



# PROJET - LIEUX DE FORMATION

Compte rendu

Baptiste Chemier & Aisi Nguyen  
POLYTECH 4ème Année APPRENTISSAGE

## Table des matières

Modélisation du problème.....	2
Méthode de résolution.....	3
Nos choix .....	3
L'algorithme .....	3
Utilisation de l'application.....	4
Résultats.....	4
100 agences.....	4
Vérification de la limite .....	5
300 agences.....	6
500 agences.....	7
Temps d'exécution .....	9
Conclusion .....	10

## Modélisation du problème

Une société organise régulièrement des sessions de formation pour les employés de ses agences situées en France métropolitaine. Pour cela, elle choisit, parmi les 1947 villes françaises de plus de 5 000 habitants un certain nombre de lieux de formations et affecte les personnes des agences à ces lieux de formations en tenant compte de la distance séparant le lieu de formation et l'agence. Chaque agence peut envoyer entre 1 et 10 personnes en formation, formation qui possède un aspect interactif, car l'interactivité améliore grandement la compréhension des formations. Cependant, cet aspect impose un maximum de 60 personnes par session de formation, donc par lieu de formation.

De plus, il existe un coût associé à chaque lieu de formation qui comprend le coût des formateurs de 2 000€ et un coût de location de la salle de 1 000€  
Cette société souhaite donc automatiser cette tâche, car celle-ci lui prend beaucoup de temps.

Lors de la modélisation de ce problème, nous devons respecter certaines contraintes :

- La capacité maximale d'un lieu de formation est de 60 personnes.
- L'ouverture d'un lieu de formation entraîne un coût de 3 000€ (2 000 € de formateur + 1000€ de restauration).
- De plus, chaque kilomètre parcouru par les membres des agences sera indemnisé 4 centimes.

Ainsi la fonction à minimiser est la suivante

$$\sum_A \sum_L (0,4 * NbPersonne * distance * 2) + 3000$$

Avec

A : le nombre d'agence

L : le nombre de lieux de formation.

Nous tenons également compte de l'aller-retour d'où le \*2 dans la formule ci-dessus.

Nous devons donc minimiser à la fois le nombre de kilomètres parcourus par les membres des agences, mais également le nombre d'agences ouvertes

## Méthode de résolution

### Nos choix

Durant nos cours, nous avons vu plusieurs méthodes permettant de résoudre ce type de problème. Nous avons donc le choix entre la méthode taboue, la méthode du recuit simulé ainsi que les algos génétiques, qui sont de bonnes méta-heuristiques. Après avoir analysé et posé le problème, nous avons choisi de nous orienter sur la méthode taboue, car il s'agissait d'une méthode donnant de bons résultats et qui nous paraissait la plus évidente à mettre en place dans cette situation.

Nous avons choisi d'utiliser la méthode Tabou afin d'optimiser ce problème de recherche de lieux de formation. Cependant, nous avons dû faire des choix pour implémenter un tel algorithme. En effet, la méthode Tabou impose de calculer le voisinage d'une solution à chaque itération. Une solution étant un ensemble de lieux de formation ayant un certain nombre d'agences rattachées à lui, le calcul du voisinage d'une solution serait énorme, car il faudrait, faire un changement du lieu de formation pour chaque agence. Ainsi, pour faire plus simple, nous avons choisi, pour chaque agence de la solution d'avoir 80% de chance pour que, dans la solution voisine, l'agence se retrouve dans une agence déjà ouverte, car, l'ouverture d'une nouvelle agence coûte 3 000€. Nous calculons donc un voisinage de taille équivalente au nombre d'agences importé.

Nous avons également souhaité limiter la taille de la liste taboue à 1 car nous avons vu en cours qu'une telle taille permet à l'algorithme de fonctionner, mais également, vu la taille du voisinage calculé, il est peu probable qu'une solution puisse retourner dans l'état précédent.

### L'algorithme

Notre algorithme suit les étapes suivantes.

Tous d'abords, nous initialisons nos agences dans les lieux de formation de manière totalement au hasard. Nous n'avons alors pas tout à fait un lieu de formation pour une agence, car en moyenne, sur les différents tests que nous avons effectués, un petit nombre de lieux de formation possède déjà plusieurs agences.

Une fois cette solution initiale générée, nous pouvons lancer l'algorithme à proprement parler. Nous itérons le calcul du voisinage autant de fois que l'utilisateur le souhaite, car il peut influencer sur ce paramètre.

Lorsque nous calculons le voisinage, nous bouclons sur l'ensemble des agences et nous essayons de les réaffecter à d'autres lieux de formations avec, 80 % de chance que ce soit un lieu de formation déjà ouvert. Si cette nouvelle affectation possède un coût moindre, nous acceptons cette solution.

## Utilisation de l'application

Pour utiliser l'application, vous trouverez dans ce dossier ZIP, un .jar qu'il vous suffit d'exécuter. Une fois l'application lancée, il vous suffit d'importer votre fichier d'agence et votre fichier de ville. Une fois cela fait, vous pouvez choisir le nombre d'itération de l'algorithme, par défaut, l'algorithme tournera 1000 fois. Enfin, vous avez plus qu'à lancer l'algorithme.

Une fois ce dernier terminé, la liste des lieux de formations s'affiche à gauche. Vous pouvez également afficher, grâce au radio bouton, la liste des agences. Lorsque vous cliquez sur un lieu de formation ou sur une agence, des informations sur ceux-ci s'affichent à droite.

## Résultats

Nous avons testé notre algorithme sur 3 jeux de données différents :

- Un jeu contenant 100 agences
- Un jeu contenant 300 agences
- Un jeu contenant 500 agences

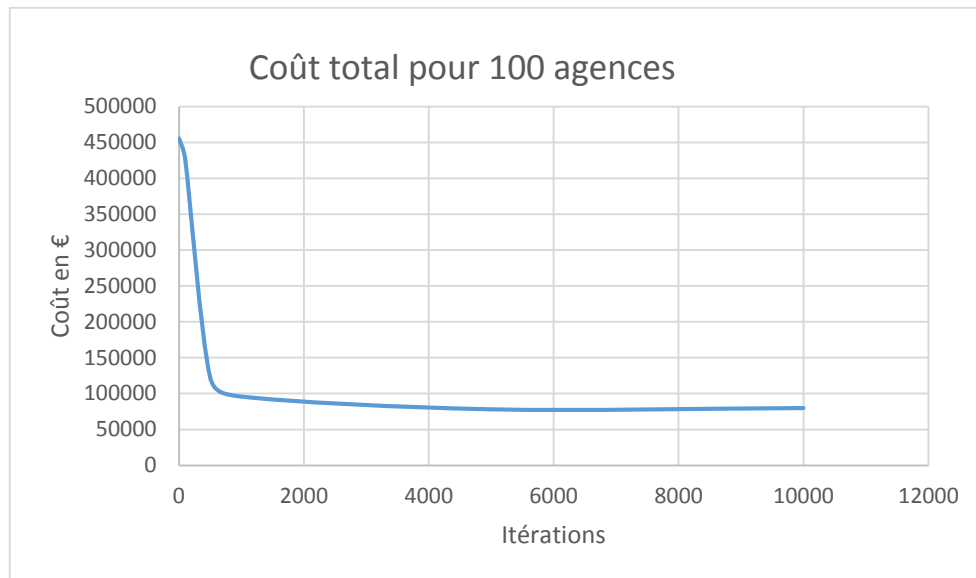
Tous les résultats présentés dans les tableaux suivants sont des moyennes de 3 exécutions à chaque fois, car comme il s'agit d'optimisation aléatoire, nous ne retrouvons jamais deux fois le même résultat, nous avons souhaité avoir une moyenne afin d'être au plus proche de la réalité. Ceci explique donc pourquoi, le nombre de lieux de formation ne soit pas toujours entier.

Afin d'avoir de meilleures courbes permettant de mieux voir comment varie le coût, nous avons décidé de représenter celles-ci sur [0-10000] itérations, car bien souvent, nous pouvons percevoir la limite de celles-ci sur cet intervalle et c'est sur celui-ci qu'il se passe le plus de choses.

### 100 agences

Nombre d'agence	100								
Itération	1	100	500	1000	5000	10000	20000	60000	100000
Nombre de lieux de formation	97	88,33	25,67	18	13,67	12,33	12,33	12,33	12
Coût total (€)	455472	427393	122933,33	95889	78066,33	79828,67	75840,33	74655	72708,33
Temps d'exécution (en milliseconde)	14,67	18,33	1361,67	358,33	9526,33	20270	44113	126349,67	158565,33

Tableau 1 - Résultats pour 100 agences



Lorsque nous testons avec 100 agences, le coût total diminue rapidement pour se stabiliser autour de 72 000€. Nous pouvons voir qu'entre 20 000 et 100 000 itérations, le coût total décroît de 3 000€ donc nous pouvons dire que le coût optimal pour 100 agences est d'environ 70 000€ avec 12 agences.

Au cours des différentes exécutions que nous avons pu faire, le meilleur résultat que nous avons obtenu est 71 986€ pour 12 agences. Ce résultat a été obtenu avec 100 000 itérations et en 165 756 millisecondes (soit 2 min 45s).

### Vérification de la limite

Afin de vérifier si la limite se situe bien aux environs de 70 000 € avec 12 agences, nous avons décidé, avec ce jeu de données, d'augmenter le nombre d'itération afin de voir si le coût continue de diminuer ou s'il converge vers ces valeurs.

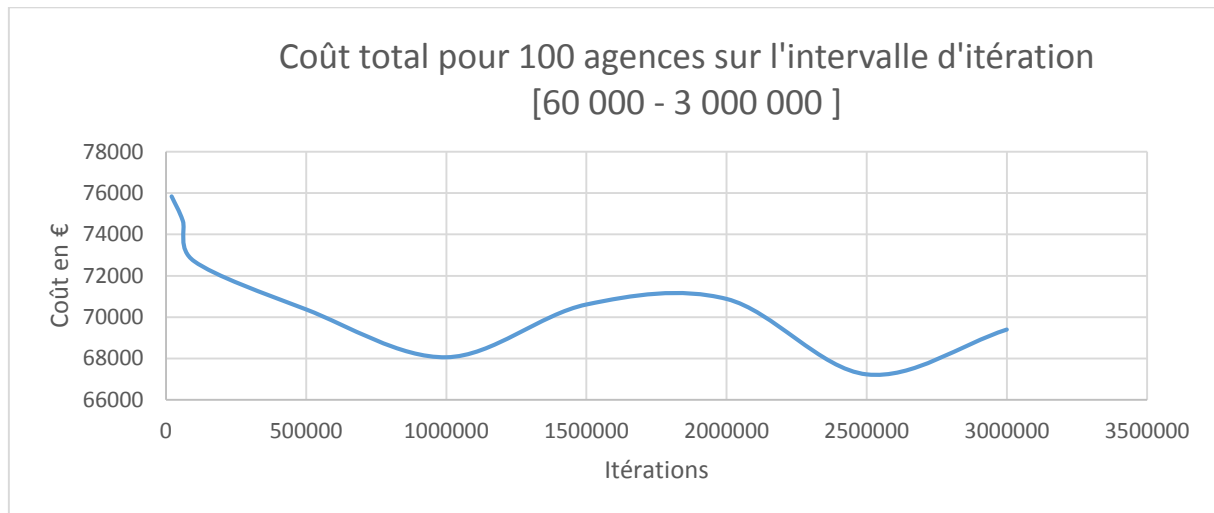
Nous avons décidé d'utiliser ce jeu de données, car c'est celui ayant d'une part le plus petit coût initial et d'autre part celui dont les temps de calculs sont les plus courts, car nous avons fait tourner l'algorithme sur plus d'un million d'itérations.

Pour cela, nous avons lancé notre algorithme avec respectivement 500 000 ; 1 ; 1,5 ; 2 ; 2,5 et 3 million d'itérations. Nous avons alors obtenu les résultats suivants.

Itérations	500000	1000000	1500000	2000000	2500000	3000000
Nombre de lieux de formation	12	12	13	12	12	12
Coût total (€)	70372	68059	70611	70881	67231	69398
Temps d'exécution (en milliseconde)	861306	1718830	2610861	3965834	5948751	8923126

Nous nous rendons alors bien compte que d'augmenter le nombre d'itération ne sert à rien, car à partir d'un million d'itérations, l'algorithme converge et nous n'avons alors plus d'écart significatif entre les différents coûts. Nous voyons même que le nombre de lieux de formations ne change pas, et ce, depuis 100 000 itérations, ainsi les variations de coût proviennent uniquement de l'affectation de certaines agences à des lieux de formations plus ou moins proches.

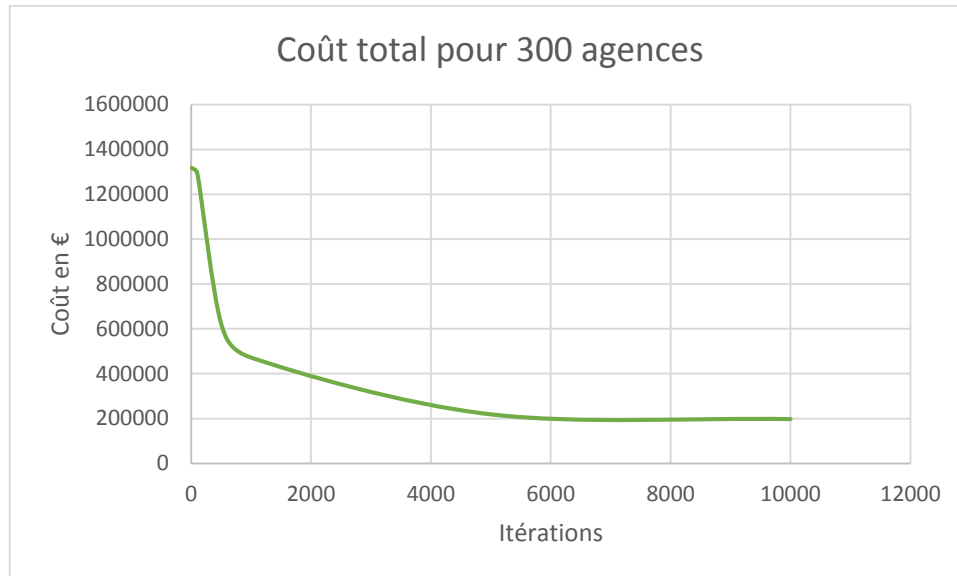
Nous voyons cela sur la courbe suivante où sur l'intervalle d'itération assez grand, le coût ne varie quasiment pas.



### 300 agences

Nombre d'agence	300								
Itération	1	100	500	1000	5000	10000	20000	60000	100000
Nombre de lieux de formation	275,33	271,33	131	100,33	46,33	41,33	39	40	35,67
Coût total	1317390,67	1293983,33	621381	471643	218823	197676	194436	194845	184462
Temps d'exécution (ms)	50,67	57,67	2627	4664	21607	40726	84162,6	262736	468880

Tableau 2 - Résultats pour 300 agences



Pour 300 agences, nous avons également le même type de courbe que pour 100 agences, le prix décroît assez rapidement, pour passer de 1,3 million € à 400 000 €, puis finit par se stabiliser un peu en dessous de 200 000 € pour environ 35 agences d'ouvertes.

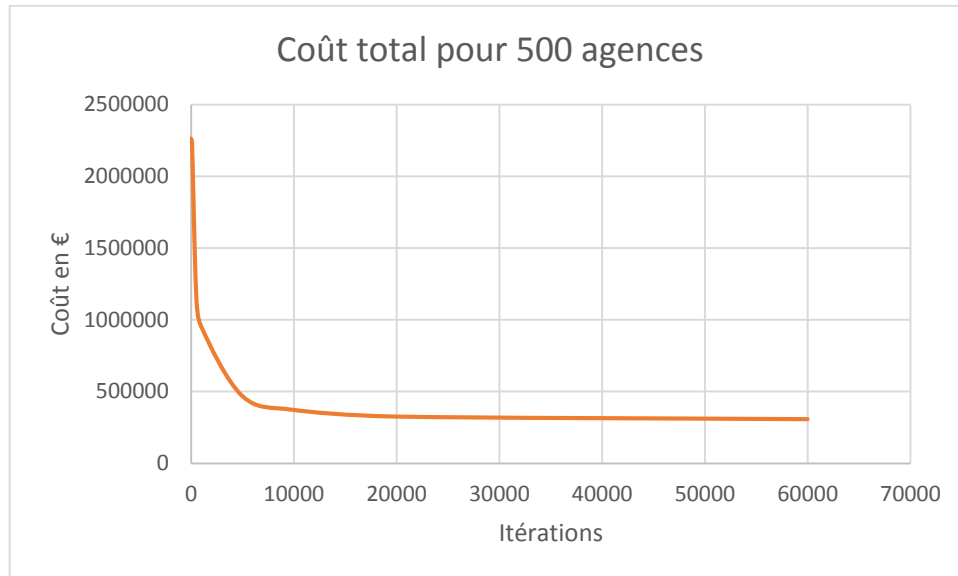
Lors de nos tests, le meilleur coût que nous avons obtenu a été de 181 751 € pour 32 agences. Ce coût a été atteint en 100 000 itérations en 500 800 millisecondes (soit 8 min 20 s). On peut donc supposer que cela ne va pas varier si l'on augmente le nombre d'itération.

### 500 agences

Nombre d'agence	500								
Itération	1	100	500	1000	5000	10000	20000	60000	100000
Nombre de lieux de formation	443	434	256,67	200,67	104	81,67	69,33	66,33	64
Coût total	2261420,67	2236949,67	1188746,33	951660	467980,33	371360,33	325163	307085	296361,33
Temps d'exécution (ms)	80	99	5388,33	7722,33	42917,33	85363,67	159839,33	446538,33	722481,33

Tableau 3 - Résultats pour 500 agences





Comme pour les autres tests, lorsque nous avons 500 agences, nous remarquons une courbe qui décroît en fonction du nombre d'itérations. Cependant, nous pouvons voir que la courbe continue à décroître plus légèrement après l'intervalle [0-10000]. Contrairement aux autres courbes, nous avons pris, cette fois un plus grand intervalle, car la courbe continuait à décroître pour converger vers 300 000 € lorsque nous faisons 60 000 itérations. Le nombre de lieux de formation converge également vers 65 lorsque nous arrivons à 60 000 itérations.

Notre meilleur résultat pour 500 agences a été 291 369 € pour 61 agences. Il a été atteint en 100 000 itérations en 735 045 millisecondes (soit 12 min 15s)

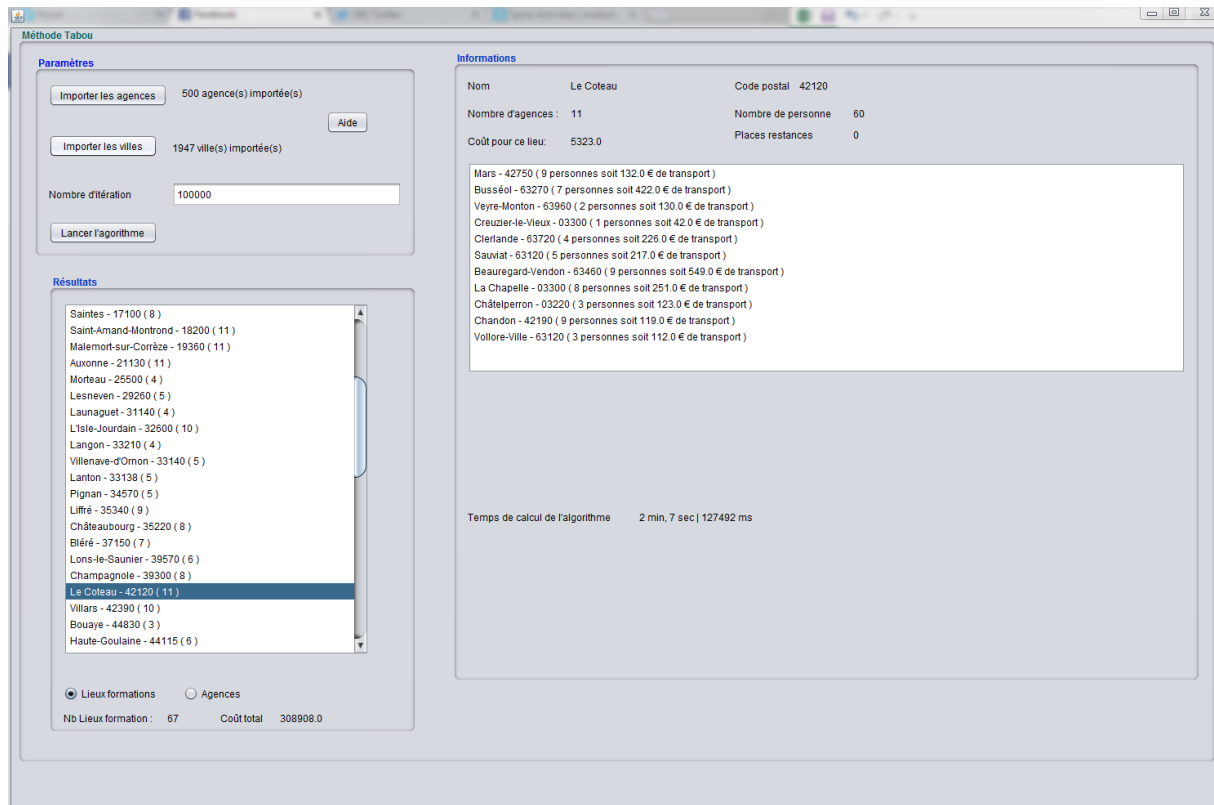
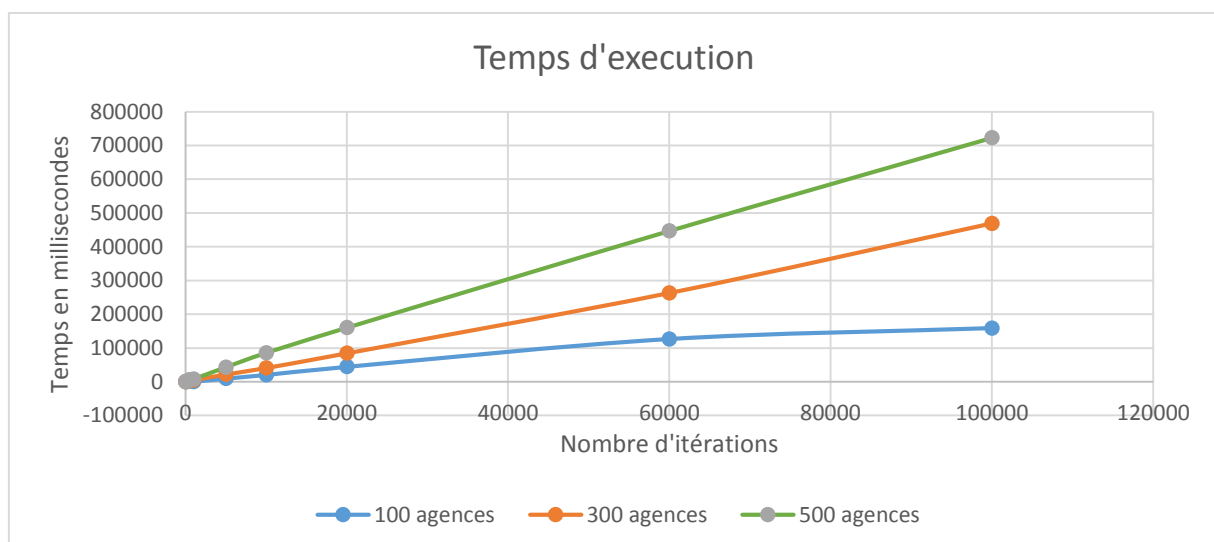


Figure 1 - Exemple de résultats pour 500 agences

## Temps d'exécution



D'après ces courbes affichant les temps d'exécution de l'algorithme en fonction du nombre d'itération, nous avons des temps plutôt linéaires et non-exponentiels, ce

qui est une bonne chose. En effet, si nous prenons un autre jeu de données avec un grand nombre d'itérations, le temps sera certes important, mais l'algorithme ne bouclera pas pendant plusieurs jours, cela devrait rester de l'ordre de quelques heures (tout dépend du nombre d'itérations et du nombre d'agences bien entendu).

De plus, sur le graphe, nous pouvons voir que le temps d'exécution est proportionnel au nombre d'agences testé.

### Conclusion

Grâce à ces différents jeux de données, nous avons pu nous rendre compte que notre algorithme se comporte de la même manière et que nous arrivons à trouver une limite qui nous donne le coût optimal permettant de minimiser à la fois le nombre de kilomètres parcourus par les membres des agences, mais également le nombre de lieux de formation ouvert.

Les temps d'exécution de notre algorithme sont assez bons puisque pour 100 000 itérations, ils ne dépassent pas le quart d'heure, et ce, pour tous nos jeux de données.

Si nous devons refaire un tel projet, nous ajouterions certainement de nouveaux paramètres afin de permettre à l'utilisateur de faire varier ces résultats. En effet, il pourrait être intéressant par exemple de faire varier, lorsque nous calculons le voisinage, la probabilité d'ajouter l'agence dans un lieu de formation déjà ouvert. Lors du développement, nous avons inversé un signe et nous avons ainsi 80 % de chance que l'agence se retrouve dans un nouveau lieu de formation, nous avons alors des résultats cohérents mais légèrement plus hauts qu'en faisant l'inverse.

De plus, nous pourrions également laisser à l'utilisateur la possibilité de faire varier la taille de la liste taboue afin de voir si notre hypothèse, qui était que vu le voisinage que nous avons, une taille de 1 était suffisante.