paire. Quand on veut obtenir un graphe simple, une séquence n'est pas toujours réalisable, i.e. on ne peut pas toujours trouver un graphe simple satisfaisant la séquence de degrés prescrite. Par exemple la séquence 3,3,1,1 n'est pas réalisable.

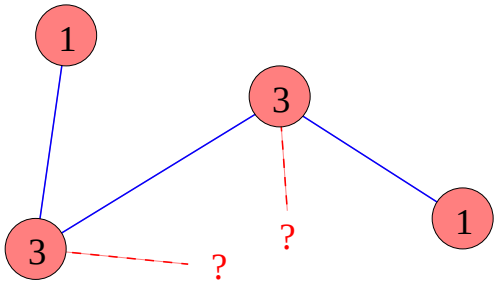

FIG. 2.2 – Non-réalisabilité du graphe 3-3-1-1

Le théorème d'Erdös-Gallai [8, 11] règle la question de la réalisabilité grâce à une démonstration constructive produisant un graphe selon une séquence de degrés donnée, l'échec de la méthode impliquant que la séquence n'est pas réalisable. Cette méthode, connue sous le nom d'algorithme de Havel-Hakimi [7, 6], est la suivante :

Algorithme (Havel-Hakimi, 1955)

. Attribuer à chaque sommet le bon nombre de demi arêtes
. Brancher les sommets un à un, dans l'ordre qu'on veut. Pour brancher un sommet, on lie ses demi arêtes avec les demi arêtes encore libre des sommets de plus haut degré *résiduel*, i.e. ayant le plus de demi arêtes libres.

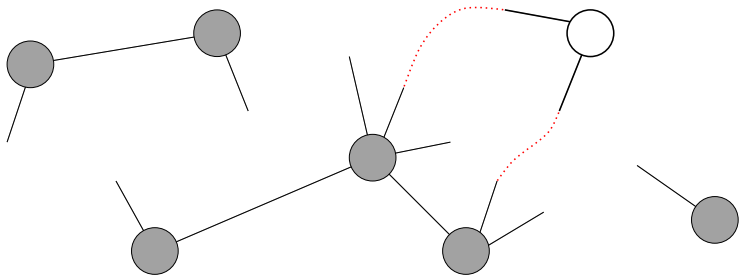

FIG. 2.3 – Branchement d'un sommet (en blanc) sur les sommets de plus haut degré résiduel. Les nouvelles arêtes sont en pointillé.

Théorème (Erdös-Gallai, 1960)

L'algorithme de Havel-Hakimi réussit si et seulement si la séquence de degrés est réalisable.

Théorème

Le complexité de l'algorithme de Havel-Hakimi est linéaire.

Démonstration

Nous présentons pour preuve notre implémentation de l'algorithme. La principale difficulté vient du branchement d'un sommet qui doit se faire en temps $O(d)$, où d est le degré du sommet à brancher.
L'idée est maintenir un tableau T des sommets disponibles, groupés par degré. $T[d]$ sera donc la liste des sommets de degré résiduel d.
Brancher un sommet de degré d se fait donc simplement en l'enlevant de $T[d]$ (puisqu'il n'est plus disponible), puis en le liant aux d sommets de plus haut degré de T (obtenus en parcourant $T[d_{max}]$ puis $T[d_{max}-1]$ etc. . .) Ces sommets voyant leur degré résiduel diminuer de 1, ils sont retirés de leur liste pour être ajoutés à la liste des sommets de degré inférieur.
Chaque liaison se fait en temps constant, assurant un coût total linéaire en la taille du graphe.

On a donc une génération en temps linéaire, mais qui n'est pas aléatoire uniforme sur l'espace des graphes ayant les propriétés voulues : séquence de degrés respectée et simplicité.

2.2.2 Une opération élémentaire degré-invariante

Introduisons l'*échange élémentaire* d'arêtes : étant données deux arêtes (a, b) et (c, d), l'échange de ces arêtes consiste à les remplacer par les arêtes (a, d) et (c, b). Cet échange ne modifie pas les degrés des sommets, et se fait en temps constant.

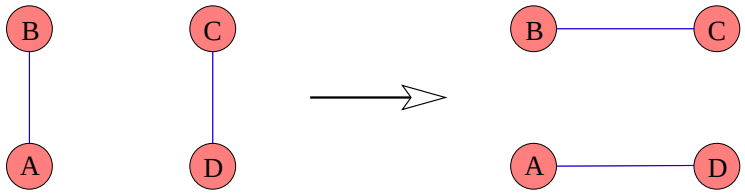

FIG. 2.4 – Échange élémentaire d'arêtes

Si on traite un graphe simple, il est utile de ne considérer que les échanges conservant toutes les propriétés du graphe. Pour que l'échange soit *valide*, le graphe doit rester simple : on ne considère donc que les couples $((a, b), (c, d))$ tels que a, b, c, d soit deux à deux distincts et tels que (a, d) et (c, b) ne soit pas des arêtes du graphe. Ce test est en temps constant également.

2.2.3 Connexion

Le graphe obtenu après la première étape n'est pas nécessairement connexe, en fait il n'est (en pratique) presque *jamais* connexe pour une distribution de degrés réelle. Il est donc nécessaire de le rendre connexe.

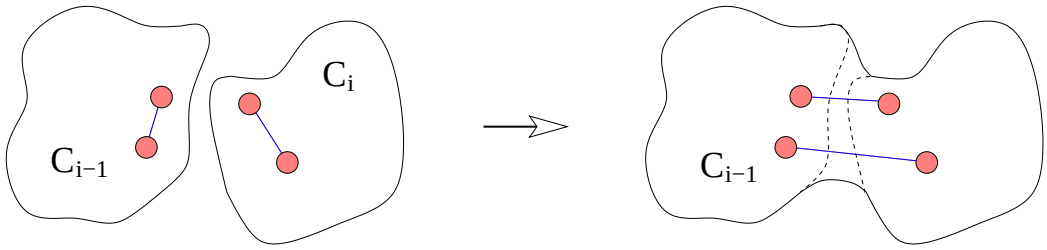

FIG. 2.5 – Fusion de deux composantes connexes par échange d'arêtes superflues

Algorithme

. Décomposer le graphe en composantes connexes $C_0, \ldots, C_i$, $T_0, \ldots, T_j$ où les C_k sont les composantes connexes contenant au moins un cycle, et les T_k les autres. Notons que chaque C_k a au moins une arête *superflue*, i.e. qu'on peut supprimer sans déconnecter la composante, et que les T_k sont des arbres.

. Fusionner C_i avec C_{i-1} en échangeant une arête superflue de l'une avec une arête superflue de l'autre (Fig. 2.5). Les deux arêtes créées connectent les deux composantes, l'échange produit donc une composante connexe C'_{i-1} gardant au

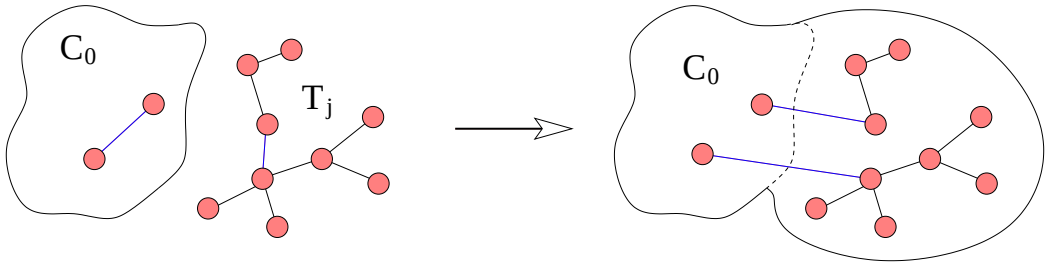

FIG. 2.6 – Fusion de la composante connexe C_0 avec un arbre T_j

moins une arête superflue (l'une des deux arêtes crées). On peut donc continuer ainsi jusqu'à n'obtenir qu'une seule composante connexe C_0.

. Fusionner C_0 avec les T_k : On fusionne C_0 avec T_j en échangeant une arête superflue de C_0 avec une arête quelconque de T_j. Comme le montre la Fig. 2.6 la nouvelle composante C_0 qui en résulte est connexe.

Théorème L'algorithme ne change pas les degrés du graphe. Il le rend connexe si les deux conditions suivantes sont vérifiées [1] :

1. Le graphe doit avoir au moins $n-1$ arêtes (où n est le nombre de sommets)
2. Aucun sommet ne doit être isolé, i.e. la séquence de degrés ne doit pas contenir de 0

Preuve

Les deux premières étapes de l'algorithme ne posent pas de problèmes. Pour la troisième étape, il faut simplement vérifier que les fusions entre C_0 et les arbres sont possibles jusqu'au bout. D'une part, les T_k ont au moins une arête (condition 2 : aucun sommet n'est isolé). D'autre part, si la fusion n'est pas possible c'est que C_0 n'a pas d'arêtes superflue, et donc C_0 est un arbre. Cela contredit la condition 1 car le nombre d'arêtes d'un arbre de taille K est toujours égal à $K-1$. Il est donc impossible que plusieurs arbres totalisent n sommets et au moins $n-1$ arêtes. On peut donc continuer jusqu'à n'obtenir qu'une seule composante connexe C_0. Notons que tous les échanges considérés ici sont valides (laissent le graphe simple) car on fusionne des composantes connexes séparées.

L'algorithme termine donc avec succès et on obtient une unique composante connexe. Il se fait en temps et place linéaires en la taille du graphe : dans la première phase, on calcule les C_k et les T_k, en mémorisant les arêtes superflues des C_k, et dans la deuxième phase, chaque fusion est en temps constant. On sait donc générer un graphe simple connexe avec séquence de degrés donnée en temps linéaire.

[1] Noter que ces deux conditions sur la séquence de degrés sont nécéssaires pour tout graphe connexe.

2.2.4 Mélange

Pour que la génération soit aléatoire, on doit mélanger le graphe connexe obtenu. On applique le processus suivant : on choisit deux arêtes au hasard parmi les m arêtes du graphe ; si l'échange de ces deux arêtes est *valide* et s'il laisse le graphe connexe, on fait l'échange, sinon on ne change rien ; puis on recommence. On a là une chaîne de Markov sur l'espace C des graphes connexes ayant la séquence de degrés donnée au départ, dont les transitions sont les échanges d'arêtes valides.

Définition Une chaîne de Markov (et par extension, sa matrice) est dite *irréductible* quand tout état est atteignable à partir de n'importe quel état initial.

Définition Une chaîne de Markov (et par extension, sa matrice) est dite *apériodique* quand dans le graphe orienté (V,A) associé à la chaîne (les sommets sont les états, les arcs sont les transitions réalisables), on a :

$$\bigwedge_{v \in V} \left(\bigvee_{C \ cycle | v \in C} |C| \right) \ = \ 1 \tag{2.1}$$

où $\bigvee$ et $\bigwedge$ désignent le ppcm et le pgcd, et $|C|$ est la taille du cycle C

Théorème

La matrice M de transition associée à notre chaîne de Markov vérifie :

- (**i**) Ses éléments non diagonaux valent soit 0, soit $\frac{2}{m(m-1)}$
- (**ii**) Elle est symétrique
- (**iii**) Elle est irréductible
- (**iv**) Elle est apériodique

Démonstration

(**i**) L'élément M_{ij} est la probabilité que l'on passe du graphe G_i au graphe G_j. Si il existe un échange d'arêtes permettant de transformer G_i en G_j, alors c'est le seul : aucun autre échange ne peut produire le même résultat. La probabilité de choisir cet échange est $M_{ij} = \frac{2}{m(m-1)}$: il s'agit de choisir la bonne paire d'arêtes à échanger. Sinon c'est qu'aucun échange ne marche, et alors $M_{ij} = 0$.

(**ii**) est immédiat par (**i**) et par réversibilité de tout échange d'arêtes : si un échange transforme G_i en G_j, alors l'échange inverse transforme G_j en G_i.

(**iii**) est un résultat de Taylor [5] : on peut passer de n'importe quel graphe connexe à n'importe quel autre ayant les mêmes degrés par des échanges élémentaires. Notons que le résultat est vrai aussi sans contrainte de connexité (facile).

(**iv**) vient du fait que les éléments diagonaux m_{ii} sont strictement positifs : en effet pour chaque graphe il existe un couple d'arêtes définissant un échange non valide (prendre deux arêtes adjacentes à un même sommet). La transition $G_i \to G_i$ a donc une probabilité supérieure à $\frac{2}{m(m-1)}$. Un résultat standard de la théorie des matrices positives et irréductibles assure alors que notre matrice est apériodique

Corollaire La chaîne de Markov converge vers la distribution uniforme sur l'ensemble de ses états. (théorie standard des chaînes de Markov)

Il suffit donc de poursuivre le mélange assez longtemps pour rendre le graphe "aussi aléatoire que l'on veut". Le point faible de cette méthode est qu'on a aucun résultat théorique sur la vitesse de convergence. Toutefois les études approfondies menées indépendamment par Gkantsidis et al. [1] et Milo et al. [2] montrent qu'un nombre d'itérations de l'ordre de m est empiriquement suffisant :

Résultat empirique Il suffit d'itérer la chaîne de Markov $10|G|$ fois pour avoir une convergence satisfaisante, i.e. un résultat "suffisament aléatoire". Les expériences n'ont montré aucune déviation statistique entre des graphes mélangés pendant $10|G|$ itérations et des graphes mélangés plus longtemps.

À titre indicatif, un résultat de T.G. Will [12] montre que la distance entre deux graphes (sans la contrainte de connexité), c'est-à-dire le nombre d'échanges élémentaires nécéssaire pour passer de l'un à l'autre, ne dépasse pas m. Le principal inconvénient du processus est de fait la complexité, car chaque transition se fait en $O(m)$ à cause du test de connexité, faute de technique permettant de maintenir la connexité de manière dynamique. La coût du mélange est donc quadratique.

2.3 Accélération

Gkantsidis et al. [1] proposent une méthode simple d'accélération de ce processus. Au lieu de faire un test de connexité après chaque échange élémentaire, on va effectuer T échanges, puis tester la connexité. Si alors le test échoue, on annule les T échanges. T est appelée la *fenêtre*. Ils ne précisent pas pourquoi il reviennent en arrière plutôt que d'espérer que le graphe se reconnecte plus tard, mais nous l'expliquerons nous-mêmes par la suite (Chapitre 5). Précisons qu'au cours des tests que nous avons menés, jamais un graphe qui s'était déconnecté ne s'est reconnecté "par chance".

Entre deux tests de connexité on a toujours une chaîne de Markov, mais sur l'espace E des graphes simples pas forcément connexes (toujours avec la bonne séquence de degrés). Mais l'état initial est un graphe connexe, et l'état final également (s'il ne l'est pas, on revient à l'état initial). Comme la fenêtre T est prédéterminée, on peut considérer ces T échanges comme une seule transition d'une chaîne de Markov A_T sur l'espace C des graphes vérifiant toutes les propriétés voulues, connexité comprise.

Théorème

Les chaînes de Markov A_T vérifient les conditions (**ii**), (**iii**) et (**iv**) du théorème précédent (section 2.2.4)

Demonstration

(**ii**) La probabilité $P_{G_i \to G_f}$ d'arriver à l'état final $G_f \in C$ en partant de l'état initial $G_i \in C$ après T échanges est la somme des probabilités des chemins $G_i \to G_1 \to \cdots \to G_{t-1} \to G_f$, où $G_1 \cdots G_{t-1} \in E$. Un chemin et son symétrique

étant équiprobables (grâce à la symétrie de la chaîne de Markov sur l'espace E), on a bien $P_{G_i \to G_f} = P_{G_f \to G_i}$ pour tout $(G_i, G_f) \in C^2$.

(**iii**) L'irréductibilité est conservée à fortiori, puisqu'on ne fait qu'élargir les possibilités de transitions.

(**iv**) L'apériodicité est elle aussi conservée à fortiori puisque $\forall G \in C, P(G \to G) > 0$ (il suffit de ne faire que des échanges invalides, et donc non effectués, pour rester sur le même état).

Corollaire

La concaténation des chaînes A_{T_0}, A_{T_1}, ... converge vers la distribution uniforme.

Démonstration

La théorie des chaînes de Markov assure qu'une concaténation de chaînes de Markov vérifiant les conditions (**ii**), (**iii**) et (**iv**) converge vers la distribution uniforme, si on les prend dans un ensemble fini. Pour cela, on peut borner T par T_{max} (typiquement $T_{max} = 10|G|$). On a alors un ensemble fini de chaînes de Markov $\{A_T \mid T \leq T_{max}\}$.

Cela n'a pas d'incidence néfaste sur l'algorithme car pour des valeurs de T de l'ordre de $|G|$, le coût des tests de complexité devient minime comparé au coût des échanges d'arêtes.

Le processus accéléré converge donc toujours vers la distribution uniforme. Il faut bien choisir la fenêtre T, car si elle est trop grande le graphe aura de fortes chances de s'être déconnecté après les T échanges, que l'on devra annuler. L'heuristique utilisée par Gkantsidis et al., schématisée dans la Fig. 2.7, consiste à ajuster la fenêtre dynamiquement, comme pour la fenêtre du protocole TCP. Si les T échanges ont abouti à un graphe connexe, on incrémente la fenêtre : $T := T + 1$. Si au contraire le graphe a été déconnecté, on divise la fenêtre par deux : $T := T/2$

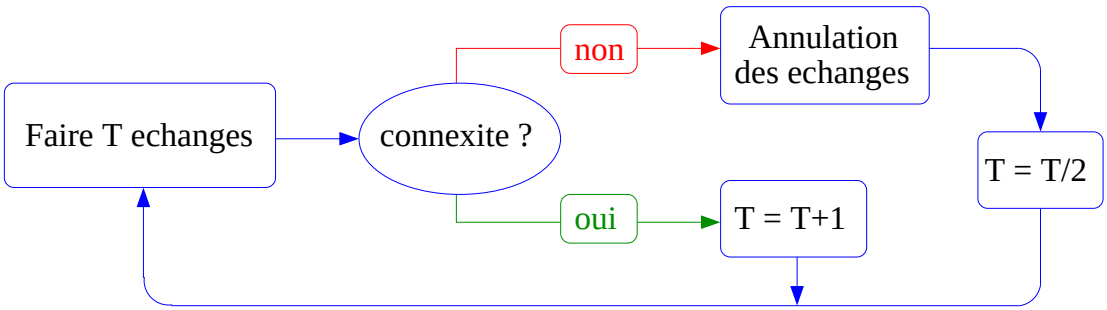

FIG. 2.7 – Heuristique d'ajustement de la fenêtre de Gkantsidis et al.

Ainsi, T s'adapte automatiquement à une valeur convenable. Toutefois les auteurs ne poussent pas l'étude plus loin, cette accélération très efficace permettant une division déjà considérable de la complexité par **T**. Le complexité de la génération de

graphes, par leur méthode, reste néanmoins quadratique. Dans la suite, nous analyserons le comportement de leur heuristique d'auto-évaluation de la fenêtre, puis nous proposerons une amélioration appréciable.

Chapitre 3

Analyse de l'accélération

3.1 Définir la fenêtre idéale

Introduisons la *probabilité de déconnexion* p, mesurant à une étape donnée la probabilité qu'un échange élémentaire déconnecte le graphe. Cette probabilité varie avec le graphe, toutefois il est légitime de la supposer constante au cours de T échanges, en première approximation (pourvu que T soit petit). De p on déduit donc la probabilité P que les T échanges laissent le graphe connexe :

$$P = (1-p)^T \tag{3.1}$$

P est appelé le *taux de réussite*.

Définissons à présent le *rendement* comme le rapport entre échanges efficaces effectués et le nombre de tests de connexité. Les échanges annulés par un retour en arrière (à cause d'une déconnexion) ne sont pas comptabilisés. Le mélange du graphe consistant en un nombre prédéfini d'échanges élémentaires à effectuer, le rendement mesure la vitesse de l'algorithme, puisque la quasi-totalité du temps de calcul est monopolisée par les tests de connexité. Dans la suite, on va considérer le *rendement instantané* θ de l'accélération, défini par l'espérance du nombre d'échanges efficaces pour **un** test de connexité. Sachant qu'à une étape donnée on effectue T échanges, qui seront efficaces seulement si le graphe reste connexe :

$$\theta \;=\; T \cdot P \;=\; T \cdot (1-p)^T \tag{3.2}$$

La probabilité de déconnexion p dépendant du graphe, à nous donc d'optimiser la fenêtre T pour obtenir le meilleur rendement possible. Cet optimum est atteint pour $T = \frac{1}{p}$, ce qui équivaut à un taux de réussite $P = \frac{1}{e}$.

Le rendement maximal est donc

$$\theta_{max} = \frac{1}{pe} \tag{3.3}$$

3.2 Étude dynamique

Pour évaluer l'efficacité de l'heuristique de Gkantsidis et al. qui rappelons-le détermine la fenêtre T dynamiquement en fonction du résultat des T échanges précédents,

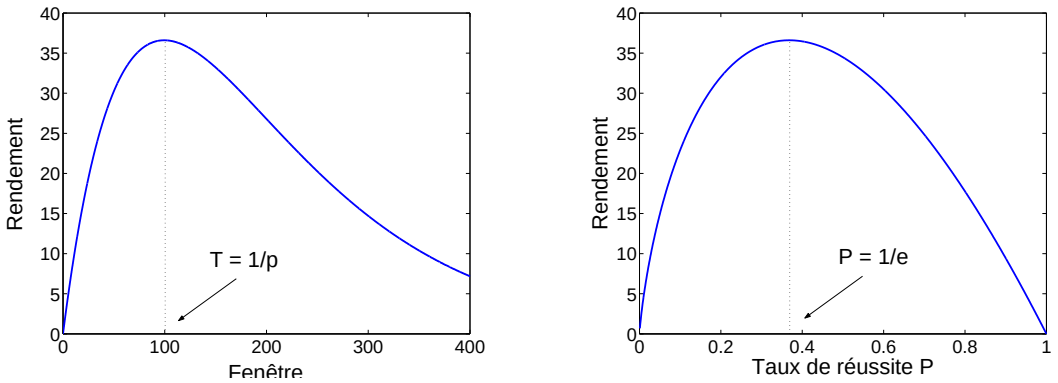

FIG. 3.1 – Rendement en fonction de la fenêtre T (à gauche) ou du taux de réussite P (à droite). Valeur utilisée : $p = 1\%$

nous allons tenter de déterminer la valeur que prend T avec leur heuristique.

Pour cela, introduisons la notion d'*équilibre*. Si T est trop grande, la probabilité que le graphe le graphe se déconnecte sera grande. Elle va donc diminuer. Inversement elle va augmenter si elle est trop faible, et que le graphe a une faible probabilité de se déconnecter. On peut ainsi supposer que T va "converger" vers une valeur d'équilibre, déterminée uniquement par la probabilité de déconnexion p. Le terme "converger" est abusif dans leur dynamique puisqu'en cas d'échec T est divisée par deux, mais on peut tout de même espérer que T oscillera autour d'une valeur moyenne $\mathbf{T}$ appelée fenêtre à l'*équilibre*.[1]

Dans le processus de Gkantsidis et al. on peut évaluer $\mathbf{T}$ en considérant qu'à l'équilibre on a une fenêtre $\mathbf{T}$, et que l'espérance de la fenêtre à la prochaine étape vaudra aussi $\mathbf{T}$ (puisqu'on est à l'équilibre). Ainsi,

$$\mathbf{T} \;=\; \mathbf{P}(\mathbf{T}+1) + (1-\mathbf{P})\frac{\mathbf{T}}{2} \tag{3.4}$$

Pour p petit (i.e. inférieur à 5%), on en déduit :

$$\mathbf{T} \;\sim\; \sqrt{\frac{2}{p}} \tag{3.5}$$

Ce raisonnement qualitatif a le mérite de nous donner le bon ordre de grandeur, puisqu'on peut montrer rigoureusement le résultat asymptotique suivant :

$$\forall \epsilon > 0, \quad \mathbf{T} = o(p^{-1/2-\epsilon}) \quad \text{quand } p \to 0 \tag{3.6}$$

.

On est bien loin de l'optimal $\mathbf{T} = \frac{1}{p}$. En fait on obtient en rendement de l'ordre de $\sqrt{\theta_{max}}$, donc pouvant être bien plus faible que le rendement maximal. Cela met

[1] Dans la suite, une grandeur en **gras** indiquera qu'on parle de la valeur moyenne de cette grandeur à l'équilibre.

en évidence un manque à gagner certain pour cette heuristique : quand le rendement pourrait être excellent, il est seulement moyen. De plus, le principe de "linear increase" de T pêche par son manque d'adaptabilité : si on sous-évalue la fenêtre ou si p diminue significativement au cours du mélange, l'équilibre peut être atteint très tard.

3.3 Mesure de la probabilité de déconnexion

Nous avons mis en place une mesure de la probabilité de déconnexion du graphe. Cette mesure a pour but de comparer la fenêtre T obtenue par notre heuristique avec la fenêtre optimale théorique $1/p$. Ne sachant pas estimer cette probabilité de manière exacte en moins de $O(|G|^3)$, nous avons opté pour une mesure approchée de type Monte-Carlo : nous appliquons l'algorithme de mélange, mais en revenant au graphe de départ après chaque test de connexité. L'algorithme va donc tenter de mélanger le graphe en effectuant un certain nombre T d'échanges, variant selon l'heuristique développée plus haut, et mémoriser le succès ou l'échec de ces mélanges.

En procédant ainsi sur N tests de connexité, nous disposons d'un panel de N mesures de type (nombre d'échanges effectués, réussite), que nous utilisons pour évaluer p par maximum de vraisemblance. Pour avoir une mesure précise il nous suffira de choisir N assez grand. La complexité de cette mesure est $O(N \cdot |G|)$.

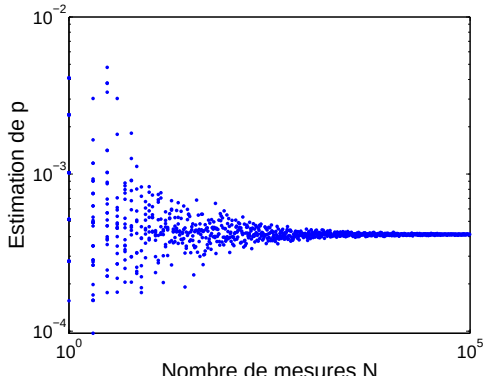

FIG. 3.2 – Convergence de la mesure de p sur un graphe. La distribution de degrés est en loi de puissance d'exposant 2.5

Pour valider cette méthode, nous avons effectué des tests de convergence en faisant grandir N (Fig 3.2). Il apparaît que 1000 mesures suffisent en général pour avoir une bonne estimation de la probabilité de déconnexion (à 10% près).

De cette manière, nous avons mesuré la probabilité de déconnexion au cours du mélange (Fig. 3.3). Cela nous permet de mieux connaître la dynamique de p, que nous avons supposé constante pour l'étude théorique, et de vérifier la validité de cette assertion.

Dégageons trois remarques sur l'evolution de p au cours du mélange :

– Le graphe initial a une probabilité de déconnexion plus grande que le graphe mélangé. Cela confirme que la méthode de génération est biaisée si l'on se prive

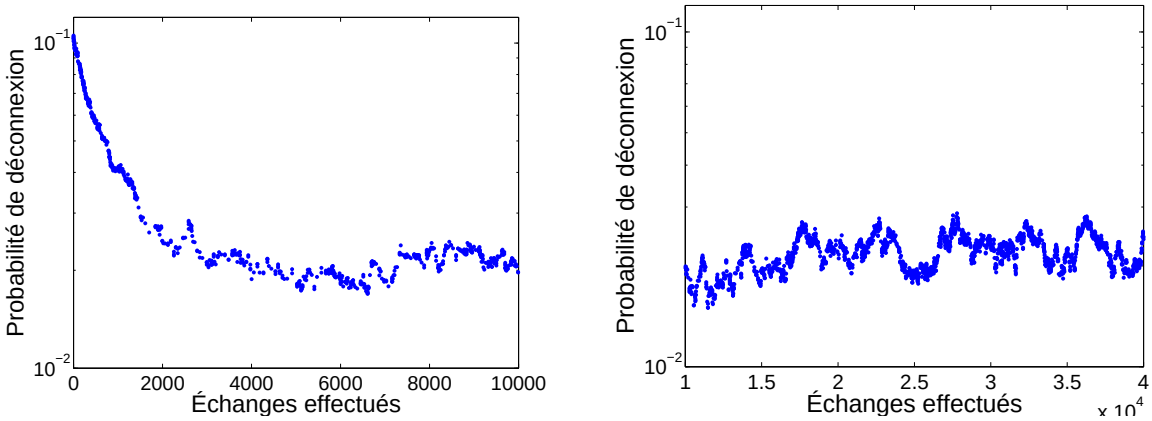

FIG. 3.3 – Évolution de la probabilité de déconnexion au cours du début du mélange d'un graphe (à gauche) et longtemps après le début (à droite)

du mélange.

- La probabilité de déconnexion atteint une valeur d'équilibre au bout de $|G|$ échanges (en ordre de grandeur). Cela vient appuyer les résultats de convergence de la chaîne de Markov déjà connus [1, 2].
- La probabilité de déconnexion fluctue même après convergence, oscillant d'autant plus qu'elle est faible. Elle peut être divisée ou multipliée par deux en relativement peu d'échanges. La notion d'équilibre est donc fragile, mais reste justifiée dans la mesure ou les oscillations de la probabilité restent centrées sur une certaine valeur limite, et ne changent pas son ordre de grandeur.

Ces observations nous conduisent à séparer le mélange en deux phases. Les $|G|$ premiers échanges constituent la première phase, pendant laquelle p est encore loin de sa valeur limite et le graphe est encore fortement biaisé. Puis vient la deuxième phase, pendant laquelle le graphe est suffisament mélangé pour que p ne subisse que des fluctuations de nature statistique.

Au cours de ce chapitre, nous avons introduit un modèle simple, basé sur la probabilité de déconnexion, permettant d'anticiper le rendement de l'heuristique de Gkantsidis et al. Nous avons également montré que ce rendement était loin d'être optimal. Puis nous avons vérifié empiriquement le fondement de notre modèle. Il nous paraît à présent nécessaire de définir une autre heuristique plus performante, en s'appuyant sur nos résultats formels.

Chapitre 4

Une nouvelle heuristique

4.1 Description

Notre heuristique est la suivante : en cas de réussite, T sera multipliée par un facteur constant $1 + q^+$, et en cas d'échec T sera multipliée par un autre facteur constant $1 - q^-$. Bien sûr, q^+ et q^- sont strictement positifs. Nous espérons ainsi rééquilibrer l'évolution de T, qui dans l'ancienne heuristique diminuait beaucoup plus vite qu'elle n'augmentait, et dans le même temps garder une grande adaptabilité permettant une convergence rapide de T vers sa valeur d'équilibre. Reste à choisir q^- et q^+ pour obtenir un rendement optimal.

Si q^+ et q^- sont pris assez petits, l'approximation donnée en (Eq 3.2) devient tout à fait valable, et à l'équilibre on a donc :

$$\mathbf{T} \;=\; \mathbf{P} \cdot (1 + q^+)\mathbf{T} \;+\; (1 - \mathbf{P}) \cdot (1 - q^-)\mathbf{T} \tag{4.1}$$

Cette fois on obtient :

$$(1-p)^{\mathbf{T}} \;=\; \mathbf{P} \;\sim\; \frac{q^-}{q^+ + q^-} \;=\; (1 + \frac{q^+}{q^-})^{-1} \tag{4.2}$$

Pour obtenir un rendement optimal, on a vu qu'il fallait avoir un taux de réussite $P = \frac{1}{e}$. On a donc une condition sur le rapport $\frac{q^+}{q^-}$, nécessaire et suffisante pour que la fenêtre d'équilibre soit la fenêtre optimale théorique :

$$\frac{q^+}{q^-} = e - 1 \tag{4.3}$$

En théorie, cette heuristique présente le double avantage d'une convergence *rapide* vers la fenêtre *optimale* pour le rendement (et donc la rapidité) du mélange.

4.2 Amplitude : un compromis difficile

Nous avons mis en évidence dans la section 4.1 une condition d'optimalité sur le rapport $\frac{q^+}{q^-}$. Il fixe en effet le rapport entre la vitesse à laquelle la fenêtre peut

augmenter et la vitesse à laquelle elle peut diminuer. Mais, à rapport fixé, ces deux vitesses peuvent être faibles, ou fortes. Pour quantifier cela, la moyenne géométrique $\sqrt{q^+q^-}$ peut convenir. Nous l'appelons *amplitude* car elle représente la vitesse (ou amplitude) de variation de la fenêtre T.

Fixer le rapport *et* l'amplitude revient à fixer q^+ et q^-. Notons au passage, comme $q^- < 1$ (sinon on multiplierait la fenêtre par $1 - q^- \leq 0$), on a nécessairement, pour un rapport valant $\frac{1}{e}$, une amplitude dans $]0, \sqrt{e-1}[$. Plus l'amplitude est faible, plus les oscillations de T autour de $\mathbf{T}$ seront faibles, garantissant un rendement quasi-optimal à l'équilibre. En contrepartie, une faible amplitude implique une plus faible adaptabilité en cas de variations de p

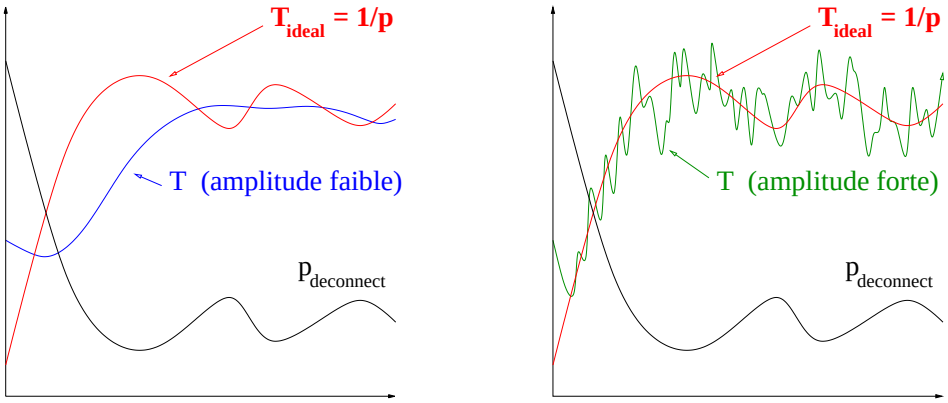

FIG. 4.1 – Schéma comparatif de la fenêtre idéale et de la fenêtre ajustée par l'heuristique au cours du mélange. À gauche, avec une faible amplitude, à droite avec une forte amplitude

Pendant la première phase, nous garderons une amplitude forte (supérieure à 5%) car la probabilité de déconnexion est encore loin de sa valeur limite. On peut ainsi s'adapter à sa diminution. Dans la deuxième phase, p fluctue autour de sa valeur d'équilibre. L'amplitude doit donc correspondre au meilleur compromis entre adaptabilité et stabilité, pour que T suive les variations de p sans trop osciller autour de sa valeur idéale.

Notons aussi que le rapport $\frac{q^+}{q^-}$ est en fait beaucoup plus libre qu'on pourrait le penser. Pour garder un rendement supérieur ou égal à 80% de son maximum théorique, on peut s'autoriser l'intervalle [0.5 , 5] autour de l'optimal $e-1 \approx 1.718$. La condition formelle d'optimalité est donc relativement faible en pratique. La discrétisation de T, qui doit prendre des valeurs entières, ou encore l'inexactitude du modèle peuvent avoir pour conséquence un déplacement substantiel du rapport optimal.

Il nous a donc semblé nécessaire de tester de nombreux couples de paramètres $(rapport, amplitude)$ pour la deuxième phase, qui représente l'essentiel du mélange. Nous ésperons ainsi valider notre approche théorique concernant le rapport $\frac{q^+}{q^-}$ et déterminer expérimentalement une bonne valeur pour l'amplitude. Les résultats sont tracés Fig. 4.3 .

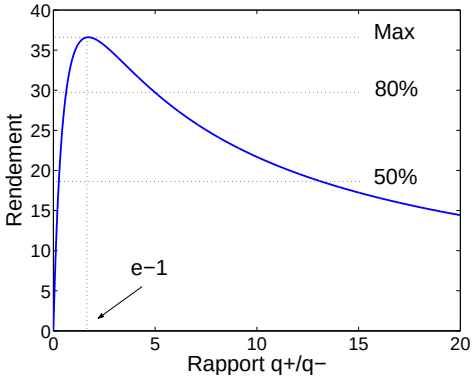

FIG. 4.2 – Rendement maximal théorique à l'équilibre en fonction de q^+/q^-

On obtient bien un rendement maximal pour un rapport proche de $e-1$. L'amplitude idéale semble être autour de 10^{-1}, ce qui semblait en théorie un bon compromis entre réactivité (T peut doubler en seulement 7 étapes) et stabilité (des oscillations de 10% autour du $\mathbf{T}$ optimal occasionnent une perte théorique de rendement inférieure à 1%). On remarque aussi que le choix est large : en faisant varier ces deux paramètres d'un facteur 2 autour de leur valeur optimale, le rendement reste supérieur à 80% de son maximum.

En conséquence, pour la plupart des graphes, nous utiliserons les paramètres suivants :

$$\frac{q^+}{q^-} = e - 1 \qquad \sqrt{q^+ q^-} = 0.1 \tag{4.4}$$

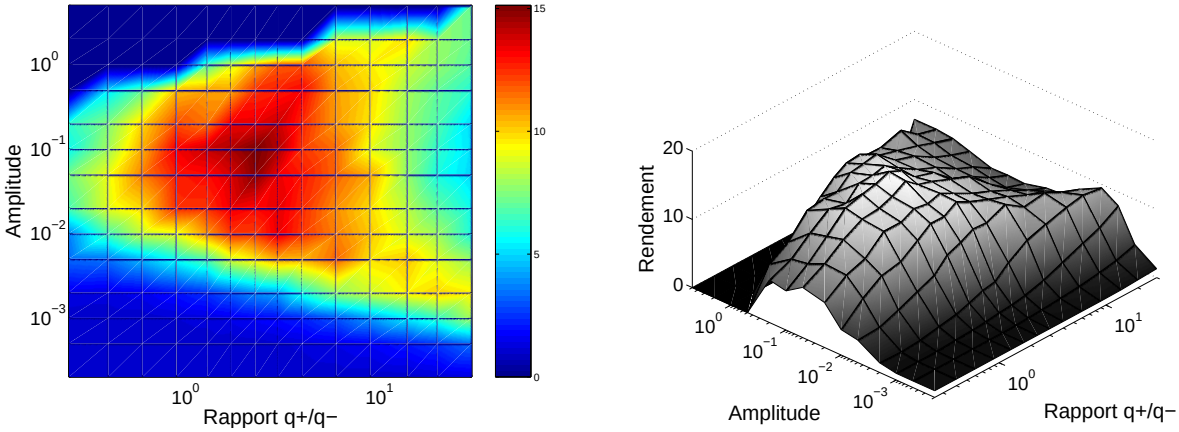

FIG. 4.3 – Rendement obtenu en fonction du rapport $\frac{q^+}{q^-}$ et de l'amplitude $\sqrt{q^+q^-}$ sur un graphe de 1000 sommets, 3500 arêtes, de degrés distribués en loi de puissance d'exposant 2.5

4.3 Comparaison avec l'heuristique optimale

Pour évaluer la performance de notre méthode avec les paramètres retenus plus haut, nous avons évalué empiriquement le rendement maximal théorique que l'on pouvait espérer au cours du mélange du graphe (nous parlons alors d'*heuristique optimale*). Pour ce faire, on se base sur l'éq. 3.3 : $\theta_{max} = \frac{1}{p\bar{c}}$ et sur l'évaluation de la probabilité de déconnexion p décrite précédemment. Au cours du mélange, on calcule p à chaque étape et on en déduire le rendement qu'on aurait pu obtenir en ajustant T à sa valeur optimale $1/p$.

Définition Le rapport entre le rendement obtenu et ce rendement idéal est appelé *facteur de qualité* de l'heuristique.

Le tableau ci-dessous (Tab. 4.1) résume les rendements obtenus pour différentes distributions de degrés, toutes en loi de puissance d'exposant $\alpha \in [2, 3]$, de tailles n et de degré moyen z. Les graphes sont mélangés $10m$ fois. Les rendements listés sont ceux obtenus avec l'heuristique de Gkantsidis et al. (θ_{Gk}), ceux obtenus avec notre heuristique (θ), et les rendements obtenus en ajustant T à sa valeur optimale (θ_{opt}).

n	z	θ_{Gk}	θ	θ_{opt}
10^4	2.1	0.79	0.88	0.90
10^4	3	3.00	5.00	5.19
10^4	6	20.9	112	117
10^3	6	26.4	215	227
10^4	20	341	35800	37000

TAB. 4.1 – Rendements mesurés lors du mélange de différents graphes

Notre heuristique apporte un progrès appréciable par rapport aux résultats obtenus avec l'heuristique de "linear increase, exponential decrease" utilisée par Gkantsidis et al. Le temps de calcul pour la génération aléatoire d'un graphe de 10^4 sommets, de degrés répartis en loi de puissance d'exposant 2.1 et de moyenne 10 passe de 24 secondes à 2 secondes avec notre implémentation.

Les rendements obtenus sont supérieurs à 95% du rendement qu'on obtiendrait avec la meilleure heuristique possible pour l'évaluation de T. On peut donc estimer que notre heuristique et les paramètres retenus (Eq. 4.4) sont satisfaisants.

Dans ce chapitre, nous avons mis en place une nouvelle heuristique, que nous avons ajustée, selon notre modèle formel, pour qu'elle tende vers le comportement optimal. Nous avons également comparé les résultats obtenus par notre heuristique avec les résultats d'une heuristique idéale. L'écart relatif étant inférieur à 5%, il paraît inutile de pousser plus loin d'éventuelles optimisations.

Chapitre 5

Freiner la déconnexion

Si on sait écarter d'office des échanges dont on sait qu'ils déconnecteront le graphe, on diminuera p, ce qui augmentera directement le rendement de l'accélération. Or les distributions de degrés des graphes réels sont en lois de puissance, et comportent donc une proportion importante de sommets de très petit degré, pouvant facilement former des petites composantes isolées au cours du mélange. C'est sans doute ce qui explique que nos graphes réalistes se déconnectent si facilement. Dans ce chapitre, nous allons nous pencher plus en détail sur cette question : la déconnexion vient-elle des sommets de faible degré ?

5.1 Déconnexion de paires isolées

Si au cours d'un échange on connecte ensemble deux sommets de degré 1, on obtient une paire isolée, et le graphe n'est plus connexe. Or les graphes réels ont souvent un nombre important de sommets de degré 1 (voir Tab. 5.1)

Degré	Web	Protéines	ARXIV	Internet	Acteurs
$= 1$	48%	48%	17%	57%	61%
≤ 2	58%	68%	36%	78%	73%
≤ 3	64%	79%	51%	85%	78%

TAB. 5.1 – Répartition des degrés dans des graphes réels

La probabilité q de créer une paire isolée lors d'un échange aléatoire sur un graphe de m arêtes ayant n_1 sommets de degré 1 est la probabilité de choisir deux des n_1 arêtes liées à un sommet de degré un et de les échanger manière à relier le deux sommets de degré un :

$$q = \frac{1}{2} \cdot \frac{\binom{n_1}{2}}{\binom{m}{2}} \approx \frac{n_1^2}{2m^2} \tag{5.1}$$

ce qui n'est pas négligeable pour les distributions citées ci-dessus : pour le graphe d'Internet par exemple on trouve $q = 8.25\%$.

On peut donc diminuer p à faible coût, puisque lors de l'échange le test "paire isolée" se fait en temps constant. Nous avons mesuré (Tab. 5.2), sur les distributions de degrés des graphes réels cités plus haut, la nouvelle probabilité de déconnexion en ne considérant comme valides que les échanges ne créant pas de paires isolées.

	Web	Protéines	ARXIV	Internet
Sans	0.0016	0.064	0.0025	0.097
Avec	0.00014	0.022	0.00037	0.041

TAB. 5.2 – Probabilité de déconnexion avec ou sans détection des paires isolées

Nous avons également refait les mesures du *facteur de qualité* (Fig 5.1), rapport entre rendement empirique et rendement maximal théorique, en faisant varier le rapport $\frac{q^+}{q^-}$ et l'amplitude $\sqrt{q^+q^-}$.

On obtient un rendement maximal pour un rapport de l'ordre de 15, ce qui est contraire à la théorie : si $\frac{q^+}{q^-} = 15$, le rendement ne devrait pas excéder 50% de sa valeur maximale (voir Fig. 4.1). De plus ce rendement est alors 3.5 fois plus grand que le maximum théorique ! Les résultats pratiques montrent donc une nette amélioration (d'un facteur 10 pour ce graphe) du rendement grâce à la prise en compte des paires isolées.

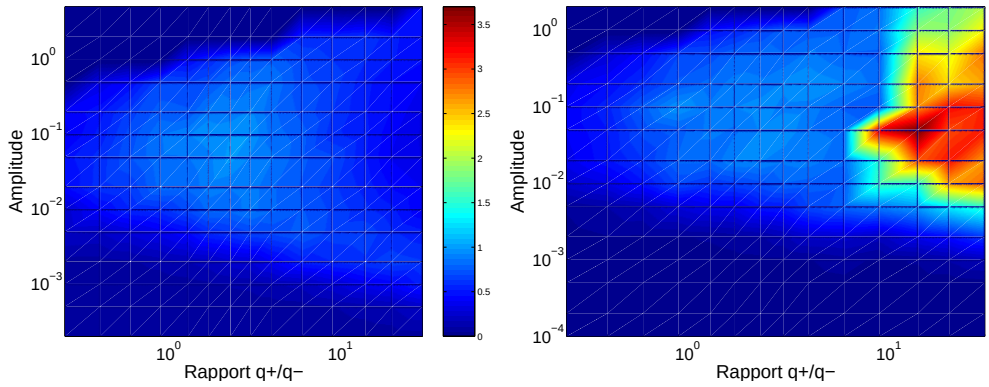

FIG. 5.1 – Comparaison des facteurs de qualité obtenus avec (à droite) ou sans (à gauche) prise en compte des paires isolées.

5.2 Reconnexion

L'explication des résultats surprenants cités ci-dessus est la reconnexion : en évitant la formation de paires isolées, on déconnecte moins facilement le graphe, qui se

reconnecte aussi plus facilement. On peut ainsi proposer une interpretation de la Fig. 5.1 :

. Pour de petits rapports q^+/q^-, la reconnexion est négligeable car la fenêtre T est trop petite pour que le graphe ait une chance significative de se reconnecter. Le facteur de qualité est donc le même que sans la prise en compte des paires isolées.

. Pour des rapports $q^+/q-$ importants, la fenêtre va être excessivement grande : on fait beaucoup d'échanges, mais à l'arrivée le graphe est presque toujours déconnecté. La reconnexion pourra alors s'exprimer, et augmenter le taux de réussite, et donc le facteur de qualité.

Ainsi, toute l'étude théorique menée aux chapitres 3 et 4 n'est plus valable puisque le taux de réussite doit à présent prendre en compte une possible reconnexion du graphe après déconnexion : l'heuristique ne fait plus tendre T vers sa valeur optimale.

Notons que le théorème vu en section 2.3 reste vrai malgré les déconnexions et reconnexions non détectées : on a toujours convergence de la concaténation de chaînes de Markov vers la distribution *uniforme* sur l'espace des graphes ayant la propriété voulue.

En conclusion, en détectant la formation de paires isolées, on accélère notablement le mélange grâce à la diminution de p, même si on perd le caractère idéal de notre heuristique.

5.3 Aller plus loin

Étant donné les bons résultats obtenus en détectant la formation de paires isolées, nous avons tenté de généraliser cette approche en détectant la formation de petites composantes connexes. Après chaque échange, nous vérifions que les sommets concernés par l'échange n'appartiennent pas à une composante connexe de taille $\leq K$, pour un certain K. C'est le *test d'isolement*. La complexité des échanges est accrue puisqu'elle passe de $O(1)$ à $O(K)$, mais on peut éspérer une multiplication du rendement grace à la baisse de la probabilité de déconnexion et à la hausse de la probabilité de reconnexion.

Pour quantifier les effets du test d'isolement, nous avons mesuré (Fig. 5.2) la *fenêtre caractéristique* $T_{1/2}$, pour laquelle le taux de réussite P vaut $1/2$[1]

Pour cela, nous lançons notre heuristique (voir section 4.1) en ajustant le rapport $\frac{q^+}{q^-}$ de manière à converger vers un taux de réussite $P = 1/2$ (voir Eq. 4.2), et nous relevons la valeur moyenne de T après un nombre suffisant d'itérations.

Précisons que nous avons rencontré des comportements semblables avec toutes les distributions de degrés testées (toutes en loi de puissance, d'exposant et de moyenne variables). On remarque que la croissance de $T_{1/2}$ passe par trois phases :

1. Pour des petites valeurs de K, $T_{1/2}$ augmente très rapidement.

[1] $T_{1/2}$ est donc telle qu'après T échanges, le graphe a une chance sur deux de s'être déconnecté. À cause de la reconnexion, l'Eq. 3.1 n'est plus verifiée, ce qui explique qu'on n'essaie pas de mesurer la probabilité de déconnexion p, mais seulement le taux de réussite P.

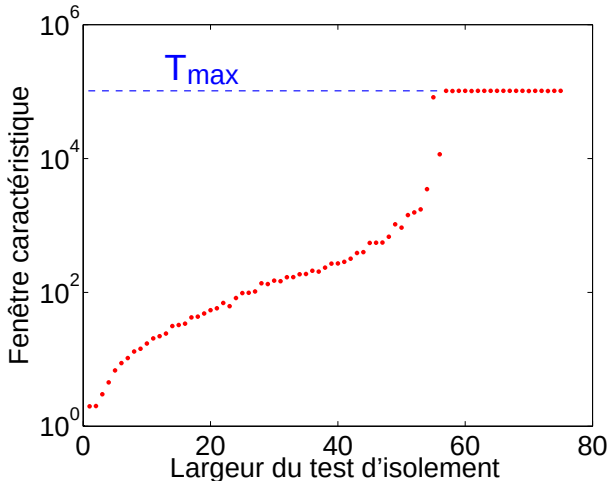

FIG. 5.2 – Evolution de la fenêtre $T_{1/2}$ en fonction de K, largeur du test d'isolement, sur un graphe généré.

2. Pour des valeurs plus grandes de K, et tant que $T_{1/2}$ reste petite relativement au nombre d'arêtes m, la croissance de $T_{1/2}$ est exponentielle en K.
3. Quand K est assez grand pour que $T_{1/2}$ s'approche de m, la reconnexion rentre en jeu et $T_{1/2}$ croît très vite vers sa *valeur crête* T_{max} (voir définition en section 5.4).

Cette croissance *au moins* exponentielle nous incite à ajuster K de manière à obtenir un taux de réussite élevé, sur une fenêtre T aussi grande que possible. Typiquement, on veut obtenir au moins 90% de réussite sur une fenêtre T de l'ordre de $|G|$. Il serait en effet inutile d'avoir $T >> |G|$ puisqu'alors les tests de connexité ne représenteraient plus qu'une part négligeable de la complexité, supplantés par les échanges d'arêtes eux-mêmes.

L'heuristique d'ajustement de T n'a alors plus lieu d'être, car modifier K permet de prendre T aussi grand que l'on veut. Un nouvel algorithme est nécessaire, qui devra gérer à la fois K et T pour des performances optimales.

5.4 Un nouvel algorithme

Il nous a paru plus judicieux de fixer une borne supérieure T_{max} à T, telle que le coût de T_{max} échanges d'arêtes vale X fois le coût d'un test de connexité. [2] Elle permet de garder un certain contrôle sur le mélange, et ce sans perte importante puisque la part des tests de connexité dans le coût total est de l'ordre de $1/X$ quand $T = T_{max}$.

Nous avons élaboré une heuristique (Fig. 5.3) pour ajuster T et K pendant le mélange selon le cahier des charges suivant :

[2] X est une constante arbitraire. Typiquement, $X = 50$ donne de bons résultats, les tests de connexité représentant alors 2% du temps de calcul quand $T = T_{max}$

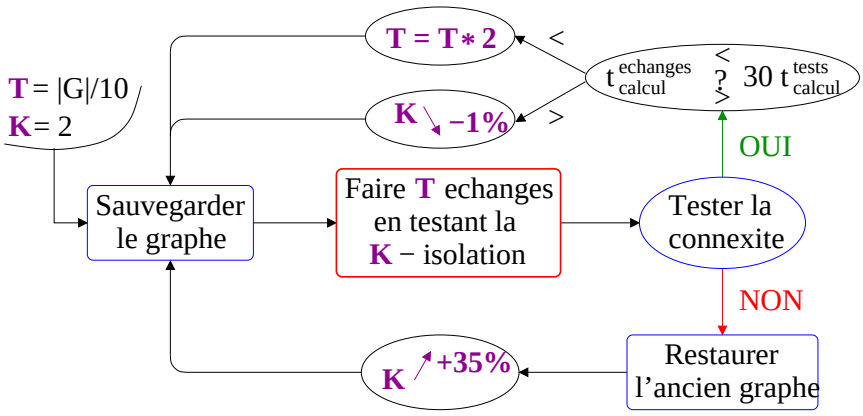

FIG. 5.3 – Evolution de la fenêtre $T_{1/2}$ en fonction de K, largeur du test d'isolement, sur un graphe généré.

- Au cours de l'essentiel du mélange, T doit être grand, avec des taux de réussite élevés. En conséquence, K doit être assez grand.
- K ne doit pas être inutilement grand : la complexité des échanges augmente avec lui.
- Au début du mélange, quand l'heuristique ajuste K et que le taux de réussite est encore faible, il est préférable de garder T petit.

Les tests de cette heuristique ont été satisfaisant, les paramètres indiqués sur la Fig. 5.3 étant ceux ayant produit les meilleurs résultats. Nous donnons quelques chiffres ci-dessous.

Résultats empiriques Pour la génération d'un graphe de 10^5 sommets, de degrés répartis en loi de puissance d'exposant $\alpha = 2.5$ et de degré moyen $z = 2.1$.

. L'heuristique donnera à K une valeur autour de 90.
. Elle ne consacre que 10% de son temps de calcul aux tests de connexité et sauvegardes/restauration du graphe, et elle perd 7% de son temps par les retours en arrière dû aux déconnexions.
. Presque un échange sur deux étant annulé à cause du test d'isolement, le mélange converge deux fois plus lentement vers la distribution aléatoire.
. La génération se fait en 36 secondes sur notre machine de test, contre plus de 10 heures par la méthode de Gkantsidis et al. Voir le tableau 5.3 pour des tests sur des topologies de graphes différentes.

Ces résultats empiriques confirment le bon comportement de notre heuristique, dans la mesure ou plus de 80% du temps de calcul est consacré aux échanges eux-mêmes. Les pertes dûes aux vérifications en cours de mélange, qui étaient considérables avec la simple accélération vue aux chapitres 3 et 4, deviennent presque négligeables. Ajoutons que nous avons retrouvé des résultats qualitativement identiques, sinon meilleurs, en modifiant la distributions de degrés prescrite (on fait varier α sur $[2, 3]$ et z sur $]2, 100]$).

D'autres tests ont confirmé que pour des distribution de degrés réalistes, K ne dépasse que très rarement la dizaine, et qu'on a toujours $K < 100$. Notre algorithme permet donc de s'affranchir de la complexité quadratique causée par les tests de connexité : sa complexité est $O(K \cdot |G|)$. Le tableau 5.3 résume les temps de calcul pour la génération de graphes de plus en plus gros, dont les paramètres correspondent au *backbone* de l'Internet.

Taille	Brut	Gkantsidis	Section 4.2	Section 5.1	**Section 5.4**
1000	2.96 s	0.41 s	0.26 s	0.10 s	**0.13 s**
10^4	6 min	27 s	7.2 s	1.9 s	**0.7 s**
10^5	≈10 h	2 h	51 min	6 min 31	**16.5 s**
10^6	≈40 jours	≈15 jours	≈10 jours	≈2 jours	**7 min 25**

TAB. 5.3 – Récapitulatif des temps de génération, sur notre machine de test. De gauche à droite : sans accélération ; avec accél. de Gkantsidis ; avec notre heuristique (4.2) ; en evitant les paires isolées (5.1) ; et avec le test d'isolement (5.4)

Chapitre 6

Conclusion

Nous avons repris la méthode d'accélération utilisée par Gkantsidis et al. pour la génération de graphes à séquences de degrés fixée, simples, connexes et aléatoires. Pour ajuster le paramètre T de l'accélération, ils utilisent une heuristique que nous nous sommes proposés d'améliorer afin de multiplier le rendement.

Nous avons montré que notre heuristique tendait vers le comportement optimal, et obtenu un ordre de grandeur théorique de l'amélioration apportée : la réduction du temps de calcul passe d'un facteur K à un facteur K^2.

Nous avons également effectué des tests pratiques et obtenu des rendements supérieurs à 95% de ceux obtenus en choisissant le paramètre T optimal au cours de l'algorithme. Nous avons donc obtenu une heuristique satisfaisante pour l'ajustement de T.

Nous avons ensuite introduit le *test d'isolement*, permettant de passer d'une complexité quadratique à une complexité linéaire (aux variations de K près). Nous avons donc obtenu une amélioration *qualitative* des méthodes précédemment utilisées. Les expériences montrent que cette amélioration est aussi quantitative, permettant la génération de graphes 100 fois plus gros dans un temps raisonnable (voir Tab. 5.3).

Le vide à combler pour une généralisation directe aux graphes orientés est essentiellement théorique, malgré de nombreux travaux dans cette direction [13, 14, 15, 16]. Les idées contenues dans les chapitres 3 et 4 peuvent aussi se généraliser à l'accélération d'autres chaînes de Markov présentant des caractéristiques similaires. Il est probable que dans le domaine de la génération aléatoire, notre heuristique puisse s'appliquer à des objets totalement différents des graphes.

Bibliographie

[1] Christos Gkantsidis, Milena Mihail, Ellen Zegura, "The Markov Chain Simulation Method for Generating Connected Power Law Random Graphs", SIAM Alenex'03

[2] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon, "Uniform generation of random graphs with arbitrary degree sequences", Phys. Rev. E64(2), 2001

[3] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence", Random Structures and Algorithms, pp. 161-179, 1995

[4] M. Molloy and B. Reed, "The size of the giant component of a random graph with a given degree sequence", Combin. Probab. Comput., pp. 295-305, 1998

[5] R.Taylor, "Constrained Switchings in graphs", 1982

[6] S.L. Hakimi, "On the realizability of a set of integers as degrees of the vertices of a graph", SIAM Journal of Appl. Math, 10, 1962

[7] V.Havel, "A remark on the existence of finite graphs", Caposis Pest. Mat. 80, 1955

[8] P. Erdös et T. Gallai, "Graphs with prescribed degree of vertices", *Mat. Lapok* 11 , 1960

[9] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications"

[10] M. E. J. Newman, "The structure and function of complex networks", 2003

[11] Claude Berge, "Graphes et Hypergraphes", Dunod, Paris, 1970

[12] Todd G. Will, "Switching distance between graphs with the same degrees", SIAM Journal of Discrete Math, Vol 12, No. 8, pp. 298-306

[13] Dhruv Mubayi, Todd G. Will, Douglas B. West, "Realizing Degree Imbalances in Directed Graphs", 2003

[14] A. Ramachandra Rao, Rabindranath Jana, and Suraj Bandyopadhyay, "A Markov Chain Monte Carlo Method for Generating Random (0,1)-Matrices with Given Marginals", *The Indian Journal of Statistics*, vol. 58, series A, Pt. 2, pp. 225-242, 1996

[15] John M. Roberts Jr., "Simple methods for simulating sociomatrices with given marginal totals", Social Networks 22, pp. 273-283, 2000

[16] Narsingh Deo, "An algorithm for removing surplus edges from a directed graph", 1974

[17] P. Erdös and A. Rényi, "On Random Graphs I", Publ. Math. Debrecen, vol. 6, pp. 290-297, 1959

[18] R.Albert and A-L. Barabási, "Emergence of Scaling in Random Networks", Science, vol. 286, pp. 509-512, 1999

[19] S.N. Dogorotsev and J.F.F Mendes, "Evolution of Networks", Adv. Phys. 51, pp. 1079-1087, 2002

[20] Jean-Loup Guillaume and Matthieu Latapy, "Relevance of Massively Distributed Explorations of the Internet Topology : Simulation Results", Algotel 2004