

RAPPORT DE STAGE

ALGORITHMIQUE POUR LES ORDRES STOCHASTIQUES SUR LES DISTRIBUTIONS DISCRÈTES

Rédigé par :

Gilles Christian Noukela

Sous l'encadrement académique de :

NIHAL PEKERGNIN

Professeur informatique

Année académique :

2016-2017

Remerciements

Mes remerciements vont à l'endroit de mes encadreurs : Nihal Pekergin, professeur d'informatique à l'Université Paris-Est Créteil, Jean-Michel Fourneau, professeur d'informatique à l'Université de Versailles Saint-Quentin pour leurs précieux conseils et mon encadrement pendant toute la période du stage. Je vous remercie du fond du coeur pour le soutien que vous m'avez apporté ainsi que pour vos différentes remarques dans la compréhension du sujet.

Je remercie profondemment Johanne Cohen, Directrice de Recherche au sein du LRI pour avoir accepté mon adhésion au sein du LRI dans le cadre de ce projet.

Je remercie Alexandre Fauquette, Étudiant à CentraleSupélec, pour son aide et ses explications dans la compréhension du sujet.

Je remercie Daniele Varacca, Responsable du Master I à l'Université Paris-Est Créteil pour son aide dans la recherche du stage.

Je voudrais exprimer ma profonde gratitude à toute l'équipe du LRI pour leur accueil chaleureux et pour m'avoir permis de passer des moments agréables en leur compagnie.

Je remercie tous ceux qui, de près ou de loin ont contribué à la réalisation de mon travail et dont les noms ne sont pas mentionnés. Veuillez recevoir, l'expression de ma profonde gratitude.

Table des figures

1.1	Graphe de tâches.	8
1.2	Exemple de graphe de tâches avec des variables aléatoires en entrée.	9
1.3	Exemple de graphe de tâches avec des entrées déterministes.	9
2.1	Distribution du temps d'achèvement du graphe de tâches pour la borne supérieure avec 5 atomes comme distributions d'entrées.	15
2.2	Distribution du temps d'achèvement du graphe de tâches pour la borne inférieure avec 5 atomes comme distributions d'entrées.	15
2.3	Distribution de la borne supérieure avec 5 atomes. la distribution initiale est uniforme avec 10 atomes	16
2.4	Distribution de la borne supérieure avec 4 atomes. la distribution initiale est uniforme avec 10 atomes	16
2.5	Fusion de deux atomes pour la borne inférieure.	17
2.6	Borne inférieure dans le premier cas	18
2.7	Les distributions cumulées de F_{D1} and F_{D2}	19
2.8	Borne inférieure dans le second cas	20
2.9	Distribution du temps d'achèvement du graphe de tâches pour la borne inférieure avec 5 atomes comme distributions d'entrées.	22
2.10	Distribution de la borne supérieure avec 5 atomes. la distribution initiale est Zip d'exposant 1,4267 avec 10 atomes	23
2.11	Distribution de la borne supérieure avec 4 atomes. la distribution initiale est Zipf d'exposant 1,4267 avec 10 atomes	23

Table des matières

1	Problèmes d'optimisation	6
1.1	Déroulement du stage	6
1.1.1	Équipe dirigeante	6
1.1.2	Tâches effectuées	6
1.1.3	Classes de problèmes d'optimisation	7
1.2	Grâphe de tâches	7
1.2.1	Illustration du problème d'optimisation	7
2	Algorithmes de bornes	11
2.1	Définitions et propriétés des ordres stochastiques	11
2.2	Algorithme pour l'ordre Strong	12
2.2.1	Algorithmes des bornes optimales pour l'ordre St	12
2.3	Borne inférieure convexe	16
2.3.1	Construction de la borne inférieure convexe	17
2.3.2	Heuristique pour une borne inférieure convexe	21

Introduction

Dans la théorie des graphes, les coûts des noeuds ou des arcs sont généralement considérés comme étant déterministes. Malheureusement, cela n'est pas toujours vrai car de nombreux problèmes émanant de divers domaines (informatique, intelligence artificielle, ingénierie, etc) peuvent se présenter en termes de relations entre les différents éléments (noeud ou arc). Ainsi, on peut donc décider d'associer à chaque noeud ou arc une probabilité que cet élément soit entièrement ou partiellement disponible. Notre objectif est celui d'analyser la distribution des résultats du problème d'optimisation connaissant la distribution des probabilités sur les arcs et noeuds.

Partant d'un graphe de tâches acyclique, nous allons calculer le temps requis pour exécuter toutes les tâches du graphe. Ce problème a une complexité polynomiale lorsque les entrées sont déterministes. Mais cela n'est plus valable lorsque les noeuds sont associées à des variables aléatoires. Ainsi, Notre approche est basée sur les distributions de probabilités au niveau des noeuds et les propriétés des opérateurs plutôt que sur la structure du graphique.

Notre rapport comporte deux chapitres structurés en plusieurs sections. Dans un premier temps, nous décrirons dans le premier chapitre, le déroulement du stage, nous présenterons ensuite la classe des problèmes d'optimisation. Enfin, nous étudierons le problème d'optimisation du graphe de tâches. Dans l'étape suivante, nous présenterons dans le second chapitre, certaines propriétés des ordres stochastiques fort et convexe. Ensuite, nous analyserons les différentes actions dans la conception des bornes inférieures. Enfin, Nous présenterons un algorithme pour la borne inférieure et nous analyserons à partir d'un exemple les résultats obtenus.

Chapitre 1

Problèmes d'optimisation

1.1 Déroulement du stage

Dans cette section, nous présentons le contexte dans lequel le stage s'est effectué ensuite, les différentes tâches que nous avons réalisées ainsi que les différents problèmes d'optimisation qui permettent d'illustrer notre approche.

1.1.1 Équipe dirigeante

Ce stage, d'une durée de deux mois, s'inscrit dans le cadre du projet ADOC (projet CNRS PEPS MASTADONS). Il s'est effectué sur différents sites à savoir : UPEC, LRI et UVSQ. L'équipe dirigeante était constituée de :

- Jean-Michel Fourneau, Professeur d'Informatique au UVSQ.
- Johanne Cohen, Directrice de Recherche au LRI.
- Nihal Pekergin, Professeur d'Informatique à l'UPEC.
- Alexandre Fauquette, Étudiant à CentraleSupélec.
- Gilles Christian NOUKELA, Étudiant à l'UPEC.

1.1.2 Tâches effectuées

Au cours de cette période de stage, nous avons travaillé principalement sur la recherche d'une distribution bornante inférieure au sens convexe et parmi les tâches effectuées, nous avons eu à :

- prouver les différents lemmes qui permettent de construire une borne inférieure selon $\mathbf{cx}$.
- implémenter les algorithmes de construction des bornes optimales pour l'ordre stochastique $\mathbf{St}$ et analyser les résultats obtenus.
- implémenter l'heuristique qui permet de trouver une borne inférieure au sens $\mathbf{cx}$ et analyser les résultats obtenus.
- implémenter l'algorithme de distribution du temps d'achèvement du graphe de tâches.
- implémenter les algorithmes de comparaisons des bornes $\mathbf{cx}$ et $\mathbf{st}$.

1.1.3 Classes de problèmes d'optimisation

La généralité de notre approche peut être illustrée à partir de certains problèmes d'optimisation à savoir :

- Le flow maximal : c'est la capacité minimale des coupes dans un graphe. Une coupe est un ensemble de sommets issus de la partition du graphe en deux sous ensembles.
- Le plus court chemin qui consiste à trouver un chemin de longueur minimal qui sépare la source de la destination dans un graphe.
- L'arbre couvrant de poids minimal qui est un arbre couvrant dont la somme des poids des noeuds est minimale. Un arbre couvrant est un arbre qui connecte tous les noeuds du graphe et est inclus dans ce dernier.
- Le temps d'achèvement qui est un ensemble ordonné des tâches nécessaires afin d'obtenir les résultats attendus.

Tous ces problèmes sont construits avec les mêmes opérateurs convexes ou concaves tels que : "Min", "Max" ou "+".

1.2 Graphe de tâches

Nous illustrons la classe du problème considéré par le problème du graphe de tâches. Notre approche repose sur la distribution des résultats du problème d'optimisation connaissant les distributions d'entrées au niveau des tâches. Ainsi, si la taille de la distribution est importante, l'analyse des résultats du problème va être complexe. Par contre, si la taille des distributions est assez petite, l'obtention des résultats se fait en un temps fini.

1.2.1 Illustration du problème d'optimisation

On considère un graphe orienté sans circuit $G(V,E)$, avec V l'ensemble des noeuds et E l'ensemble des arcs tels que $(i,j) \in E$ si et seulement si $i < j \forall i,j \in E$. Nous supposons que les noeuds sont étiquetés avec des durées $(w_i)_{i \in V}$ en suivant un certain ordre. Pour éviter d'avoir un algorithme récursif, nous supposons également que le graphe G n'est pas un graphe série parallèle (graphe issu de la composition des autres graphes). Ainsi, le passage d'une tâche ou d'un noeud de i vers j stipule que j a besoin des résultats de i pour être exécuté. De plus, l'exécution de la tâche i passe par l'exécution de tous les prédécesseurs de i .

On désigne par b_i (resp. e_i) le temps de début (resp. de fin) de la tâche i . Le temps d'exécution w_i est donc donné par $w_i = e_i - b_i$. Ainsi, de manière récursive, le temps complet du dernier noeud du graphe e_{N_v} peut être obtenu à partir de la séquence suivante :

$$e_i = w_i + \max_{j \in \Gamma^-(i)} (e_j)$$

où $\Gamma^-(i)$ est l'ensemble des prédécesseurs de i et la séquence est initialisée par :

$$e_1 = b_1 + w_1 \text{ et } b_1 = 0$$

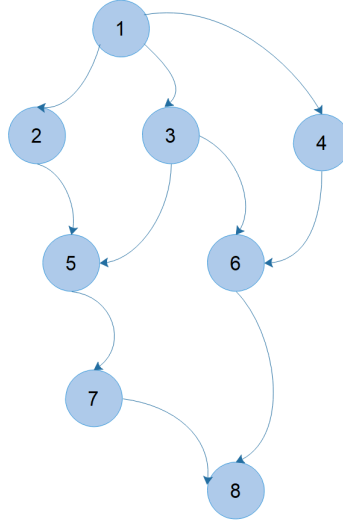


FIGURE 1.1 – Graphe de tâches.

Le temps d'achèvement du graphe de tâches est donc égale à e_{N_v} . En supposant l'indépendance des variables aléatoires définies sur les entrées de notre modèle, la probabilité de $(d_1, \dots, d_{N_v})$ est donnée par :

$$Pr(d_1, \dots, d_{N_v}) = \prod_i Pr(X_i = d_i)$$

Nous désignons par Ω l'espace d'états qui est le produit cartésien des supports des distributions d'entrée, $CompletionTime(d_1, \dots, d_{N_v})$ la variable aléatoire égale au temps d'exécution du graphe lorsque les retards sur les noeuds i sont distribués en fonction de D_i et $CT(d_1, \dots, d_{N_v})$ le temps d'achèvement du graphe de tâche à valeur dans R^+ quand les variables aléatoires d'entrée sont égales à $d_1, \dots, d_{N_v}$. Ainsi, en conditionnant les valeurs des variables aléatoires d'entrée, nous obtenons :

$$Pr(CompletionTime(D_1, \dots, D_{N_v}) = T) = \sum_{(d_1, \dots, d_{N_v}) \in \Omega} Pr(d_1, \dots, d_{N_v}) 1_{CT(d_1, \dots, d_{N_v})=T}$$

Ce problème a une complexité polynomiale lorsque les entrées sont déterministes. Par contre, lorsque les entrées sont des variables aléatoires, sa complexité devient exponentielle ce qui rend l'analyse des distributions sur les résultats complexe. Il est à noter que lorsque la taille du support des distributions est petite, la résolution du problème est encore possible. Une approche simple consiste à conditionner toutes les variables aléatoires afin d'obtenir un problème dont les entrées sont déterministes. Ainsi, en supposant que toutes les distributions ont les mêmes tailles (égales à A), la complexité de cette approche est égale à :

$$A^{N_v} B(N_v)$$

où $B(N_v)$ est égale à la complexité dans le cas déterministe pour un graphe de taille N_v .

Exemple 1.1 : Considérons le graphe acyclique $G(V,E)$ ci dessous dont les entrées sont des variables aléatoires. Soient $N_v = 4$ le nombre de noeuds du graphe et $A = 2$ la taille du support de chaque distributions d'entrée.

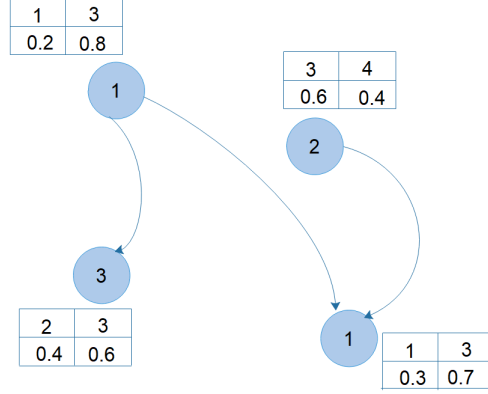


FIGURE 1.2 – Exemple de graphe de tâches avec des variables aléatoires en entrée.

À partir de ce modèle, nous pouvons obtenir $2^4 = 16$ graphes déterministes en entrées. Un exemple de graphe déterministe issu du modèle précédent est le graphe de tâches ci-dessous :

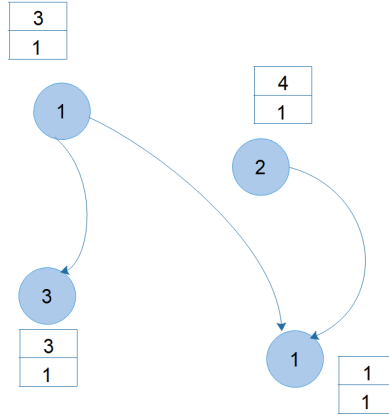


FIGURE 1.3 – Exemple de graphe de tâches avec des entrées déterministes.

où la probabilité d'obtenir cette configuration est $\Pr(d_1, d_2, d_3, d_4) = 0.8 * 0.4 * 0.3 * 0.6 = 0.0576$ et le chemin critique de cette configuration est $\text{Max}(3+3, 3+1, 4+1) = 6$. On réitère le même calcul pour les autres configurations

Ainsi, la probabilité que $\text{CompletionTime}(d_1, \dots, d_{N_v})$ soit égale à 6 est donnée par :

$$\Pr(\text{CompletionTime}(D_1, \dots, D_{N_v}) = 6) = \sum_{(d_1, \dots, d_{N_v}) \in \Omega} \Pr(d_1, \dots, d_{N_v}) 1_{CT(d_1, \dots, d_{N_v})=6} =$$

$$0.0576 + 0.1344 + 0.0864 + 0.2016 + 0.0504 + 0.0336 = 0.564$$

Comme la complexité pour cette famille de problèmes avec des variables aléatoires est NP, il n'existe pas d'algorithmes en temps polynomial sauf lorsque NP=P pour les cas généraux de ces problèmes.

Chapitre 2

Algorithmes de bornes

2.1 Définitions et propriétés des ordres stochastiques

Définition 1.1(Ordre Convexe croissante) Soient X et Y deux variables aléatoires, $X <_{icx} Y$ si pour toute fonction convexe croissante $\phi, E[(\phi(X))] \leq E[\phi(Y)]$ pourvu que l'espérance existe.

Définition 1.2(Ordre Convexe) Soient X et Y deux variables aléatoires, $X <_{cx} Y$ si pour toute fonction convexe $\phi, E[(\phi(X))] \leq E[\phi(Y)]$ pourvu que l'espérance existe.

Une définition similaire est la suivante : Soit X et Y deux variables aléatoires, $X <_{cx} Y$ si $X <_{icx} Y$ et $E[X] = E[Y]$ pourvu que l'espérance existe.

Définition 1.3(Ordre Strong) Soient X et Y deux variables aléatoires, $X <_{st} Y$ si pour toute fonction croissante $\phi, E[(\phi(X))] \leq E[\phi(Y)]$ pourvu que l'espérance existe.

Corollaire 1.1 Nous pouvons donc déduire les propositions suivantes :

- $X <_{st} X$ et $X <_{cx} X$
- si $X <_{st} Y$ alors $\forall k, E[X^k] \leq E[Y^k]$.
- si $X <_{cx} Y$ alors $\forall k > 1, E[X^k] \leq E[Y^k]$. Par conséquent, $Var[X] \leq Var[Y]$ (car $E[X] = E[Y]$)

Propriété 1.1 (Stop Loss) Soit X et Y deux variables aléatoires, $X <_{cx} Y$ si et seulement si $E[X] = E[Y]$ et pour tout $d, E[(X - d)^+] \leq E[(Y - d)^+]$.

Corollaire 1.2 Soit X et Y deux variables aléatoires, $X <_{cx} Y$ si et seulement si $E[X] = E[Y]$ et pour tout $d, E[(d - X)^+] \leq E[(d - Y)^+]$.

Propriété 1.2 (Addition) Soit X et Y deux variables aléatoires et Z une autre variable aléatoire indépendante de X et Y . Si $X <_{cx} Y$ alors $X + Z <_{cx} Y + Z$.

Propriété 1.3 (Espérance) Soit X et Y deux variables aléatoires d'espérance finie, $E[X] <_{cx} X$ et $E[X] + X <_{cx} 2X$.

Propriété 1.4 Soit X et Y deux variables aléatoires telles que : $X <_{st} Y$. Si $E[X] = E[Y]$ alors $X = Y$.

Propriété 1.5 Soit X, Y et Θ trois variables aléatoires telles que $[X|\Theta = a] \leq_{cx} [Y|\Theta = a]$ pour tout a dans le support de Θ , alors $X \leq_{cx} Y$.

Théorème 1.1 (Théorème A.344 de Shantikumar page 133) Soit X et Y deux variables aléatoires avec des espérances égales ($E[X] = E[Y]$) et avec des fonctions de répartition F_x et F_y respectivement. $X <_{cx} Y$ si et seulement si le nombre de changements de signe pour la fonction $F_x - F_y$ est égale à 1 et la séquence de changement de signe est $+, -$. Les cas où $F_x - F_y = 0$ sont rejetés (ne sont pas considérés comme un changement de signe).

2.2 Algorithme pour l'ordre Strong

L'intérêt est d'avoir une distribution bornante définie sur un support de taille plus petit ce qui permettra de réduire la complexité des opérations sur les distributions. De plus, si les opérations utilisées sont monotones alors les distributions des résultats sont également bornantes. Ce problème a fait l'objet d'une étude au préalable dans **la thèse de Farah (chapitre 6)** en utilisant l'ordre stochastique strong : Partant d'une distribution de probabilité discrète d de taille N avec une fonction de récompense positive croissante r , on a trouvé un algorithme pour la distribution bornante optimale inférieure d_2 (resp. supérieure d_1) de taille $K \leq N$ de d pour la récompense r . Cela se traduit par :

- $d_2 \leq_{st} d \leq_{st} d_1$
- Il n'existe pas de distribution d' (resp. d'') de taille K tels que $d_2 \leq_{st} d' \leq_{st} d$ et $\sum_i r[i]d_2[i] \leq \sum_i r[i]d'[i] \leq \sum_i r[i]d[i]$ (resp. $d \leq_{st} d'' \leq_{st} d_1$ et $\sum_i r[i]d[i] \leq \sum_i r[i]d''[i] \leq \sum_i r[i]d_1[i]$).

2.2.1 Algorithmes des bornes optimales pour l'ordre St

Les différents algorithmes ci dessous, présentés dans la thèse de **Farah Ait Salaht** permettent de déterminer les meilleures approximations d'une distribution pour une récompense donnée.

Algorithme de construction de la borne inférieure optimale

La détermination de la borne inférieure optimale selon **st** s'obtient à travers l'algorithme suivant :

Algorithme1 : Algorithme de construction de la borne inférieure optimale

Entrées :

d : Distribution de probabilités discrète définie sur l'espace d'état **H**, $|\mathbf{H}| = N$;

r : Récompense positive croissante ;

K : Taille du support F_2 associée à la distribution bornante.

Sorties :

d2 : Borne inférieure optimale de d définie sur F_2 , $|\mathbf{F}_2| = K$;

R_{d2} : Espérance des récompenses de $d2$;

1. $R_d = \sum_i r[i]d[i]$ /*espérance de récompense de d^* /
2. $X[i, j] = X[i, j - 1] + d[j](r[j] - r[i]), \forall i \in \{1, \dots, N\}, \forall j \in \{i+1, \dots, N\}$;
3. $T[i, j] = \infty, \forall i, j \in \{1, \dots, N\}$; $Save[i, j] = 0$ et $Save[i, 2] = \text{ÉtatMin}$ $\forall i, j \in \{1, \dots, N\}$;
4. $T[i, 2] = X[1, i-1], \forall i \in \{2, \dots, N\}$;
5. $l \in \{3, \dots, N\}, i \in \{2, \dots, N\}, j \in \{2, \dots, i-1\}$
 if $T[j, l-1] + X[j, i-1] < T[i, l]$ then
 $T[i, l] = T[j, l-1] + X[j, i-1]$;
 $Save[i, l] = f_j$;
6. $T[i, K] = T[i, K] + X[i, N], \forall i \in \{1, \dots, N-1\}$;
7. $m_i = \text{argmin}_s \{T[s, K]\}$; /* indice de l'état ayant le résidu minimal */
8. $F_2 = \{f_{m_i}\}$ /* construction de F_2 associé à la distribution $d2$ */
9. for $l=K$ to 2 do
 $F_2 = F_2 \cup Save[m_i, l]$;
 $m_i = Save[m_i, l]$;
10. $d2[m] = \sum_{s \in \{\Gamma^- H(F_2[m+1]) \cap \Gamma^+ H(F_2[m])\} \cup F_2[m]} d[s] \forall m \in \{1, \dots, K-1\}$;
 $d2[K] = \sum_{s \in \{\Gamma^+ H(F_2[K]) \cup H(F_2[K])\}} d[s]$;
11. $R_{d2} = R_d - \min_s T[s, K]$.

Algorithme de construction de la borne supérieure optimale

La détermination de la borne supérieure optimale selon **st** s'obtient à travers l'algorithme suivant :

Algorithme2 : Algorithme de construction de la borne supérieure optimale

Entrées :

d : Distribution de probabilités discrète définie sur l'espace d'état **H**, $|\mathbf{H}| = N$;

r : Récompense positive croissante ;

K : Taille du support F_1 associée à la distribution bornante.

Sorties :

d2 : Borne supérieure optimale de d définie sur F_1 , $|\mathbf{F}_1| = K$;

R_{d1} : Espérance des récompenses de $d1$;

1. $R_d = \sum_i r[i]d[i]$ /*espérance de récompense de d^* /

2. for $i=N$ to 1 do
 - for $j=i-1$ to 1 do

$$X[i,j]=X[i,j+1]+d[j](r[i]-r[j]) ;$$
 3. $T[i,j] = \infty, \forall i, j \in \{1,...,N\}$; $Save[i,j] = 0$ et $Save[i,2]=\acute{E}tatMax \forall i, j \in \{1,...,N\}$;
 4. $T[i,2]=X[N,i+1], \forall i \in \{1,...,N-1\}$;
 5. $l \in \{3, ..., K\}, i \in \{N-1, ..., 1\}, j \in \{N-1, ..., i+1\}$
 if $T[j, l-1] + X[j, i+1] < T[i, l]$ then

$$T[i, l] = T[j, l-1] + X[j, i+1];$$

$$Save[i, l] = f_j;$$
 6. $T[i,K]=T[i,K]+X[i,1], \forall i \in \{2,...,N\}$;
 7. $ma = argmin_s \{T[s, K]\}$; /* indice de l'état ayant le résidu minimal */
 8. $F_1 = \{f_{ma}\}$ /* construction de F_2 associé à la distribution d_2 */
 9. for $l=K$ to 2 do

$$F_l = F_1 \cup Save[ma, l]; \quad ma = Save[ma, l];$$
 10. $d1[l] = \sum_{s \in \{\Gamma^- H(F_1[l]) \cap \Gamma^+ H(F_1[l-1])\} \cup F_1[l]} d[s] \forall m \in \{1,...,K-1\}$;

$$d1[1] = \sum_{s \in \{\Gamma^+ H(F_1[1]) \cup H(F_1[1])\}} d[s];$$
 11. $R_{d_1} = R_d + T[ma, K]$.
-

Avec :

X[i,j] qui correspond à la différence de récompense engendrée par la suppression de l'état f_j en sachant que l'état f_i est conservé.

T[i,l] qui correspond au résidu minimal associé au chemin de longueur $l-1$ dans la séquence $\{f_1, ..., f_i\}$.

Save[i,j] qui correspond au dernier état conservé dans la séquence $\{f_1, ..., f_i\}$, pour un chemin de longueur $l-1$.

Ces algorithmes ont été implémentés en Matlab et ont une complexité en $O(N^2K)$ qui devient cubique lorsque K a le même ordre que N . Mais nous l'avons repris en Java (afin d'effectuer le calcul plus rapidement) et représenter sous forme graphique les résultats obtenus en Python.

Exemple 2.1 : On considère notre graphe de tâches G de départ, associé à des distributions de probabilités discrètes définies sur les variables aléatoires indépendantes. Nous avons trois instances qui sont générées pour ce graphique. La première instance, le temps d'achèvement suit une distribution uniforme entre $\{1,...,10\}$. Dans la seconde(resp. la troisième) instance le temps d'achèvement suit la distribution de **Zipf1**(resp.**Zipf2**) avec un exposant égale à **1,4267**(resp. **2**) ayant pour support $\{1,...,10\}$. La figure ci-dessous présente la distribution des limites supérieure et inférieure pour l'ordre strong pour les distributions ayant seulement 5 atomes. Les distributions d'entrée pour les nœuds sont la distribution Zipf avec un exposant égal à **1.4267**.

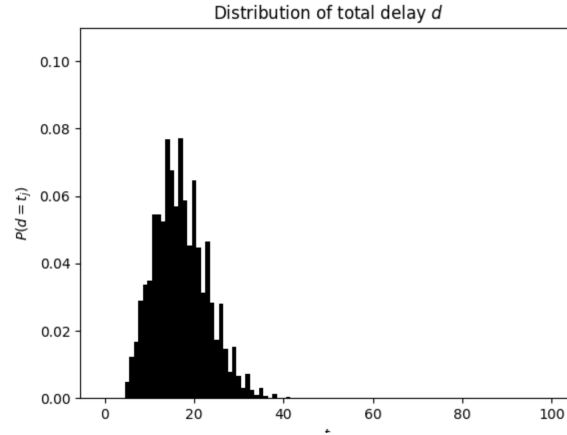


FIGURE 2.1 – Distribution du temps d’achèvement du graphe de tâches pour la borne supérieure avec 5 atomes comme distributions d’entrées.

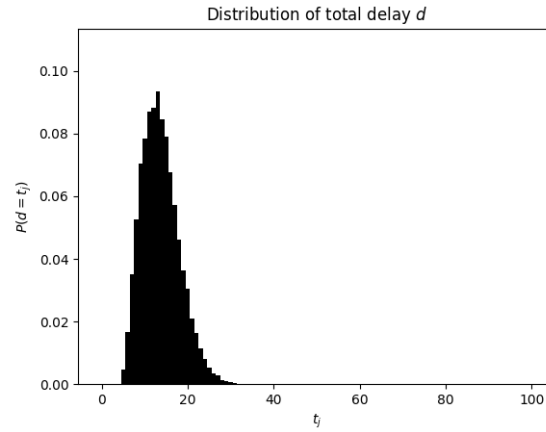


FIGURE 2.2 – Distribution du temps d’achèvement du graphe de tâches pour la borne inférieure avec 5 atomes comme distributions d’entrées.

Sur la figure 6 (resp. figure 7), nous montrons les limites optimales sur 5 (resp. sur 4) atomes lorsque la distribution initiale est une distribution uniforme sur 10 atomes.

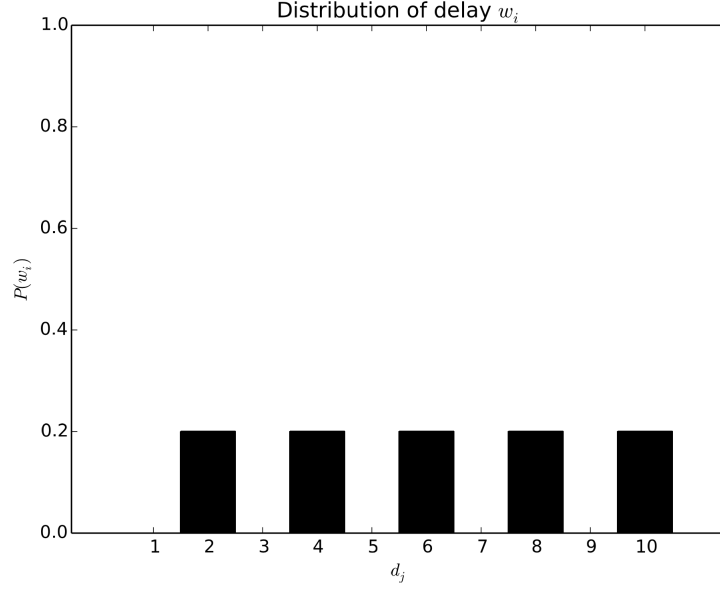


FIGURE 2.3 – Distribution de la borne supérieure avec 5 atomes. la distribution initiale est uniforme avec 10 atomes

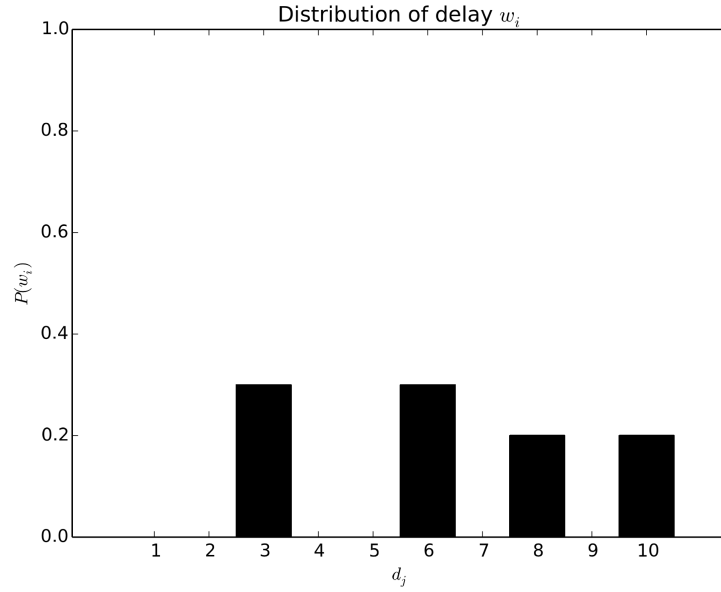


FIGURE 2.4 – Distribution de la borne supérieure avec 4 atomes. la distribution initiale est uniforme avec 10 atomes

2.3 Borne inférieure convexe

Dans la suite, nous allons prouver que nous pouvons réduire la taille du support des distributions lorsqu'elle est trop grande, tout en conservant la limite inférieure selon l'ordre stochastique convexe.

2.3.1 Construction de la borne inférieure convexe

À travers les différents lemmes ci-dessous, nous essayons de construire une borne inférieure convexe. Il est à noter que nous n'avons pas procédé ainsi pour la construction de la borne inférieure. Mais, cette construction nous permet de voir comment procéder afin d'obtenir une borne inférieure pour l'ordre stochastique convexe à partir d'un ensemble d'atomes.

Lemme 2.1 *On considère une distribution arbitraire discrète **D1** sur deux atomes a et b (avec $a < b$) définis par les probabilités strictement positives p_a et p_b . Alors la distribution **D2** avec un unique atome M égale à $\frac{ap_a + bp_b}{p_a + p_b}$ est une borne inférieure convexe. On note $D2 \leq_{cx} D1$*

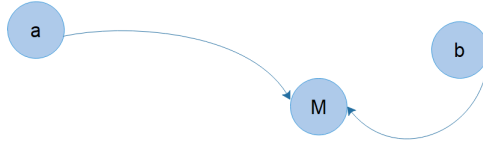


FIGURE 2.5 – Fusion de deux atomes pour la borne inférieure.

Preuve 2.1 : L'application de la propriété 1.3

Lemme 2.2 *On considère une distribution **D1** sur trois atomes a, b, c (avec $a < b < c$) définis par les probabilités strictement positives p_a, p_b, p_c . Supposons que $b > \frac{ap_a + cp_c}{p_a + p_c}$. Alors la distribution **D2** avec deux atomes en a et b avec les probabilités q_a et q_b données par*

$$q_a + q_b = 1$$

et

$$aq_a + bq_b = ap_a + bp_b + cp_c$$

est une borne inférieure convexe de **D1**.

Preuve 2.2 : La preuve repose sur deux étapes : calculer q_a et q_b et vérifier qu'il s'agit bien d'une distribution. Puis, on doit vérifier que la distribution est une borne inférieure au sens convexe.

Comme $a \neq b$, le déterminant est non nul donc le système a une unique solution. Cette solution est :

$$q_a = p_a + p_c \frac{c-b}{a-b}$$

et

$$q_b = p_b + p_c \frac{a-c}{a-b}$$

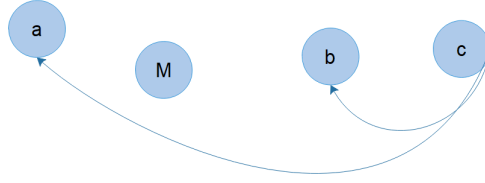


FIGURE 2.6 – Borne inférieure dans le premier cas

Comme $a < b < c$, il est clair que $q_b > 0$. Par contre $q_a < p_a$ et on doit vérifier le signe de q_a .

On a

$$(a - b)q_a = (a - b)p_a + (c - b)p_c = ap_a + cp_c - b(p_a + p_c) \leq 0$$

La dernière inégalité provient de la condition du lemme. Puisque $a < b$ on a donc $q_a > 0$. La distribution est bien définie. On vérifie maintenant que c'est une borne inférieure au sens convexe en décomposant selon les valeurs de d par rapport aux trois points dans le support de **D1**.

- $d \geq c$. $(X - d)^+$ et $(Y - d)^+$ sont nuls. L'inégalité est vérifiée.
- $b \leq d < c$. $(Y - d)^+$ est nul. Par contre, $(X - d)^+$ est non nul quand X vaut c . Et on a

$$0 = \mathbf{E}[(Y - d)^+] \leq \mathbf{E}[(X - d)^+] = (c - d)p_c$$

car $d < c$.

- $a \leq d < b$. $(X - d)^+$ est non nul quand $X = c$ ou $X = b$. Donc

$$\mathbf{E}[(X - d)^+] = (c - d)p_c + (b - d)p_b$$

$(Y - d)^+$ est non nul quand $Y = b$. Donc

$$\mathbf{E}[(Y - d)^+] = (b - d)q_b$$

Comme $q_b = p_b + p_c \frac{a-c}{a-b}$, cela revient, après simplifications, à comparer $(c-d)$ et $\frac{(b-d)(a-c)}{a-b}$:

Par construction, on a : $a \leq d < b$ Donc $(c - b)(a - d) \leq 0$.

En effet,

$$(c - b)(a - d) = (c - d)(a - b) + (c - a)(b - d) \leq 0.$$

$$\Rightarrow (c - d)(a - b) \leq (a - c)(b - d)$$

Comme $a < b$, alors

$$(c - d) \geq \frac{(b - d)(a - c)}{a - b}$$

Donc $\mathbf{E}[(Y - d)^+] \leq \mathbf{E}[(X - d)^+]$ pour tout d entre a et b .

- $d < a$. $(X - d)^+$ et $(Y - d)^+$ sont non nuls. Donc $\mathbf{E}[(X - d)^+] = \mathbf{E}[X - d] = \mathbf{E}[X] - d$.

De même, $\mathbf{E}[(Y - d)^+] = \mathbf{E}[Y] - d$. Comme $\mathbf{E}[X] = \mathbf{E}[Y]$ grâce à la propriété 1.1, l'inégalité est vérifiée.

Exemple 2.2 Soit **D1** une distribution discrète définie sur $\mathcal{H}1 = \{1, 3, 4\}$ ($a=1, b=3, c=4$) avec les probabilités suivantes $[0.2, 0.6, 0.2]$. Ainsi, $E[D1] = 2.8$, et le barycentre entre 1 et 4 est $M = 2.5$. ($b > M$). En appliquant le lemme 2.2, la distribution de la borne inférieure **D2** peut être construite. **D2** est définie sur $\mathcal{H}2 = \{1, 3\}$ avec les probabilités $q_2 = 0.1$ et $q_4 = 0.9$, ainsi $E[D2] = 2.8$.

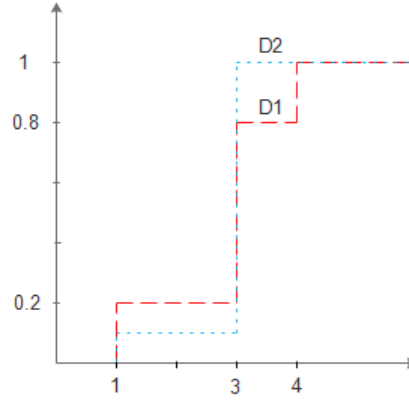


FIGURE 2.7 – Les distributions cumulées de F_{D1} and F_{D2}

Nous allons par la suite considérer les différents cas possibles. Supposons J l'intervalle de cette valeur.

- $J < 3$, alors $F_{D1}(3^-) - F_{D2}(3^-) = 0.2 - 0.1 > 0$.
- $3 \leq J < 4$, la comparaison de $F_{D1}(4^-) - F_{D2}(4^-)$ nous donne :

$$p_1 + p_2 - q_2 - q_4 = 0.2 + 0.6 - 0.1 - 0.9 < 0$$

- $J \geq 4$, la fonction $F_{D1} - F_{D2} = 0$.

Comme le nombre de modifications du signe est égale à 1 et la séquence est +,- alors d'après le théorème 2.1 nous obtenons

$$D2 \leq_{cx} D1$$

Lemme 2.3 On considère une distribution **D1** sur trois atomes a, b, c (avec $a < b < c$) définis par les probabilités strictement positives p_a, p_b, p_c . Supposons que $b < \frac{ap_a + cp_c}{p_a + p_c}$. Alors la distribution **D2** avec deux atomes en b et c avec les probabilités q_b et q_c données par

$$q_b + q_c = 1$$

et

$$cq_c + bq_b = ap_a + bp_b + cp_c$$

est une borne inférieure convexe de **D1**.

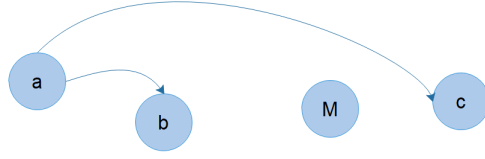


FIGURE 2.8 – Borne inférieure dans le second cas

Preuve 2.3 : La preuve est analogue à celle du lemme précédent : calculer q_b et q_c et vérifier qu'il s'agit bien d'une distribution. Puis, on doit vérifier que la distribution est une borne inférieure au sens convexe.

Comme $b \neq c$, le déterminant est non nul donc le système a une unique solution. Cette solution est :

$$q_b = p_b + p_a \frac{c-a}{c-b}$$

et

$$q_c = p_c + p_a \frac{a-b}{c-b}$$

Comme $a < b < c$, il est clair que $q_b > 0$. De même, $q_c > 0$ car d'après l'hypothèse de départ on a :

$$b < \frac{ap_a + cp_c}{p_a + p_c}$$

$$\Rightarrow b(p_a + p_c) < ap_a + cp_c$$

$$\Rightarrow p_c(b - c) + p_a(b - a) < 0$$

$$\Rightarrow p_c(c - b) + p_a(a - b) > 0$$

Puisque $(c - b) > 0$, on a donc $q_c > 0$. La distribution est bien définie. On vérifie maintenant que c'est une borne inférieure au sens convexe en décomposant selon les valeurs de d par rapport aux trois points dans le support de **D1**.

- $d \geq c$. Comme précédemment $(X - d)^+$ et $(Y - d)^+$ sont nuls. L'inégalité est vérifiée.
- $b \leq d < c$. $(X - d)^+$ est non nul pour $X = c$ et $\mathbf{E}[(X - d)^+] = (c - d)p_c$. De même, $(Y - d)^+$ est non nul pour $Y = c$ et $\mathbf{E}[(Y - d)^+] = (c - d)q_c$. Clairement $q_c \leq p_c$, et $(c - d)$ est positive, donc l'inégalité est vérifiée.
- $a \leq d < b$. $(X - d)^+$ est non nul quand $X = c$ ou $X = b$. Donc

$$\mathbf{E}[(X - d)^+] = (c - d)p_c + (b - d)p_b$$

$(Y - d)^+$ est non nul quand $Y = c$ ou $Y = b$. Donc

$$\mathbf{E}[(Y - d)^+] = (c - d)q_c + (b - d)q_b$$

Comme $q_b = p_b + p_a \frac{c-a}{c-b}$ et $q_c = p_c + p_a \frac{a-b}{c-b}$, après simplifications on obtient :

$$\mathbf{E}[(Y - d)^+] = (c - d)p_c + (b - d)p_b + p_a \frac{(c-d)(a-b) + (c-a)(b-d)}{c-b}$$

Puisque $(c - b) > 0$, il suffit de vérifier le signe de $(c-d)(a-b) + (c-a)(b-d)$.

En effet,

$$(c - d)(a - b) + (c - a)(b - d) = (c - b)(a - d) \leq 0.$$

Donc $\mathbf{E}[(Y - d)^+] \leq \mathbf{E}[(X - d)^+]$ pour tout d entre a et b .

— $d < a$. $(X - d)^+$ et $(Y - d)^+$ sont non nuls. Donc $\mathbf{E}[(X - d)^+] = \mathbf{E}[X - d] = \mathbf{E}[X] - d$.

De même, $\mathbf{E}[(Y - d)^+] = \mathbf{E}[Y] - d$. Comme $\mathbf{E}[X] = \mathbf{E}[Y]$ grâce à la propriété 2.1, l'inégalité est vérifiée.

2.3.2 Heuristique pour une borne inférieure convexe

Contrairement à la borne supérieure, nous n'avons pas trouvé un algorithme pour la borne inférieure optimale de l'ordre convexe. Néanmoins, nous avons proposé un heuristique basé sur le lemme 2.1 qui permet de calculer une borne inférieure. Ainsi, à chaque étape de l'heuristique, nous remplaçons deux atomes par un nouvel atome et par construction, nous conservons les mêmes attentes et le second moment (ie $f(x) = r^2$) diminue. Plus précisément, soient a et b deux atomes arbitraires, nous cherchons le couple (a, b) qui minimise la différence pour le second moment Δ afin de conserver le second moment et la variance.

$$\Delta(a, b) = p_a a^2 + p_b b^2 - (p_a + p_b) \left(\frac{p_a a + p_b b}{p_a + p_b} \right)^2$$

Après plusieurs réductions, nous obtenons : $\Delta(a, b) = \frac{p_a a p_b b}{p_a + p_b} (a - b)^2$. La complexité de cet heuristique est N^2 .

Algorithme 3 : Heuristique pour la borne inférieure pour l'ordre convexe

Entrées :

d1 : Distribution de probabilités discrète définie sur l'espace d'état $\mathbf{H}$, $|\mathbf{H}| = N$;

Sorties :

d2 : Borne supérieure de d définie sur F_1 , $|\mathbf{F}_1| = K$;

1. D2=D1
2. pour tout atome a faire
 - pour tout atome b faire
 - Calcule $\Delta(a, b)$ et mettre le résultat dans une structure de données.
3. finpour
4. pour i de 1 à $K+1$ faire
 - Cherche le couple (a, b) qui minimise $\Delta(a, b)$.
 - Faire la fusion de a et de b dans **D2**. Soit c , l'atome résultant de cette fusion.

5. Mettre à jour la matrice $\Delta(x, y)$ (supprimer les entrées liées à a et b , ajouter des entrées liées à c).
 6. finpour
-

Nous pouvons également obtenir une borne inférieure en utilisant n'importe quelle fonction convexe r et en modifiant la valeur de Δ tout en gardant la même heuristique. Δ reste toujours la différence de récompenses et est égale à :

$$\Delta(a, b) = p_a r(a) + p_b r(b) - (p_a + p_b) r\left(\frac{p_a a + p_b b}{p_a + p_b}\right)$$

Exemple 2.3 : Nous reprenons l'exemple 1 pour l'adapter à l'ordre convexe. Soit le graphe de tâches G initial, associé à des distributions de probabilités discrètes définies sur les variables aléatoires indépendantes. Nous avons également trois instances qui sont générées pour ce graphique. La première instance, le temps d'achèvement suit une distribution uniforme entre $\{1, \dots, 10\}$. Dans la seconde (resp. la troisième) instance le temps d'achèvement suit la distribution de Zipf1 (resp. Zipf2) avec un exposant égale à 1,4267 (resp. 2) ayant pour support $\{1, \dots, 10\}$. La figure ci-dessous présente la distribution limite inférieure pour l'ordre convexe pour les distributions ayant seulement 5 atomes. Les distributions d'entrée pour les nœuds sont la distribution Zipf avec un exposant égal à 2.

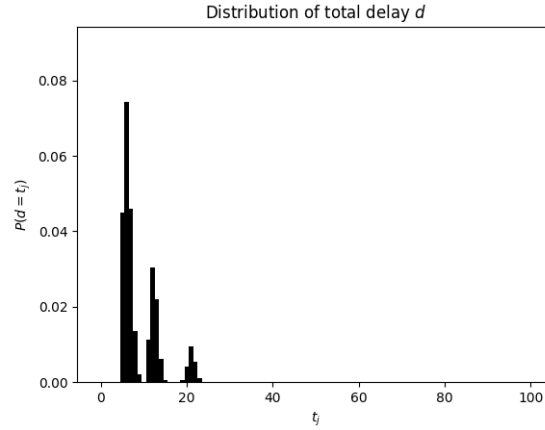


FIGURE 2.9 – Distribution du temps d'achèvement du graphe de tâches pour la borne inférieure avec 5 atomes comme distributions d'entrées.

Sur la figure 13 (resp. figure 14), nous montrons les limites optimales sur 5 (resp. sur 4) atomes lorsque la distribution initiale est une distribution Zipf d'exposant 1,4267 sur 10 atomes.

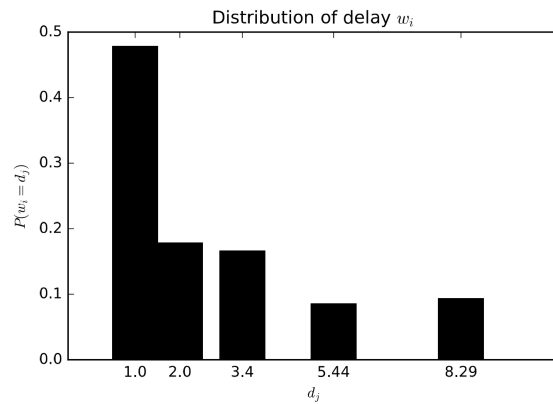


FIGURE 2.10 – Distribution de la borne supérieure avec 5 atomes. la distribution initiale est Zip d'exposant 1,4267 avec 10 atomes

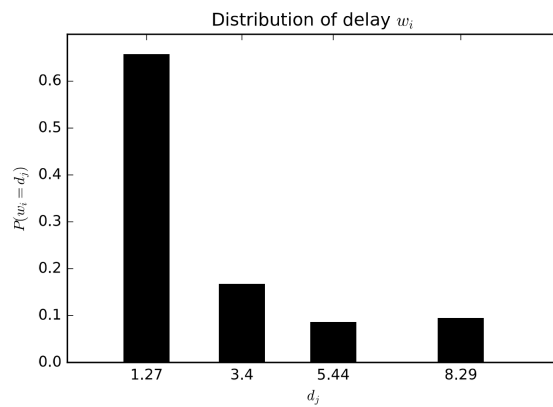


FIGURE 2.11 – Distribution de la borne supérieure avec 4 atomes. la distribution initiale est Zipf d'exposant 1,4267 avec 10 atomes

Conclusion

Ce stage a été très enrichissant pour moi car il m'a permis de travailler dans différents laboratoires de recherche. J'ai particulièrement apprécié. Ce stage m'a également permis d'assister à des séminaires de présentations des différents travaux de recherche. J'ai pu développer une vision globale du domaine de la recherche et je suis ravi d'avoir pu contribuer à un tel sujet.

il a été question pour nous d'analyser les distributions des résultats du problème d'optimisation en considérant les distributions de probabilités sur les tâches. Parmi les différents problèmes d'optimisation, notre approche a été illustrée par le problème d'optimisation sur le grâphe de tâches.

Nous avons trouvé une heuristique permettant le calcul d'une borne inférieure au sens convexe. Mais, nous n'avons pas de preuve pour l'optimalité de la borne inférieure. Cet ainsi que plusieurs questions restent encore en suspens. Pour la suite de la recherche, nous allons étudier une nouvelle méthode de partitionnement (K-Moyennes) sur les distributions d'entrées des noeuds dans le but d'obtenir un algorithme optimal pour la borne inférieure.

Bibliographie

J-M Fourneau, N. Pekergin, J. Cohen, A. Fauquette, and G.C. Noulkela. *Convex Stochastic Bounds and Stochastic Optimisation on Graphs*. Ninth International Workshop on Practical Applications of Stochastic, PASM 2017.

Farah Ait Salaht. *Chaines de Markov Incomplètement Spécifiées : analyse par comparaison stochastique et application à l'évaluation de performance des réseaux*. Thèse PHD, 2014.

J-M Fourneau. *Agrégation de données par comparaisons stochastiques*. DAVID, Université de Versailles, 2017.

Wikipédia : K-moyennes,
<https://fr.wikipedia.org/wiki/K-moyennes> 20-08-2017, 2017