

PROJET
UE Méthodes de ranking et recommandations
Sujet 5 : Simulation d'un Google Bombing

Maxime Gonthier - Laureline Martin

22 avril 2019

Table des matières

1	Introduction	4
2	Manuel utilisateur	4
3	Plan	5
4	I. Tests initiaux et hypothèses	6
4.1	Explications du code	6
4.2	Résultats des tests initiaux	6
4.2.1	Stanford.txt sans modification	6
4.2.2	Résultats	7
4.3	Conclusion des tests initiaux	7
4.3.1	Hypothèses	7
5	II. Tests avec un nombre de graphes générés aléatoirement	8
5.1	Explication du code	8
5.2	Résultats des tests	9
5.3	Analyse graphes en anneau	9
5.4	Analyses Graphes complet	11
5.5	Analyses Arbres	13
5.6	Conclusion sur l'impact de chaque structure	15
6	III. Expériences avec un graphe de degré aléatoire	15
6.1	Résultats des tests	15
6.2	Analyses	17
7	IV. Expériences sur l'efficacité	18
7.1	Résultats des tests	18
7.2	Analyses	18
8	Conclusions	19

1 Introduction

Le but de ce projet est de simuler, d'évaluer et de proposer la méthode la plus efficace pour un Google Bombing. Il s'agit d'une technique d'attaque sur le graphe du web afin d'augmenter la pertinence d'une page cible. Pour cela, plusieurs attaquants créent des pages et y insèrent des liens vers une page cible à attaquer. Plusieurs structures de pages attaquantes sont possibles :

- seul
- complet
- anneau
- arbre

Nous allons tester ces différentes structures sur des pages de pertinences différentes et déterminer quelle est la structure la plus avantageuse pour obtenir la pertinence la plus forte possible.

2 Manuel utilisateur

Notre application se lance grâce à la commande **make all** suivie de fichier ".txt" à modifier. **make clean** pour nettoyer et **make compil** pour simplement compiler.

Après le **make all** voilà ce qui s'affiche :

```

user@debian:~/Bureau/Ranking$ make clean
user@debian:~/Bureau/Ranking$ make compil
user@debian:~/Bureau/Ranking$ make all Stanford/Stanford.txt
./ranking Stanford/Stanford.txt
Random cible si besoin : 241859

pages : 281903
lien : 2312497
Entrez la structure que vous voulez insérer :
1 pour un sommet seul
2 pour un anneau
3 pour un graphe complet
4 pour un arbre
5 pour un nombre d'anneaux aléatoire
6 pour un nombre de graphes complet aléatoire
7 pour un nombre d'arbres aléatoire
8 pour un arbre a attaquant unique
9 pour un graphe de 100 sommets avec un degré aléatoire
10 pour un graphe avec une certaine probabilité de connecter les sommets entre eux
2
Entrez le nombre de structure que vous voulez ajouter :
1
Entrez le nombre de sommets par structure a ajouter :
150
Entrez la pertinence de la cible :
1 pour une pertinence forte
2 pour une pertinence moyenne
3 pour une pertinence faible
4 Pour entrer manuellement le sommet cible
1
Entrez le numéro du sommet cible :
1
282053 pages
2312797 liens
File read!
132 itérations
Done!
La pertinence du sommet cible est : 0.000207197
Le temps d'execution est de : 34.565403 secondes

```

Dans un premier temps l'utilisateur choisit la structure qu'il veut ajouter au graphe du web. Il y a des structures simples (arbre, anneau, graphe complet, sommet seuls), des structures moins conventionnelles (arbre à attaquant unique, un arbre où seul le sommet racine pointe vers la cible). On peut aussi ajouter un nombre de structures aléatoires (avec une limitation de 100 sommets). On peut également insérer un graphe de degré aléatoire ou un graphe dont le degré est une probabilité. Ensuite l'utilisateur va entrer le nombre de structures qu'il veut, cela ne s'applique pas pour les structures numéros 5, 6 et 7. Ensuite on entre le nombre de sommets de la ou les structures à ajouter. Puis on entre le sommet cible. Le numéro du sommet cible est demandé une seconde fois pour savoir quel sommet sera donné en résultats. Enfin le programme affiche le nombre de pages/liens, le temps d'exécution et la pertinence du sommet cible.

3 Plan

Dans un premier temps nous allons effectu  des tests simples en utilisant quatre type de structures diff rentes ainsi que trois cibles de pertinences diff rentes. Nous en d duirons des hypoth ses sur l'efficacit  de chaque structure et l'impact de la pertinence de la cible sur cette m me efficacit . Dans un second temps nous  tudierons l'impact sur la pertinence de graphes g n r s en nombre al atoire pour  tudier qu'elle structure est globalement la plus efficace. Puis nous testerons la meme choses avec cette fois ci un graphe dont les arcs le reliant a lui meme sont g n r es al atoirement. Enfin nous ferons varier le nombre de sommets du graphe ataquant pour en d duire son efficacit  et nous conclurons.

4 I. Tests initiaux et hypoth ses

L'algorithme power calculant les pertinences est contenu dans le fichier *ranking.c*. Il n'est pas d taill  dans ce rapport mais le fichier est comment . On consid re ici le graphe du web Stanford.txt. Ce graphe est modifi  par l'ajout de sommets et d'arcs afin d'augmenter la valeur d'un sommet cibl . On consid re que :

- Les sommets repr sentent les pages du web.
- Les arcs repr sentent les liens dirigeant vers d'autres pages.
- Les valeurs des sommets repr sentent les pertinences calcul s par l'algorithme pagerank.
- Le sommet cible repr sente la page dont on souhaite augmenter la pertinence.
- 100 sommets sont ajout s pour chaque test.

4.1 Explications du code

A faire.

4.2 R sultats des tests initiaux

4.2.1 Stanford.txt sans modification

281903 pages
2312497 liens
132 it rations
27.627466 secondes

Voici les pertinences de base des trois sommets  tudi s :

Pertinence forte : Page 280545 - 9.96199e-05

Pertinence moyenne : Page 281466 - 7.53954e-06

4.2.2 Résultats

Pertinences des sommet cible	Forte	Moyenne	Faible
Sans modification	9,96E-05	7,54E-06	6,05E-07
Difference sans modification	0,00E+00	0,00E+00	0,00E+00
Attaque a 100 sommets seuls	1,82E-04	5,89E-05	5,19E-05
Differences avec 100 sommets seuls	8,25E-05	5,13E-05	5,13E-05
Attaque avec un anneau de 100 sommets	1,71E-04	5,22E-05	4,52E-05
Differences avec un anneau a 100 sommets	7,17E-05	4,46E-05	4,46E-05
Attaque avec un graphe complet de 100 sommets	1,05E-04	1,08E-05	3,84E-06
Differences avec un graphe complet a 100 sommets	5,17E-06	3,23E-06	3,24E-06
Attaque avec un arbre de 100 sommets	1,72E-04	5,25E-05	4,55E-05
Differences avec un arbre a 100 sommets	7,22E-05	4,49E-05	4,49E-05
Attaque avec un arbre a attaquant unique de 100 sommets	1,38E-04	3,17E-05	2,47E-05
Differences avec un arbre a attaquant seul a 100 sommets	3,88E-05	2,41E-05	2,41E-05

4.3 Conclusion des tests initiaux

Cinq structures différentes sont ajoutées : des sommets seuls, un graphe complet, un anneau, un arbre et un arbre a attaquant unique. L'arbre a attaquant unique est un arbre dont seul le sommet racin est relié à la cible. L'objectif de cette structure est de minimiser le nombre de sommets reliés à la cible pour ainsi minimiser les chances de se faire repérer lors de l'attaque. On observe pour chaque cible que la structure la plus efficace est le sommet seul, suivis de l'arbo, de l'anneau, de l'arbre a attaquant unique et enfin du graphe complet.

La raison de cette ordre serait que pour des sommets seuls, la probabilité de pointer sur la cible est de 1/1 alors qu'elle est de 1/n pour un graphe complet avec n le nombre de sommets. Ainsi la probabilité est diluée et donc la cible en sera moins modifiée.

On observe que pour chaque structure la différence de modification de la pertinence est presque identique entre les cibles faibles et moyennes. Ainsi on peut en déduire que modifier la pertinence d'une cible à e-06 ou e-07 est de même difficulté.

Pour le sommet seul on observe que la modification de pertinence (la différence) pour une cible forte, moyenne ou faible sont respectivement 8.25e-05, 5.13e-05 et 5.13e-05. Ainsi on peut en déduire que modifier la pertinence est plus significatif quand la cible est déjà forte.

4.3.1 Hypothèses

On suppose que les structures les plus efficaces sont les sommets seuls et les arbres. On suppose que changer la pertinence d'une cible devient plus difficile à partir d'une cible de pertinence Xe-05 et que c'est assez identique pour les cibles de pertinences Xe-06 et Xe-07.

5 II. Tests avec un nombre de graphes générés aléatoirement

Nous allons désormais insérer des graphes générés aléatoirement. Ce qui est aléatoire n'est pas la structure des graphes mais le nombre de graphes générés.

L'objectif de cette démarche est de déterminer quel structure est la plus efficace globalement.

C'est à dire quelle structure influe le plus sur la pertinence quelle que soit la situation.

De plus les resultats nous aiderons aussi a determiner l'impact qu'a une certaine structure sur la cible de manière plus générale que dans les cas prédéfinis de la partie précédente. Le nombre de sommet des graphes ajoutées est fixé a 100. La cible est également fixé ainsi que la structure des graphes ajoutées. On va par exemple insérez 5 graphes complet de nombre de sommet respectivement : 25, 10, 5, 6 et 4. Tous reliées au sommet cible. Dans un premier temps nous allons expliquer le code derrière cette démarche puis nous analyserons les résultats.

5.1 Explication du code

Tous est dans le fichier *ajoutsommetsattanquants.c*. Les fonctions utilisées sont *ajoutanneaualeatoire*, *ajoutcompletaleatoire* et *ajoutarbrealeatoire*. Ces fonctions reprennent en partie le code des trois fonctions presque eponyme décrite précédemment. Regardons ce qui a changé.

```
while (nbsommetrestant > 0) {  
    nbajout = rand()%(100-nouveausommets);  
    if (nbajout <= 3) { nbajout = 3; }  
    nbsommetrestant -= nbajout;
```

nbajout représente le nombre de sommet que l'on va ajouter dans la première structure créer. Il choisit donc un nombre aléatoire entre 0 et 100 car le nombre de sommet total est limité a 100. Si le nombre choisis est inférieur a 3 on le fixe a 3 car créer des anneau ou des graphes complets de tailles inférieure a 3 reviens juste a créer des sommets seuls. *nbajout* est enlevé au nombre de sommet restant a ajouter. Puis on lance la construction de la structure comme vu précédemment avec *nbajout* représentant le nombre de

sommets. A la fin de cette itération *nbajout* reprends un nombre aléatoire qui cette fois prend une valeur entre 0 et 100 moins le nombre de sommets ajouté précédemment représenté par *nouveausommets*.

5.2 Résultats des tests

Type de graphe	Sans modification	Anneau	Graphe complet	Arbre
Pertinences	9,96E-05	1,7279E-04	1,1072E-04	1,7433E-04
	9,96E-05	1,7207E-04	1,1198E-04	1,7345E-04
	9,96E-05	1,7279E-04	1,2006E-04	1,7417E-04
	9,96E-05	1,7135E-04	1,1040E-04	1,7326E-04
	9,96E-05	1,7135E-04	1,1179E-04	1,7372E-04
	9,96E-05	1,7135E-04	1,1042E-04	1,7387E-04
	9,96E-05	1,7279E-04	1,1173E-04	1,7378E-04
	9,96E-05	1,7207E-04	1,1234E-04	1,7456E-04
	9,96E-05	1,7207E-04	1,1073E-04	1,7265E-04
	9,96E-05	1,7279E-04	1,1796E-04	1,7477E-04
	9,96E-05	1,7207E-04	1,1729E-04	1,7273E-04
	9,96E-05	1,7207E-04	1,1960E-04	1,7287E-04
	9,96E-05	1,7135E-04	1,2463E-04	1,7450E-04
	9,96E-05	1,7207E-04	1,1456E-04	1,7338E-04
	9,96E-05	1,7207E-04	1,1443E-04	1,7474E-04
	9,96E-05	1,7207E-04	1,1335E-04	1,7332E-04
	9,96E-05	1,7135E-04	1,1246E-04	1,7427E-04
	9,96E-05	1,7207E-04	1,1346E-04	1,7272E-04
	9,96E-05	1,7279E-04	1,2473E-04	1,7454E-04
	9,96E-05	1,7207E-04	1,1858E-04	1,7256E-04
	9,96E-05	1,7279E-04	1,1057E-04	1,7301E-04
	9,96E-05	1,7135E-04	1,1457E-04	1,7436E-04
	9,96E-05	1,7135E-04	1,1483E-04	1,7329E-04
	9,96E-05	1,7279E-04	1,1558E-04	1,7326E-04
	9,96E-05	1,7207E-04	1,1781E-04	1,7281E-04
	9,96E-05	1,7279E-04	1,2091E-04	1,7390E-04
	9,96E-05	1,7279E-04	1,2048E-04	1,7202E-04
	9,96E-05	1,7279E-04	1,1457E-04	1,7362E-04
	9,96E-05	1,7279E-04	1,1460E-04	1,7410E-04
	9,96E-05	1,7135E-04	1,1446E-04	1,7333E-04

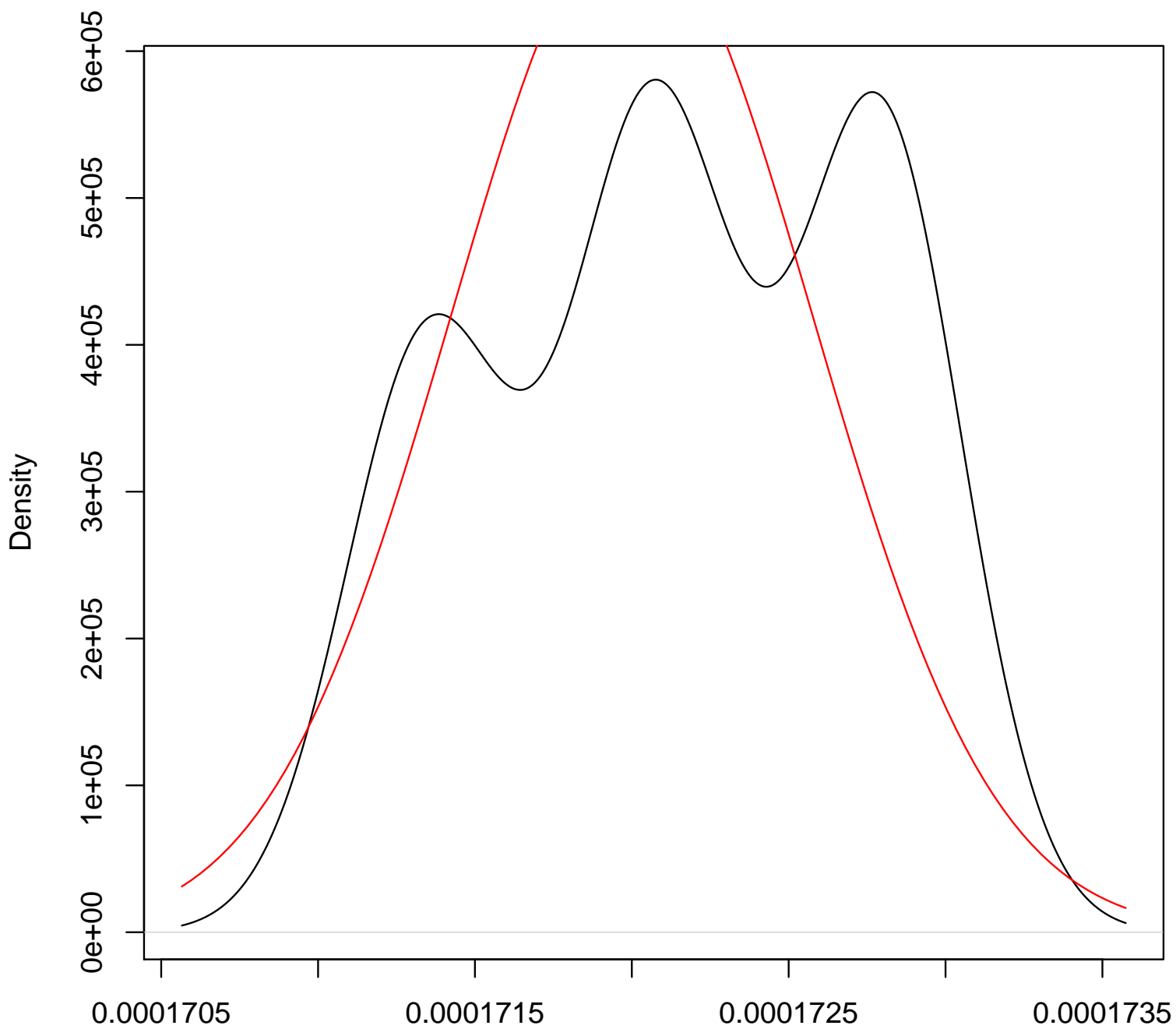
La cible est de pertinence forte.

Nb tirages	30	30	30	30
Espérance	9,96E-05	1,72E-04	1,15E-04	1,74E-04
Ecart type	0	5,75759E-07	4,06743E-06	7,347E-07
Intervalle de confiance : 95% de la moyenne				
t-normale	1,645	1,645	1,645	1,645
Erreur	0	1,7292E-07	1,22159E-06	2,2065E-07

En regardant les espérances on observe que c'est l'anneau est l'arbre qui impactent le plus la pertinence de la cible. On observe aussi que les erreurs sont inférieures à 0,000015 on peut donc faire confiance aux résultats.

5.3 Analyse graphes en anneau

density.default(x = Anneau_aleatoire[, 1])

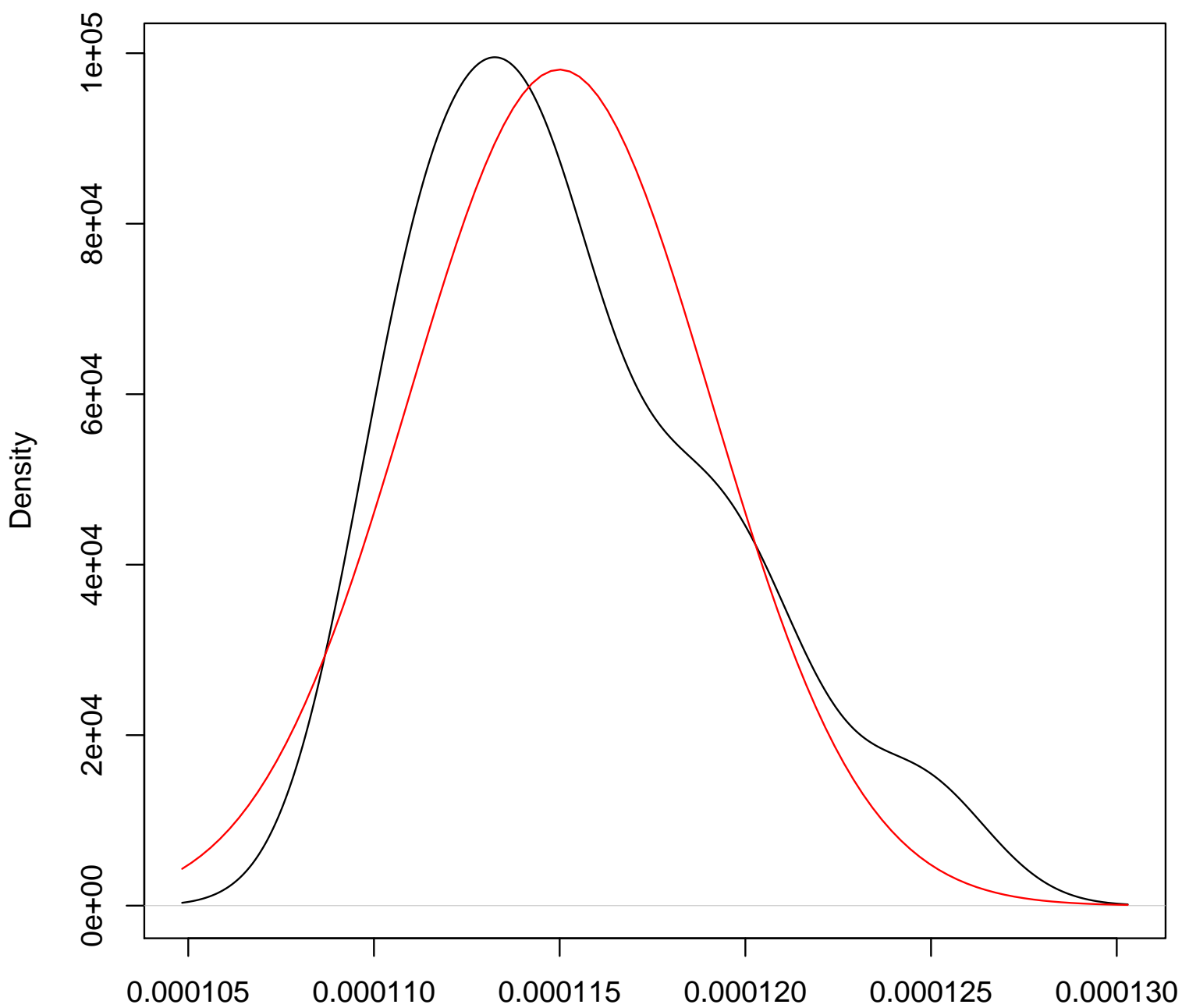


N = 30 Bandwidth = 2.625e-07

La courbe rouge représente la loi normale, la noire est la fonction de densité.

5.4 Analyses Graphes complet

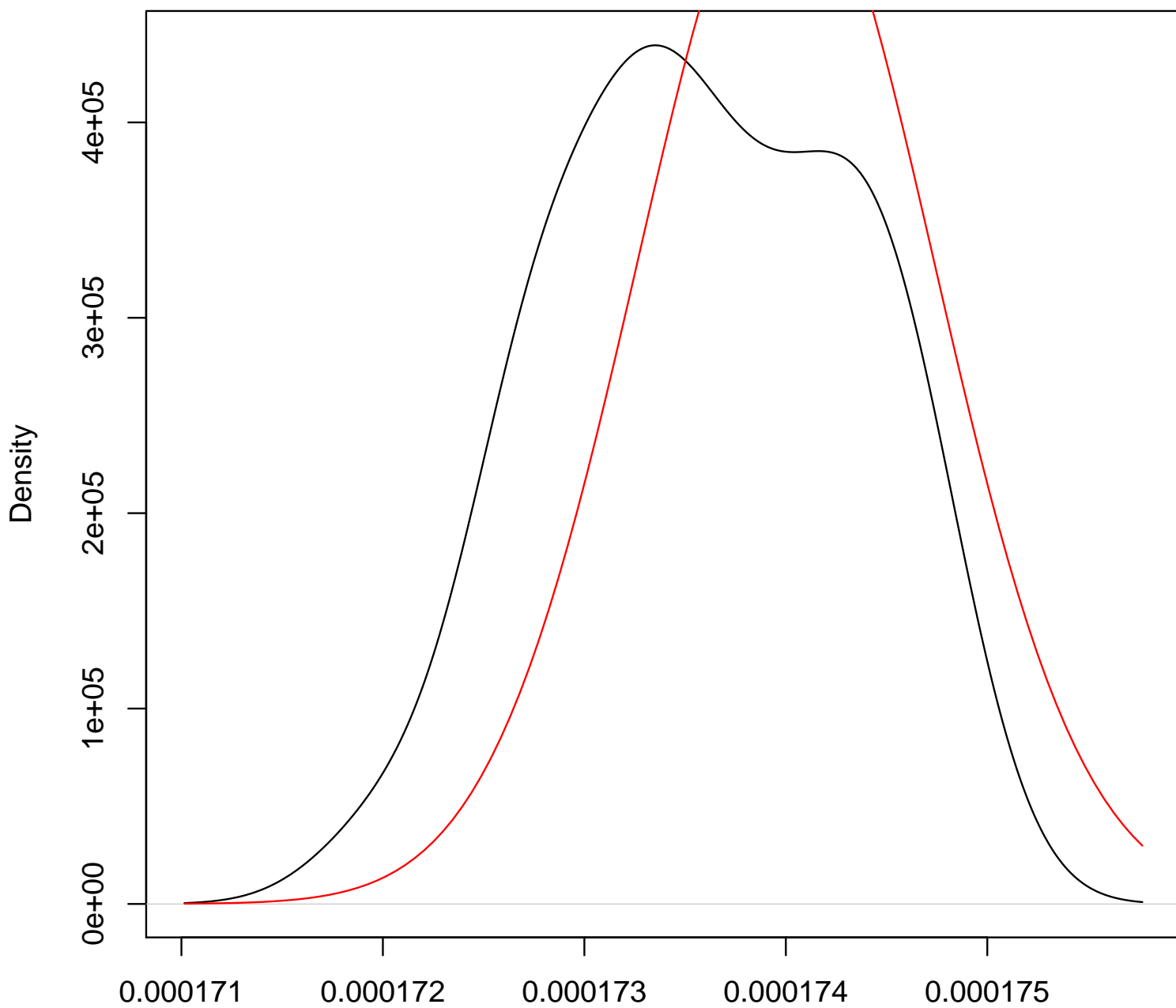
density.default(x = Anneau_aleatoire[, 1])



N = 30 Bandwidth = 1.854e-06

5.5 Analyses Arbres

density.default(x = Anneau_aleatoire[, 1])



N = 30 Bandwidth = 3.349e-07

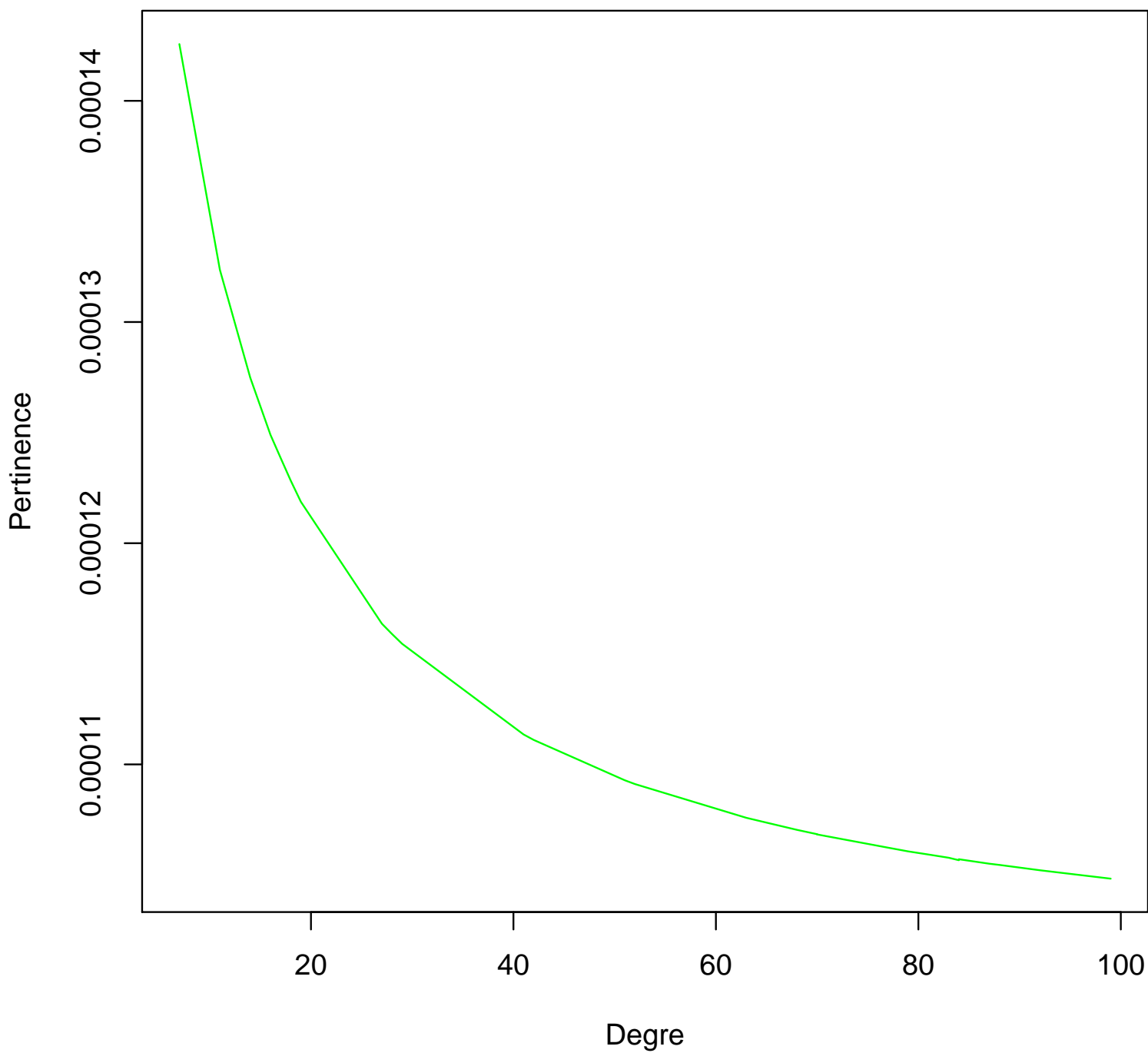
5.6 Conclusion sur l'impact de chaque structure

6 III. Expériences avec un graphe de degré aléatoire

Ici nous allons créer un graphe de 100 sommets de degré aléatoire. C'est à dire que chaque sommet sera relié au même nombre de sommets au sein du graphe.

6.1 Résultats des tests

	Pertinence initiale	Attaque avec un graphe de 100 sommets de degré aléatoire	Degré
	9,96E-05	1,0567E-04	84
	9,96E-05	1,0593E-04	81
	9,96E-05	1,0607E-04	79
	9,96E-05	1,0911E-04	52
	9,96E-05	1,1636E-04	27
	9,96E-05	1,2187E-04	19
	9,96E-05	1,0704E-04	68
	9,96E-05	1,3236E-04	11
	9,96E-05	1,0522E-04	92
	9,96E-05	1,0737E-04	65
	9,96E-05	1,0759E-04	63
	9,96E-05	1,0483E-04	99
	9,96E-05	1,2490E-04	16
	9,96E-05	1,1636E-04	27
	9,96E-05	1,1111E-04	42
	9,96E-05	1,1590E-04	28
	9,96E-05	1,2749E-04	14
	9,96E-05	1,1136E-04	41
	9,96E-05	1,2283E-04	18
	9,96E-05	1,0928E-04	51
	9,96E-05	1,0546E-04	88
	9,96E-05	1,0685E-04	70
	9,96E-05	1,1545E-04	29
	9,96E-05	1,0624E-04	77
	9,96E-05	1,0578E-04	83
	9,96E-05	1,0571E-04	84
	9,96E-05	1,0551E-04	87
	9,96E-05	1,4257E-04	7
	9,96E-05	1,2386E-04	17
	9,96E-05	1,0683E-04	70
Nb tirages	30	30	30
Espérance	9,96E-05	1,13E-04	
Ecart type	0	9,61535E-06	
Intervalle de confiance : 95% de la moyenne			
t-normale	1,645	1,645	
Erreur	0	2,88782E-06	



6.2 Analyses

On observe que plus le degré est fort, moins la pertinence de la cible est modifié. Ici l'espérance de la pertinence de la cible apres attaque est de $1.13e-04$, ce qui est meilleur que la pertience de la cible lors des tests initiaux avec un graphe complet a 100 sommets ($1.05e-04$). Ainsi cette structure est plus efficace que le graphe complet. De plus cette structure a l'avantage d'etre moins detectable que l'anneau ou le graphe complet car le degré varie. En effet détecter un anneau ou un graphe complet est beaucoup plus aisé que de détecter un graphe de degré X.

7 IV. Expériences sur l'efficacité

7.1 Résultats des tests

Type de graphe attaquant	Pertinence de la cible	Nombre de sommets	Pertinence après attaque	Différence avec la pertinence initiale	% de modification	Efficacité
Anneau	9,96E-05	100	1,71E-04	7,17E-05	72,0%	7,17E-07
		250	2,79E-04	1,79E-04	179,9%	7,17E-07
		500	4,58E-04	3,58E-04	359,5%	7,16E-07
		1000	8,14E-04	7,15E-04	717,5%	7,15E-07
	7,54E-06	100	5,22E-05	4,46E-05	591,8%	4,46E-07
		250	1,19E-04	1,12E-04	1479,3%	4,46E-07
		500	2,31E-04	2,23E-04	2957,4%	4,46E-07
		1000	4,53E-04	4,46E-04	5910,4%	4,46E-07
	6,05E-07	100	4,52E-05	4,46E-05	7373,0%	4,46E-07
		250	1,12E-04	1,12E-04	18428,2%	4,46E-07
		500	2,24E-04	2,23E-04	36842,8%	4,46E-07
		1000	4,46E-04	4,46E-04	73630,6%	4,46E-07
Complet	9,96E-05	100	1,05E-04	5,17E-06	5,2%	5,17E-08
		250	1,05E-04	5,28E-06	5,3%	2,11E-08
		500	1,05E-04	5,23E-06	5,3%	1,05E-08
		1000	1,05E-04	5,05E-06	5,1%	5,05E-09
	7,54E-06	100	1,08E-05	3,23E-06	42,9%	3,23E-08
		250	1,09E-05	3,34E-06	44,2%	1,33E-08
		500	1,09E-05	3,36E-06	44,6%	6,72E-09
		1000	1,09E-05	3,36E-06	44,5%	3,36E-09
	6,05E-07	100	3,84E-06	3,24E-06	534,8%	3,24E-08
		250	3,95E-06	3,34E-06	552,2%	1,34E-08
		500	3,98E-06	3,38E-06	557,7%	6,75E-09
		1000	3,99E-06	3,39E-06	559,5%	3,39E-09
Arbre	9,96E-05	100	1,72E-04	7,22E-05	72,5%	7,22E-07
		250	2,79E-04	1,80E-04	180,4%	7,19E-07
		500	4,58E-04	3,59E-04	360,0%	7,17E-07
		1000	8,15E-04	7,15E-04	718,0%	7,15E-07
	7,54E-06	100	5,25E-05	4,49E-05	595,7%	4,49E-07
		250	1,19E-04	1,12E-04	1483,6%	4,47E-07
		500	2,31E-04	2,23E-04	2962,0%	4,47E-07
		1000	4,54E-04	4,46E-04	5915,2%	4,46E-07
	6,05E-07	100	4,55E-05	4,49E-05	7421,3%	4,49E-07
		250	1,12E-04	1,12E-04	18481,8%	4,47E-07
		500	2,24E-04	2,23E-04	36899,3%	4,47E-07
		1000	4,47E-04	4,46E-04	73689,8%	4,46E-07
Arbre à attaquant unique	9,96E-05	100	1,38E-04	3,88E-05	38,9%	3,88E-05
		250	1,79E-04	7,96E-05	79,9%	7,96E-05
		500	2,35E-04	1,36E-04	136,3%	1,36E-04
		1000	3,31E-04	2,31E-04	231,8%	2,31E-04
	7,54E-06	100	3,17E-05	2,41E-05	319,9%	2,41E-05
		250	5,71E-05	4,95E-05	657,2%	4,95E-05
		500	9,21E-05	8,45E-05	1121,2%	8,45E-05
		1000	1,51E-04	1,44E-04	1908,1%	1,44E-04
	6,05E-07	100	2,47E-05	2,41E-05	3985,3%	2,41E-05
		250	5,02E-05	4,96E-05	8187,8%	4,96E-05
		500	8,52E-05	8,45E-05	13969,5%	8,45E-05
		1000	1,44E-04	1,44E-04	23772,2%	1,44E-04

7.2 Analyses

Ici nous avons testé chaque structure sur trois cibles différentes, tout en faisant varier le nombre de sommets par structure.

8 Conclusions