

Chapitre6 : Simulations sur données aléatoires

Wilfried Ehounou

February 1^{er} 2018

Contents

1 Simulation des algorithmes sur des réseaux théoriques	5
1.1 Objectifs	5
1.2 Définitions	5
1.3 Données: Génération aléatoires de graphes	7
1.3.1 Génération de réseaux de flots	7
1.3.2 Génération de line graphes sous jacent aux réseaux de flots non orienté	8
1.3.3 Distribution des valeurs de corrélation dans la matrice <i>matE</i>	8
1.3.4 Description du protocole d'étude	10
1.4 Analyses et interprétations	12
1.4.1 Distribution de la méthode de permutation aléatoire	13
1.4.2 Relation entre la distance line et la distance de Hamming	16
1.4.3 Comparaison des méthodes de correction	17
1.4.4 Influence des erreurs de corrélations sur les distributions	19
1.4.5 Impact de la fonction de coût sur les distributions	20
1.5 Simulation des graphes Iourtes	22

Chapter 1

Simulation des algorithmes sur des réseaux théoriques

Dans ce chapitre, nous allons procéder de manière contraire au problème à résoudre en supposant que les réseaux électriques de flots sont connus. À partir de ces réseaux, nous produisons leurs line graphes associés et en déduisons leurs matrices d'adjacences. Nous modifions certains valeurs dans ces matrices pour introduire des erreurs d'adjacences entre arêtes.

Nous attribuons les corrélations entre arêtes à partir de différentes lois de distributions, tout en tenant compte des erreurs de corrélations introduites dans ces matrices d'adjacences.

Les algorithmes proposés sont exécutés sur des matrices de corrélation construites par les corrélations entre arêtes et nous allons comparer les line graphes produites par les algorithmes avec ceux déduits des réseaux électriques de flots.

1.1 Objectifs

Les travaux réalisés dans cette partie ont pour but de montrer que les algorithmes proposés (couverture et correction) fournissent un line graphe de distance de Hamming minimale lorsque la matrice d'adjacence de ce graphe contient plus de corrélations *fausses positives* que de corrélations *fausses négatives* et peu d'erreurs de corrélations et cela malgré l'ordre du line graphe. Précisons aussi le nombre d'erreurs de corrélation doit être inférieur à 6 pour un seuil de corrélation supérieure à 0.8

Pour parvenir à un tel résultat, nous montrons que les sommets, n'appartenant à aucune couverture ($sommets \in sommets_1$) doivent être corrigés avec la méthode de **permutation aléatoire minimum** pour une **fonction de coût normale** pour de meilleurs résultats. Nous montrons également la relation existante entre la distance de Hamming et la distance line.

1.2 Définitions

Soient le graphe non orienté G du réseau de flots, le line graphe LG associé à G et la matrice d'adjacence $matE$ de LG . La matrice de corrélation entre arêtes de G , appliquée à une valeur de seuil, donne la matrice d'adjacence $matE$ dont les sommets sont les arêtes de G .

Le line graphe dont on ajoute des erreurs de corrélations (modification de cases) dans sa matrice d'adjacence est noté LG' . La matrice d'adjacence de LG' est $matE'$.

Définition 1 Une corrélation entre arêtes (ou arcs) est l'existence d'un sommet commun aux arêtes (ou arcs). Ce sommet commun peut être source, destination ou intermédiaire comme présenté dans la figure 1.1

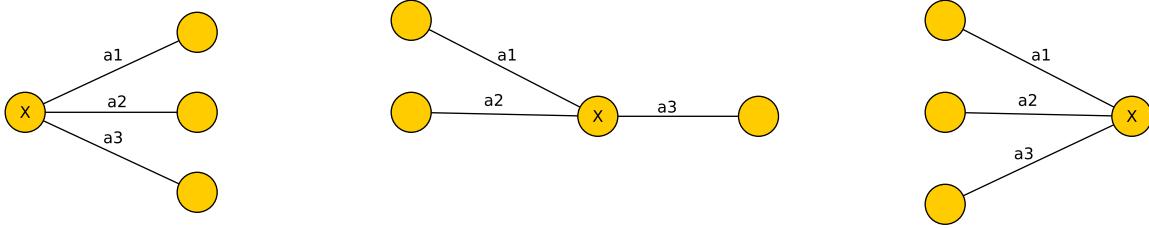


Figure 1.1: De la gauche à la droite: sommet X source, sommet X intermédiaire, sommet X destination

Dans la matrice d'adjacence $matE$ du ligne graphe formé par les corrélations, chaque case reçoit 1 quand deux arêtes partagent un sommet alors que la case reçoit 0 dans le cas contraire.

Définition 2 Une erreur de corrélation est la modification de la valeur d'une case de la matrice d'adjacence $matE$ de LG .

On distingue quatre catégories d'erreurs de corrélation, regroupées dans le tableau 1.1. Il s'agit des corrélations

- **vrai positives** : Il s'agit de cases à 1 n'ayant pas été modifiées dans la matrice $matE$.
- **vrai négatives** : Il s'agit de cases à 0 n'ayant pas été modifiées dans la matrice $matE$.
- **faux positives** : Il s'agit de cases à 0 modifiées dans la matrice $matE$.
- **faux négatives** : Il s'agit de cases à 1 modifiées dans la matrice $matE$.

Propriété 1 La corrélation fausse négative (l'absence de corrélation) est désignée par la valeur 0 dans la matrice d'adjacence $matE'$ tandis que la corrélation fausse positive (l'existence de corrélation) a une valeur 1 dans cette matrice. (voir tableau 1.1)

LG	LG'	corrélations
0	0	→ vrai négative
0	1	→ fausse positive
1	0	→ fausse négative
1	1	→ vrai positive

Table 1.1: Valeurs de corrélations selon le type d'erreurs dans les matrices d'adjacence de LG et LG'

Définition 3 matrice de corrélation

La matrice de corrélation est une matrice de relation entre arêtes dans laquelle, à chaque case, est associé une valeur de probabilité correspondante à la corrélation entre ces arêtes.

Les valeurs de probabilités sont définies, en fonction du type d'erreurs de corrélation, par différentes lois de distributions.

Définition 4 : Hypothèse de corrélations des arcs

Les corrélations entre arcs ou arêtes, proche de 1, ont tendance à partager un sommet tandis que celles proche de 0 ne partageront jamais de sommets

À partir de la définition 4, nous définissons une valeur de seuil pour dissocier les arêtes adjacentes des arêtes non adjacentes. En effet les corrélations entre arêtes, inférieure au seuil, sont transformées en 0 et celles, supérieure ou égale au seuil, sont modifiées par 1 dans la matrice de corrélation. En appliquant la valeur de seuil, on transforme notre matrice probabiliste en une **matrice de corrélation binaire** qui est aussi la matrice d'adjacence *matE* entre arêtes du graphe *LG*.

Définition 5 métrique: distance de Hamming

La métrique utilisée pour différencier deux graphes est la distance de Hamming. La distance de Hamming est le nombre d'arêtes (ou arcs) différentes entre deux graphes ayant le même ensemble de sommets.

Ainsi, une distance de Hamming égale à 0 signifie que les deux graphes sont identiques. Tandis que une distance de Hamming égale à k signifie qu'il a k arêtes différentes entre ces deux graphes.

Définition 6 fonction de coût d'un sommet

La fonction de coût d'un sommet est le coût de chaque arête ajoutée ou supprimée lorsqu'on applique l'algorithme de correction sur ce sommet.

Le coût d'une arête peut être

- unitaire : l'ajout et la suppression valent 1.
- normal : la suppression coûte la corrélation entre l'arête et une autre et l'ajout vaut 1 moins cette corrélation.
- quadratique : la suppression coûte la corrélation au carré entre l'arête et une autre et l'ajout vaut 1 moins cette corrélation au carré.
- puissance 4 : la suppression est la corrélation entre l'arête et une autre à la puissance 4 et l'ajout vaut 1 moins cette corrélation à la puissance 4.
- en cloche : l'ajout et la suppression dependent d'une fonction polynomiale de degré 2 dont les valeurs autour d'un seuil sont proche de 0. Nous y reviendrons dans la partie 1.4.5

1.3 Données: Génération aléatoires de graphes

1.3.1 Génération de réseaux de flots

La structure de données utilisée, pour le graphe du réseau de flots, est une *matrice d'adjacence*. Cette matrice d'adjacence est une matrice creuse carrée de n sommets et de degré moyen α . La probabilité d'existence d'une arête est de $proba = \frac{\alpha}{n}$.

La matrice d'adjacence forme un graphe connexe. Toutefois, si le graphe obtenu n'est pas connexe, on choisit aléatoirement un sommet de chaque composante connexe et on ajoute une arête entre ces sommets.

Pour orienter les arêtes, on réalise un tri topologique avec un parcours en largeur *Breadth First Search (BFS)* du graphe non orienté généré, à partir de certains sommets choisis comme des sources. Chaque sommet x a un ordre topologique D_x et l'arête a_{xy} devient soit l'arc a_{xy} si $D_x < D_y$ soit l'arc a_{yx} si $D_x > D_y$. Le graphe obtenu est alors orienté (un *Directed Acyclic Graph DAG*).

L'ajout des flots sur chaque arc se fait aussi par un parcours en largeur (BFS). On définit les valeurs minimales et maximales des grandeurs physiques. Ces valeurs sont sélectionnées selon le réseau énergétique à modéliser. À titre d'exemple, les valeurs choisies pour des grandeurs électriques sont : les intensités $I = [150, 200]$, les tensions $U = [220, 250]$, les puissances $P = [33000, 62500]$.

On débute par les sommets sources dont on génère une valeur aléatoire comprise dans l'un des intervalles de ces grandeurs. Chaque arc sortant du sommet source reçoit un flot égal à la somme des flots sur les arcs entrants du sommet source multipliée par le facteur ϵ (désignant les pertes par effets joules) et divisée par le degré sortant de ce sommet si nous avons comme grandeurs les intensités et les puissances. Dans le cas de grandeurs comme les tensions, le flot de chaque arc sortant est le flot multiplié par le facteur ϵ . On propage les valeurs des grandeurs physiques jusqu'à ce qu'on arrive aux sommets puits. L'application de ces règles de flots permettent de vérifier la *loi de conservation des noeuds*.

1.3.2 Génération de line graphes sous jacent aux réseaux de flots non orienté

Nous nommons les arcs du graphe et nous construisons la matrice d'adjacence des arcs du graphe (0 aucun sommets en commun sinon 1). Cette matrice d'adjacence est aussi la matrice de corrélation binaire entre arcs.

D'après la définition d'un line graphe ??, la matrice de corrélation binaire est la *matrice d'adjacence du line graphe* sous jacent au graphe non orienté de notre réseau de flots généré. Cette matrice est symétrique et est notée *matE*. Si la matrice *matE* ne contient *aucune erreur de corrélation* alors elle admet une line-couverture (couverture en cliques) c'est-à-dire que chaque sommet est contenu dans deux cliques au maximum et chaque arête est couverte par une seule clique. Dans le cas contraire, nous utilisons l'algorithme de correction pour fournir une couverture pour chaque sommet n'appartenant à aucune clique.

Notre objectif est de proposer une line-couverture (couverture en cliques) aux graphes dont les matrices d'adjacence *matE* possèdent des erreurs de corrélation. Cette line-couverture fournit un line graphe dont la distance line/Hamming entre ce line graphe et le graphe de matrice d'adjacence *matE* est minimale. Rappelons que le graphe de matrice d'adjacence *matE* peut contenir des erreurs de corrélations.

1.3.3 Distribution des valeurs de corrélation dans la matrice *matE*

Supposons que la matrice d'adjacence *matE* ne contienne pas d'erreurs de corrélations. Cela implique que 1) il n'aurait pas d'erreurs *fausses positives et fausses négatives* 2) toutes les corrélations *vrai négatives* tendent vers 0 et 3) toutes les corrélations *vrai positives* tendent vers 1. Cela donne, sur une figure (figure 1.2), des demi paraboles croissantes (*vrai positives*) et décroissantes (*vrai négatives*) à partir du seuil s . Cependant notre matrice *matE* contient des erreurs de corrélations

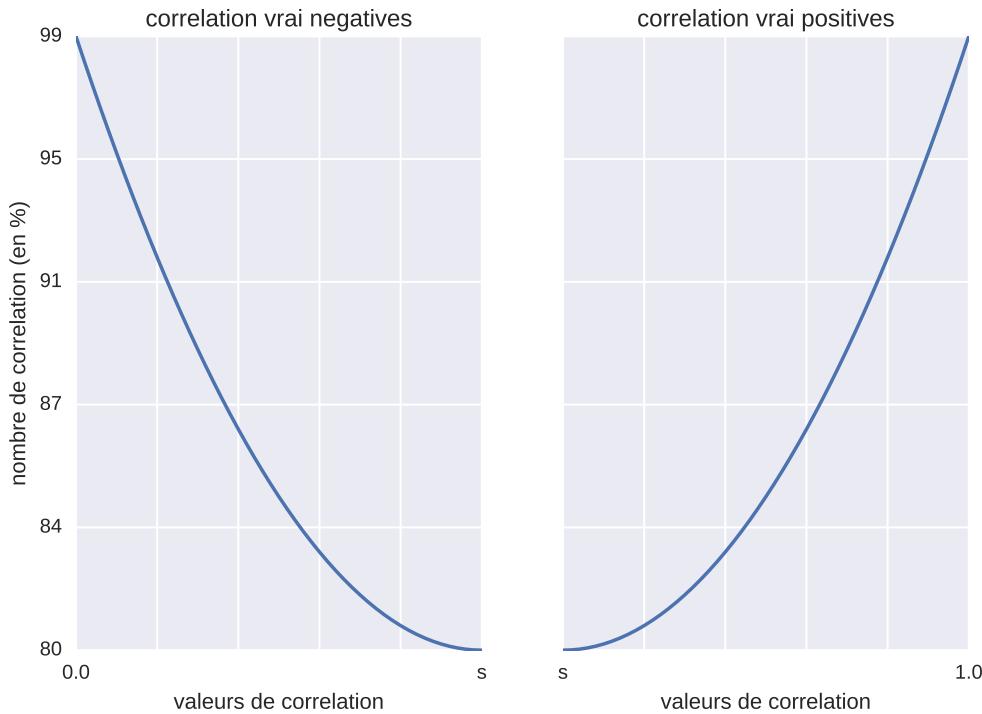


Figure 1.2: À gauche : Parabole croissante pour les erreurs vrai positives dans l'intervalle $[s, 1]$, à droite : Parabole décroissante pour les erreurs vrai négatives dans l'intervalle $[0, s]$. L'ordonnée désigne le taux de corrélation pour une valeur donnée.

impliquant la présence de corrélations *fausses négatives* et *fausses positives* autour de la valeur de seuil p_correl plutôt que du seuil s . La variable p_correl est un seuil à partir duquel une corrélation est assez significative pour indiquer que deux arêtes partagent un sommet. En effet, le seuil p_correl est une valeur arbitraire choisie en fonction de l'hypothèse de corrélation entre arcs (définition 4).

Ainsi, nous définissons quatre intervalles qui désignent l'ensemble de valeurs pour les corrélations:

- *vrai négatives* $\rightarrow int_vn = [0, p_correl - 0.2[$
- *fausses négatives* $\rightarrow int_fn = [p_correl - 0.2, p_correl[$
- *fausse positives* $\rightarrow int_fp = [p_correl, s[$
- *vrai positives* $\rightarrow int_vp = [s, 1]$

Ainsi, les arcs adjacents ayant une corrélation comprise dans l'intervalle int_fn sont considérés comme *fausses négatives* alors que pour une corrélation comprise dans l'intervalle int_fp , les arcs non adjacents ont des corrélations *fausses positives*. Les intervalles sont resumés dans la figure 1.3. Ainsi, si p_correl tend vers le seuil s , on a plus de corrélations *fausses négatives* que des corrélations *fausses positives* dans la matrice $mateE$. Par contre, le nombre de corrélations *fausses positives* devient très élevé quand p_correl tend vers 0.

Nous retiendrons que les erreurs de corrélations se localisent autour du seuil p_correl et peuvent suivre différentes lois de probabilités. La figure 1.4 présente les courbes des distributions de

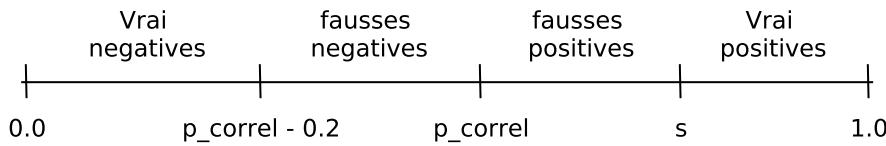
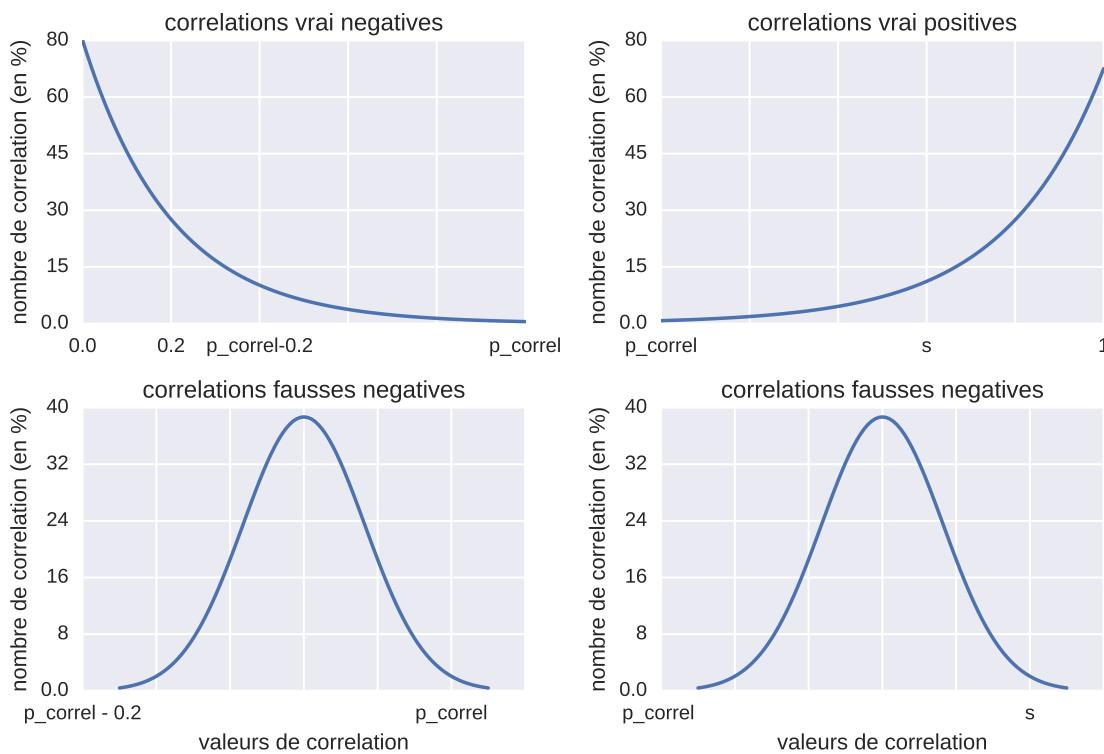


Figure 1.3: Correspondance entre valeurs et types de corrélations

corrélation. Les corrélations *fausses positives* et *fausses négatives* suivent des lois normales alors que celles *vrai positives* suivent des lois exponentielles de paramètres $\lambda \geq 2$ et celles *vrai négatives* avec des lois exponentielles inversée avec $\lambda \leq -2$.

Figure 1.4: En haut : loi exponentielle pour les erreurs *vrai positives* et *vrai négatives*, en bas : loi uniforme pour les erreurs *fausses positives* et *fausses négatives*

L'utilisation de lois de probabilité a pour but de montrer l'impact de seuil p_correl sur la ligne couverture et aussi calculer les coûts de correction des sommets n'appartenant à aucune cliques.

1.3.4 Description du protocole d'étude

redécrire la partie en dessous et reprogrammer

La réalisation de notre étude passe par la génération de 500 graphes de flots G de 30 sommets ayant un degré maximal moyen $\Delta(G) = 5$ qui simulent le fonctionnement d'un réseau électrique. Nous en déduisons également 500 line graphes de 150 sommets et 470 arêtes, en moyenne.

Nous introduisons trois paramètres $k, p_correl, prob$:

1. k désigne le nombre de corrélations erronées à ajouter dans la matrice $matE$. Dans notre étude, $k \in [1, 9]$.
2. p_correl désigne la probabilité d'ajouter des erreurs de corrélations, soit corrélation *fausses positives* (ajout d'arêtes) soit corrélation *fausses négatives* (suppression d'arêtes) soit les deux. Cette variable $p_correl \in [0, 1]$ varie par pas de 0.1. Par exemple
 - si $p_correl = 0 \rightarrow$ on a que des corrélations *fausses positives* dans la matrice $matE'$ car on supprime uniquement des arêtes dans le ligne graphe de matrice d'adjacence $matE$.
 - si $p_correl = 0.5 \rightarrow$ le nombre de corrélations *fausses négatives* et *fausses positives* est approximativement semblable dans la matrice $matE'$ car on ajoute et supprime équitablement des arêtes dans la matrice $matE$.
 - si $p_correl = 1.0 \rightarrow$ la matrice $matE'$ ne contient que des corrélations *fausses négatives* car on ajoute des arêtes dans le ligne graphe de matrice $matE$.
3. $prob$ désigne la probabilité associée à une corrélation selon le type d'erreurs effectué dans $matE'$. En un mot, $prob$ est la valeur de corrélation entre arêtes. Ce paramètre est important car les valeurs de corrélations calculées ne sont pas binaires mais dépendent d'une loi de probabilité donc probabilistes. Nous en reparlerons également dans la section 1.4.5.

Pour ajouter des erreurs de corrélation à la matrice $matE$ correcte, on tire aléatoirement k cases non encore modifiées. Nous mettons chaque case et sa case symétrique à 1 si la probabilité de la case est inférieure ou égale à p_correl . Si cette case est déjà à 1, on choisit une autre case. Selon le type d'erreurs de chaque case (vrai négatives, vrai positives), on lui attribue une valeur de corrélation selon des distributions prédéfinies.

Considérons le graphe G_k de matrice d'adjacence $matE_k$ dans laquelle on ajoute $k \in [1, 9]$ erreurs de corrélation selon $p_correl = 0.5$, la probabilité d'ajouter autant d'erreurs *fausses négatives* et *fausses positives*. À la fin de l'exécution de l'algorithme de couverture, s'il existe des sommets de G_k non couverts par *une ou deux cliques*, on les ajoute à l'ensemble des sommets à corriger *sommets_1* et nous appliquons l'algorithme de correction sur chaque sommet de *sommets_1* selon les méthodes suivantes:

- méthode 1 : degré minimum avec remise.
Elle consiste à sélectionner le sommet de degré minimum dans l'ensemble *sommets_1*, à appliquer l'algorithme de correction afin de modifier $matE$ et enfin à re-exécuter les deux algorithmes sur la matrice $matE$ modifiée.
- méthode 2 : coût minimum avec remise.
Elle consiste à sélectionner le sommet de coût minimum dans l'ensemble *sommets_1*, à appliquer l'algorithme de correction afin de modifier $matE$ et enfin à re-exécuter les deux algorithmes sur la matrice $matE$ modifiée.
- méthode 3 : coût minimum avec permutation des sommets de *sommets_1*.
Elle consiste à choisir une permutation dont les sommets sont classés par ordre croissant de leur coût de modification de la matrice $matE$ et à appliquer l'algorithme de correction sur cette permutation.

- méthode 4 : degré minimum avec permutation des sommets de *sommets_1*.
Elle consiste à choisir une permutation dont les sommets sont classés par ordre croissant de leur degré et à appliquer l'algorithme de correction sur cette permutation.
- méthode 5 : permutation aléatoire des sommets de *sommets_1*.
Elle consiste à choisir aléatoirement N permutations puis à appliquer l'algorithme de correction et à sélectionner la permutation ayant un coût et une distance de Hamming minimum.

Considérons $\alpha \in [1, 5]$ le nombre de fois qu'on applique k erreurs dans la matrice *matE* du line graphe *LG*, $G_{k,\alpha}$ le graphe de matrice d'adjacence *matE_{k,α}* dont on a modifié k corrélations α fois et *LG_{k,α}* le line graphe de matrice d'adjacence *matE_{k,α}* fourni par les algorithmes de couverture et de correction à partir du graphe $G_{k,\alpha}$.

En comparant

1. *LG* et *LG_{k,α}*, on obtient la distance de Hamming notée $DH_{k,\alpha}$.
2. $G_{k,\alpha}$ et *LG_{k,α}*, on a la distance line notée $DL_{k,\alpha}$.

On définit par les variables *moy_DH* et *moy_DL*, les moyennes respectives des distances de Hamming (notée $DH_{k,\alpha}$) et des distances line (notée $DL_{k,\alpha}$) pour une valeur donnée de k et pour tout $\alpha \in [1, 5]$.

$$moy_DH_k = \sum_{\alpha=1}^5 DH_{k,\alpha} \quad moy_DL_k = \sum_{\alpha=1}^5 DL_{k,\alpha} \quad (1.1)$$

Le protocole d'étude est recapitulé dans la figure 1.5. Le réseau de flots *G* généré est transformé en un line graphe *LG*, puis sont ajoutés k erreurs de corrélations α fois dans *LG* pour obtenir $G_{k,\alpha}$. Nous appliquons les algorithmes (couverture et correction) pour déterminer la line-couverture (ou la plus proche possible) du graphe $G_{k,\alpha}$ car ce graphe n'est pas nécessairement un line graphe. Cette line-couverture correspond au line graphe *LG_{k,α}* qui est utilisé pour calculer les distances line ($DL_{k,\alpha}$) et de Hamming ($DH_{k,\alpha}$). Les valeurs moyennes (*moy_DL*, *moy_DH*) des distances line et de Hamming sont calculées pour chaque erreur k et leurs distributions sont présentées dans la section suivante afin de montrer les performances des algorithmes en présence d'erreurs de corrélations de différentes lois de probabilités.

1.4 Analyses et interprétations

Les valeurs de corrélations sont définies par les lois uniformes (erreurs vrai positives et vrai négatives) et exponentielles (erreurs fausses positives et fausses négatives) comme illustré par la figure 1.4 pour une probabilité d'ajout des erreurs *p_correl* = 0.5. Cela signifie qu'il y a autant d'erreurs fausses positives que d'erreurs fausses négatives dans la matrice de corrélation du graphe G_k . Ces lois de distribution respectent l'hypothèse de corrélations des arcs (voir la définition 4). Nous décrirons d'abord les distributions des distances line et de Hamming moyennées (*moy_DL* ou *moy_DH*) pour une méthode de correction (aléatoire). Ensuite nous comparons les cinq méthodes de correction en nous basant sur les distances de Hamming moyennées. Enfin nous expliquons le choix de la méthode de *permutation aléatoire* et montrons que les algorithmes (couverture et correction) proposent de meilleurs résultats lorsque la matrice de corrélation possède plus de corrélations *faux négatives* que de corrélations *faux positives* et aussi peu d'erreurs de corrélations ($k < 6$). Nous présentons également l'impact de la fonction de coût dans les distributions de distances de Hamming et la relation existante entre la distance line et la distance de Hamming.

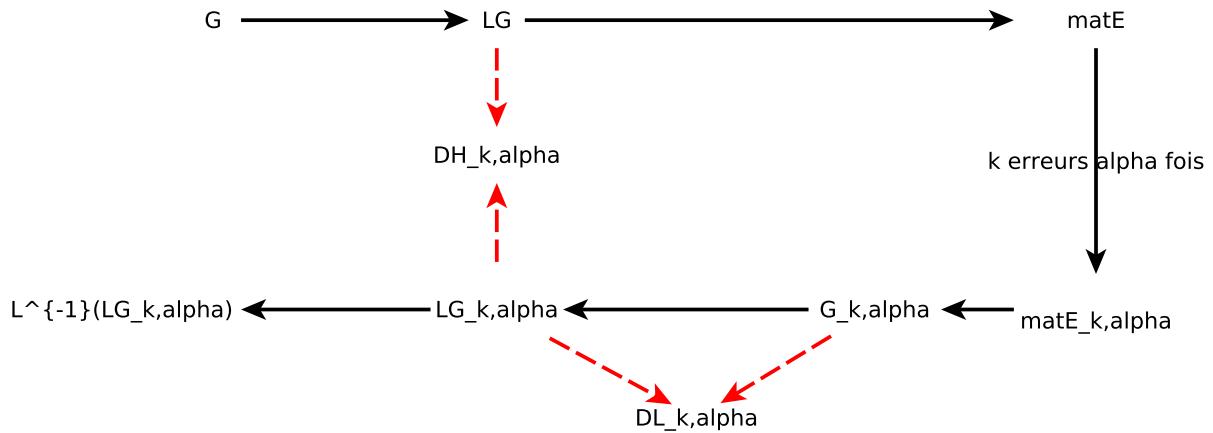


Figure 1.5: Différentes étapes du processus de simulation de nos algorithmes (traits en noir), calcul de distances entre étapes

1.4.1 Distribution de la méthode de permutation aléatoire

La figure 1.6 représente les distributions des distances line, de Hamming, des fonctions de répartition de la corrélation entre distances line et Hamming pour $k \in [0, 5]$ erreurs.

Pour $k = 0$ erreur, nous avons un batonnet sur les histogrammes de distances line et de Hamming. Ce batonnet est le pourcentage d'arêtes identiques entre les graphes G_k et LG_k (distance line) et aussi entre les graphes LG et LG_k . Nous constatons que le pourcentage est de 100% impliquant que les arêtes des graphes LG_k , découvertes par les algorithmes, sont identiques à celle du graphe LG . Ce qui est normal parce que nous n'avons ajouté aucune erreur dans le line graphe LG . Les courbes des fonctions de répartition suivent l'équation 1.2 pour la corrélation entre les distances line et de Hamming et l'équation 1.3 pour la distance de Hamming.

$$y = \begin{cases} 0 & \text{si } x < 1 \\ 100 & \text{si } x = 1 \end{cases} \quad (1.2)$$

$$y = 1 \quad \text{si } x \in [0, 1] \quad (1.3)$$

Nous nous servirons du cas de $k = 0$ erreur comme une référence de la meilleure performance de nos algorithmes.

Pour $k \in [1, 4]$, l'ensemble des batonnets, regroupés avant la droite $y = k$ (droite en rouge) de chaque histogramme, a un pourcentage supérieur à 50%. La présence de cette droite nous indique que, dans la majorité des cas, qu'il existe une différence de k arêtes entre les graphes G_k et LG_k (voir distance line figure 1.6) et ces k arêtes correspondent aux erreurs ajoutées dans la matrice d'adjacence $matE$ du line graphe LG . Cela explique la distance de hamming de 0 arête entre LG et LG_k et le pourcentage pour 0 erreur est le pic de chaque histogramme (voir distance de Hamming figure 1.6).

Au-delà de $k \geq 5$ erreurs, le pic de chaque histogramme baisse significativement quand k augmente (voir distances line et de Hamming figure 1.7). Une explication est la présence d'arêtes erronées dans les line graphs LG_k proposés parce que la majorité de ces line graphs qui ont plus de k arêtes différentes entre LG_k et G_k alors que ce nombre k doit correspondre au nombre d'erreurs ajoutées dans le line graphe LG . Il s'illustre parfaitement avec $k = 9$ erreurs dans la

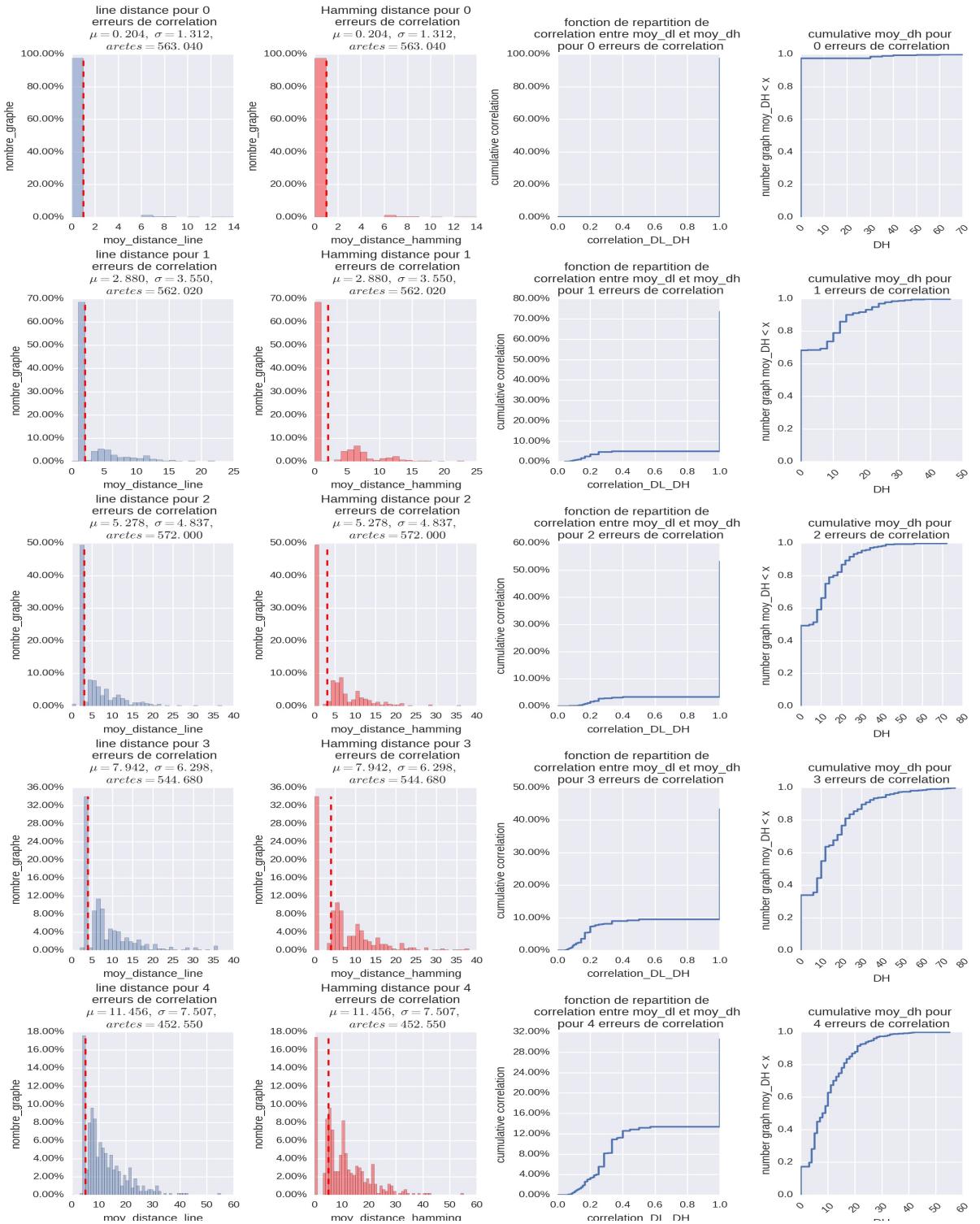


Figure 1.6: Méthode de permutation aléatoire avec une fonction de correction à coût unitaire : distribution des distances line *moy_DL* et de Hamming *moy_DH* pour $k \in [0, 5]$ corrélations alterées

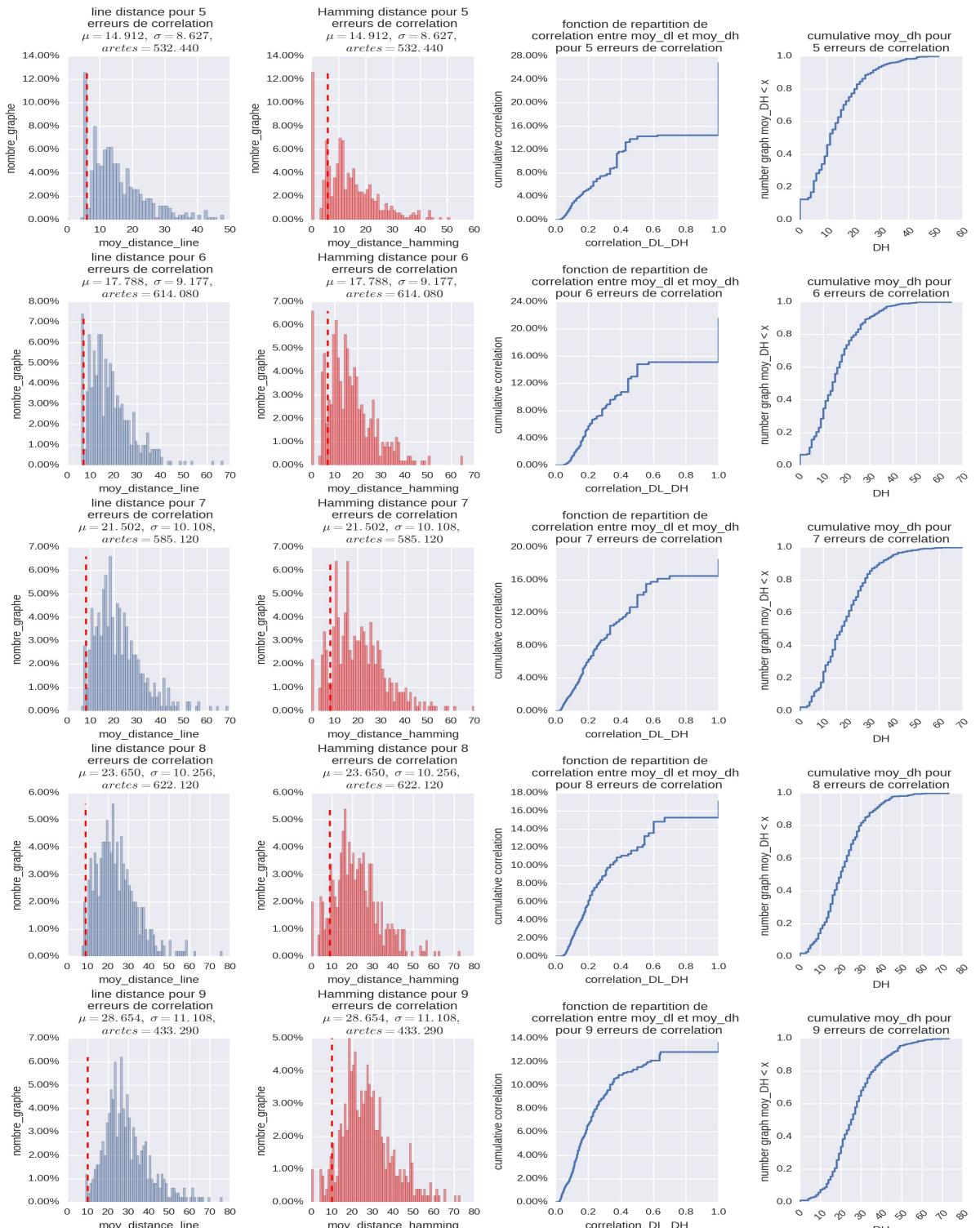


Figure 1.7: Méthode de permutation aléatoire avec une fonction de correction à coût unitaire : distribution des distances line moy_DL et de Hamming moy_DH pour $k \in [6, 9]$ corrélations alterées

figure 1.7 où on a moins de 13% de line graphes dont les arêtes sont identiques et les 87% restants ont plus d'une arête différente.

Par ailleurs, les fonctions de répartition des corrélations entre distances line et de Hamming et celle de la distance de Hamming ont des courbes qui s'éloignent de celle de $k = 0$ erreur. En effet, ces courbes se divisent en deux parties : une courbe croissante et une droite verticale (distance de Hamming) ou horizontale (distance line). Pour $k \in [1, 4]$, dans les figures des distributions cumulatives des distances de Hamming (colonne 4 de la figure 1.7), on remarque que la droite verticale pour k erreurs baisse quand k augmente. Cette droite est le pourcentage de line graphes identiques (LG et LG_k). Par exemple, on a 69% de line graphes LG_k identiques à LG pour $k = 1$ alors qu'on en a que 19% pour $k = 4$. Cela est dû à l'ajout d'arêtes dans LG_k n'appartenant pas à LG . Cela forme une courbe exponentielle croissante dans laquelle on a plus de 10 arêtes différentes pour $k \leq 5$.

De même, au-delà du nombre arêtes différentes c'est-à-dire $moy_DH > k$, nous constatons une droite verticale très courte qui baisse également quand la variable moy_DH augmente. Cela signifie qu'il y a très peu de line graphes LG_k ayant des distances de Hamming très élevées par rapport aux k erreurs (voir figures 1.6 et 1.7) pour $k \geq 5$.

Le fait que les distributions de distance line et de Hamming soient, toutes les deux, asymétriques (pour $k \leq 6$) soient symétriques (pour $k > 6$) nous interrogent sur l'évolution des distributions des distances line par rapport à celles de Hamming.

1.4.2 Relation entre la distance line et la distance de Hamming

Les réseaux réelles, dont nous possédons les mesures de flots sur les arcs, sont inconnus. Cela implique que la distance de Hamming est impossible à déterminer. Étant donnée que nous avons le line graphe $G_{k,\alpha}$ du réseau de flots dans nos simulations, nous pouvons calculer les distances de Hamming et line.

La distance line est la distance de Hamming minimum entre LG et $LG_{k,\alpha}$ pendant la correction des sommets. Elle compare les line graphes obtenus $LG_{k,\alpha}$ après correction du graphe $G_{k,\alpha}$ pour en fournir un line graphe dont la correction des sommets est de coût minimum.

Nous calculons la corrélation entre les distances line et de Hamming à partir de la formule 1.4.

$$corr_{k,\alpha} = \frac{|moy_DL_{k,\alpha} - moy_DH_{k,\alpha}|}{\max(moy_DL_{k,\alpha}, moy_DH_{k,\alpha})}; corr_k = \sum_{\alpha=1}^5 corr_{k,\alpha}; F_k(x) = P(corr_k \leq x) \quad (1.4)$$

avec $x \in [0, 1]$ une valeur de corrélation et k le nombre d'erreurs de corrélation.

Sur la figure 1.8, est représenté la fonction de répartition F_k dans laquelle nous avons, en abscisse, la corrélation entre les distances et, en ordonné, le pourcentage de graphes dont la corrélation $corr$ est inférieure à x . En effet si $corr_k = 1$ alors il n'existe aucune corrélation entre les distances line et de Hamming. Cela signifie que le line graphe fourni LG_k est le véritable line graphe de LG sur notre réseau de flots G ($LG_k = LG$). De même, si $corr_k = 0$ alors les distances line et de Hamming sont identiques. Cela signifie que ajouter/supprimer ces arêtes au line graphe LG_k produit le line graphe de notre réseau ($LG_k \neq LG$) et que LG_k et LG sont différents de k arêtes quand $k < 6$.

Donc si $F_k(1) \approx 0$ alors le nombre de corrélation $corr_k = 1$ est très élevé. Cela s'illustre sur la figure 1.8 par les courbes de $k = [2, 5]$. Par exemple $F_5(1) \approx 10\%$ signifie que nous avons $70 - 10 = 60\%$ line graphes LG_k correspondant aux line graphes LG des réseaux. ($corr_5 \approx 60\%$ et 70% le pourcentage de corrélations égale à 1).

Par contre, si $F_k(1)$ est très élevé, cela signifie que le nombre de $corr_k = 1$ est très faible entraînant

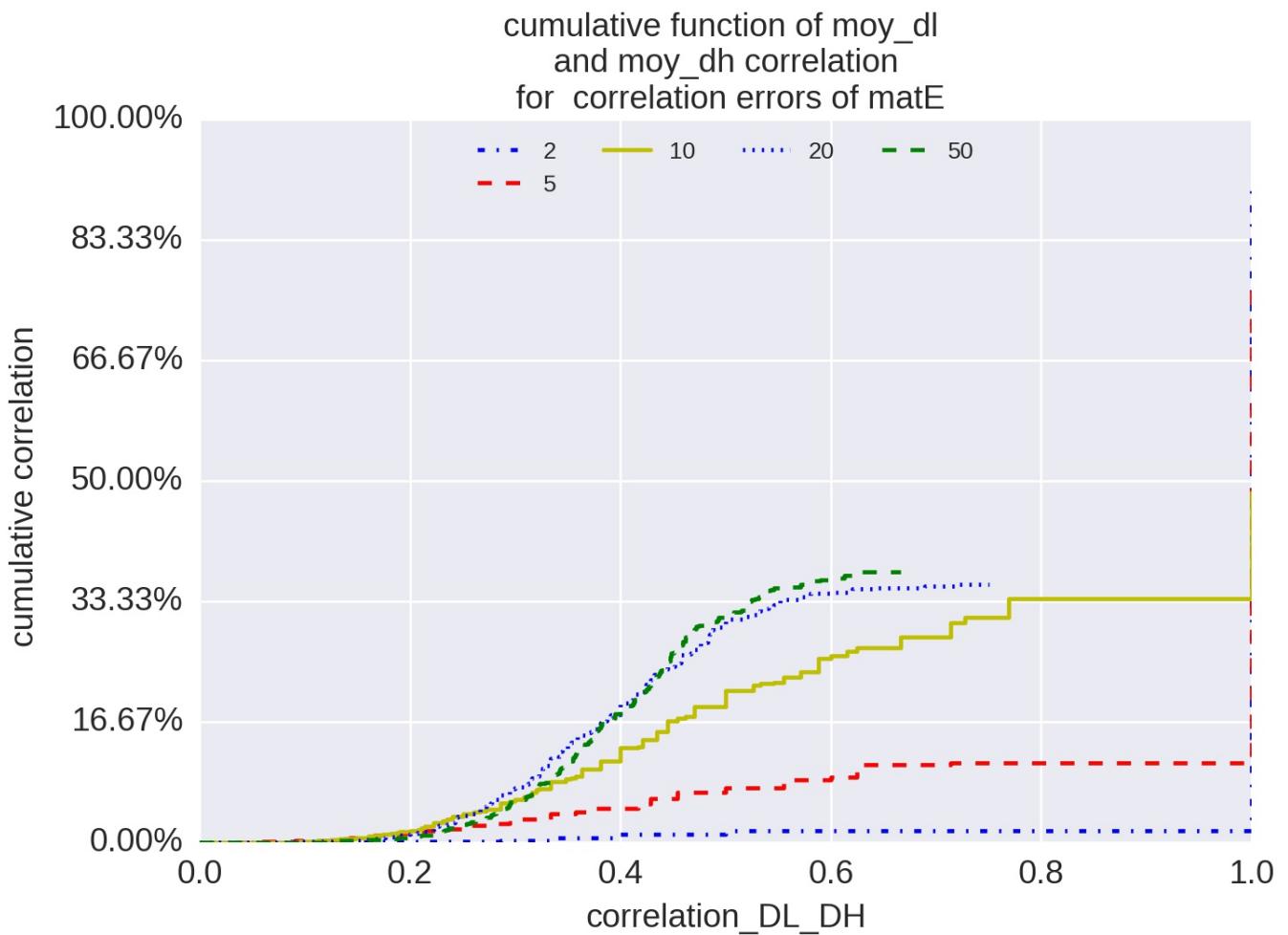


Figure 1.8: distance line versus distance de Hamming pour k erreurs de corrélation et $p = 0.5$

une corrélation très forte en les distances line et de Hamming. C'est notre constat avec les courbes de $k = [10, 20]$ dans lesquelles nous avons une croissante continue en fonction de l'augmentation des valeurs de corrélations.

Nous subdivisons nos courbes en deux catégories:

- Celle dont on a une corrélation entre distances lines et de Hamming (courbes de $k = [10, 20]$).
- celle dont on a aucune corrélation entre ces distances parce que nous fournissons le line graphe du réseau c'est-à-dire $LG = LG_k$ (courbes de $k = [2, 5]$).

Nous pouvons conclure que l'utilisation de la distance line est une bonne métrique pour juger de la qualité de notre algorithme de correction en absence de la distance de Hamming parce que une distance line inférieure à 5 fournit le line graphe LG du réseau de flots tandis qu'une distance supérieure à 10 correspond au nombre de corrélations à modifier pour livrer le line graphe LG .

1.4.3 Comparaison des méthodes de correction

Nous recherchons la meilleure méthode de correction parmi les cinq méthodes énumérées plus haut. Pour ce faire, on dispose des distributions de distances line et de Hamming, des histogrammes, des

fonctions de répartitions de ces distributions et aussi des moyennes de distances line/Hamming associées à k corrélations érronées pour chacune des méthodes regroupées dans les figures 1.6 et 1.7. Nous utilisons la moyenne des distances de Hamming pour la comparaison de méthodes parce que la distance de Hamming permet d'évaluer la différence entre le graphe de base LG et celui prédit LG_k par nos algorithmes et aussi nous connaissons les line graphs associés aux réseaux électriques.

Rappelons que nous avons la probabilité $p_correl = 0.5$ et la fonction de coût est unitaire (F_0). La figure 1.9 affiche les courbes des différentes méthodes pour des distances de Hamming moyennées en fonction des k erreurs de corrélations.

Nous constatons que la pire des méthodes est celle de degré minimum avec remise (en rouge avec un rond) car elle est au dessus des autres et la meilleure est celle de *de permutation aléatoire* (en jaune avec un triangle) car elle propose des line graphs ayant le nombre minimum d'arêtes différentes pour $\forall k$.

Nous retenons, pour la suite, la méthode de **permutation aléatoire** comme méthode de correc-

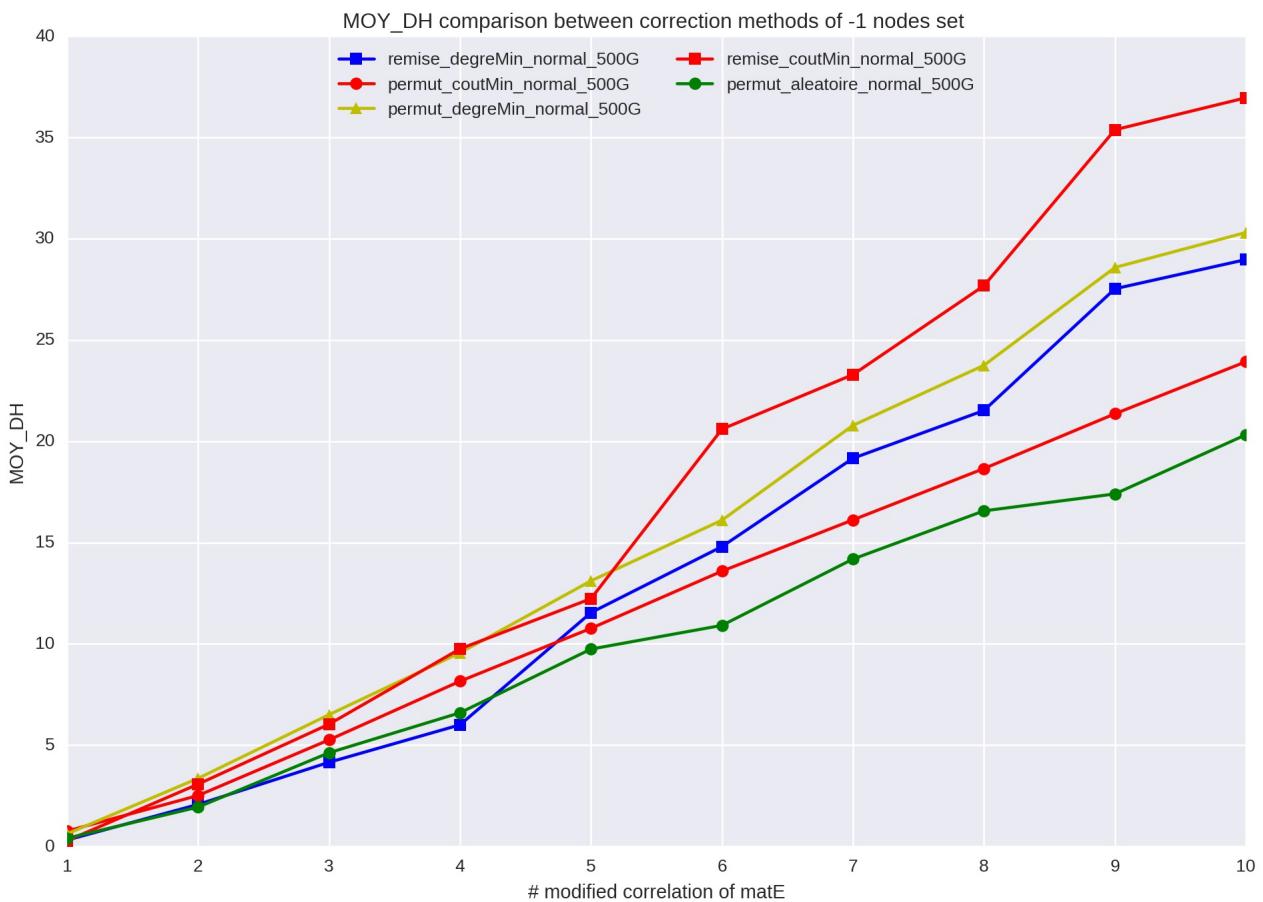


Figure 1.9: Comparaison des différentes méthodes de correction de sommets pour $k \in [1, 9]$ corrélations modifiées. Les courbes en bleu carré, rouge carrée, rouge rond, vert rond et jaune triangle sont associées respectivement aux méthodes 1, 2, 3, 5, 4

tion des sommets n'appartenant à aucune couverture ($sommets \in sommets_1$).

1.4.4 Influence des erreurs de corrélations sur les distributions

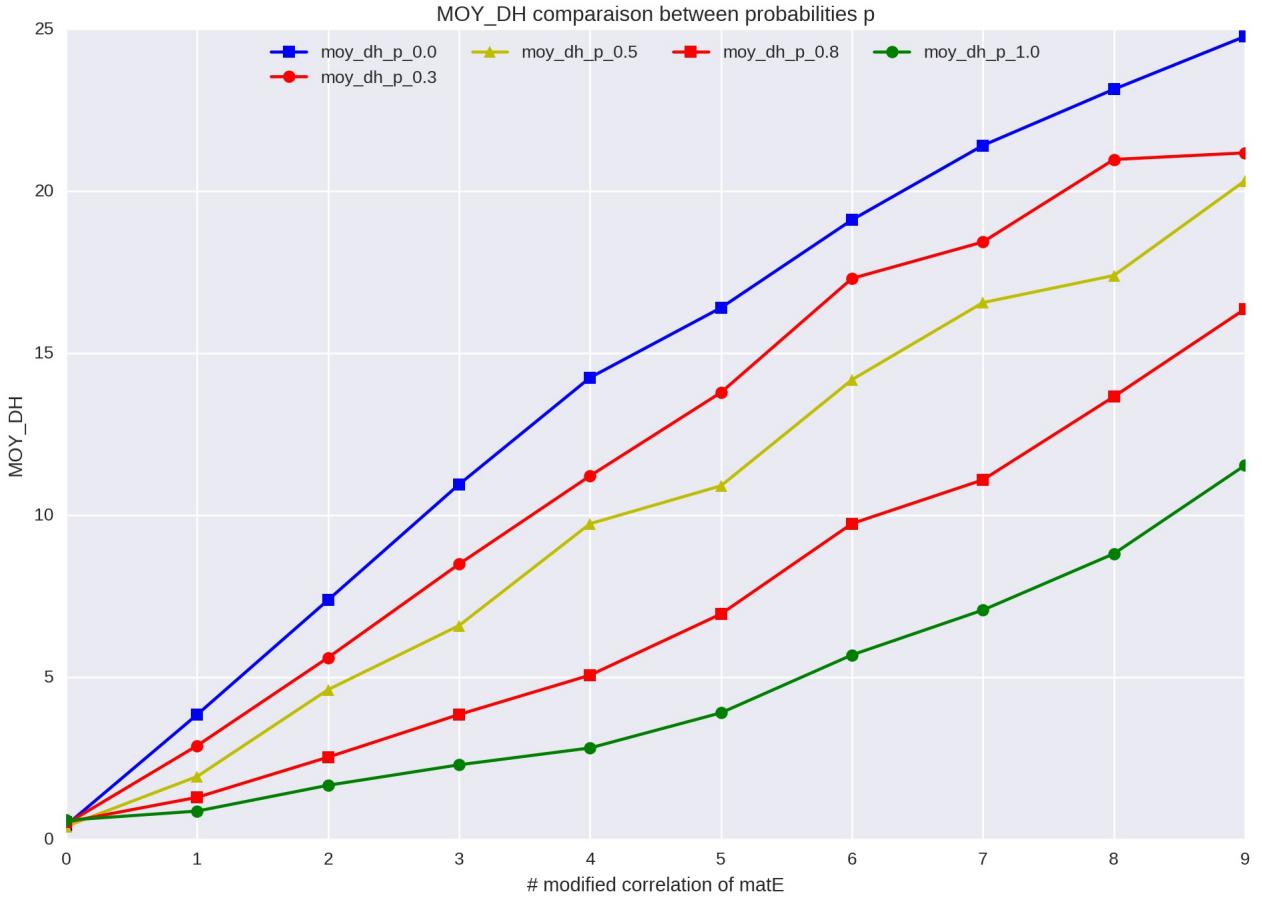


Figure 1.10: Comparaison des différentes probabilités d'ajout $k \in [1, 9]$ de corrélations fausses positives et fausses négatives sur la méthode de permutation aléatoire

Faisons varier la variable $p_correl \in [0, 1]$ par pas de 0.1 dans le but de visualiser l'impact de corrélations *fausses positives* et *fausses négatives* dans l'exécution des algorithmes. Rappelons que l'ajout et la suppression d'arêtes ont le même coût de traitement c'est-à-dire 1. La figure 1.10 résume l'évolution des types d'erreurs de corrélations (p_correl) pour des distances de Hamming moy_DH en fonction de $k \in [1, 9]$ erreurs de corrélations.

Nous constatons que les algorithmes donnent de meilleurs résultats pour $p_correl = 1$ et de mauvais résultats pour $p_correl = 0$. En d'autres termes, lorsqu'on ajoute que des corrélations *fausses négatives* i.e $p_correl = 1$ dans la matrice $matE$, les algorithmes proposent, dans la majorité des cas, un line graphe LG_k dont ces erreurs de corrélation sont supprimées. Cela s'illustre dans la figure 1.11 où l'ajout de 5 corrélations *fausses négatives* influencent très peu les line graphs

proposés LG_k puisqu'ils sont identiques aux line graphes initiales LG dans 45% des cas. En revanche, ce pourcentage baisse avec beaucoup d'erreurs de corrélations. Tel est le cas pour $k = 9$ corrélations où le taux est de 24%.

Figure 1.11: Méthode de permutation aléatoire avec une fonction de correction à coût unitaire et $p_correl = 1.0$: distribution des distances line moy_DL et de Hamming moy_DH pour $k \in [6, 9]$ corrélations alterées

Par ailleurs, le mauvais résultat obtenu pour des probabilités $p < 0.5$ s'explique par le mode de fonctionnement de l'algorithme de correction. En effet, cet algorithme consiste à ajouter des arêtes au voisinage du sommet à corriger puis de supprimer certaines arêtes pour éviter qu'un sommet n'appartienne à plus de 2 cliques (propriété du line graphe).

Nous pensons que le meilleur compromis est la probabilité $p_correl = 0.7$ parce que, pour peu de corrélations modifiées ($k < 5$), les line graphes produits LG_k et générés LG diffèrent de $k < 6$ arêtes correspondant aux k erreurs de corrélations effectuées et au-delà $k \geq 6$, le nombre d'arêtes différentes est fonction du nombre de corrélations faites multiplié par 1.5. Cela signifie qu'il faut, dans la matrice de corrélation, 30% de corrélations *fausses positives* et 70% de corrélation *fausses négatives*.

Que se passe-t-il si on priorise l'ajout de corrélations *fausses positives* à chaque traitement c'est-à-dire l'ajout d'arêtes? En d'autres termes, la suppression d'arêtes *fausses négatives* a un coût moins important que celui des *fausses positives*.

1.4.5 Impact de la fonction de coût sur les distributions

Nous définissons quatre fonctions de coût: unitaire ($n=0$), normal ($n=1$), quadratique ($n=2$), puissance 4 ($n = 4$) selon l'expression suivante

$$F_n = \begin{cases} prob[(a_i, a_j)]^n \\ (1 - prob[(a_i, a_j)])^n \end{cases} \quad (1.5)$$

ou $prob[(a_i, a_j)]^n$, la corrélation entre les arêtes a_i et a_j , correspondant au coût d'ajout de corrélations fausses négatives, $(1 - prob[(a_i, a_j)])^n$ au coût d'ajout de corrélations fausses positives.

Étant donnée que nous avons utilisé des matrices binaires dans la génération de line graphes, nous assignons des probabilités pour chaque type de corrélation tel que:

- $prob[(a_i, a_j)] = [0, 0.5[$: corrélation vrai négative i.e $0 \rightarrow 0$
- $prob[(a_i, a_j)] = [0.5, 0.79[:$ corrélation fausse négative i.e $1 \rightarrow 0$
- $prob[(a_i, a_j)] = 0.8$: corrélation fausse positive i.e $0 \rightarrow 1$
- $prob[(a_i, a_j)] =]0.8, 1]$: corrélation vrai positive i.e $1 \rightarrow 1$

La figure 1.12 affiche les courbes des différentes fonctions de coût selon les k corrélations modifiées pour $p_correl = 0.5$.

Les 4 courbes sont superposées pour $k \leq 5$ et au-delà de $k > 5$, les courbes carrée ($n=2$) et normal ($n=1$) ont les plus petites distances moyennées de Hamming. Pour $k > 9$, la courbe unitaire ($n=0$) évolue de manière exponentielle tandis que les courbes quadratique ($n=2$) et normal ($n=1$) ont

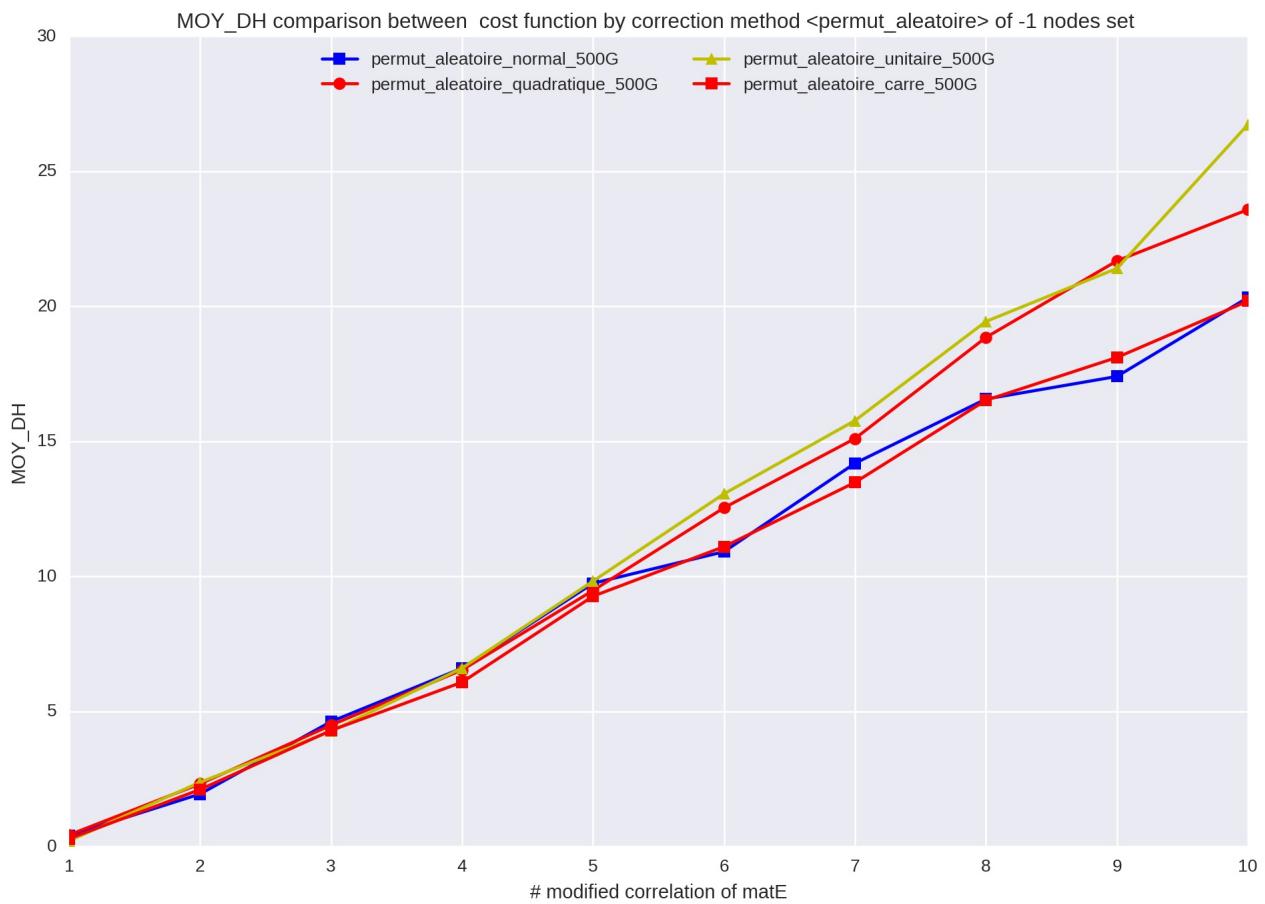


Figure 1.12: Comparaison des différentes fonctions de coût avec l’ajout de $k \in [1, 9]$ de corrélations fausses positives et fausses négatives pour une probabilité $p = 0.5$ avec la méthode de permutation aléatoire

un rapport de 2 entre les distances de Hamming et le nombre k de corrélations modifiées. On en déduit que la pire fonction de coût est la fonction unitaire ($n=0$ en jaune).

Cependant, nous ne pouvons pas choisir la meilleure fonction de coût car les fonctions des courbes *normal* et *quadratique* évoluent identiquement. On peut conclure qu’utiliser les probabilités de corrélations pour le calcul du coût améliore significativement les distances moyennées de Hamming. On choisit pour la suite la fonction de coût normal F_1 pour le coût de modification de chaque arête.

Cas particulier: fonction de coût en cloche

Souvenons nous que corriger la matrice binaire de corrélation consiste à modifier les cases de cette matrice par leurs valeurs contraires. Le cas idéal serait la modification des corrélations fausses positives et fausses négatives pendant la phase de correction. Cependant, pendant cette phase, la correction est effectuée autant sur les corrélations fausses positives et fausses négatives que sur celles vrai positives et vrai négatives. L’algorithme de correction ne fait aucune distinction entre les corrélations fausses et les bonnes corrélations. Afin de prioriser la modification des corrélations

fausses, nous définissons une nouvelle fonction de coût appelée *cloche*.

La fonction *cloche* se définit ainsi:

$$F_c = |4 \cdot ((p - s) - (s - 0.5))^2|^{1.5} \quad (1.6)$$

La fonction de coût cloche est une fonction polynomiale de degré 2 qui applique des poids aux arêtes de sorte que les arêtes, dont les corrélations avoisinent le seuil, sont moins pénalisées et les arêtes, dont les corrélations sont éloignées du seuil, ne sont quasi jamais utilisées pendant la phase de correction. Les poids minimaux sont appliqués pour les corrélations fausses tandis que les poids maximaux ($= 1$) sont appliqués aux bonnes corrélations. Cette fonction dépend de la valeur de corrélation p entre deux arêtes et du seuil s à partir duquel on introduit des corrélations erronées.

Les algorithmes (couverture et correction) sont exécutées avec la fonction cloche sur les graphes générés précédemment. Les différentes distances line/Hamming obtenues, comparées avec les fonctions de coûts unitaires F_0 et normales F_1 , sont résumées dans la figure 1.13. En effet, pour 10 et 20 erreurs de corrélation, l'algorithme de correction modifie de 11 à 22 corrélations pour la fonction normale F_1 et aussi de 38 à 65 pour la fonction en cloche respectivement. Nous en déduisons que le coût de la fonction F_1 est inférieur à celui de la fonction en cloche. La figure 1.13 est une bonne illustration par la position de la courbe rouge (F_1) en dessous de celle en jaune (*cloche*).

Ce résultat provient de l'utilisation des arêtes fausses négatives parce que le coût de ces arêtes est faible par rapport au coût des arêtes de corrélations fausses positives. En effet les corrélations fausses négatives et fausses positives appartiennent aux intervalles $[0, s[$ et $]s, 1]$ respectivement. Alors qu'on a une décroissance faible de 0 à s et une croissance forte de s à 1 dans la courbe de F_c . Ce qui explique le coût faible des fausses négatives et celui élevé des fausses positives. Par ailleurs, la courbe en bleu (fonction unitaire F_0) est au dessus des deux autres courbes (F_1 et cloche) signifiant que son coût est le plus élevé des trois fonctions.

On en conclut, par transitivité, que la fonction F_1 fournit les coûts minimum pendant la correction.

1.5 Simulation des graphes Iourtes

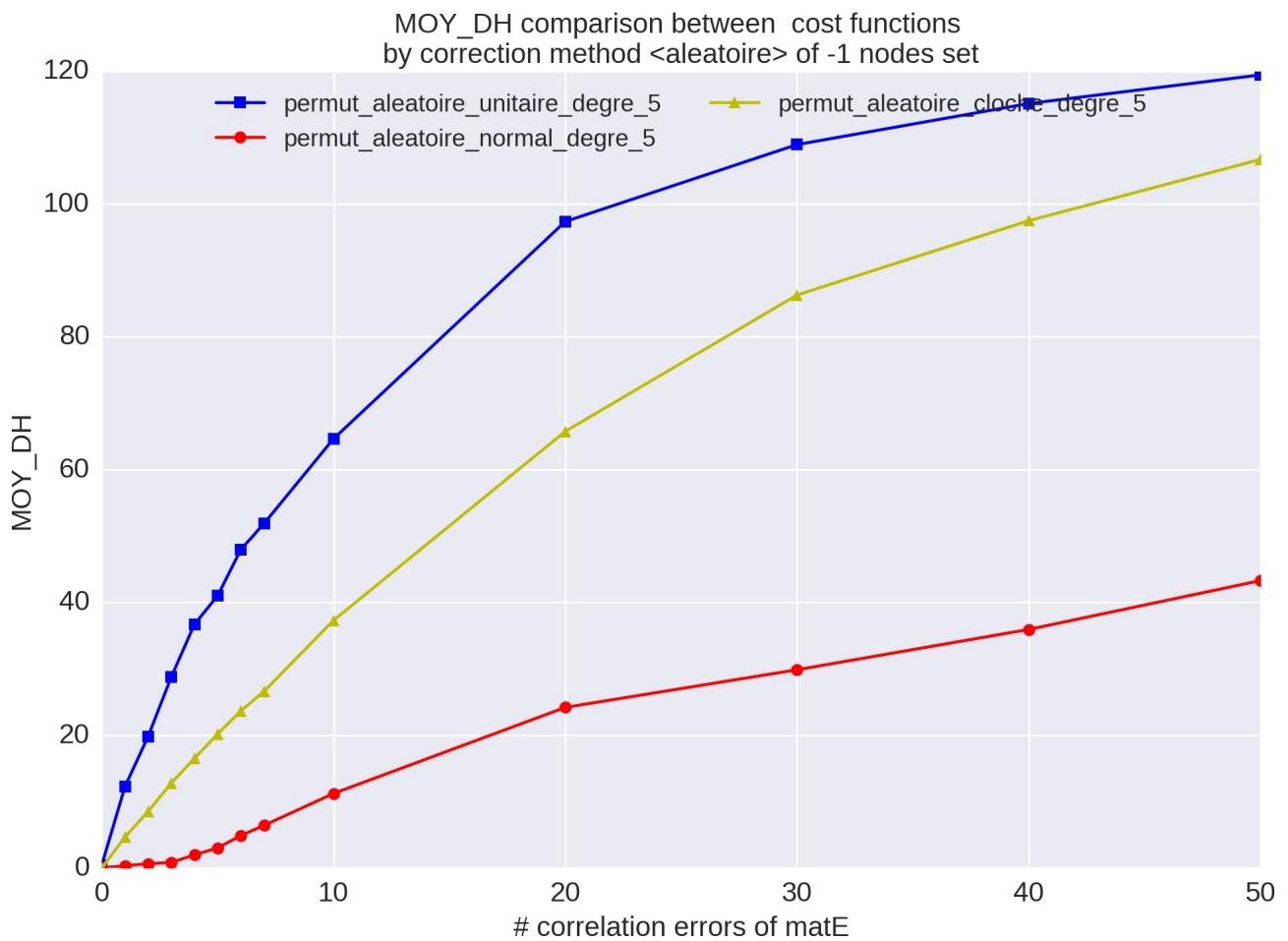


Figure 1.13: Comparaison des fonctions de coût unitaire, normal et en cloche avec l'ajout de $k \in [1, 9]$ de corrélations fausses positives et fausses négatives pour une probabilité $p = 0.5$ avec la méthode de permutation aléatoire