

Chapitre6 : Expérimentations sur des données aléatoires

Wilfried Ehounou

July 5^e 2018

Contents

1	Évaluation des performances des algorithmes	5
1.1	Génération de graphes électriques	5
1.2	Expérimentation 1 : modification de k cases de la matrice du line-graphe	6
1.2.1	Sélection de k cases et génération de la matrice $M_{k,p}$	6
1.2.2	Protocole d'expérimentation sur les graphes $G_{k,p}$	6
1.2.3	Analyses des résultats	9
1.2.3.1	Interprétation du mode de correction <i>aléatoire sans remise</i>	9
1.2.3.2	Comparaison des modes de correction	14
1.2.3.3	Influence des cases modifiées et de la fonction de coût	16
1.2.3.4	Relation entre la distance de Hamming et la distance de correction	19
1.2.4	Conclusion de l'expérimentation 1	21
1.3	Expérimentation 2 : Ajout de probabilité dans la matrice du line-graphe	22
1.3.1	Affectation de probabilités aux cases de la matrice M_{LG}	22
1.3.2	Génération du graphe de corrélation	24
1.3.3	Protocole d'expérimentation des graphes G_s de corrélation	26
1.3.4	Analyses des résultats	26
1.3.4.1	Évolution du pourcentage de cases corrigées	26
1.3.4.2	Influence de la valeur du seuil	29
1.3.4.3	Choix de la fonction de coût et impact sur les distances de Hamming	29
1.3.5	Conclusion de l'expérimentation 2	32
1.4	Expérimentation 3 : algorithmes sur les grilles bouclées	32
1.4.1	Définition des grilles bouclées et les distances line théoriques	33
1.4.1.1	Définition de la grille bouclée $G_{k,k'}$	33
1.4.1.2	Correction des grilles bouclées	33
1.4.1.3	Modification par <i>ajout d'arêtes uniquement</i>	34
1.4.1.4	Modification par <i>suppression d'arêtes uniquement</i>	35
1.4.2	Protocole d'expérimentation sur les grilles bouclées	36
1.4.3	Analyse des résultats	37
1.4.4	Conclusion de l'expérimentation 3	39
1.5	Conclusion du chapitre 1	39
1.6	Annexes	40

Chapter 1

Évaluation des performances des algorithmes

Dans ce chapitre, nous générerons de topologies de réseaux électriques qui sont des DAG sans circuits. À partir de ces topologies, nous construisons leurs line-graphes que nous modifions selon deux approches. La première approche consiste à changer les valeurs de k cases choisies aléatoirement. La seconde approche construit une matrice associée au line-graphe du DAG dont chaque case contient une valeur de probabilité puis applique une valeur de seuil sur cette matrice pour en déduire une matrice d'adjacence. De ce fait, cette matrice d'adjacence contient des cases modifiées qui seront désignées par *erreurs*.

Notre objectif est d'évaluer les performances de notre couple d'algorithmes sur ces line-graphes modifiés c'est-à-dire la capacité de nos algorithmes à corriger les *erreurs*. Pour ce faire, nous divisons ce chapitre en quatre parties. La première partie décrit la génération de graphes électriques (les DAG) et la construction de leurs line-graphes associés. Ensuite, la seconde partie présente les différentes étapes de la modification des k cases des line-graphes, le protocole d'expérimentation et l'analyse des résultats. La troisième partie analyse les performances des algorithmes sur des line-graphes modifiés par la deuxième approche. Enfin, dans la dernière partie, nous analysons ces performances sur des graphes dit *grilles bouclées*. Dans ces graphes, chaque sommet n'est couvert par aucune clique.

1.1 Génération de graphes électriques

La topologie du réseau électrique est représentée par un graphe orienté sans circuit G . Les câbles électriques sont unidirectionnels et les équipements sont toujours alimentés par une source. Ce qui implique que le courant se propage dans une direction et cette direction indique l'orientation des arcs d'un *DAG* (*Directed Acyclic Graph*). Nous allons décrire comment nous générerons le graphe G .

Considérons un graphe non orienté $G = (V, E)$ dans lequel V est l'ensemble de n sommets, E l'ensemble des m arêtes et α son degré moyen choisi. La probabilité d'existence d'une arête entre deux sommets est de $\frac{\alpha}{n}$. Afin de générer un tel graphe après avoir choisir n et α , nous sélectionnons deux sommets x et y de V et nous générerons une valeur p_{xy} qui suit une loi de probabilité uniforme. Si p_{xy} est supérieure à la probabilité d'existence d'une arête alors nous ajoutons l'arête (x, y) à E . Si G n'est pas connexe, nous choisissons aléatoirement un sommet dans chaque composante connexe et nous ajoutons une arête entre ces sommets. Nous obtenons alors m arêtes.

Afin d'orienter G comme un *DAG*, nous sélectionnons de façon aléatoire quatre sommets de degré minimum pour les définir comme les sources de notre tri topologique. Nous effectuons ce tri avec un parcours en largeur *Breadth First Search (BFS)* dans le graphe G . Chaque sommet x obtient un ordre topologique D_x et l'arête e_{xy} devient soit l'arc a_{xy} si $D_x < D_y$ soit l'arc a_{yx} si $D_x > D_y$. Les arcs a_{xy} forment l'ensemble A des arcs de G . Nous en déduisons que $G = (V, A)$ est orienté et son line-graphe LG est construit à partir de la définition ??.

Nous notons M_G la matrice d'adjacence de G et M_{LG} la matrice d'adjacence du line-graphe LG .

1.2 Expérimentation 1 : modification de k cases de la matrice du line-graphe

1.2.1 Sélection de k cases et génération de la matrice $M_{k,p}$

Nous modifions k cases de la matrice d'adjacence M_{LG} . Ces cases sont choisies de manière aléatoire. Afin de contrôler la proportion des cases à modifier dont la valeur initiale est 0 ou 1, nous introduisons la probabilité p .

Soit donc $p \in [0, 1]$ la variable qui désigne la proportion de cases à 0 sélectionnées. La proportion de cases à 1 est donc $1 - p$. Par exemple, $p = 0.5$ signifie que 50% des cases sélectionnées sont des cases à 0 et 50% des autres cases sélectionnées sont des cases à 1. De même, les k cases sont des cases à 1 si $p = 0$ et elles sont des cases à 0 si $p = 1$. Avec la répartition p , nous calculons les nombres n_0 de cases à 0 et n_1 de cases à 1. Ces cases sont à modifier dans M_{LG} . Ensuite nous sélectionnons uniformément n_0 cases à 0 et n_1 cases à 1 dans la matrice M_{LG} . Les cases à 0 sont changées en 1 et les cases à 1 sont changées en 0. La nouvelle matrice d'adjacence $M_{k,p}$ contient quatre types de cases :

- si $M_{k,p}[i, j] = M_{LG}[i, j] = 0$ alors $M_{k,p}[i, j]$ est dit *vrai négatif*.
- si $M_{k,p}[i, j] = M_{LG}[i, j] = 1$ alors $M_{k,p}[i, j]$ est dit *vrai positif*.
- si $M_{k,p}[i, j] = 0$ et $M_{LG}[i, j] = 1$ alors $M_{k,p}[i, j]$ est dit *faux négatif*.
- si $M_{k,p}[i, j] = 1$ et $M_{LG}[i, j] = 0$ alors $M_{k,p}[i, j]$ est dit *faux positif*.

La matrice $M_{k,p}$ est la matrice d'adjacence du graphe $G_{k,p}$ et ce graphe a le même ensemble de sommets que LG mais leur ensemble d'arêtes diffère de k arêtes. Généralement, $G_{k,p}$ n'est pas un line-graphe. Toutefois, s'il est un line-graphe alors il est impossible que $G_{k,p}$ soit le line-graphe de G .

1.2.2 Protocole d'expérimentation sur les graphes $G_{k,p}$

L'application de nos algorithmes de découverte et de correction débute par la génération de 500 topologies électriques G de 30 sommets, chacune ayant un degré maximal moyen de $\Delta(G) = 5$. Nous construisons 500 line-graphes LG de 150 sommets et 470 arêtes, en moyenne.

Nous avons donc introduit trois paramètres k, p, α_{max} :

1. Le paramètre k désigne le nombre de cases modifiées dans la matrice M_{LG} . Dans notre étude, $k \in \{1, \dots, 9\}$.

1.2. EXPÉRIMENTATION 1 : MODIFICATION DE k CASES DE LA MATRICE DU LINE-GRAPHE⁷

2. Le paramètre p désigne la proportion de cases à 0 sélectionnées parmi les k cases. Dans notre étude, $p \in \{0.1, \dots, 0.9\}$.
3. Le paramètre α_{max} désigne le nombre de graphes générés pour un couple (k, p) de valeurs données.

Nous choisissons $\alpha_{max} = 5$ pour des temps de calculs réalistes. Chaque graphe généré est identifié par une valeur de $\alpha \in \{1, \dots, \alpha_{max}\}$, est noté $G_{k,p,\alpha}$ et sa matrice d'adjacence est $M_{k,p,\alpha}$. La valeur α désigne l'indice de modification de k cases dans la matrice M_{LG} pour une valeur de p donnée. Il est utilisé pour faire varier les k cases choisies dans un graphe. Nous appliquons notre couple d'algorithmes sur une matrice $M_{k,p,\alpha}$ et nous obtenons la matrice $M'_{k,p,\alpha}$ qui est la matrice d'adjacence du line-graphe $LG_{k,p,\alpha}$.

Pour comparer les arêtes entre les graphes $LG_{k,p,\alpha}$ et $G_{k,p,\alpha}$, nous calculons la distance de correction (section [?]) notée $DC_{k,p,\alpha}$. De même, le nombre d'arêtes différentes entre les graphes $LG_{k,p,\alpha}$ et LG définit la distance de Hamming $DH_{k,p,\alpha}$. Nous définissons par les variables $moy_DH_{k,p}$ et $moy_DC_{k,p}$, les moyennes respectives des distances de Hamming (notée $DH_{k,p,\alpha}$) et des distances de correction (notée $DC_{k,p,\alpha}$) pour une valeur donnée de k et pour tout $\alpha \in \{1, \dots, \alpha_{max}\}$.

$$moy_DH_{k,p} = \sum_{\alpha=1}^{\alpha_{max}} DH_{k,p,\alpha} ; moy_DC_{k,p} = \sum_{\alpha=1}^{\alpha_{max}} DC_{k,p,\alpha} \quad (1.1)$$

Les différentes étapes de l'expérimentation sont résumées dans la figure 1.1. Les étapes sont représentées par des graphes et les phases de modification de ces graphes sont désignées par les flèches unidirectionnelles. Quant aux flèches bidirectionnelles (en rouge), elles indiquent le calcul de distances (de Hamming et de correction).

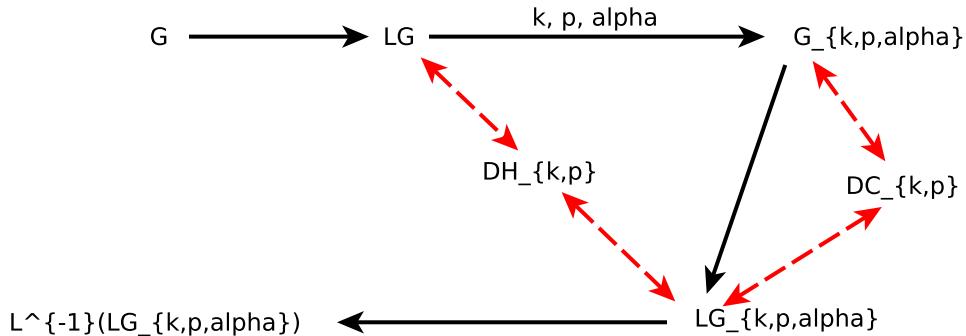


Figure 1.1: Étapes de l'expérimentation :

- 1) On génère le graphe G et son line-graphe LG ;
- 2) On modifie k cases selon la répartition p α fois pour obtenir le graphe $G_{k,p,\alpha}$;
- 3) On applique les algorithmes de couverture et de correction pour avoir un line-graphe $LG_{k,p,\alpha}$. $LG_{k,p,\alpha}$ et $G_{k,p,\alpha}$ diffèrent de $DC_{k,p,\alpha}$ arêtes. $LG_{k,p,\alpha}$ a $DH_{k,p,\alpha}$ cases modifiées par rapport à LG ;
- 4) $L^{-1}(LG_{k,p,\alpha})$ est le graphe racine de $LG_{k,p,\alpha}$.

Soit \mathcal{C} l'ensemble des sommets n'étant couverts par aucune clique après l'algorithme de couverture. La correction de la matrice $M_{k,p,\alpha}$ est nécessaire s'il existe des sommets appartenant à \mathcal{C} . Nous distinguons deux modes d'exécution de notre algorithme de correction:

1. Mode avec remise :
 1. Exécution algorithme de couverture, **return** \mathcal{C}
 2. **Tant que** \mathcal{C} n'est pas vide
 3. Correction d'un sommet de \mathcal{C}
 4. Exécution algorithme de couverture, **return** \mathcal{C}

2. Modes sans remise :
 1. Exécution algorithme de couverture, **return** \mathcal{C}
 2. **Tant que** \mathcal{C} n'est pas vide
 3. Correction d'un sommet de \mathcal{C}
 4. Mise à jour du sommet de \mathcal{C}

À chaque étape 3 dans les deux modes, un sommet de \mathcal{C} est choisi selon :

- (a) *Degré minimum* : le sommet de degré minimum est sélectionné.
- (b) *Coût minimum* : le sommet de coût de compression minimum est sélectionné. Le coût de compression est la somme des coûts de chaque case modifiée.
- (c) *Aléatoire* : le sommet est sélectionné aléatoirement parmi les sommets de \mathcal{C} .

La correction de chaque sommet de \mathcal{C} implique l'ajout et la suppression des arêtes du graphe. Nous souhaitons orienter les décisions de l'algorithme de correction de telle sorte qu'il ajoute ou supprime uniquement des arêtes ou qu'il réalise les deux opérations. Nous priorisons une opération en attribuant des coûts différents à la modification d'arêtes. Nous distinguons trois types d'opérations que nous appelons *fonctions de coût* :

- (i) *Unitaire* : l'ajout et la suppression d'une arête ont un même coût c'est-à-dire 1.
- (ii) *Ajout* : l'ajout d'une arête a un coût de 1 et la suppression a un coût de 2.
- (iii) *Suppression* : l'ajout d'une arête a un coût de 2 et la suppression a un coût de 1.

Nous rappelons que la distance de correction n'est pas la somme de toutes les modifications d'arêtes réalisées. En effet, une arête supprimée et ajoutée plusieurs fois (pour différents sommets de \mathcal{C}) n'est comptabilisée qu'une fois et son coût est appliqué selon la fonction de coût.

Étant donnée que nous avons 3 fonctions de coût, 2 modes de correction et 3 sélections possibles des sommets, nous nous retrouvons avec 18 approches de correction de sommets et il est fastidieux de les interpréter sur une même figure. Ainsi nous nous limitons à la fonction de coût *unitaire* dans un premier temps et nous considérons les approches de correction suivantes : 1a, 1b, 2a, 2b, 2c. La lecture de l'approche de correction (1a) est le suivant : 1) avec remise et a) le degré minimum. L'approche (1c) n'est d'aucune utilité car l'algorithme de correction ne parvient pas à fournir un line-graphe. En effet, l'ensemble \mathcal{C} croît linéairement à chaque étape de correction et la correction devient une boucle infinie. Le tableau 1.1 résume les approches de correction retenues dans l'analyse des performances de l'algorithme de correction.

Nous recherchons le mode qui traite le problème *Proxi-Line* c'est-à-dire le mode qui majore la distance line de $G_{k,p}$ par la distance de correction entre $LG_{k,p}$ et $G_{k,p}$. Une fois le meilleur mode trouvé, nous comparons les fonctions de coût i , $2i$ et $3i$ avec ce mode pour trouver l'influence de la fonction de coût sur les distances de correction.

1.2. EXPÉRIMENTATION 1 : MODIFICATION DE K CASES DE LA MATRICE DU LINE-GRAPHE

Table 1.1: tableau récapitulatif des approches de corrections

Mode	choix sommets	fonction de coût
Sans remise	degré minimum	unitaire ajout suppression
	coût minimum	unitaire ajout suppression
	aléatoire	unitaire ajout suppression
Avec remise	degré minimum	unitaire ajout suppression
	coût minimum	unitaire ajout suppression

1.2.3 Analyses des résultats

Nous débutons l’interprétation de nos résultats par l’analyse des distributions des distances de Hamming avec l’approche de correction *aléatoire sans remise* ($2c$) et la fonction de coût *unitaire* ($3i$). Ensuite, nous expliquons le choix de l’approche ($2c$) pour la correction des sommets. Nous présentons également le meilleur compromis dans la répartition des k cases modifiées et la relation existante entre les distances de correction et de Hamming. Enfin, nous montrons que l’approche ($2c$) fournit des distributions de distances de correction et de Hamming identiques, quelles que soient la fonction de coût (i), ($2i$), ($3i$) et la répartition k choisies.

1.2.3.1 Interprétation du mode de correction *aléatoire sans remise*

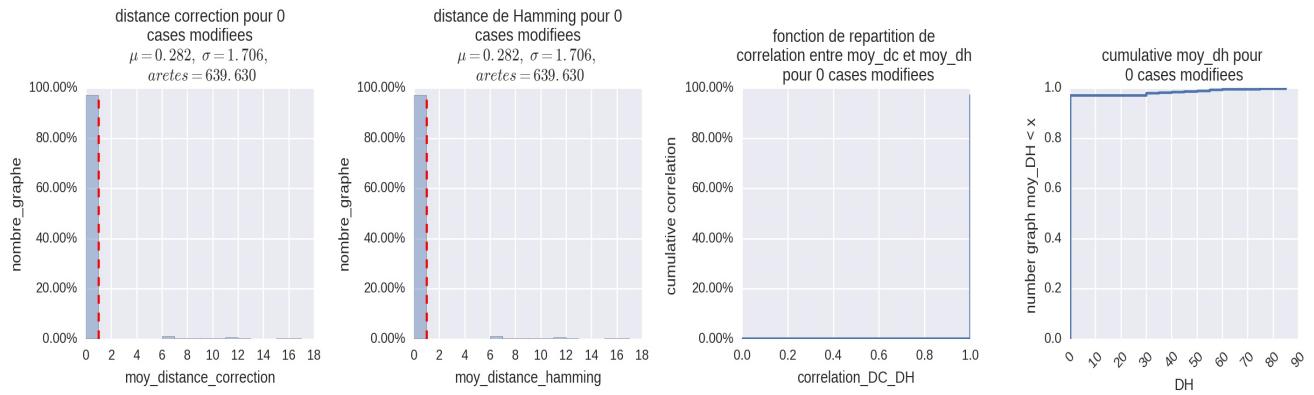


Figure 1.2: Approche de correction aléatoire sans remise à coût unitaire pour $k = 0$ case modifiée. La première colonne représente la distribution des distances correction $moy_DC_{0,0.5}$. La seconde colonne est la distribution des distances de Hamming $moy_DH_{0,0.5}$. La troisième colonne est la fonction de répartition de la corrélation entre les distances de correction et de Hamming avec en abscisse la corrélation entre ces distances ($correlation_DC_DH$). La quatrième colonne est la fonction cumulative des distances de Hamming.

Nous supposons que $p = 0.5$ c'est-à-dire qu'il y'a autant de cases *fausses négatives* que de cases *fausses positives* dans la matrice $M_{k,p}$.

Nous représentons les distributions des distances de correction et de Hamming, la fonction de répartition de la corrélation entre ces distances et la fonction cumulative de la distance de Hamming. La distribution des distances de correction indique la proportion de graphes $LG_{k,p,\alpha}$ qui ont le même ensemble d'arêtes que les graphes $G_{k,p,\alpha}$. En ce qui concerne la distribution des distances de Hamming, elle indique la proportion de graphes $LG_{k,p,\alpha}$ qui ont le même ensemble d'arêtes que les graphes LG . La corrélation entre les distances de correction et de Hamming, notée *correlation_DC_DH*, est calculée avec la formule 1.4. Sa fonction de répartition $F_k(x)$ indique le nombre de corrélations inférieures à une valeur de corrélation x donnée. Quant à la fonction cumulative de la distance de Hamming, elle montre l'évolution du nombre de cases modifiées de la matrice $M'_{k,p}$ en fonction du nombre de line-graphes construites LG .

Les figures 1.2 et 1.3 présentent les courbes respectives pour $k = 0$ et $k \in \{1, 2, 5, 9\}$ cases modifiées. La colonne 1 indique la distribution des distances de correction, la colonne 2 est la colonne de la distribution des distances de Hamming, la colonne 3 est associée à la fonction de répartition de la corrélation entre les distances de correction et de Hamming avec en abscisse la corrélation entre ces distances ($correlation_DC_DH$). Et la colonne 4 est celle de la fonction cumulative des distances de Hamming. Dans les colonnes 1 et 2, les distributions se divisent en deux zones:

- Zone *gauche ou améliorante* : elle correspond aux batonnets de l'intervalle $[0, k]$. Cet intervalle améliore l'ensemble $E'_{k,p,\alpha}$ des arêtes du graphe $LG_{k,p,\alpha}$ pour que $E'_{k,p,\alpha}$ soit identique à l'ensemble E_{LG} des arêtes du graphe LG . Si $moy_DH \rightarrow 0$ alors $LG_{k,p,\alpha}$ est très proche de LG . Si $moy_DH \rightarrow k$ alors les matrices des graphes $LG_{k,p,\alpha}$ et LG diffèrent de k cases et ces cases sont les k cases modifiées dans LG .
- Zone *droite ou dégradante* : elle correspond aux batonnets de l'intervalle $]k, +\infty[$. Cet intervalle détériore l'ensemble $E'_{k,p,\alpha}$ des arêtes du graphe $LG_{k,p,\alpha}$. Ainsi, le line-graphe $LG_{k,p,\alpha}$ s'éloigne de LG quand $moy_DH_{k,p} \rightarrow +\infty$.

1.2. EXPÉRIMENTATION 1 : MODIFICATION DE K CASES DE LA MATRICE DU LINE-GRAPHE11

Ces deux zones sont séparées par une droite en pointillée d'équation $y = k$. Cette droite désigne le nombre de cases modifiées dans le line-graphe LG .

Pour $k = 0$ case modifiée, nous vérifions que nos algorithmes sont cohérents c'est-à-dire que la phase de correction est inutile. En effet, nous avons 100% de graphes $G_{k,p,\alpha}$, $LG_{k,p,\alpha}$, LG qui ont les mêmes ensembles d'arêtes et cela implique que $moy_DH_{0,0.5} = moy_DC_{0,0.5} = 0$. D'où le seul batonnet dans les colonnes 1 et 2. Par ailleurs, la fonction de répartition de la corrélation et la fonction cumulative des distances de Hamming sont définies par les équations 1.2 et 1.3 respectivement.

$$F_k(x_1) = \begin{cases} 0 & \text{si } x_1 < 1 \\ 100 & \text{si } x_1 = 1 \end{cases} \quad (1.2)$$

$$y_{cumulDH}^0(x) = 1 \quad \text{si } x \in \mathbb{N} \quad (1.3)$$

avec x_1 la corrélation entre les distances et x le nombre d'arêtes modifiées. Les valeurs des distances de Hamming sont égales à 0 donc sa fonction cumulative $y_{cumulDH}^k$ vaut 1. L'équation 1.2 s'interprète comme suit : $F_k(x) = 100\%$ des line-graphes ont leurs distances de correction et de Hamming correlées ($x = 1$).

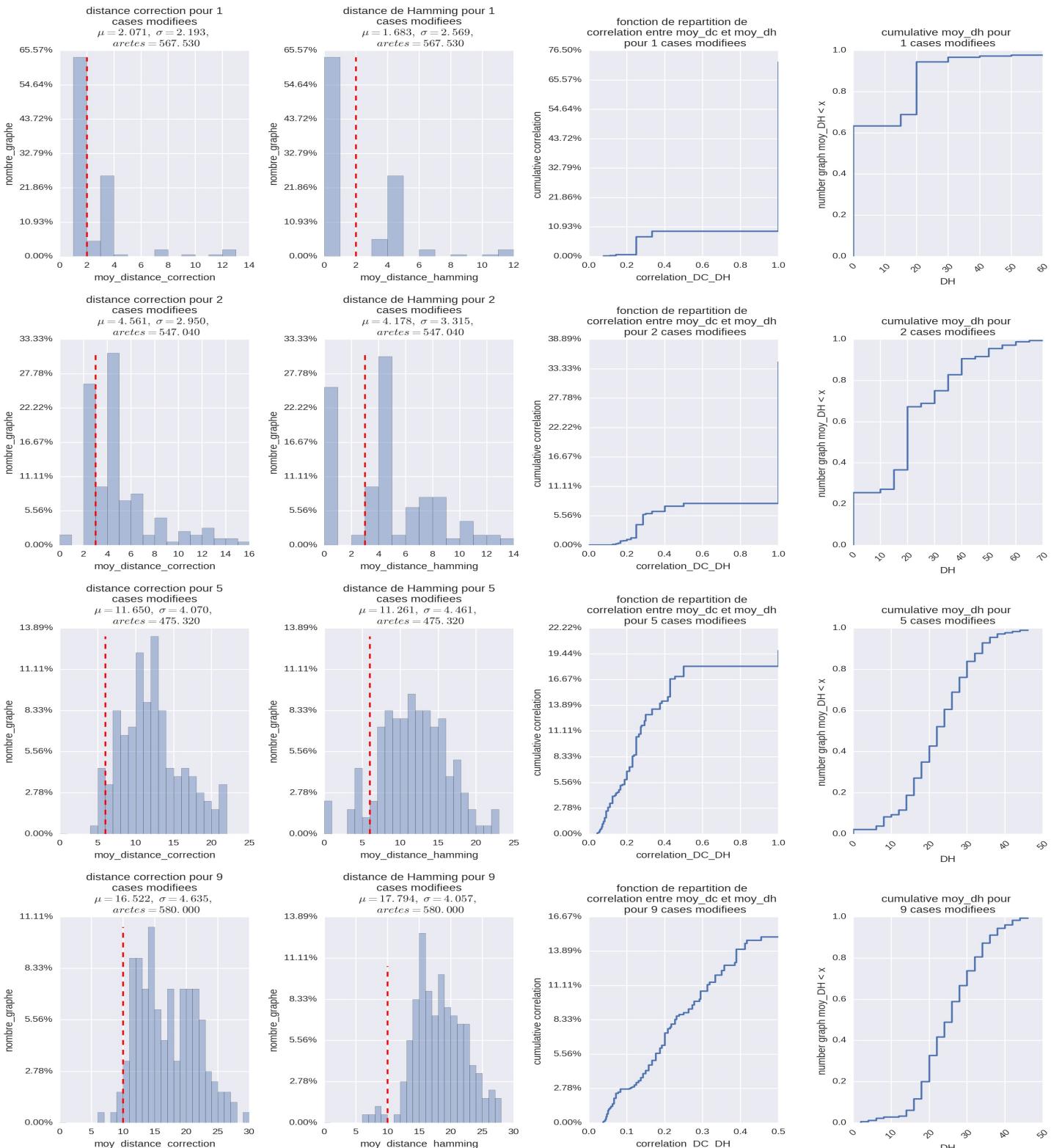


Figure 1.3: Approche de correction aléatoire sans remise à coût unitaire pour $k = \{1, 2, 5, 9\}$ cases modifiées : La première colonne représente la distribution des distances de correction $moy_DC_{k,0.5}$. La seconde colonne est la distribution des distances de Hamming $moy_DH_{k,0.5}$. La troisième colonne est la fonction de répartition de la corrélation entre les distances de correction et de Hamming avec en abscisse la corrélation entre ces distances (correlation_DC_DH). La quatrième colonne est la fonction cumulative des distances de Hamming. La première ligne est associée à $k = 1$ case modifiée, la seconde ligne à $k = 2$ cases modifiées, la troisième ligne à 5 cases modifiées et enfin la dernière à 9 cases modifiées.

1.2. EXPÉRIMENTATION 1 : MODIFICATION DE K CASES DE LA MATRICE DU LINE-GRAPHE13

Pour $k \in \{1, 2\}$, le pic des histogrammes se localise dans la zone *améliorante* des colonnes 1 et 2 de la figure 1.3 et son pourcentage est supérieur à 50%. Les autres batonnets sont dans la zone *dégradante* et leur pic a un pourcentage inférieur à 10% en moyenne. Dans la colonne 1 de la figure 1.3, le pic correspond aux k cases modifiées du graphe $G_{k,p,\alpha}$ et son pourcentage est identique à celui du pic de la colonne 2. Le pic de la colonne 2 correspond à $moy_DH_{k,0.5} = 0$ et signifie que LG et $LG_{k,p,\alpha}$ ont le même ensemble d'arêtes. Ainsi, les $k \leq 2$ cases modifiées sont supprimées de la matrice $M'_{k,p,\alpha}$ lorsque $moy_DC_{k,0.5} \leq k$. Cependant, les distances de correction et de Hamming ont approximativement les mêmes valeurs lorsque $moy_DC_{k,0.5} > k$ ($moy_DC_{k,0.5}$ est dans la partie *dégradante* de la figure 1.3). Cela s'explique par le fait que les distances $moy_DC_{k,0.5}$ et $moy_DH_{k,0.5}$ sont corrélées. Nous détaillons la notion de corrélation de distances dans la section 1.2.3.4. Ainsi, les distances de correction et de Hamming sont corrélées dans $\eta_k = 5\%$ des line-graphes $LG_{k,p,\alpha}$ et le line-graphe $LG_{k,p,\alpha}$ devient le line-graphe initial LG si nous corrigons les DC cases modifiées du graphe $LG_{k,p,\alpha}$. La variable η_k est la proportion de line-graphes dont les distances de correction et de Hamming sont fortement corrélées.

Pour $k = 5$, le pic se trouve toujours dans la zone *améliorante* des colonnes 1 et 2 de la figure 1.3 mais son pourcentage baisse significativement à 15.69% (voir la ligne 3). Le nombre de line-graphes dans la zone *dégradante* dans les colonnes 1 et 2 augmente tout comme les distances de correction et de Hamming qui atteignent jusqu'à 45 arêtes. La variable η_k est égale à 18.38% de line-graphes $LG_{k,p,\alpha}$ (voir ligne 3 de la colonne 3 de la figure 1.3). Cette augmentation provient de la baisse du pourcentage du pic de la zone *améliorante* au profit de la zone *dégradante* et la plupart des graphes appartenant à cette zone ont leurs distances de correction et de Hamming corrélées. Les 15.69% de line-graphes $LG_{k,p,\alpha}$ identiques à LG s'expliquent par le type de cases modifiées et l'emplacement des arêtes dans le graphe LG . En effet, ces cases modifiées sont des *fausses négatives* et ces arêtes supprimées n'appartiennent pas à des cliques voisines. Toutefois, quelque soit le type de cases modifiées, notre couple d'algorithmes ajoutent beaucoup d'arêtes pour obtenir le line-graphe $LG_{k,p,\alpha}$ lorsque les cases sont reparties avec $p = 0.5$. Par exemple, nous avons constaté, en moyenne, 10 à 20 arêtes différentes pour $k = 5$ cases modifiées dans nos expérimentations. Par ailleurs, nous remarquons qu'il existe des graphes dans lesquels la distance de correction est inférieure à k . Tel est le cas pour $k = 5$ ou nous avons $moy_DC_{k,0.5} = 3$ et $nombre_graphe = 0.14\%$ dans la colonne 1 de la figure 1.3. En effet, les cases modifiées sont des cases *fausses négatives*. Ces arêtes supprimées de LG appartiennent à la même clique et certaines arêtes sont ajoutées de tel sorte que la clique se partitionne en deux cliques.

Enfin, pour $k = 9$, la distribution est dans la zone *dégradante* et le pic est à $moy_DC_{k,0.5} = 23$ cases modifiées avec un pourcentage de 6% line-graphes. En comparant les pourcentages des distances $moy_DC_{k,0.5}$ et $moy_DH_{k,0.5}$, nous constatons qu'ils ne sont pas identiques comme pour $k \leq 5$. En effet, certaines cases modifiées ne sont pas corrigées car l'algorithme de correction modifie énormément de cases qui sont différentes des k cases et ce taux croît quand $moy_DC_{k,0.5}$ est élevé. Le taux de cases corrigées est de 53.38% en moyenne. La variable η_k passe à 22% de line-graphes $LG_{k,p,\alpha}$ parce que la correction a modifiée des cases différentes des k cases. Cependant les distances de correction et de Hamming restent toujours corrélées et 3% line-graphes $LG_{k,p,\alpha}$ ont le même ensemble d'arêtes que LG (voir ligne 4 de la colonne 3 de la figure 1.3). C'est pourquoi, les courbes de $F_k(x)$ et $y_{cumulDH}^k$ ont cette forme enrobée proche de la fonction sigmoïde de paramètre $\lambda \leq -15$. Pour rappel, nous précisons que ces courbes tendent vers la fonction suivante $f_\lambda(x) = \frac{1}{1+e^{\lambda*(x-0.5)}}$.

Les cases modifiées $k \in \{1, \dots, 9\}$ sont présentées dans les figures 1.21 et 1.22 de l'annexe 1.6.

L'approche de correction (2c) nous montre que les distances de correction et de Hamming se dégradent quand le nombre k de cases modifiées augmentent. En fait, pour $k \leq 5$, le nombre de line-graphes $LG_{k,p,\alpha}$ identiques à LG est supérieur à $nombre_graphe = 25\%$ avec des distances de correction $moy_DC_{k,0.5} = k$. Pour des distances $moy_DC_{k,0.5} = 2 \times k$, les k cases modifiées sont corrigées. Des cases érronnées sont ajoutées pendant l'exécution de l'algorithme de correction mais elles sont peu nombreuses. Nous pouvons considérer ces cases comme la précision de notre algorithme car ces cases indiquent le nombre de cases à modifier pour obtenir LG . Ainsi la distance de correction est majorée par le nombre de cases k modifiées. Cependant, au-delà de $k > 5$, la distance de correction $moy_DC_{k,0.5}$ double et moins de 50% des k cases sont corrigées. Les cases érronnées ajoutées proviennent de l'ajout et la suppression d'arêtes dans le line-graphe $LG'_{k,p,\alpha}$ étant donnée que la fonction de coût est *unitaire*. La distance de correction est majorée par $2 \times k$ en moyenne.

Conclusion : l'approche de correction *aléatoire sans remise* (2c) propose des line-graphes $LG_{k,p}$ identiques à LG quand $k \leq 5$. Dans ce cas, le problème *Proxi-Line* est traité car la distance line est majorée par k . Toutefois, le nombre de cases à corriger après l'exécution de notre couple d'algorithmes est faible lorsque la distance de correction est inférieure au double de k ($moy_DC_{k,0.5} = 2 \times k$) avec $k > 5$. Cela conduit à majorer la distance line par le double des cases érronnées. En outre, pour $k > 5$, le pourcentage η_k de corrélation entre $moy_DC_{k,0.5}$ et $moy_DH_{k,0.5}$ croît. Nous allons étudier l'évolution de η_k dans le paragraphe 1.2.3.4 mais nous commençons par la comparaison des différentes approches de correction pour en déduire celle qui minimise les distances de correction ou de Hamming.

1.2.3.2 Comparaison des modes de correction

Nous recherchons la meilleure approche de correction parmi les cinq énumérées dans le tableau 1.1. Pour ce faire, nous disposons des distributions des distances de correction et de Hamming, des fonctions de répartition de ces distributions et aussi des moyennes de distances de correction et de Hamming associées aux k cases modifiées. Les distributions des distances de correction et de Hamming sont obtenues avec $p = 0.5$ et la fonction de coût est *unitaire*. Les distributions de distances de chaque approche sont regroupées dans les colonnes 1 et 2 dans les figures en annexes 1.6. Nous décidons d'utiliser la moyenne des distances de Hamming pour la comparaison de approches de correction parce qu'il est facile de déterminer le nombre de cases modifiées étant donnée que nous connaissons les line-graphes LG et $LG_{k,p}$.

Soit $G_{k,p,\alpha}^i$ le i^{ieme} graphe généré contenant k cases modifiées la α^{ieme} fois avec la répartition $p = 0.5$ des k cases. Nous le notons $G_{k,\alpha}^i$ avec $0 \leq i \leq 500$ et $\alpha \leq \alpha_{max}$. Le line-graphe de $G_{k,\alpha}^i$ obtenu après l'algorithme de correction est noté $L(G_{k,\alpha}^i)$.

Soit DH_k^i la distance de Hamming entre LG et $L(G_{k,\alpha}^i)$.

La variable $moy_DH_k^i$ est la moyenne de DH_k^i pour les α_{max} graphes $G_{k,\alpha}^i$ et la variable moy_DH_k est la moyenne de $moy_DH_k^i$ pour 500 graphes contenant k cases modifiées. La figure 1.4 affiche les courbes des différents approches de correction pour des distances de Hamming moyennées moy_DH_k en fonction des k cases modifiées. En ordonné, nous avons le nombre de cases différentes entre deux line-graphes.

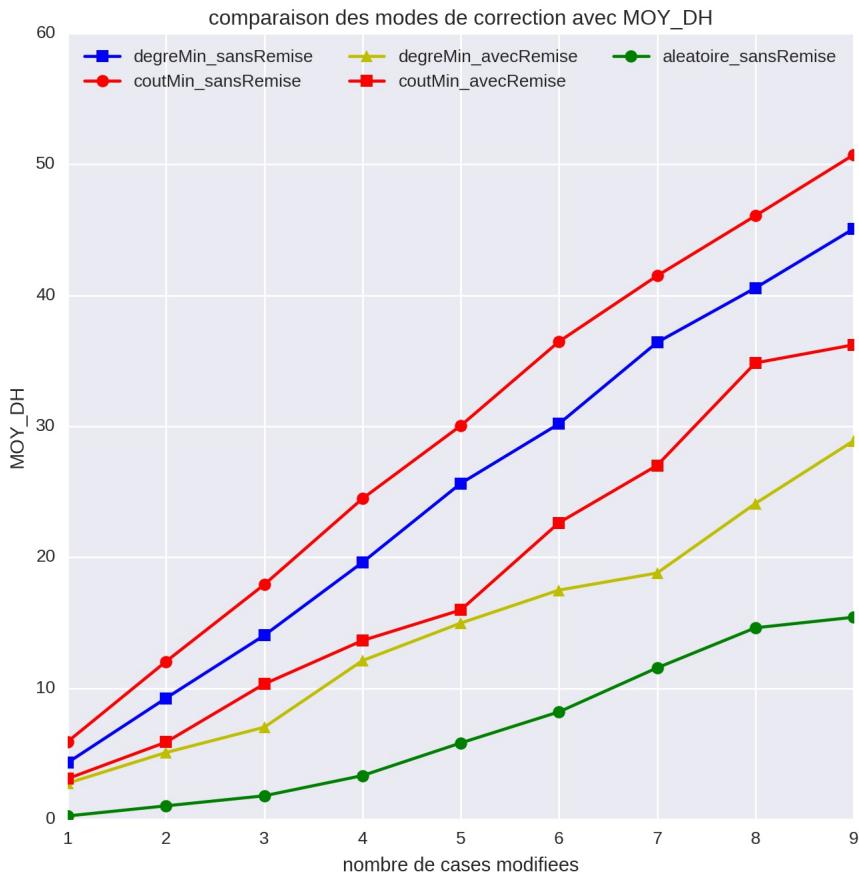


Figure 1.4: Comparaison des différentes approches de correction de sommets pour $k \in \{1, \dots, 9\}$ cases modifiées. Les courbes en bleu carré : approche degré minimum sans remise (2a), rouge carrée : approche coût minimum avec remise (1b), rouge rond : approche coût minimum sans remise (2b), vert rond : approche aléatoire sans remise (2c) et jaune triangle : approche degré minimum avec remise (1a)

Considérons des courbes associées aux approches (2b), (2c) et (1b). En choisissant les nombres de cases modifiées $k \in \{4, 8\}$, nous avons $moy_DH \in \{4, 15\}$ cases pour l'approche (2c), $moy_DH \in \{13, 36\}$ cases pour l'approche (1b) et $moy_DH \in \{25, 46\}$ cases pour l'approche (2b). Pour $k = 4$ cases modifiées, l'approche (1b) modifie, en moyenne, 9 cases de plus que l'approche (2c). En revanche, ce nombre moyen de cases modifiées augmente à 21 cases quand $k = 8$. De même, l'approche (2b) modifie 12 cases de plus que l'approche (1b) pour $k = 4$ cases modifiées et 10 cases pour $k = 8$ cases. L'approche (2c) donne de meilleures résultats par rapport aux approches (1b) et (2b).

Conclusion : l'approche *aléatoire sans remise* propose de meilleurs résultats que les approches (2a), (2b), (1a) et (1b) parce que les distances de correction sont minimales pour toute valeur de k comme le montre la figure 1.4. Nous retenons, pour la suite, l'approche *aléatoire sans remise* comme l'approche de correction des sommets de \mathcal{C} , sommets n'appartenant à aucune couverture.

1.2.3.3 Influence des cases modifiées et de la fonction de coût

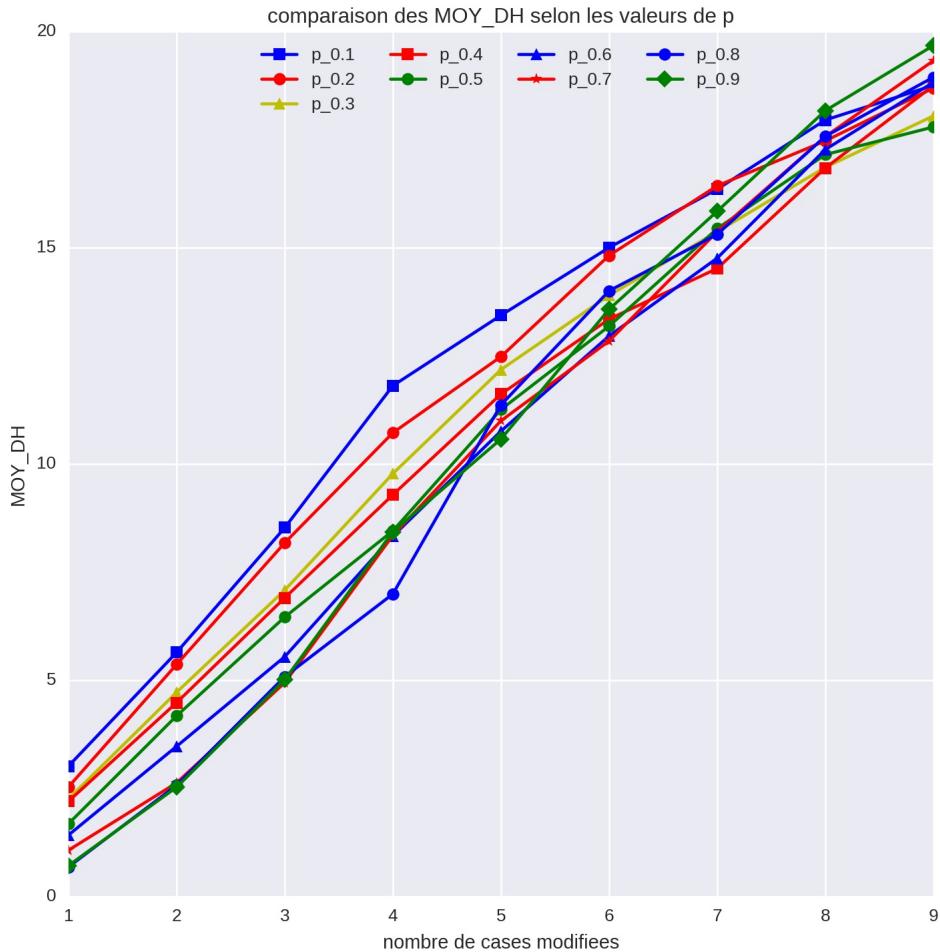


Figure 1.5: Comparaison des différentes répartitions des $k \in \{1, \dots, 9\}$ cases fausses positives et fausses négatives avec le mode *aléatoire sans remise* et la fonction de coût *unitaire*.

Nous mesurons l'influence des fonctions de coût sur nos distances de Hamming. Pour ce faire, nous appliquons les fonctions de coût *unitaire*, *ajout* et *suppression* (voir tableau 1.1) pour en déduire les valeurs de p qui sont favorables à l'algorithme de correction c'est-à-dire qui minimisent les distances de Hamming.

Nous considérons d'abord la fonction *unitaire*. La figure 1.5 représente l'évolution des distances de Hamming selon les différentes valeurs de p . Les courbes de p sont éloignées pour $k \leq 5$ et au-delà de $k > 5$, les courbes se rapprochent. En effet, pour $k \leq 5$, 43.4% des cases *fausses négatives* en moyenne sont corrigées pour $p \in \{0.7, 0.9\}$ et pour $p = 0.8$, cette moyenne est de 45.5%. Quant aux cases *fausses positives*, seulement 10% des cases sont corrigées. La différence entre k et MOY_DH correspond aux cases non corrigées et expliquent les valeurs moyennées du nombre de cases corrigées. Ces moyennes sont plus élevées que celles obtenues avec $p < 0.7$. En effet,

1.2. EXPÉRIMENTATION 1 : MODIFICATION DE K CASES DE LA MATRICE DU LINE-GRAPHE17

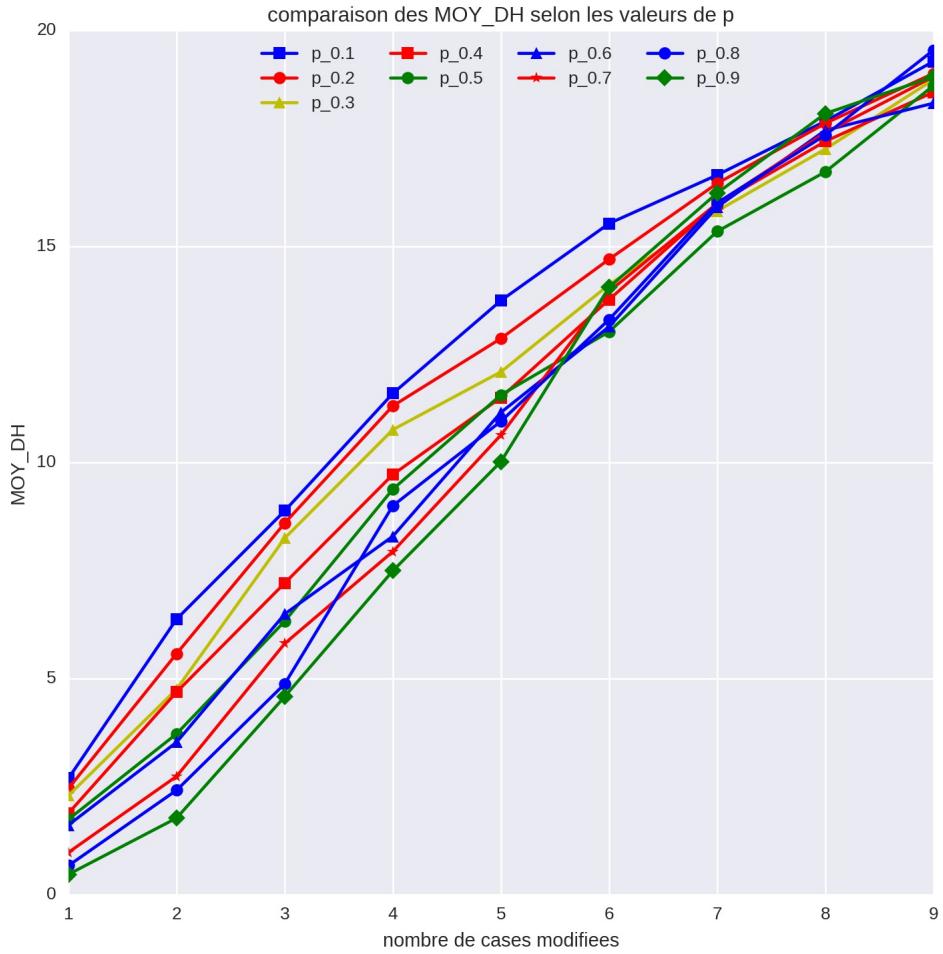


Figure 1.6: Comparaison des différentes répartitions des $k \in \{1, \dots, 9\}$ cases fausses positives et fausses négatives avec le mode *aléatoire sans remise* et la fonction de coût *suppression*.

seulement 19.32% des cases *fausses positives* et 38% des cases *fausses négatives* sont corrigées avec $p < 0.7$. L'algorithme ajoute beaucoup d'arêtes pour $p \geq 0.7$ par rapport $p < 0.7$ parce que le nombre de cases *fausses négatives* dans la matrice du graphe de corrélation est élevé pour $p < 0.7$ et le coût d'ajout d'arêtes qui est de 1.

De même, pour $k > 0.5$, 80% des cases érronnées sont des cases *fausses négatives* après l'algorithme de correction. L'algorithme privilégie l'ajout à la suppression d'arêtes.

La fonction *unitaire* a peu d'influence sur les variations des distances de Hamming quelque soit les valeurs de p .

La figure 1.6 correspond à la fonction *suppression* et elle a le même comportement que la fonction *unitaire* parce que toutes les courbes divergent quand $k \leq 5$ et convergent quand $k > 5$. Ici nous remarquons aussi que nous avons en moyenne beaucoup de cases *fausses négatives* en moyenne à la fin de l'algorithme de correction. Avec la fonction *suppression*, nous constatons que le nombre moyen de cases *fausses négatives* à la fin de l'algorithme de correction est largement

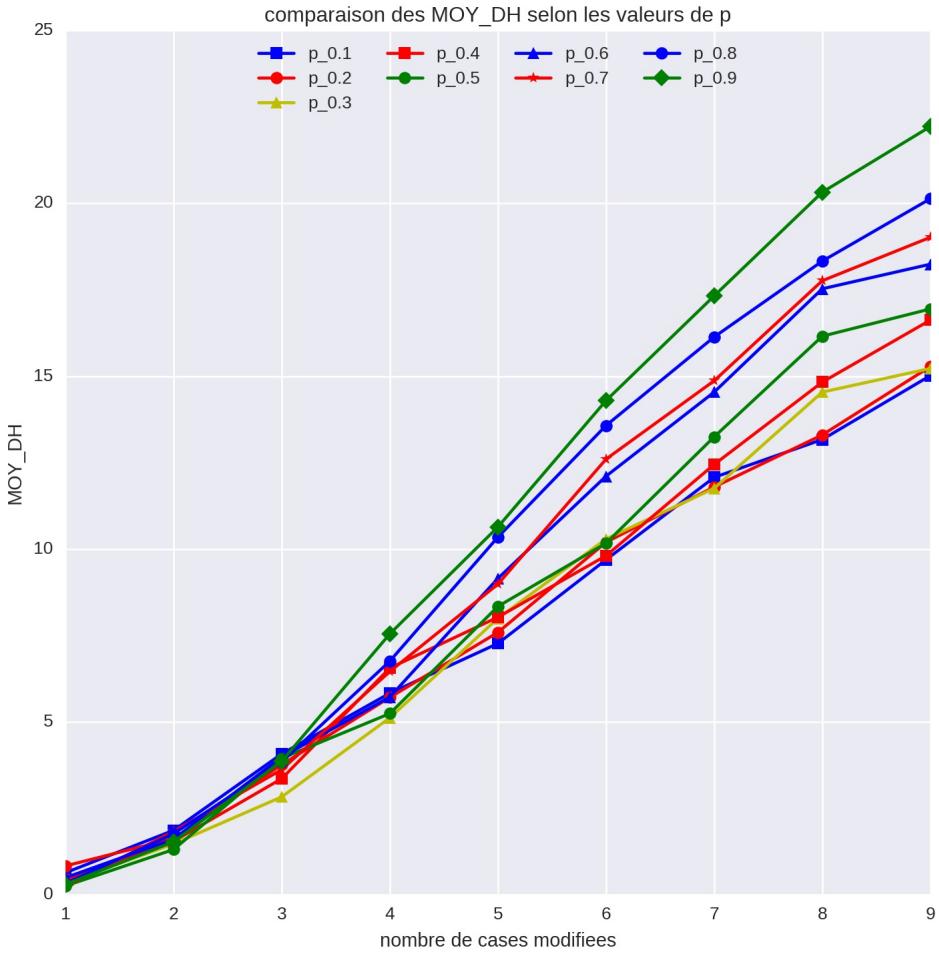


Figure 1.7: Comparaison des différentes répartitions des $k \in \{1, \dots, 9\}$ cases fausses positives et fausses négatives avec le mode *aléatoire sans remise* et la fonction de coût *ajout*.

supérieure au nombre de cases *fausses négatives* modifiées dans la matrice $M_{k,p,\alpha}$. Cependant, le nombre de ces cases *fausses négatives* (39.4% en moyenne) à la fin de l’algorithme de correction est inférieur au nombre de cases corrigés *fausses négatives* (47.7% en moyenne) avec la fonction *unitaire*. Cette baisse provient majoritairement de la position des sommets à corriger dans \mathcal{C} . En effet, il existe des chaînes simples de longueur 2 entre certains sommets de \mathcal{C} . Une arête de chacune des chaînes est supprimée afin que la compression de deux cliques fournisse une nouvelle clique de \mathcal{CC} . Cela fait que les arêtes incidentes ont leurs sommets de \mathcal{CC} et ces arêtes sont supprimées une fois sur deux au minimum. La fonction *suppression* donne le même résultat que la fonction *unitaire*.

Dans la figure 1.7 associée à la fonction *ajout*, les courbes divergent quand k est croissant. En effet, des arêtes *fausses négatives* sont ajoutées à $G_{k,p,\alpha}$ pour $p \geq 0.4$ pendant la correction. Alors les distances $moy_DH_{k,p}$ augmentent car le nombre de cases *fausses négatives* augmente et aussi plus de 40% des cases *fausses positives* ne sont pas supprimées. Par ailleurs, les courbes convergent vers 0 quand $k \leq 4$. La baisse des valeurs moyennes des distances de Hamming est le résultat de

1.2. EXPÉRIMENTATION 1 : MODIFICATION DE k CASES DE LA MATRICE DU LINE-GRAPHE19

la correction des cases modifiées dans $M_{k,p}$. Néanmoins, l'algorithme corrige majoritairement des cases *fausses négatives* lorsque toutes les cases modifiées ne parviennent pas à être corrigées.

Conclusion : nous ne pouvons pas conclure que les valeurs de p ont une influence sur la correction des k cases modifiées car les fonctions de coût *unitaire* et *suppression* font converger l'une vers l'autre leurs distances de Hamming avec l'augmentation du nombre de cases modifiées. Toutefois, la fonction de coût *ajout* fait diverger nos courbes sans créer un écart significatif de distances entre ses courbes.

1.2.3.4 Relation entre la distance de Hamming et la distance de correction

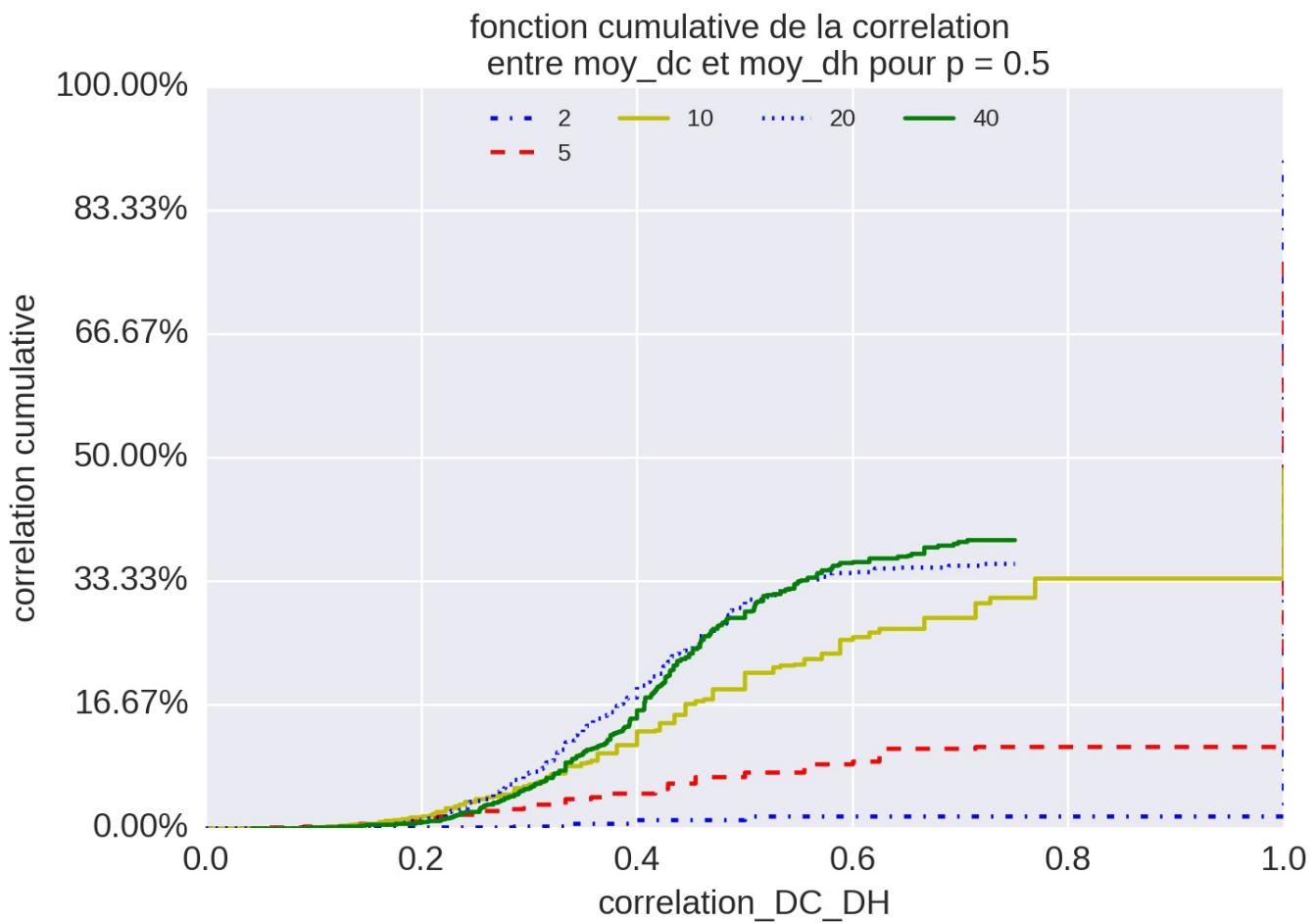


Figure 1.8: La corrélation de la distance de correction versus la distance de Hamming pour k cases modifiées et $p = 0.5$

Nous allons calculer les corrélations entre les distances de correction et de Hamming en se basant sur le graphe G , son line-graphe LG et le line-graphe $LG_{k,p,\alpha}$ obtenu par notre couple d'algorithmes. Notre objectif est de montrer la relation entre ces deux distances.

Considérons les distances de correction et de Hamming obtenues par l'approche *aléatoire sans remise*, la variable $p = 0.5$ et la fonction de coût *unitaire* (voir tableau 1.1). Nous calculons la

corrélation entre les distances de correction et de Hamming avec la formule 1.4.

$$\text{corr}_{k,p,\alpha} = \frac{|\text{moy_DC}_{k,p,\alpha} - \text{moy_DH}_{k,p,\alpha}|}{\max(\text{moy_DC}_{k,p,\alpha}, \text{moy_DH}_{k,p,\alpha})}; \text{corr}_{k,p} = \sum_{\alpha=1}^5 \text{corr}_{k,p,\alpha}; F_k(x, p) = P(\text{corr}_{k,p} < x) \quad (1.4)$$

avec $x \in [0, 1]$ une valeur de corrélation et k le nombre de cases modifiées. La fonction $\text{corr}_{k,p,\alpha}$ retourne l'écart entre ces deux distances sous la forme de valeurs probabilistes. Ainsi $\text{corr}_{k,p,\alpha} = 1$ indique qu'il n'existe aucune corrélation entre les distances de correction et de Hamming c'est-à-dire que $\text{moy_DC}_{k,p,\alpha} = k$ et $\text{moy_DH}_{k,p,\alpha} = 0$. De même, $\text{corr}_{k,p} = 0$ indique que ces distances sont identiques c'est-à-dire $\text{moy_DC}_{k,p,\alpha} = \text{moy_DH}_{k,p,\alpha}$. En moyenne, $\text{moy_DH}_{k,p,\alpha}$ tend vers k cases modifiées.

La figure 1.8 représente 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 moyenne $\text{corr}_{k,p}$ est inférieure à $x \in [0, 1]$. Si $\text{corr}_{k,p,\alpha} = 0$ alors les matrices de $LG_{k,p,\alpha}$ et LG sont différentes de k cases quand $k < 6$ ($LG_{k,p,\alpha} \neq LG$). Si $\text{corr}_{k,p,\alpha} = 1$ alors le line-graphe $LG_{k,p,\alpha}$ est le line-graphe du graphe G ($LG_{k,p,\alpha} = LG$) et $F_k(1) \approx 0$. La corrélation $\text{corr}_{k,p} = 1$ est sa valeur maximale. Ce cas est illustré dans la figure 1.8 par les courbes de $k \in \{2, 5\}$. Par exemple $F_5(1) \approx 10\%$ signifie que nous avons $70 - 10 = 60\%$ des line-graphes LG_k qui ont le même ensemble d'arêtes que les line-graphes LG (70% est le pourcentage de corrélations égales à 1 $|\text{corr}_{5,p} = 1| = 70\%$).

En revanche, une valeur de $F_k(1)$ très élevée signifie que le nombre x de $\text{corr}_{k,p} = 1$ est très faible. Ce nombre x implique une corrélation très forte entre les distances de correction et de Hamming. C'est l'observation faite avec les courbes de $k \in \{10, 20, 40\}$ de la figure 1.8 dans lesquelles nous avons une croissance continue en fonction de l'augmentation des valeurs de corrélations.

Nous subdivisons nos courbes en deux catégories:

- Celles pour lesquelles il y'a une corrélation entre les distances de correction et de Hamming (courbes de $k \in \{10, 20, 40\}$).
- Celles pour lesquelles il y'a a aucune corrélation entre ces distances parce que $LG = LG_{k,p}$ (courbes de $k \in \{2, 5\}$).

Conclusion : il existe de fortes corrélations entre les distances de correction et de Hamming lorsque $k \geq 10$. Dans ce cas, nous pouvons utiliser la distance de correction pour calculer les écarts de cases modifiées pendant l'algorithme de correction. Dans le cas où $k \leq 5$, les distances de correction inférieure ou égale à 5 cases propose le line-graphe LG et cette distance entraîne une corrélation proche de 0. Pour $k \in \{6, 7, 8, 9\}$, les valeurs de $\text{corr}_{k,p}$ avoisine 0.5. Ces valeurs sont faibles et nous ne pouvons rien conclure. Ainsi, pour juger de la qualité de notre algorithme de correction en l'absence de la distance de Hamming, la distance de correction est alors une bonne métrique.

1.2.4 Conclusion de l'expérimentation 1

Les performances de notre couple d'algorithmes ont été testées sur des graphes non-orientés sans circuits qui représentent la topologie de réseaux électriques. Nous avons construit les line-graphes de ces graphes et avons modifié k cases dans les matrices des line-graphes. Les cases modifiées sont reparties en deux ensembles (cases *fausses positives* et *fausses négatives*) selon la variable

$p \in [0, 1]$. L'analyse des performances compare les distances de correction et de Hamming en fonction du nombre de cases modifiées selon 5 approches de correction et 3 fonctions de coût (voir tableau 1.1). Nous concluons que l'approche *aléatoire sans remise* donne de meilleurs résultats lorsque le nombre k de cases modifiées est faible ($k \leq 5$). La distance line est alors majorée par la distance de correction. Le problème *Proxi-Line* a une solution mais elle n'est pas optimal. En revanche, au-delà de $k > 5$, il est alors difficile de déterminer une bonne supérieure à la distance line car l'algorithme de correction modifie des cases n'étant pas contenues dans les k cases modifiées préalablement. Par ailleurs, nous avons montré que la distance de correction peut être utilisée comme une métrique pour connaître le nombre de cases modifiées dans la matrice d'un graphe à condition que cette distance soit supérieure à 10 arêtes. Enfin, les fonctions de coût ont peu d'influences sur les distances de correction pendant la phase de correction.

1.3 Expérimentation 2 : Ajout de probabilité dans la matrice du line-graphe

Notre seconde expérimentation a le même but que la première sauf qu'ici la modification des cases est faite à partir de valeurs de seuils appliquées à des matrices de corrélation. Les valeurs de ces matrices sont définies selon la distribution des valeurs de corrélation d'un réseau réel, celui du datacenter *Champlan*.

Notre but est de déterminer la valeur de seuil qui minimise les distances de correction et de Hamming obtenues avec l'approche *aléatoire sans remise* (tableau 1.1).

Nous décrivons tout d'abord l'affectation des valeurs de corrélation aux cases de la matrice du line-graphe. Ensuite nous présentons les différentes étapes de notre expérimentation et enfin nous analysons les résultats obtenus.

1.3.1 Affectation de probabilités aux cases de la matrice M_{LG}

Soient la matrice M_{LG} d'adjacence du line-graphe LG du DAG non orienté de *Champlan* et M_c sa matrice de corrélation obtenue avec la *distance de Pearson*. La distribution de valeurs de M_c est représentée par la figure 1.9. Le graphique à gauche de la figure 1.9 est la distribution des valeurs de M_c selon les cases à 1 dans M_{LG} et le graphique à droite est celui des cases à 0 dans M_{LG} . Par exemple, la valeur de corrélation 0.1 désigne toutes les valeurs appartenant à $[0.1, 0.2[$.

La densité des corrélations est décroissante pour les cases à 0 quand les valeurs de corrélation tendent vers 1. De même, cette densité est croissante pour les cases à 1 quand les corrections tendent vers 1. La matrice M_c contient des cases érronnées et ces cases ont une influence sur le calcul des valeurs de corrélation. C'est pourquoi les distributions ne sont pas linéairement croissantes ou décroissantes. Nous décidons, avec nos constatations, que la génération des distributions des valeurs de corrélation des cases à 0 et 1 suivent des lois normales asymétriques.

Nous générions des valeurs de corrélation qui sont similaires à la distribution conjecturée des valeurs de corrélation du graphe de *Champlan*. Nous supposons que les valeurs de corrélation des cases à 0 de M_{LG} suivent la loi asymétrique de paramètre $\alpha = 5$ et les valeurs de corrélation des cases à 1 de M_{LG} suivent cette même loi avec $\alpha = -5$ comme illustré dans la figure 1.10.

Soient n_0 la cardinalité de l'ensemble des cases à 0 de M_{LG} et n_1 la cardinalité de l'ensemble des cases à 1 de M_{LG} . Pour obtenir des valeurs de corrélation de M_c , nous générions des valeurs réelles

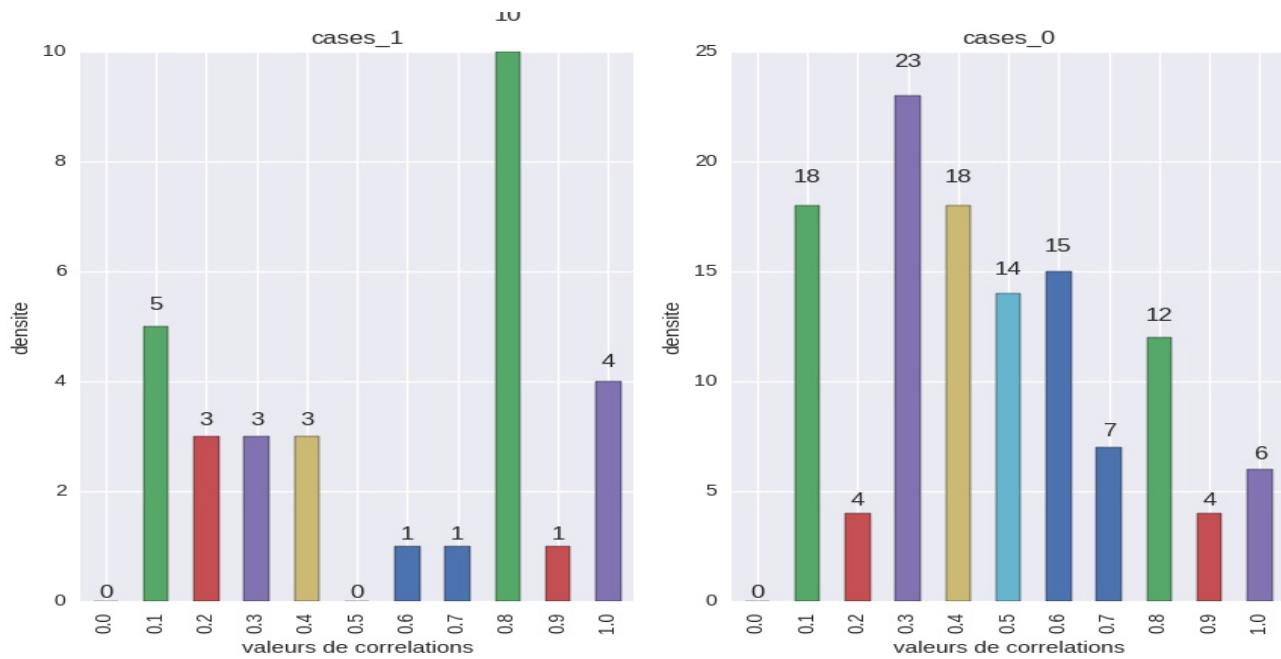


Figure 1.9: Distribution des valeurs de corrélation de M_c selon les cases à 0 (à gauche) et à 1 (à droite) de M_{LG} . La corrélation à 0.1 désigne les valeurs de corrélation comprises entre 0.10 et 0.199.

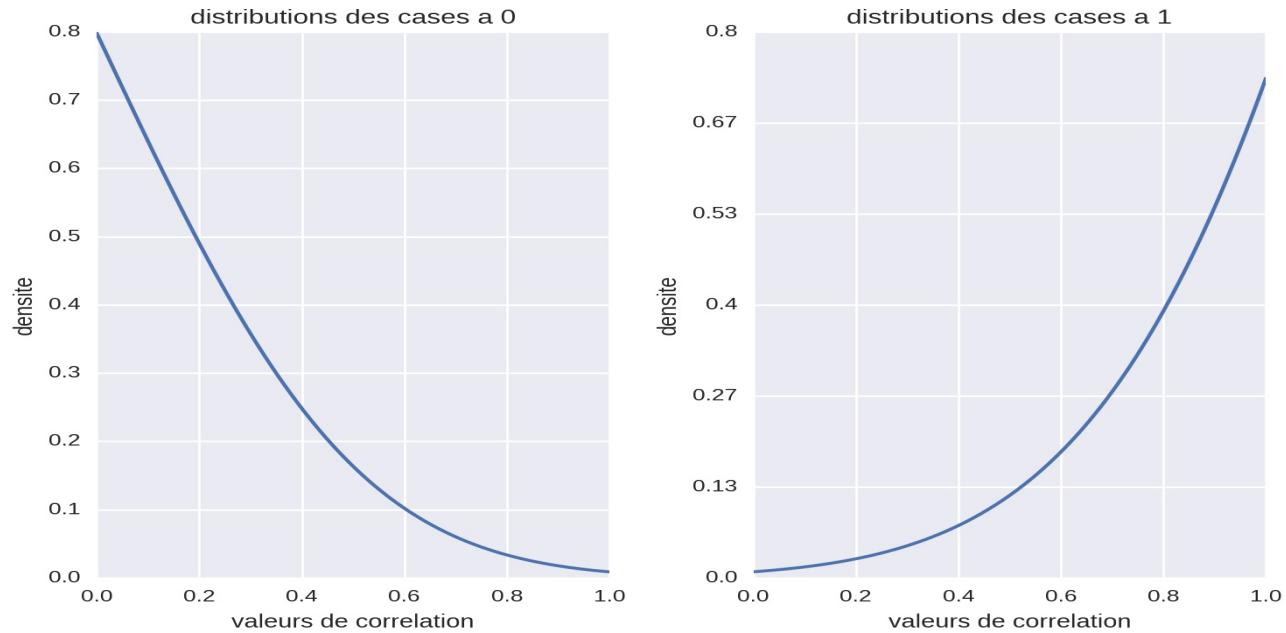


Figure 1.10: À gauche : loi asymétrique de coefficient d'asymétrie $\alpha = 5$ pour les cases à 0. À droite : loi asymétrique de coefficient d'asymétrie $\alpha = -5$ pour les cases à 1.

val appartenant à $[0, 1]$ selon la loi asymétrique de coefficient $\alpha = 5$. Nous divisons l'intervalle $[0, 1]$ en 10 sous-intervalles. Chaque sous-intervalle est noté $[x_i, x_{i+1}]$ avec x_i une valeur de corrélation et $i \in [0, 9]$. Nous calculons la densité de chaque sous-intervalle et l'ensemble des densités forme un

histogramme qui suit une loi asymétrique. Nous calculons la fonction de répartition P de chaque densité de telle sorte que

$$\forall x_i, x_{i+1}, P(X \leq x_i) \leq P(X \leq x_{i+1}) \text{ et } P(X \leq x_{10}) = 1.$$

Pour chaque case à 0, nous tirons aléatoirement n_0 nombres réels compris entre 0 et 1 uniformément. Si un nombre appartient à $]P(X \leq x_i), P(X \leq x_{i+1})]$ alors sa valeur de corrélation est x_{i+1} . Nous répétons les mêmes étapes pour les cases à 1 en générant les valeurs val avec une loi asymétrique de coefficient $\alpha = -5$. Nous obtenons la matrice de corrélation M_c du line-graphe LG . Une distribution des valeurs de corrélation est représentée par la figure 1.11 dans laquelle nous avons 189 cases à 0 et 111 cases à 1 dans LG .

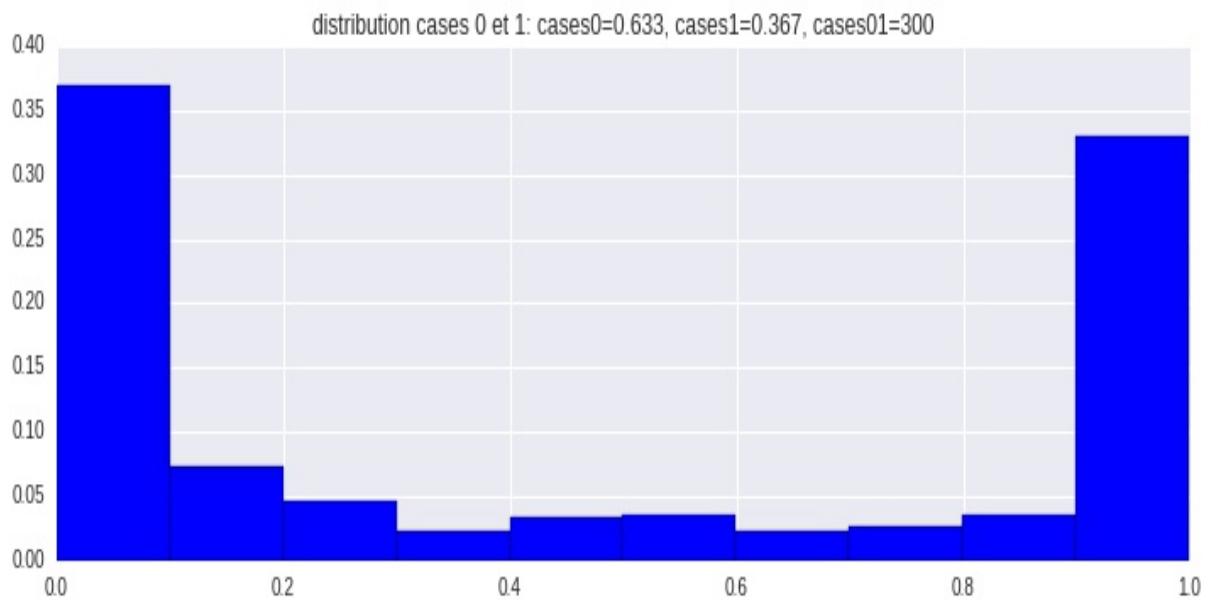


Figure 1.11: Un exemple de la distribution des valeurs de corrélations générées. 63% des cases sont des cases à 0 et 37% sont des cases à 1 dans LG .

1.3.2 Génération du graphe de corrélation

Nous allons déterminer la matrice d'adjacence du graphe de corrélation à partir des valeurs de M_c . Soit $s = \{0.1, \dots, 0.9\}$, une valeur de seuil choisie. Nous construisons la matrice M_s selon les règles suivantes :

- Si $M_c[i, j] \geq s$ alors $M_s[i, j] = 1$.
- Si $M_c[i, j] < s$ alors $M_s[i, j] = 0$.

La matrice M_s est la matrice d'adjacence du graphe G_s dit *graphe de corrélation*. Ces cases peuvent contenir des cases érronnées. Ces cases érronnées proviennent de la sélection du seuil s et de la génération de valeurs de corrélation pour les cases à 0 et à 1 du line-graphe LG . En effet ,

- Si $M_s[i, j] = M_{LG}[i, j] = 0$ alors $M_s[i, j]$ est dit *vrai négatif*.
- Si $M_s[i, j] = M_{LG}[i, j] = 1$ alors $M_s[i, j]$ est dit *vrai positif*.
- Si $M_s[i, j] = 0$ et $M_{LG}[i, j] = 1$ alors $M_s[i, j]$ est dit *faux négatif*.
- Si $M_s[i, j] = 1$ et $M_{LG}[i, j] = 0$ alors $M_s[i, j]$ est dit *faux positif*.

Un exemple de distributions des cases érronnées selon les valeurs de seuils est présenté dans la figure 1.12. Par exemple, le graphe de corrélation G_s contient 17 cases *vrai positives*, 151 cases

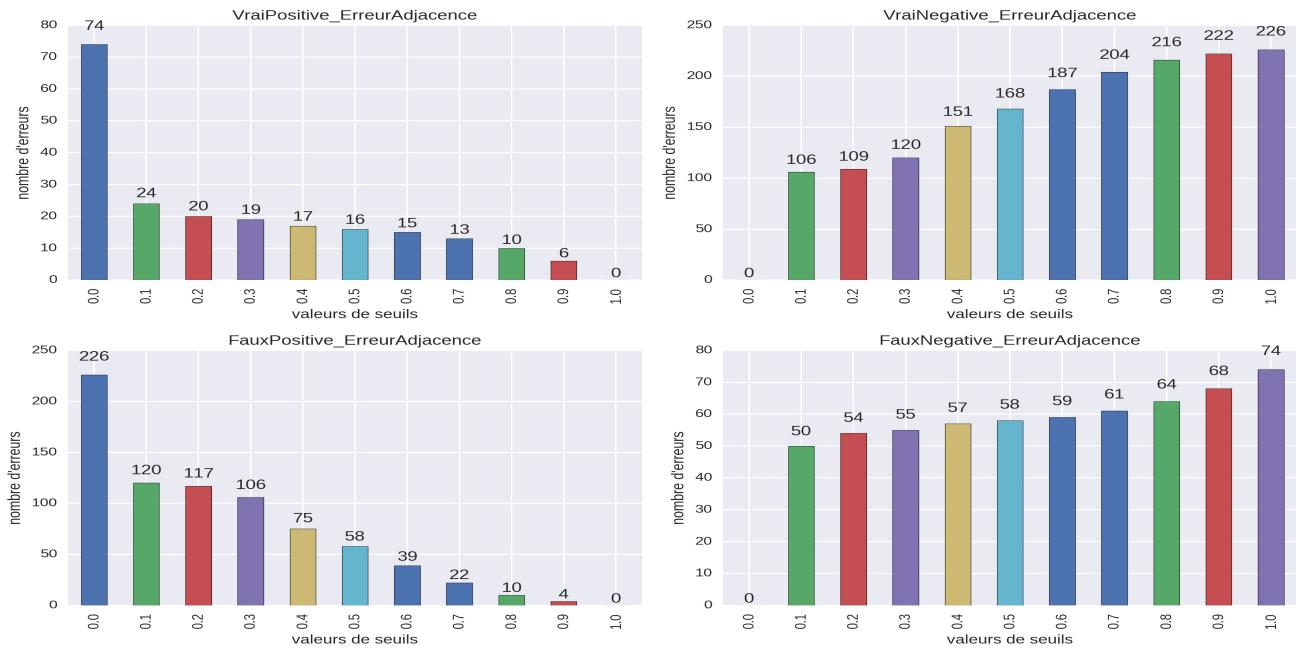


Figure 1.12: Distribution des valeurs de corrélation sur un graphe généré de 30 sommets et de degré maximal de 5. .

vrai négatives, 75 cases *fausses positives* et 57 cases *fausses négatives* pour un seuil $s = 0.4$.

1.3.3 Protocole d’expérimentation des graphes G_s de corrélation

Le but de notre couple d’algorithmes est de corriger les cases érronnées dans M_s afin que la matrice proposée M'_s soit la matrice d’adjacence d’un line-graphe LG_s et que la distance de Hamming entre LG_s et LG soit minimale. Pour ce faire, nous recherchons la valeur du seuil s qui minimise la distance de Hamming entre LG_s et LG .

Nous générerons les graphes dans les mêmes conditions que l’expérimentation 1 de la section 1.2 avec de petites modifications. D’abord, le nombre de graphes générés est de 150. Ensuite, nous construisons une matrice de corrélation dont les étapes sont décrites dans la section 1.3.1. Et enfin, l’ajout des cases érronnées est réalisé à partir d’un seuil s dans la matrice de corrélation M_e comme cela est expliqué dans la section 1.3.2. Les étapes de l’expérimentation sont résumées dans la figure 1.13.

Nous considérons que l’approche *aléatoire sans remise* (tableau 1.1) pendant l’algorithme de correction. Mais les fonctions de coûts des arêtes utilisent les valeurs de corrélation comme suit:

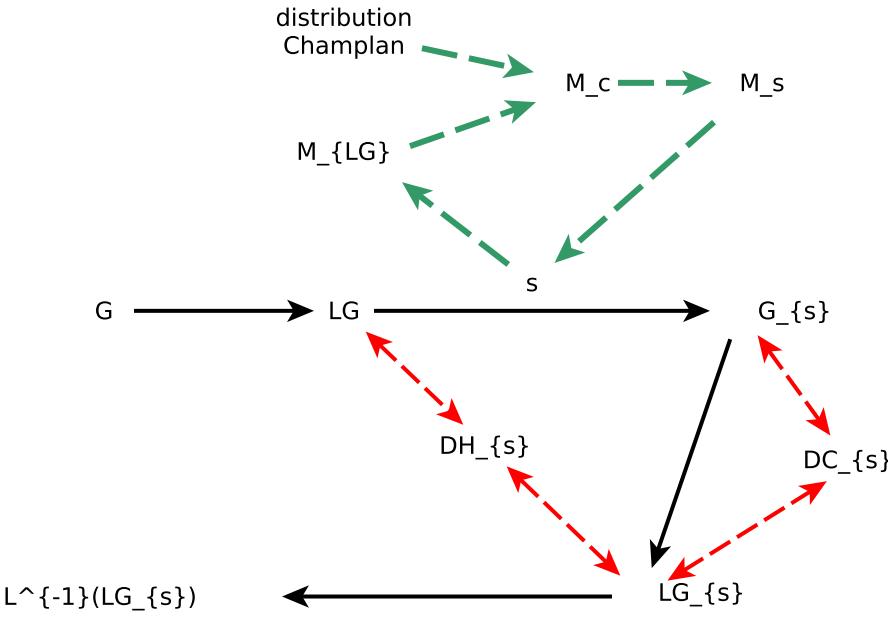


Figure 1.13: Étapes de l’expérimentation : 1) on génère le graphe G et son line-graphe LG , 2) on génère la matrice de corrélation M_c du line-graphe LG à partir de la distribution des valeurs de corrélation du graphe de Champlan puis on lui applique une valeur de seuil s pour obtenir le graphe G_s , 3) on applique les algorithmes de découverte et de correction pour avoir un line-graphe LG_s . LG_s et G_s diffèrent de DC_s arêtes. LG_s a DH_s cases modifiées par rapport à LG , 4) $L^{-1}(LG_s)$ est le graphe racine de LG_s .

- (a) *Unitaire* : l’ajout et la suppression d’une arête ont un coût de 1.
- (b) *Normale* : l’ajout d’une arête à la case $M_s[i, j]$ a un coût égal à sa valeur de corrélation $M_c[i, j]$ et la suppression d’une arête a un coût $1 - M_c[i, j]$.
- (c) *Ajout* : l’ajout d’une arête à la case $M_s[i, j]$ a un coût $M_c[i, j]$ alors que la suppression à cette case à un coût $2 \times (1 - M_c[i, j])$.
- (d) *Suppression* : la suppression d’une arête à la case $M_s[i, j]$ a un coût $1 - M_c[i, j]$ alors que l’ajout d’une arête à cette case a un coût $2 \times M_c[i, j]$.

Nous allons comparer les distances de Hamming et le pourcentage de cases corrigées pour en déduire la bonne valeur de seuil.

1.3.4 Analyses des résultats

Nous allons décrire l’évolution du pourcentage des cases corrigées en fonction de la valeur du seuil pour la fonction de coût *normale*. Ensuite nous déterminons l’influence de la valeur de seuil sur le nombre de cases corrigées et enfin nous recherchons la meilleure fonction de coût et l’influence des fonctions de coût sur l’évolution des distances de Hamming.

1.3.4.1 Évolution du pourcentage de cases corrigées

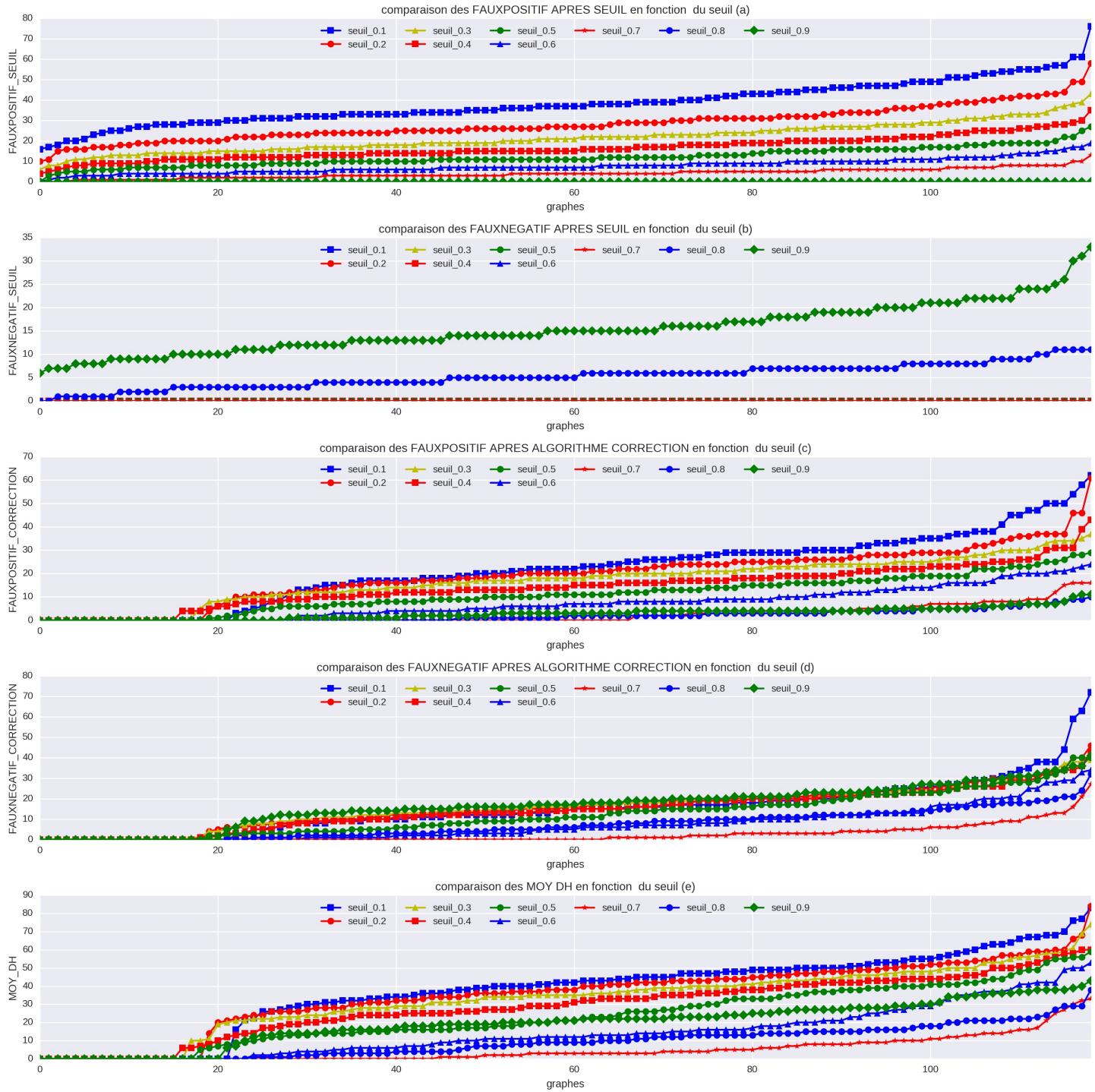


Figure 1.14: Choix du seuil: (a) cases *fausses positives* dans la matrice M_s ; (b) cases *fausses négatives* dans la matrice M_s , (c) cases *fausses positives* dans la matrice M'_s ; (d) cases *fausses négatives* dans la matrice M'_s ; (e) comparaison des seuils selon *moy_DH*

Pour mesurer le pourcentage de cases corrigés, nous allons procéder comme suit :

- Considérer les cases *fausses positives* dans la matrice M_s puis représenter le nombre de cases *fausses positives* pour chaque valeur de seuil. Le graphique (a) de la figure 1.14 correspond à

la comparaison des seuils par rapport au nombre de cases *fausses positives* avant la correction. Nous remarquons qu'il n'y a aucune case *fausses positives* dans M_s pour $s \in \{0.8, 0.9\}$. Ce nombre croît quand le seuil s décroît ($s \rightarrow 0$).

- (b) Considérer les cases *fausses négatives* dans la matrice M_s puis représenter le nombre de cases *fausses négatives* pour chaque valeur de seuil. Le graphique (b) de la figure 1.14 correspond à la comparaison des seuils par rapport au nombre de cases *fausses négatives* avant la correction. Le nombre des cases *fausses négatives* est nulle pour $s \notin \{0.8, 0.9\}$.
- (c) Considérer les cases *fausses positives* après la correction de G_s (matrice M'_s) puis représenter le nombre de cases *fausses positives* pour chaque valeur de seuil. Le graphique (c) de la figure 1.14 correspond à la comparaison des seuils par rapport au nombre de cases *fausses positives* après la correction. Le nombre de cases baisse quand $s \leq 0.7$ puis augmente $s > 0.7$.
- (d) Considérer les cases *fausses négatives* après la correction de G_s (matrice M'_s) puis représenter le nombre de cases *fausses négatives* pour chaque valeur de seuil. Le graphique (d) de la figure 1.14 correspond à la comparaison des seuils par rapport au nombre de cases *fausses négatives* après la correction. Le nombre de cases varie peu quand $s < 0.4$ puis baisse quand $s = \{0.5, 0.6\}$ avant d'atteindre sa valeur minimum à $s = 0.7$. Il augmente $s > 0.7$.
- (e) Représenter les distances de Hamming moyennes de chaque graphe pour chaque un seuil. Le graphique (e) de la figure 1.14 correspond à la comparaison des seuils en fonction de la distance de Hamming de chaque seuil. La distance de Hamming baisse quand $s \rightarrow 0.7$ avec sa valeur minimum à $s = 0.7$ puis augmente quand $s > 0.7$.

Rappelons que les différents graphiques sont rangés par ordre croissant et les distances de Hamming sont obtenues à partir la fonction de coût *normale*.

Nous distinguons trois types de seuils:

- Les seuils $s < 0.7$ qui baissent le nombre de cases *fausses positives* et augmentent *fausses négatives* après l'algorithme de correction. Ils n'ont pas d'effets réels sur la distance de Hamming car il y'a un transfert d'éléments de l'ensemble des cases *fausses positives* à celui des *fausses négatives* et vice-versa. À cet effet, on remarque qu'il y'a 20% de cases *fausses négatives* après la correction alors qu'il n'en existait aucune case *fausses négatives* avant la correction. Il en est de même avec les cases *fausses positives* dont le nombre diminue de 20% également pendant la correction. Ces seuils n'ont aucune influence sur les cases érronnées.
- Les seuils $s > 0.7$ qui augmentent le nombre de cases *fausses positives* et baisse celui des cases *fausses négatives* après l'algorithme de correction. En effet, le nombre de cases *fausses positives* est nul dans M_s parce qu'il n'y a aucune case en 1 ayant une valeur inférieure à s dans la distribution. Dans M'_s , le nombre moyen de cases *fausses positives* est de 1.76 pour $s = 0.8$ et de 2.33 pour $s = 0.9$. Il y'a ajouté d'arêtes dans le graphe LG_s parce que le nombre d'arêtes à supprimer pour corriger un sommet est très élevé et cela implique que le coût de la correction par la suppression d'arêtes est très onéreux par rapport à celui de l'ajout d'arêtes. Ainsi à chaque sommet à corriger, l'algorithme ajoute plus d'arêtes qu'il en supprime et certaines arêtes ajoutées appartiennent à l'ensemble des arêtes de LG . Cela fait baisser les cases *fausses négatives* comme nous le constatons avec les chiffres suivants : le nombre moyen de ces cases passe de 6.7 à 2.5 pour $s = 0.8$ et de 17.7 à 12.5 pour $s = 0.9$.

- Le seuil $s = 0.7$ qui diminue le nombre de cases *fausses négatives* et *fausses positives*. En effet, le nombre moyen de cases érronnées est faible < 10 avant la phase de correction et il est ≤ 5 après la correction. Nous remarquons aussi les cases érronnées sont majoritairement des cases *fausses négatives* après la correction. La présence de ces cases provient du coût de modification des arêtes car la compression π_1, π_2, π_s de coût minimale nécessite la suppression d'arêtes existantes. En effet, la matrice M_s contient que des cases *fausses positives*. Pour trouver des bipartitions cohérentes autour des sommets à corriger, il faut ajouter beaucoup d'arêtes pour chaque partition et cela fait croître le coût de la correction. Nous avons remarqué que la suppression de quelques arêtes permet d'obtenir des cliques qui correspondent à des sommets de G . L'algorithme préfère alors supprimer des arêtes car elles sont peu par rapport aux arêtes à ajouter et aussi le cout de la suppression d'arêtes est faible. La distance de Hamming obtenue avec ce seuil est minimale par rapport aux autres seuils. Ce seuil fournit alors une meilleure correction des sommets de \mathcal{C} .

Conclusion : nous pouvons conclure que la répartition des cases érronnées avant la phase de correction a une influence sur l'algorithme de correction. Ainsi le choix du seuil dans le bon intervalle permet de réduire le nombre de cases érronnées et fournit une excellente correction des sommets de \mathcal{C} . Cependant, les seuils différents du bon seuil entraîne que la fonction de coût n'a aucune influence sur les corrections parce que l'algorithme corrige peu de cases érronnées mais ajoute aussi des cases *fausses positives* et *fausses négatives* dans la même proportion.

1.3.4.2 Influence de la valeur du seuil

Les seuils inférieurs à 0.7 correspondent à une réduction des cases *fausses positives* et une augmentation de cases *fausses négatives* dans la matrice de correction. Dans ce cas, nous constatons que le nombre des cases *fausses positives* baisse énormément quand $s \rightarrow 0.7$. Ce phénomène s'explique par le coût de modifications des arêtes (normale) et le nombre faible de cases érronnées au voisinage de 0.7. L'augmentation des cases *fausses négatives* provient du fonctionnement de notre algorithme de correction. L'algorithme doit ajouter des arêtes dans une partition π_1 ou π_2 au voisinage d'un sommet pour en faire une clique. Le nombre élevé de cases *fausses négatives* nécessite l'ajout de beaucoup d'arêtes.

Les seuils supérieurs à 0.7 correspondent à l'augmentation des cases *fausses positives* et la baisse de celles *fausses négatives*. La présence de cases *fausses positives* en grand nombre entraîne l'algorithme de correction dans deux cas : l'ajout de peu d'arêtes et la suppression de beaucoup d'arêtes pour atteindre un line-graphe. Dans le premier cas, la distance de Hamming est faible mais le line-graphe est différent du line-graphe de G_s . Dans ce second cas, la distance de Hamming est très élevée et nous avons très peu de chance de retrouver le line-graphe de G_s . Quant à la baisse des cases *fausses négatives*, elle réduit le nombre de cases à modifier pour obtenir une case.

Conclusion : le meilleur compromis de seuil est le seuil qui baisse les cases *fausses positives* et les cases *fausses négatives* après l'algorithme de correction. Le seuil capable d'atteindre ce résultat est dans l'intervalle $s =]0.6, 0.7]$. Dans la suite du chapitre, nous retenons $s = 0.7$. Avec ce seuil, les distances de Hamming sont aussi les plus faibles (graphique (e) figure 1.14).

1.3.4.3 Choix de la fonction de coût et impact sur les distances de Hamming

Nous rappelons que la fonction de coût est définie par la somme des coûts des cases à modifier pour corriger chaque sommet de \mathcal{C} pendant l'algorithme de correction. Le coût de correction d'un

1.3. EXPÉRIMENTATION 2 : AJOUT DE PROBABILITÉ DANS LA MATRICE DU LINE-GRAPHE29

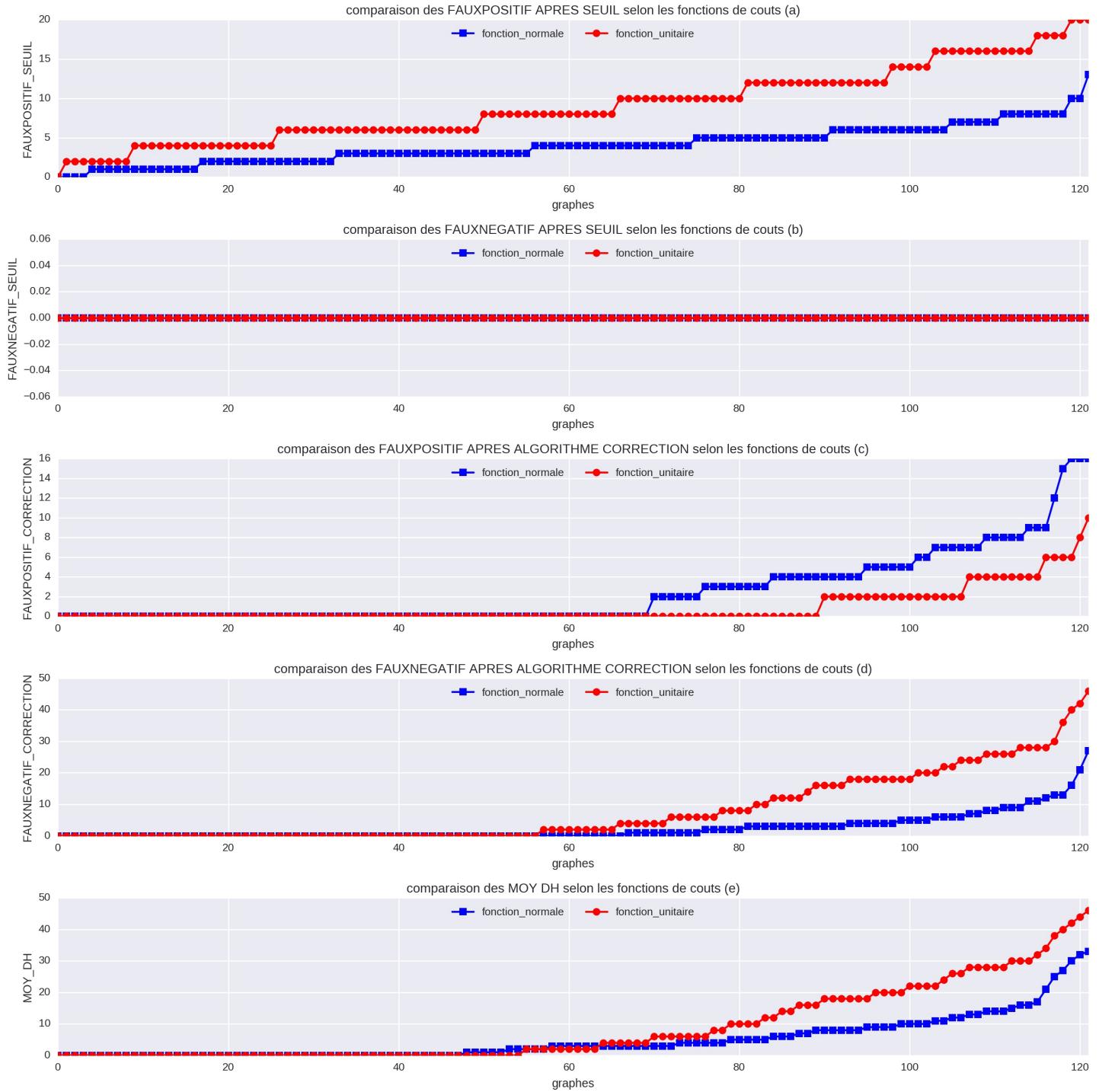


Figure 1.15: comparaison entre les fonctions de coût *unitaire* et *normale*: (a) cases *fausses positives* dans la matrice M_s ; (b) cases *fausses négatives* dans la matrice M_s ; (c) cases *fausses positives* dans la matrice M'_s ; (d) cases *fausses négatives* dans la matrice M'_s ; (e) comparaison des fonctions de coût selon *moy_DH*

sommet est le coût minimal de toutes les cases modifiées. Nous recherchons alors la fonction de coût qui minimise globalement le coût de correction des sommets de \mathcal{C} . Pour ce faire, nous comparons d'abord les fonctions de coût *unitaire* et *normale* parce que nous souhaitons savoir

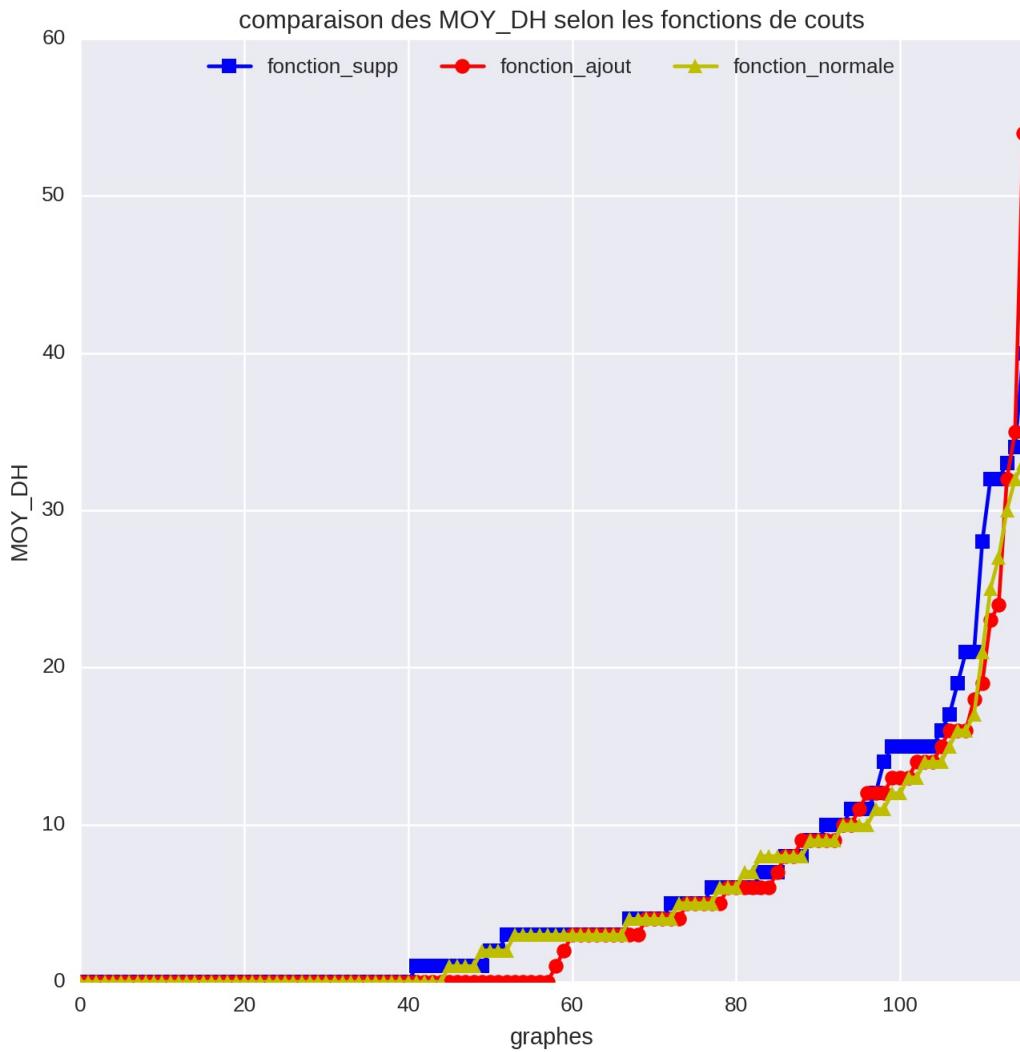


Figure 1.16: Comparaison entre les fonctions de coût *normale*, *ajout* et *suppression* : la fonction *ajout* est la courbe en rouge, la fonction *normale* est en jaune et la fonction *suppression* est en bleu

s'il est préférable d'utiliser les valeurs de corrélations dans les couts des opérations. Puis nous comparons la meilleure des deux fonctions avec celles *ajout* et *suppression*. La figure 1.15 contient

- La comparaison des distances de Hamming entre les fonctions de coût *unitaire* et *normale* (graphique (e)).
- La comparaison du nombre de cases *fausses négatives* entre les 2 fonctions de coût avant l'algorithme de correction (graphique (b)).
- La comparaison du nombre de cases *fausses négatives* entre les 2 fonctions de coût après l'algorithme de correction (graphique (d)).

- La comparaison du nombre de cases *fausses positives* entre les 2 fonctions de coût avant l'algorithme de correction (graphique (a)).
- La comparaison du nombre de cases *fausses positives* entre les 2 fonctions de coût après l'algorithme de correction (graphique (a)).

Les distances de Hamming et les nombres de cases sont ordonnées par ordre croissant.

Nous constatons que la courbe de la fonction *unitaire* est au dessus de celle de la fonction *normale* dans le graphique (e) de la figure 1.15. Avec $s = 0.7$, l'ensemble de cases érronnées sont des cases *fausses positives*(graphique (c) de la figure 1.15). En appliquant les fonctions *unitaire* et *normale*, nous avons l'introduction de cases *fausses négatives* et le nombre de ces cases est plus important dans la fonction *unitaire*. Ce nombre fait croître la distance de Hamming *moy_DH* parce que la correction a réduit le nombre de cases *fausses positives* dans la fonction *normale* (voir les graphiques (a) et (c) de la figure 1.15). Ce qui explique les faibles distances *moy_DH* de la fonction *normale*. La fonction *normale* donne de meilleurs résultats et nous la choisissons dans le calcul des coûts de correction.

Par ailleurs, dans la figure 1.16, les courbes des fonctions *normale*, *ajout* et *suppression* sont entremêlées et il ne se dégage aucun écart significatif entre elles. Il est donc difficile de juger de l'influence d'une des fonctions sur la correction des sommets de \mathcal{C} .

Conclusion : prioriser l'*ajout* à la *suppression* et vice versa n'a aucune influence sur les distances de Hamming quand nous utilisons les valeurs de corrélations. Toutefois, il est préférable de considérer les corrélations dans le calcul du coût de la modification d'une case parce que cela améliore les distances *moy_DH* comme il est indiqué dans le graphique (e) de la figure 1.15.

1.3.5 Conclusion de l'expérimentation 2

Nous avons générée des valeurs de corrélations pour toutes les cases de la matrice M_{LG} en considérant la distribution des valeurs de corrélation du réseau électrique du datacenter *Champlan*. Les valeurs de corrélation suivent des lois normales asymétriques de paramètre $\alpha = 5$ pour les cases à 0 et de paramètre $\alpha = -5$ pour les cases à 1. La matrice de corrélation M_c est construite à partir de ces corrélations. Un ensemble $s \in S$ de seuils est appliqué à la matrice M_c pour la transformer en la matrice d'adjacence M_s du graphe G_s . La matrice M_s contient des cases *fausses négatives* et *fausses positives*. Nous cherchons à minimiser la distance de Hamming en corrigeant le maximum de cases érronnées pendant l'algorithme de correction. Cela passe par la sélection adéquate du seuil et de la fonction de coût. Après l'exécution de notre couple d'algorithmes, nous avons déduit que le bon seuil est contenu dans l'intervalle $s =]0.6, 0.7]$ et que la fonction *normale* est la meilleure fonction de coût. L'utilisation du seuil s et de la fonction *normale* ne garantissent pas la suppression totale des cases érronnées. Mais elles minimisent leur nombre de telle sorte qu'un expert du métier puisse effectuer les corrections manuellement qui conduisent au line-graphe *LG* recherché. Dans la section suivante, nous nous intéressons aux graphes dans lesquelles tous les sommets ne peuvent être couverts par 1 ou 2 cliques. Ces graphes sont dits *grilles bouclées*.

1.4 Expérimentation 3 : algorithmes sur les grilles bouclées

Nous considérons des graphes dans lesquels le voisinage d'un sommet peut être couvert par une ou deux cliques. L'exécution de l'algorithme de couverture sur chacun de ces graphes fournit une

couverture vide. Cette famille de graphes est désignée graphes *grilles bouclées*. Après l'exécution de l'algorithme de couverture, tous les sommets de la *grille bouclée* sont dans l'ensemble \mathcal{C} des sommets à corriger.

Dans cette section, nous évaluons les performances de nos algorithmes, particulièrement l'algorithme de correction en effectuant des opérations de *suppression* et d'*ajout* d'arêtes uniquement. Nous déterminons une borne supérieure de la distance line de ces graphes. Nous comparons cette borne avec les résultats obtenus par l'algorithme de correction avec l'approche *aléatoire sans remise* (voir tableau 1.1).

Nous débutons notre analyse par la définition et la construction d'une grille bouclée. Ensuite, nous décrirons le protocole d'expérimentation sur des grilles bouclées d'ordres différents. Enfin nous interprétons les résultats obtenus pour chaque opération.

1.4.1 Définition des grilles bouclées et les distances line théoriques

1.4.1.1 Définition de la grille bouclée $G_{k,k'}$

Chaque sommet de $G_{k,k'}$ est identifié par le couple (i, j) avec $0 \leq i < k$ et $0 \leq j < k'$. Le sommet (i, j) est adjacent au sommet :

- $(i, j + 1)$ si $j < k' - 2$
- $(i + 1, j)$ si $i < k - 2$
- $(i, j - 1)$ si $j > 0$
- $(i - 1, j)$ si $i > 0$

De plus, les sommets $(0, 0)$ et $(0, k - 1)$, $(0, 0)$ et $(0, k' - 1)$, $(k - 1, 0)$ et $(k - 1, k' - 1)$, $(0, k' - 1)$ et $(k - 1, k' - 1)$ sont adjacents. On remarque que tout graphe induit par un sommet et son voisinage forme un graphe étoile $K_{1,4}$. La figure 1.17 est un exemple de *grille bouclée* $G_{4,4}$. Ce graphe contient 16 sommets, 28 arêtes. Les sommets $(0, 0), (0, 1), (1, 1), (1, 0)$ forme une cellule et le graphe $G_{4,4}$ contient 10 cellules.

Définition 1. Une cellule est un cycle de longueur 4 identifié par les sommets (i, j) , $(i, j + 1)$, $(i + 1, j)$ et $(i + 1, j + 1)$ avec $i < k - 1$ et $j < k' - 1$. Nous notons une telle cellule $C_{i,j}$.

Si $k = k' = 2$, la grille bouclée $G_{2,2}$ est la cellule $C_{0,0}$.

Propriété 1. Le graphe $G_{k,k'}$ possède $k \times k'$ sommets, $k \times (k' - 1) + k' \times (k - 1) + 4$ arêtes et $(k - 1) \times (k' - 1) + 1$ cellules.

1.4.1.2 Correction des grilles bouclées

Nous allons considérer les modifications de l'ensemble des arêtes de $G_{k,k'}$ afin d'obtenir un line-graphe. La première modification se base uniquement par ajout d'arêtes et la seconde sur la suppression d'arêtes uniquement. Nous supposons que les deux opérations conduisent sur la même borne supérieure de $DL(G_{k,k'})$.

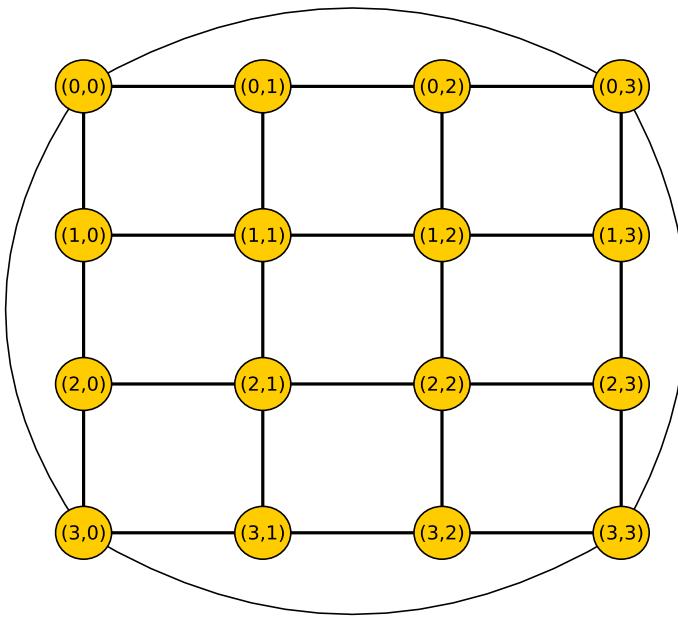


Figure 1.17: La grille bouclée $G_{4,4}$: il est composé de 16 sommets, 28 arêtes et 10 cellules.

1.4.1.3 Modification par ajout d'arêtes uniquement

Soit le graphe $G_{k,k'}$ contenant $k \times k' + 1$ cellules. Pour transformer chaque cellule en cliques comme cela est illustré dans la figure 1.18, nous ajoutons 2 arêtes. Nous considérons le sommet $(0, 0)$ contenu dans les cellules $C_{0,0}$ et $C_{k-1,k'-1}$. Nous ajoutons 2 arêtes dans $C_{0,0}$ et $C_{k-1,k'-1}$. Les cellules deviennent des cliques K_4 .

L'arête $\{(i, j+1), (i+1, j)\}$ appartient aux cellules $C_{i,j}$ et $C_{i,j+1}$. Or cette arête est déjà couverte par une clique K_4 de la cellule $C_{i,j}$. Alors nous ne pouvons pas ajouter d'arêtes dans la cellule $C_{i,j+1}$. L'arête $\{(i, j+1), (i, j+2)\}$ forme une clique K_2 . Le sommet $(i, j+1)$ est couvert par une clique K_4 et une clique K_2 . Le sommet $(i+1, j)$ est aussi couvert par une clique K_4 et une clique K_2 parce que les cellules $C_{i,j}$ et $C_{i+1,j}$ partagent l'arête $\{(i+1, j), (i+1, j+1)\}$ et cette arête forme une clique K_4 avec la cellule $C_{i,j}$. Les cellules $C_{i,j}$ et $C_{i+1,j+1}$ ne partagent que le sommet $(i+1, j+1)$. En plus les arêtes $\{(i+1, j+1), (i+2, j+1)\}$ de $C_{i+1,j}$ et $\{(i+1, j+1), (i+1, j+2)\}$ de $C_{i,j+1}$ ne sont pas couverts par une clique K_4 . Nous pouvons alors transformer $C_{i+1,j+1}$ en une clique K_4 en ajoutant 2 arêtes.

Ainsi, dans des cellules successives en lignes (avec k) ou en colonnes (avec k'), nous ajoutons des arêtes dans $\lceil \frac{k \times k'}{2} \rceil + 1$ cellules. L'arête d'une cellule qui n'est pas contenue par une clique K_4 forme une clique K_2 . Les cellules ayant un seul sommet en commun sont transformées en des cliques K_4 . À la fin de la correction, la grille bouclée $G_{k,k'}$ est partitionnée en des cliques finies K_4 et K_2 . Dans cette construction, on remarque que chaque sommet est couvert par 2 cliques. De cette construction découle le lemme suivant :

Lemma 1. *La distance line d'un graphe cellule $G_{k,k'}$ avec l'opération ajout uniquement est*

$$DL(G_{k,k'}) \leq k \times k' + 3 \quad (1.5)$$

Dans la figure 1.18, nous réalisons la correction de distance line $DL(G_{4,4}) \leq 12$ en transformant les cellules partageant un sommet en cliques K_4 . C'est le cas des cellules $C_{1,2}$ et $C_{0,2}$ qui ont le

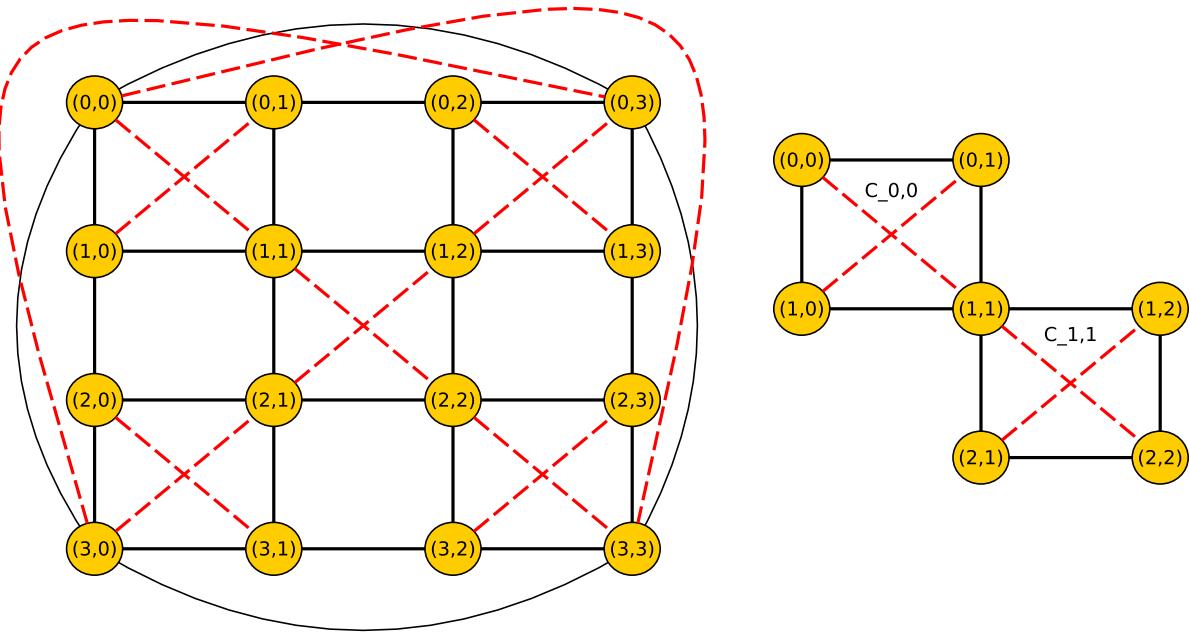


Figure 1.18: La grille bouclée $G_{4,4}$: il est composé de 16 sommets, 33 arêtes. Il contient 4 cliques K_2 et 6 cliques K_4 . Les arêtes ajoutées sont les traits de couleur rouge.

sommet $(1, 2)$ en commun. Les arêtes partagées entre deux cellules ont un des sommets couvert par une clique K_2 et l'autre sommet couvert par une clique K_4 . Tel est le cas avec le sommet $(3, 2)$ qui forme l'arête $\{(2, 2), (3, 2)\}$ et cette arête est contenue par une clique K_4 .

1.4.1.4 Modification par suppression d'arêtes uniquement

Soit le graphe $G_{k,k'}$ contenant $k \times k' + 1$ cellules. Nous supprimons les arêtes $\{(0, 0), (k - 1, 0)\}, \{(0, 0), (0, k' - 1)\}, \{(k - 1, 0), (k - 1, k' + 1)\}$ de la cellule $C_{k-1, k'-1}$. Cette cellule contient uniquement l'arête $\{(0, k' - 1), (k - 1, k' - 1)\}$ et cette arête forme la clique k_2 . Nous supprimons également les arêtes $\{(0, k' - 2), (0, k' - 1)\}$ et $\{(k - 2, k' - 1), (k - 1, k' - 1)\}$ incidentes respectivement aux sommets $(0, k' - 1)$ et $(k - 1, k' - 1)$ de sorte que ces sommets soient couverts par deux cliques k_2 . Les sommets de degré minimums $(0, 0), (k - 1, 0)$. Ils sont couverts par deux cliques K_2 c'est-à-dire $\{(0, 0), (1, 0)\}$ et $\{(k - 2, 0), (k - 1, 0)\}$.

Considérons les sommets de degré 4. Soit (i, j) un tel sommet. Pour former une bipartition autour de ce sommet, nous allons supprimer 2 arêtes. chaque arête appartient à deux cellules voisines. Dans notre cas, nous supprimons l'arête $\{(i - 1, j), (i, j)\}$ entre les cellules $C_{i-1, j-1}$ et $C_{i-1, j}$ et aussi l'arête $\{(i, j), (i + 1, j)\}$ entre les cellules $C_{i, j-1}$ et $C_{i, j}$. Ces sommets sont couverts aussi par des cliques K_2 .

Les arêtes incidentes à un sommet de degré 3 et n'étant pas des cliques K_2 sont aussi supprimées. À la fin de l'algorithme de correction, la couverture de corrélation ne contient que des cliques K_2 . Le graphe $G_{k,k'}$ est un cycle hamiltonien de taille $(k \times k' + 1) + k' + 1$. La distance de correction entre $G_{k,k'}$ et $L(G_{k,k'})$ est $DC_{k,k'} = k \times k' + 3$ et cette distance est minimale.

Lemma 2. La distance line d'une grille bouclée $G_{k,k'}$ avec l'opération suppression uniquement d'arêtes est

$$DL(G_{k,k'}) = k \times k' + 3 \quad (1.6)$$

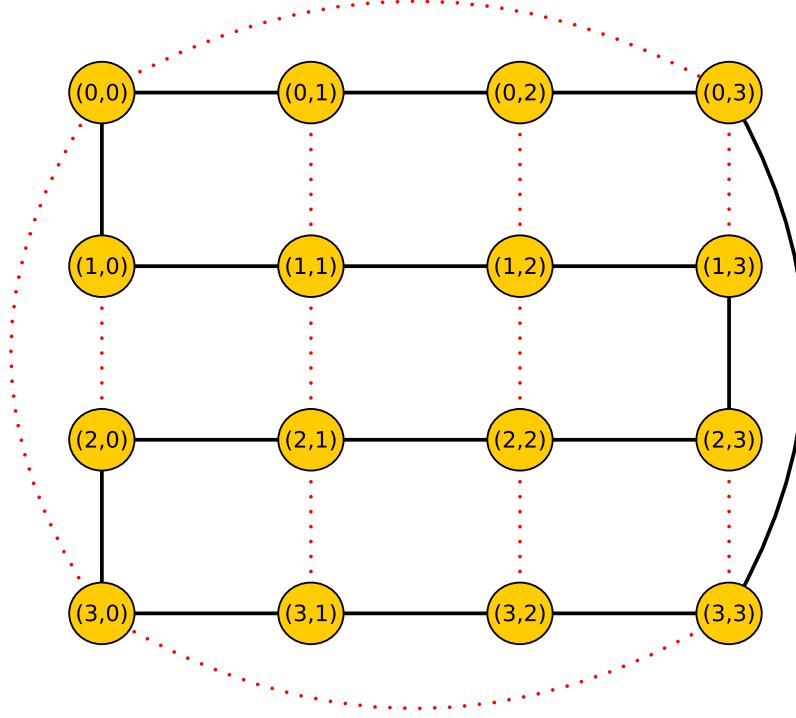


Figure 1.19: La grille bouclée $G_{4,4}$: il est composé de 16 sommets, 16 arêtes et 16 cliques K_2 . Les arêtes supprimées sont les traits en pointillés rouges

Une illustration de la correction avec l'opération *suppression uniquement* est faite avec la grille $G_{4,4}$ dans la figure 1.19. La grille $G_{4,4}$ possède $k \times k' = 16$ cliques K_2 et 12 arêtes sont supprimées.

Conclusion : nous avons déterminé deux types de modifications d'arêtes qui ont une borne supérieure de la distance line pour chaque opération. En revanche, les cliques formant la couverture de corrélation sont différentes selon la modification. Dans la modification *ajout d'arêtes uniquement*, le grille bouclé $G_{k,k'}$ contient des cliques K_4 et K_2 alors que $G_{k,k'}$ ne contient que des cliques K_2 dans la *suppression d'arêtes uniquement*.

1.4.2 Protocole d'expérimentation sur les grilles bouclées

Nous allons comparer cette borne supérieure de l'équation 1.6 avec les distances de correction obtenues par l'algorithme de correction.

Considérons $G_{k,k'}$ une grille bouclée dans lequel le nombre de sommets par lignes est identique le nombre de sommets par colonnes ($k = k'$). Nous le notons G_k .

Nous construisons 48 grilles bouclées contenant chacune $k \times k + 1$ cellules, avec $k \in \{2, \dots, 98\}$. Dans chaque graphe G_k , nous exécutons 50 fois notre couple d'algorithmes avec chaque modification d'arêtes et la distance de correction obtenue est comparée avec la borne supérieure (équation 1.6).

Soient $\phi^+(u,v)$ le coût de l'opération *ajouter une arête* et $\phi^-(u,v)$ le poids de l'opération *supprimer une arête* (voir section ??).

La modification *ajout d'arêtes uniquement* est tel que

- L'ajout d'arêtes a un coût $\phi^+(u, v) = 1$,
- La suppression d'arêtes a un coût $\phi^-(u, v) = 10$

Quant à la modification *suppression d'arêtes uniquement*, elle se définit comme suit :

- L'ajout d'arêtes a un coût $\phi^+(u, v) = 10$,
- La suppression d'arêtes a un coût $\phi^-(u, v) = 1$

Nous allons comparer l'évolution des distances de correction des 48 graphes en fonction la borne supérieure pour chaque modification réalisée.

1.4.3 Analyse des résultats

Notre objectif est de présenter les variations des distances de correction par rapport à la borne supérieure de la distance line pour chaque modification réalisée sur les graphes pendant l'algorithme de correction. Pour ce faire, nous regroupons notre analyse en 5 expérimentations.

Les deux premières expérimentations comparent les distances de correction avec la borne supérieure pour la modification *ajout d'arêtes uniquement* (figure 1.20 (a)) et pour la modification *suppression d'arêtes uniquement* figure 1.20 (c)). Nous constatons que les courbes des distances de correction et celle de la borne supérieure sont croissantes. La courbe de la borne supérieure, désignée par *borneSup* dans les graphiques (a) et (c), évolue lentement par rapport aux courbes des distances et l'écart entre ces courbes croît linéairement. Pour comprendre cet écart croissant, nous vérifions le pourcentage d'arêtes supprimées pour chaque modification. Ce sont les deux autres expérimentations faites et représentées par la figure 1.20 (b)) pour la modification *ajout d'arêtes uniquement* et la figure 1.20 (d)) pour la modification *suppression d'arêtes uniquement*. En effet, nous avons choisi les arêtes supprimées parce que le nombre d'arêtes est déjà connu c'est-à-dire 0 pour l'ajout uniquement et la borne supérieure pour la suppression uniquement.

Nous remarquons que les arêtes de G_k supprimées avoisinent en moyenne de 40% quand le nombre k de cellules est petit ($k \leq 14$) dans la modification *ajout arêtes uniquement*. Au-delà de $k > 14$, une arête sur deux du graphe G_k est supprimée. La courbe *aretes_supprimees_ajout_1* dans le graphique (b) représente le pourcentage d'arêtes supprimées. En effet, nous expliquons ces chiffres par l'ajout d'arêtes entre des sommets de cellules voisines et ces sommets ne sont pas partagés entre les cellules voisines. Ces arêtes ajoutées impliquent la suppression des arêtes de G_k parce que la partition π_s (voir chapitre ??) des arêtes à supprimer pendant la compression n'est pas vide et contient des arêtes de G_k en général. Ainsi ces arêtes provoquent l'abandon des arêtes diagonales ajoutées à partir du sommet commun entre des cellules comme indiqué dans l'exemple du graphe $G_{4,4}$ dans la figure 1.18.

En ce qui concerne la modification *suppression d'arêtes uniquement*, les arêtes supprimées proviennent aussi des cellules voisines. Dans les cellules partageant un sommet, l'algorithme ajoute généralement une arête diagonale à partir de ce sommet commun dans une seule cellule. Les arêtes diagonales ajoutées sont responsables de l'augmentation de la distance de correction (courbe *dc_supp_1* dans le graphique (c)). Pour des cellules partageant une arête, l'algorithme supprime certaines arêtes communes comme indiqué dans la section 1.4.1.4. Les autres arêtes communes non supprimées sont dues à la présence des arêtes diagonales. Cela explique pourquoi nous avons en moyenne 50% des arêtes de G_k qui sont supprimées dans le graphique (d). Ce pourcentage est identique à celui des arêtes supprimées de la borne supérieure lorsque le nombre de cellules devient

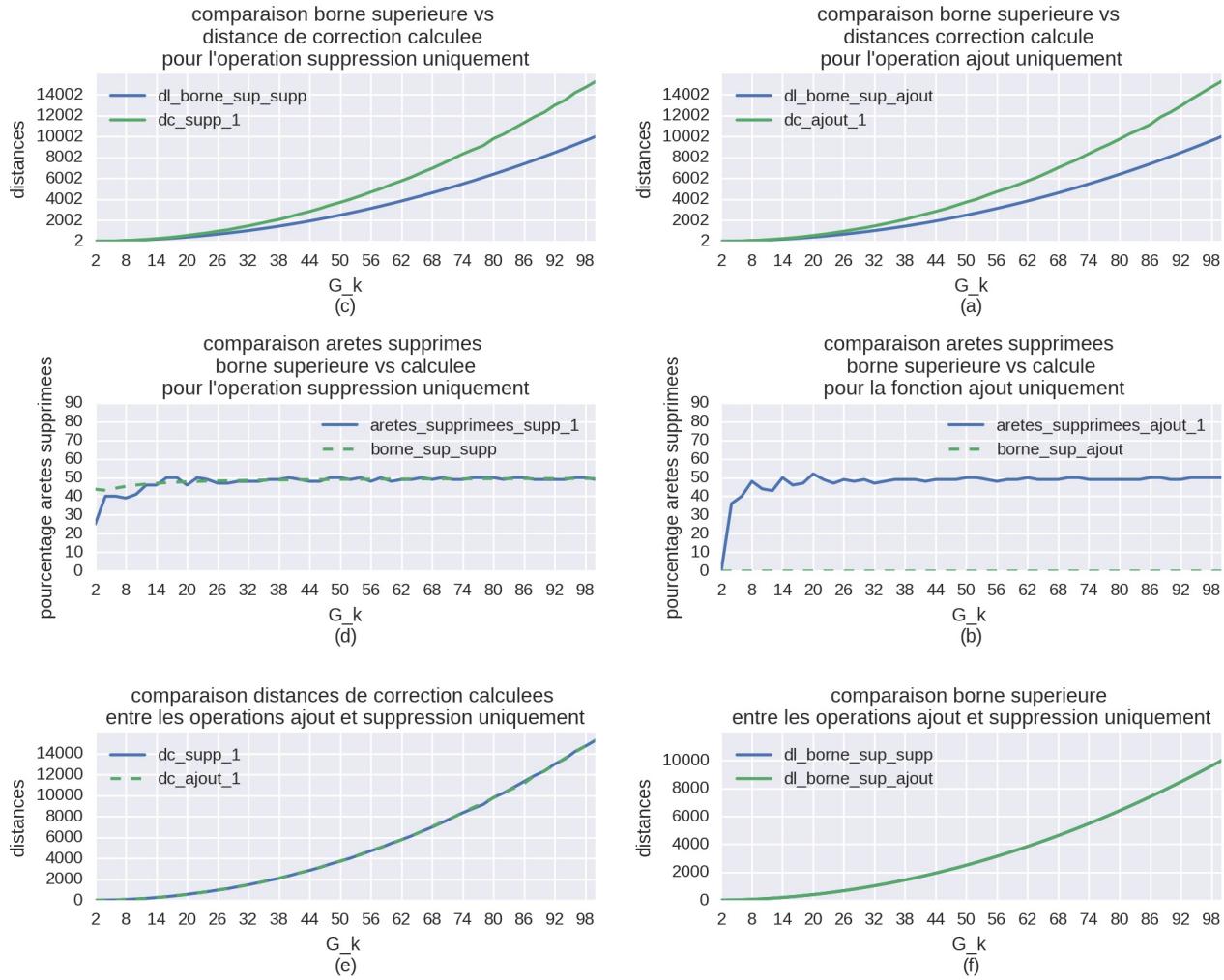


Figure 1.20: Comparaison de la borne supérieure de la distance line et calculées selon des fonctions de coût *suppression* et *ajout* : La figure (a) désigne la comparaison entre les distances de correction et la borne supérieure de l'équation 1.6 avec la modification *ajout d'arêtes uniquement*, La figure (c) désigne la comparaison entre les distances de correction et la borne supérieure de l'équation 1.6 avec la modification *suppression d'arêtes uniquement*, La figure (b) compare le pourcentage d'arêtes supprimées dans les graphes bouclées avec celui de la borne supérieure de l'équation 1.6 dans la modification *ajout d'arêtes uniquement*, La figure (d) compare le pourcentage d'arêtes supprimées dans les graphes bouclées avec celui de la borne supérieure de l'équation 1.6 dans la modification *suppression d'arêtes uniquement*, La figure (e) compare les distances de correction entre les différentes modifications.

élevé $k \geq 18$ et il ne signifie pas que l'algorithme de correction supprime les arêtes de la section 1.4.1.4.

Par ailleurs, les courbes `dc_ajout_1` de la modification *d'ajout d'arêtes uniquement* et celle `dc_supp_1` de la modification *suppression d'arêtes uniquement* ont les mêmes tendances et sont superposées (figure (e)) parce que dans la modification *d'ajout*, l'algorithme ajoute la même proportion d'arêtes qu'il supprime dans la modification *suppression*.

Conclusion : l’expérimentation montre que le line-graphe $L(G_k)$ proposé par l’algorithme de correction pour un k donnée est identique en terme de distance de correction quelle que soit la modification réalisée comme illustre la figure 1.20(e). Toutefois, les graphes corrigés ne sont pas optimaux en terme de distances de correction parce que l’algorithme priorise dans certains cas les modifications *ajout d’arêtes uniquement* et *suppression uniquement* et cela leur permet d’atteindre des minimums locaux. Cependant le minimum global n’est pas atteint généralement.

1.4.4 Conclusion de l’expérimentation 3

Les grilles bouclées ont la particularité d’avoir des arêtes qui ne peuvent être partitionnées en cliques. Nous avons décrit la construction de ces graphes et nous définissons deux méthodes pour les corriger. La première méthode est la modification *d’ajout uniquement* qui consiste à ajouter uniquement des arêtes et la seconde méthode est la modification *suppression uniquement* qui supprime uniquement des arêtes des grilles bouclées. Avec ces méthodes, nous avons trouvé une borne supérieure unique de la distance line des grilles bouclées et nous avons comparé cette borne supérieure avec les distances de correction obtenues après l’algorithme de correction.

Nous remarquons que les distances de correction varient très peu entre les modifications *ajout uniquement* et *suppression uniquement*. Les graphes corrigés n’ont pas de distances de corrections optimales parce que l’algorithme supprime des arêtes pendant la modification *ajout d’arêtes uniquement* et ajoute des arêtes pendant la modification *suppression uniquement*.

1.5 Conclusion du chapitre 1

Le chapitre 1 analyse les performances de nos algorithmes de couverture et de correction selon 3 expérimentations.

La première expérimentation consiste à modifier les k cases de la matrice d’adjacence du line-graphe d’un réseau électrique. Ces cases modifiées sont divisées en deux sous-ensembles disjoints (cases *fausses négatives* et cases *fausses positives*) selon une variable $p \in [0, 1]$. Si $p = 0$ alors l’ensemble des cases modifiées est composé que de cases *fausses positives* et si $p = 1$ alors nous avons que des cases *fausses négatives*. Le but est de borner le nombre de cases corrigées par nos algorithmes. Ainsi, nous avons défini les distances de correction et de Hamming. La distance de correction est le nombre minimum de cases à modifier dans un graphe de k cases érronnées pour en faire un line-graphe. Quant à la distance de Hamming, elle est la différence de cases entre les matrices de line-graphe proposé par nos algorithmes et le line-graphe du réseau électrique. Nous avons comparée le nombre de cases corrigées avec 5 approches de correction qui sont : *aléatoire sans remise* (2c), *degré minimum sans remise* (2a), *coût minimum sans remise* (2b), *degré minimum avec remise* (1a), *coût minimum avec remise* (1b). À chaque approche, nous avons 3 coûts de modification d’une case : *unitaire*, *ajout* et *suppression*. Nous avons conclut que l’approche *aléatoire sans remise* (2c) proposait des distances de correction minimales quelle que soit la répartition effectuée p et la fonction de coût utilisée. Ces distances constituent la borne supérieure de la distance line quand le nombre k de cases modifiées est faible $k \leq 5$. D’autre part, nous avons montré que les distances de correction et de Hamming deviennent très corrélées quand le nombre de cases modifiées initiales est élevé $k > 10$. Dans ce cas où $k \leq 5$, le line-graphe proposé par l’algorithme de correction est celui de réseau électrique. La distance de correction peut être utilisée comme une métrique lorsque la topologie initiale du réseau est inconnue.

La seconde expérimentation considère que chaque case de la matrice du line-graphe est associée

à une valeur de corrélation. Les valeurs de corrélation sont générées en tenant compte des distributions des valeurs de corrélation du réseau électrique d'un datacenter *Champlan*. Ces corrélations sont calculées avec la *distance de Pearson*. Les valeurs de corrélation dans la matrice forment la *matrice de corrélation*. Nous avons défini un ensemble de seuil dans lequel chaque seuil est appliqué à la matrice de corrélation pour en construire la matrice d'adjacence du graphe G_s . Le graphe G_s contient des cases *fausses négatives* et des cases *fausses positives*. Notre objectif est de minimiser le nombre de cases érronnées dans le graphe G_s après l'exécution de nos algorithmes et cela nécessite la sélection d'une valeur adéquate du seuil. Nous avons considéré l'approche de correction *aléatoire sans remise* (*2c*) et nous avons sélectionné quatre fonctions de coût : *unitaire*, *ajout* et *suppression* et *normale*. Les fonctions de coût sont fonction des cases modifiées par l'algorithme de correction. Nous avons déduit que le bon seuil appartient à l'intervalle $s \in]0.6, 0.7]$ et la fonction *normale* donne de bons résultats pour le calcul dans les distances de correction et de Hamming.

La dernière expérimentation se focalise sur les graphes dans lesquelles un sommet et son voisinage ne peuvent être couverts par une ou deux cliques. Un exemple de ces graphes est la famille des graphes *grilles bouclées*. Une grille bouclée de k lignes et k' colonnes est composé de $k \times k' + 1$ cellules avec une cellule un graphe biparti $K_{2,2}$ non orienté. Nous avons montré que la correction de ces graphes peut se faire selon deux méthodes (modification *ajout d'arêtes uniquement* et *suppression d'arêtes uniquement*). Les deux modifications admettent la même borne supérieure de ses distances line. Notre objectif est de vérifier que la convergence des distances de correction vers la borne supérieure quelque soit la modification réalisée. Les résultats obtenus montrent que les distances de correction sont invariantes peu importe les modifications et que les graphes corrigés n'ont pas de distances de corrections optimales.

Au terme de ces 3 expérimentations, nous pouvons conclure nos algorithmes ont un comportement optimal lorsque le mode de correction est *aléatoire sans remise* et le seuil de corrélation est contenu dans l'intervalle $]0.6, 0.7]$ peu importe la répartition des cases érronnées et la fonction de coût. Ces conditions garantissent que la matrice du graphe contienne peu de cases érronnées et que ces cases seront corrigées pendant l'algorithme de correction. Cependant pour la famille des graphes dont aucun sommet ne peut être couvert par une clique (cas des grilles *bouclées*), les distances line calculées ne convergent pas vers les bornes supérieures prédefinies. Toutefois, le type de correction (modifications *ajout uniquement* et *suppression uniquement*) sur ces graphes n'influence pas les valeurs de distances de correction.

1.6 Annexes

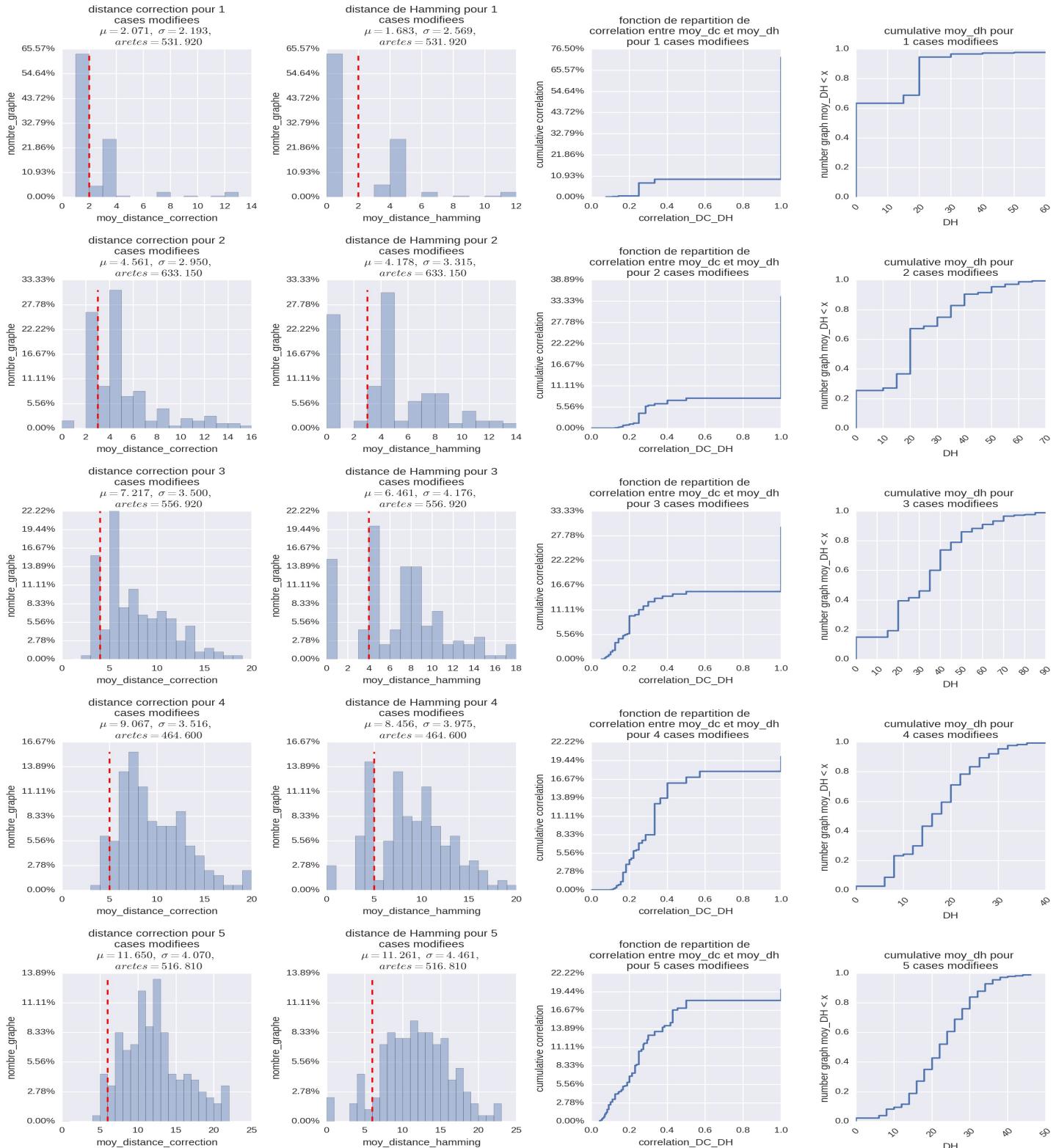


Figure 1.21: Mode de correction aléatoire sans remise à coût unitaire pour $k = \{1, 2, 3, 4, 5\}$ cases modifiées : La première colonne représente la distribution des distances de correction $moy_DC_{k,0.5}$. La seconde colonne est la distribution des distances de Hamming $moy_DH_{k,0.5}$. La troisième colonne est la fonction de répartition de la corrélation entre les distances de correction et de Hamming avec en abscisse la corrélation entre ces distances (correlation_DC_DH). La quatrième colonne est la fonction cumulative des distances de Hamming. La première ligne est associée à $k = 1$ case modifiée, la seconde ligne à $k = 2$ cases modifiées, la troisième ligne à 5 cases modifiées et enfin la dernière à 9 cases modifiées

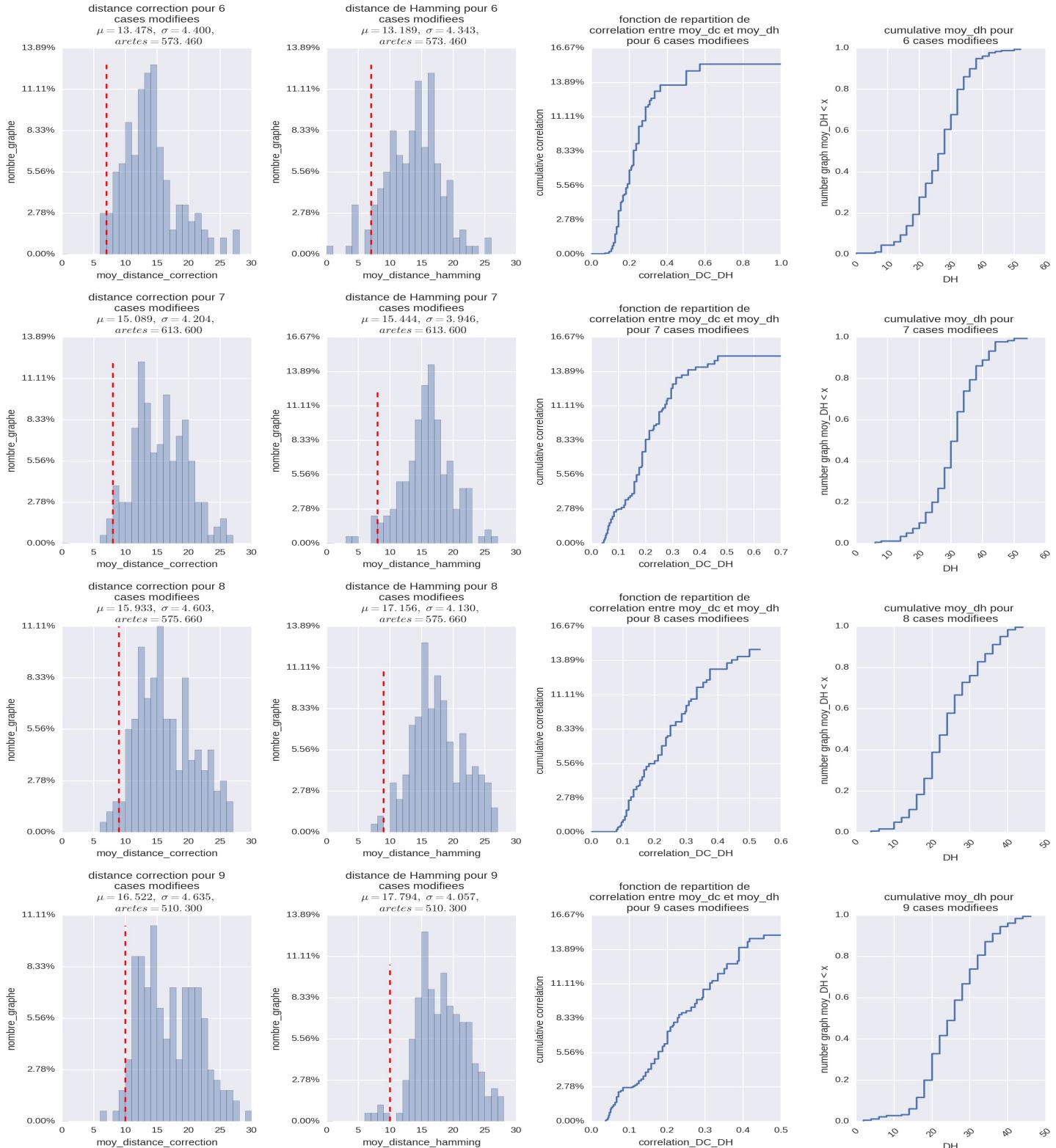


Figure 1.22: Mode de correction aléatoire sans remise à coût unitaire pour $k = \{6, 7, 8, 9\}$ cases modifiées : La première colonne représente la distribution des distances de correction $moy_DC_{k,0.5}$. La seconde colonne est la distribution des distances de Hamming $moy_DH_{k,0.5}$. La troisième colonne est la fonction de répartition de la corrélation entre les distances de correction et de Hamming avec en abscisse la corrélation entre ces distances (correlation_DL.DH). La quatrième colonne est la fonction cumulative des distances de Hamming. La première ligne est associée à $k = 1$ case modifiée, la seconde ligne à $k = 2$ cases modifiées, la troisième ligne à 5 cases modifiées et enfin la dernière à 9 cases modifiées