

METAHEURISTIQUE

RECUIT SIMULE



NOM	Prénom
HMAM	Aymen

Master SDTDS

Enseignant : M. SIARRY

Année universitaire 2020-2021

1. Objectif de TP

L'objectif de ce TP est de créer, en langage de programmation PYTHON, une application de l'algorithme du recuit simulé pour résoudre le problème du placement des composants électroniques.

Donc il s'agit de :

- de placer 25 composants électroniques (numérotés de 1 à 25) (configuration optimale) (Figure1)

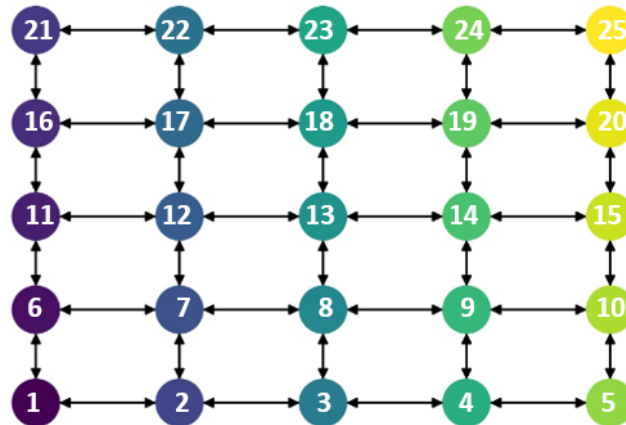


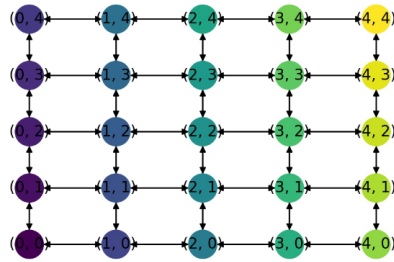
Figure 1: Configuration optimale

- Les composants sont reliés entre eux d'une façon prédéterminée (Table ci-dessous)

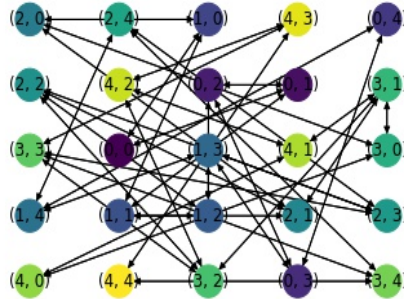
Composant	Est relié à	Composant	Est relié à
1	2,6	13	8,12,14,18
2	1,3,7	14	9,13,15,19
3	2,4,8	15	10,14,20
4	3,5,9	16	11,17,21
5	4,10	17	12,16,18,22
6	1,7,11	18	13,17,19,23
7	2,6,8,12	19	14,18,20,24
8	3,7,9,13	20	15,19,25
9	4,8,10,14	21	16,22
10	5,9,15	22	17,21,23
11	6,12,16	23	18,22,24
12	7,11,13,17	24	19, 23, 35
		25	20,24

- Le composant numéro i peut être placé sur n'importe quel nœud.
- La distance entre un composant et son voisin comme montre le tableau précédant est 5.
- La distance globale pour une configuration optimale est 200
- Exemple de calcul de la distance globale d'une configuration :

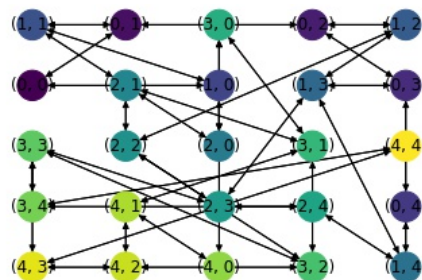
Exemple 1 : distance globale =200



Exemple 2 : distance globale =790



Exemple 3 : distance globale =490



2. Problématique

A partir d'une configuration initiale au hasard, on applique l'algorithme du recuit simulé pour trouver la configuration optimale. La seule "opération" autorisée pour trouver la solution optimale est une seule permutation de deux composants. La solution optimale est connue à l'avance (il s'agit uniquement de démontrer l'efficacité du recuit simulé).

3. Définition de l'algorithme du recuit simulé

Le recuit simulé est une méthode de programmation empirique (métaheuristique) inspirée d'un processus utilisé en métallurgie. On alterne dans cette dernière des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau. Cette méthode est transposée en optimisation pour trouver les extrema d'une fonction.

4. Pseudo code de l'algorithme du recuit simulé

```
1: On donne une configuration initiale quelconque
2: On donne une température initiale T
3: while (Le système n'est pas figé) do

    On fait une modification élémentaire du système qui nous
    donne une variation d'énergie  $\Delta E$ 

    if  $\Delta E \leq 0$  then Modification acceptée
    else Modification acceptée avec la probabilité  $\exp -\Delta E/T$ 
    end if

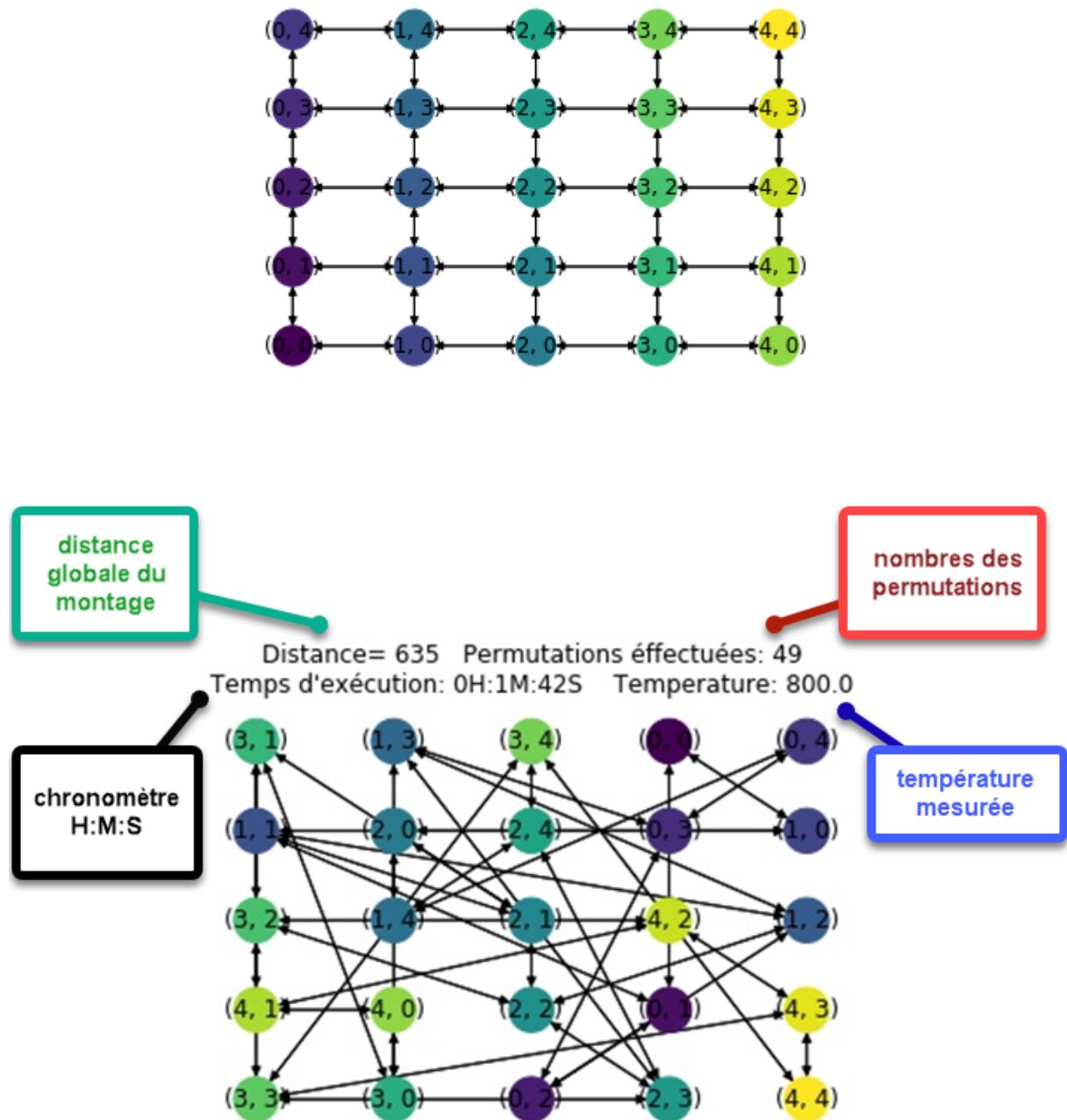
    if Equilibre thermodynamique then Diminution lente de T
    end if

4: end while
```

5. Notation (pour le code en python)

On désigne par :

Composant[1] = (0,0)	Composant[14] = (2,3)
Composant[2] = (0,1)	Composant[15] = (2,4)
Composant[3] = (0,2)	Composant[16] = (3,0)
Composant[4] = (0,3)	Composant[17] = (3,1)
Composant[5] = (0,4)	Composant[18] = (3,2)
Composant[6] = (1,0)	Composant[19] = (3,3)
Composant[7] = (1,1)	Composant[20] = (3,4)
Composant[8] = (1,2)	Composant[21] = (4,0)
Composant[9] = (1,3)	Composant[22] = (4,1)
Composant[10] = (1,4)	Composant[23] = (4,2)
Composant[11] = (2,0)	Composant[24] = (4,3)
Composant[12] = (2,1)	Composant[25] = (4,4)
Composant[13] = (2,2)	



6. Application

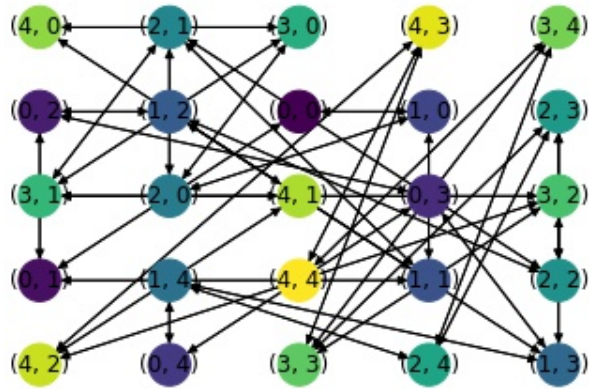
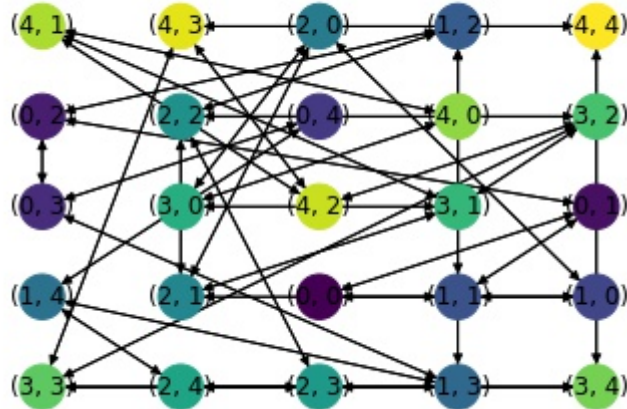
Le code source pour notre application de l'algorithme du recuit simulé pour résoudre le problème du placement des composants électroniques contient les fonctions suivantes :

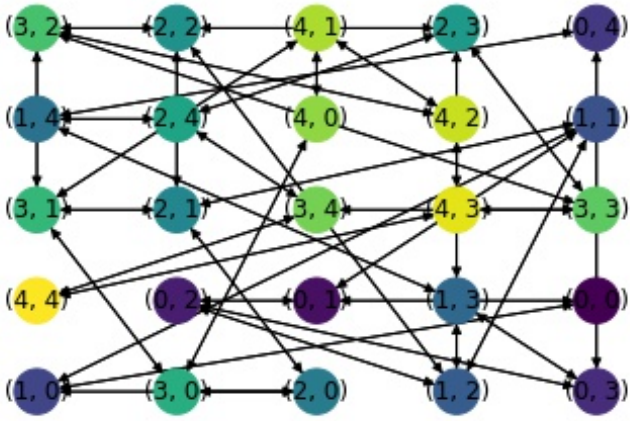
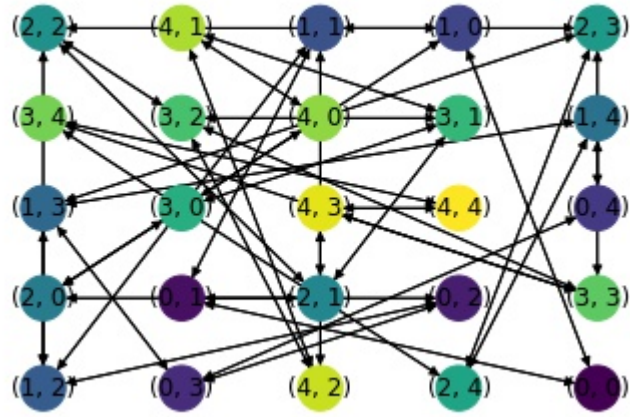
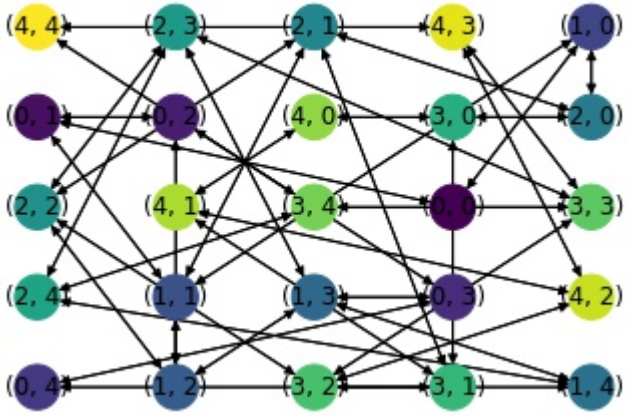
(code complet en annexe)

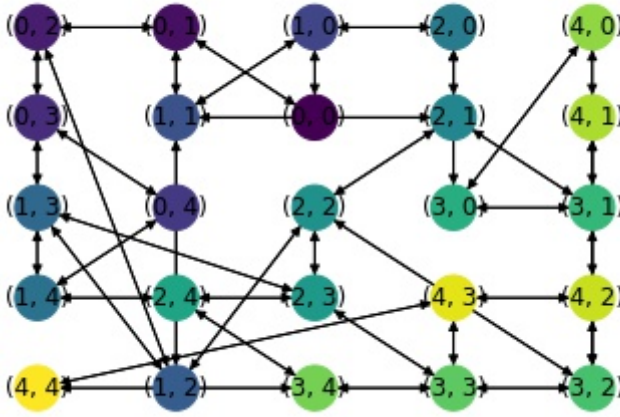
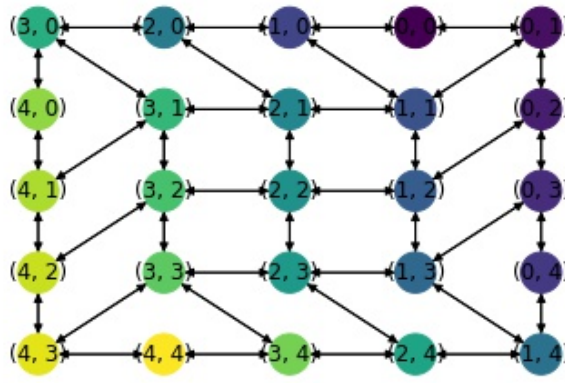
Nom de la fonction	Rôle de la fonction
Configuration_depart():	Assurer la génération d'une configuration de départ dont la distance globale est dans l'intervalle [700,800]
Dist_totale(configuration)	Calculer la distance globale entre les 25 composants d'une configuration en entrée tout en respectant les voisinages de chaque composants
recuit_simule(Configuration_depart)	Appliquer l'algorithme de recuit simulé

Essai 1 :

Température initiale	1000
Taux de refroidissement	0.9
Equilibre thermodynamique	<p>Il est atteint :</p> <ul style="list-style-type: none"> - $12*N$: mouvements acceptés - $100*N$ mouvements tentés <p>Avec $N=25$ nombres des composants</p>
Système figé	3 paliers de température successifs sans aucun mouvements

<p>Solution de départ (d=730)</p>	<p>Montage Départ</p> 
<p>Après une minute (d=640)</p>	<p>Distance= 640 Permutations effectuées: 23 Temps d'exécution: 0H:0M:59S Temperature: 1000</p> 

<p>Après une heure (distance=590)</p>	<p>Distance= 590 Permutations effectuées: 1442 Temps d'exécution: 1H:0M:05 Temperature: 656.1</p> 
<p>Après deux heures (distance=710)</p>	<p>Distance= 710 Permutations effectuées: 2584 Temps d'exécution: 2H:0M:15 Temperature: 430.46721</p> 
<p>Après 9 heures (distance=630)</p>	<p>Distance= 630 Permutations effectuées: 12786 Temps d'exécution: 9H:0M:35 Temperature: 20.27556</p> 

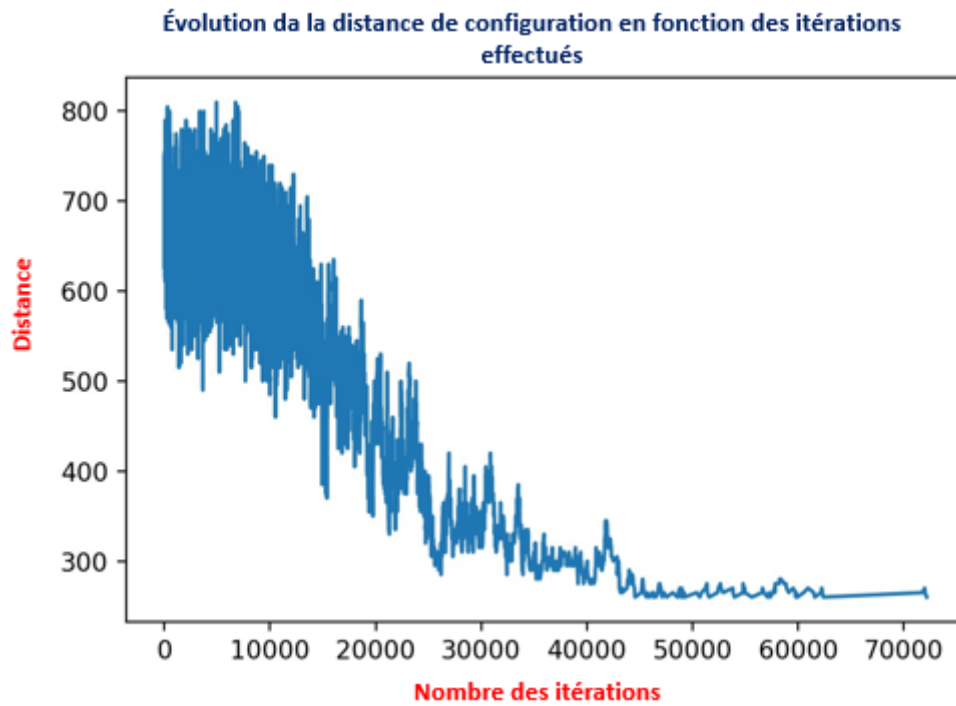
<p>Après 11 heures (distance=365)</p>	<p>Distance= 365 Permutations effectuées: 19563 Temps d'exécution: 11H:0M:2S Temperature: 8.72796</p> 
<p>Solution finale optimisé Après 12H :24M :42S 72164 permutations (distance=260)</p>	<p>Distance= 260 Permutations effectuées: 72164 Temps d'exécution: 12H:34M:42S Temperature: 0.8595</p> 

Interprétation :

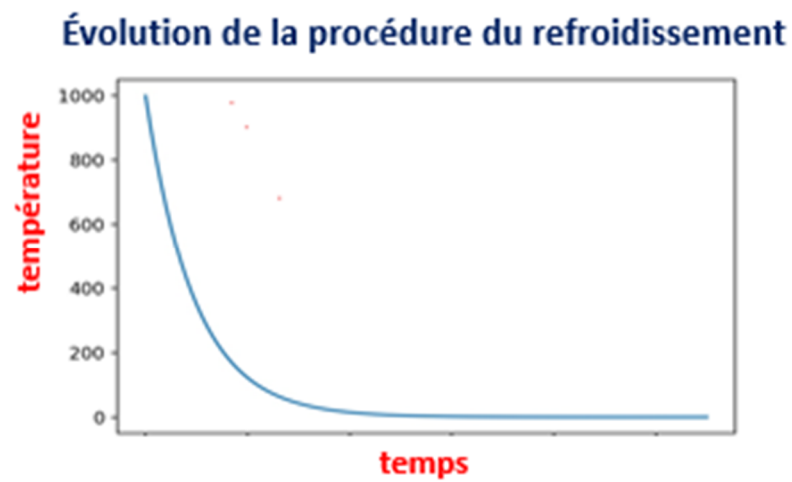
Après 72164 itérations effectués (12 heures,24 minutes, 42 secondes), on obtient une solution proche de la solution optimale avec une distance totale = 260. On est donc piégé dans un minimum local.

On constate que le recuit simulé est très long. C'est grâce au choix de la température initiale (1000) très forte avec un taux de refroidissement très faible ($\alpha=0,9$).

Evolution de la distance en fonction des itérations effectués



Refroidissement en fonction de temps

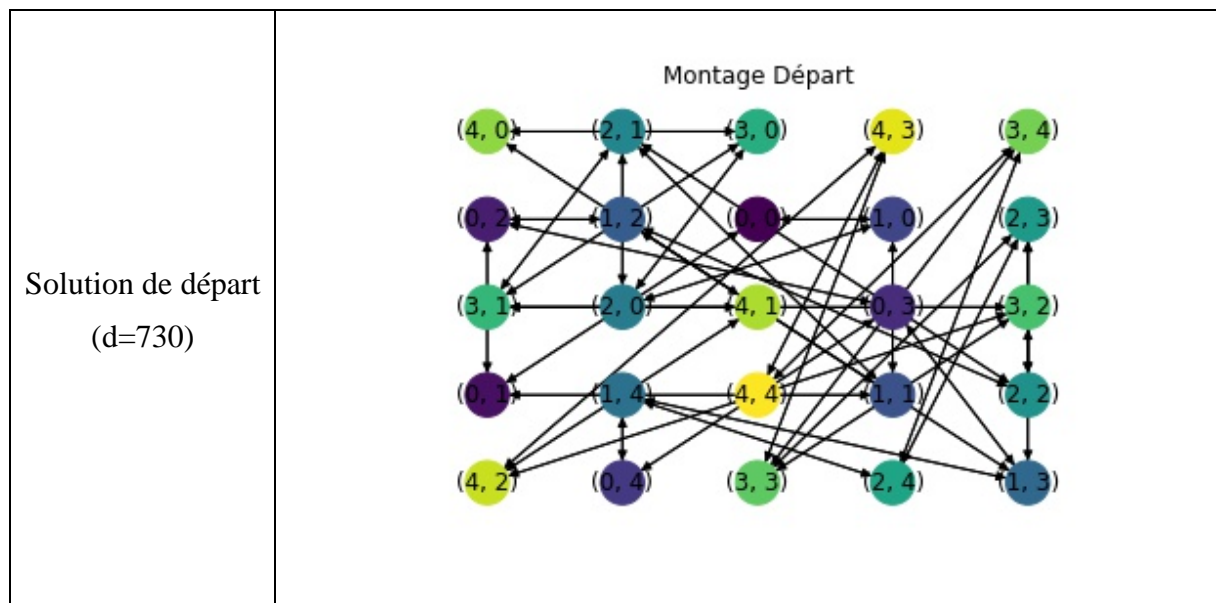


Essai 2:

Dans l'essai n° 2, on a essayé de :

- diminuer la température initiale (500 au lieu de 1000)
- garder le taux de refroidissements égale 0.9
- modifier la condition de l'équilibre thermodynamique
 - mouvements acceptés 50
 - mouvements tentés 1250

Température initiale	500
Taux de refroidissement	0.9
Equilibre thermodynamique	<p>Il est atteint :</p> <ul style="list-style-type: none"> - 2*N : mouvements acceptés - 50*N mouvements tentés <p>Avec N=25 nombres des composants</p>
Système figé	3 paliers de température successifs sans aucun mouvements



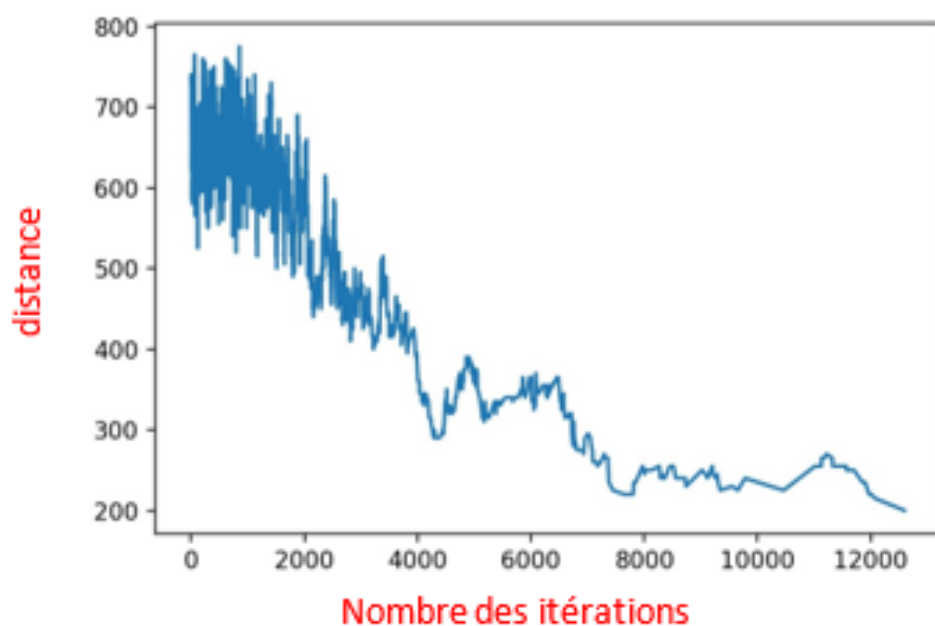
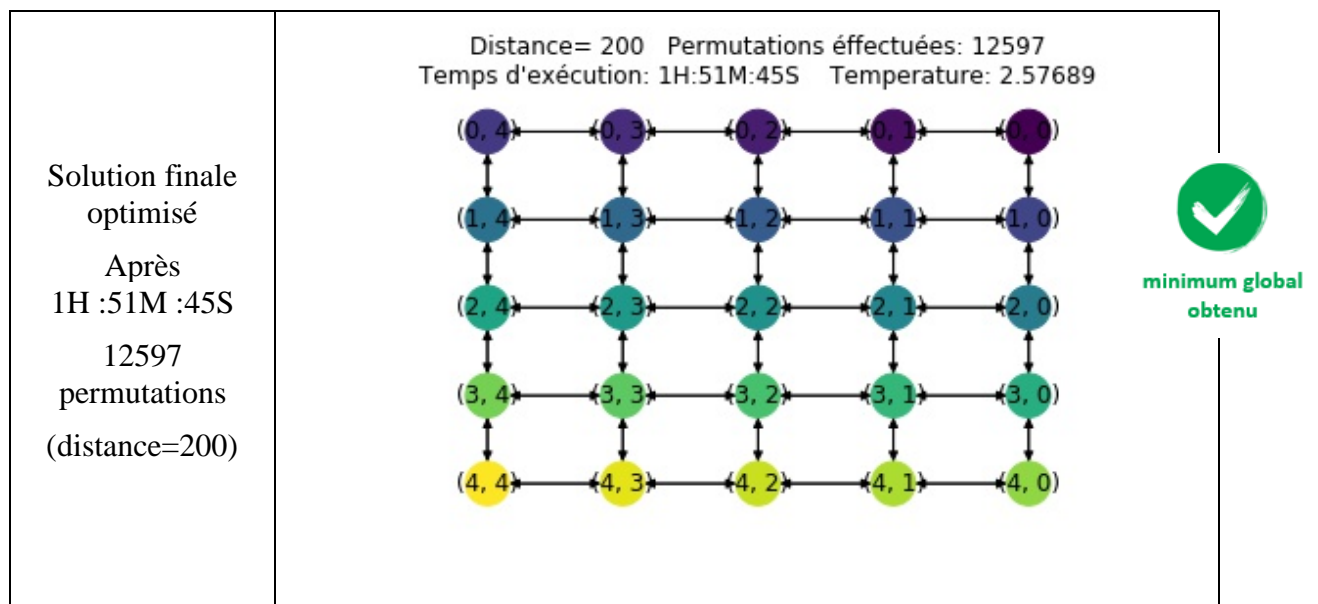


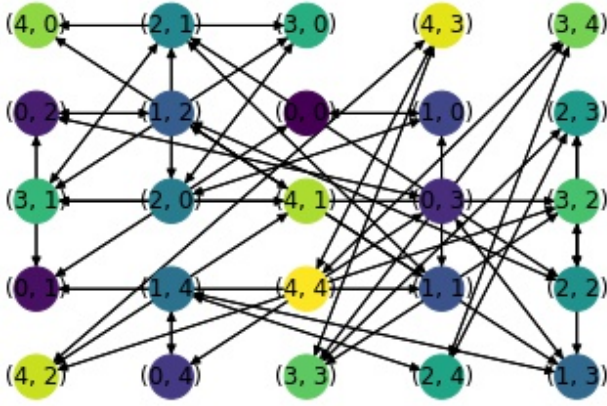
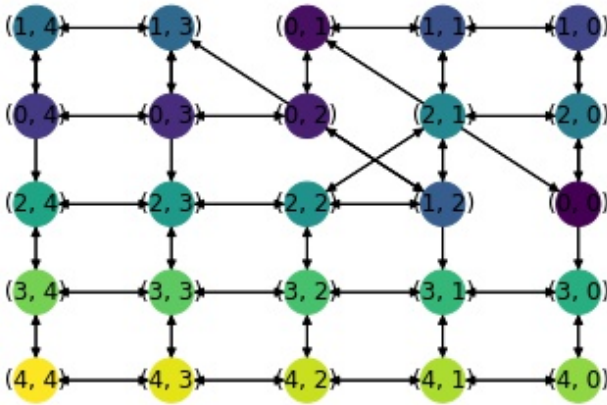

Figure 2: Evolution de la distance en fonction des itérations effectués

Interprétation :

Après 12597 permutations (1H :51M :45S), on obtient le minimum global de la fonction objectif du problème. Donc, cette configuration de recuit simulé est efficace. Dans l'essai n°3 et n°4, on vérifiera cette efficacité en diminuant le taux de refroidissement tout en gardant la solution initiale.

Essai 3:

Température initiale	500
Taux de refroidissement	0.8
Equilibre thermodynamique	Il est atteint : <ul style="list-style-type: none"> - $2*N$: mouvements acceptés - $50*N$ mouvements tentés Avec $N=25$ nombres des composants
Système figé	3 paliers de température successifs sans aucun mouvements

<p>Solution de départ (d=730)</p>	<p>Montage Départ</p> 
<p>Solution finale optimisé</p> <p>Après 0H :42M :26S</p> <p>3116 permutations (distance=270)</p>	<p>Distance= 270 Permutations effectuées: 3116 Temps d'exécution: 0H:42M:26S Temperature: 2.57689</p>  <div style="text-align: right;">  <p>piégé dans un minimum local</p> </div>

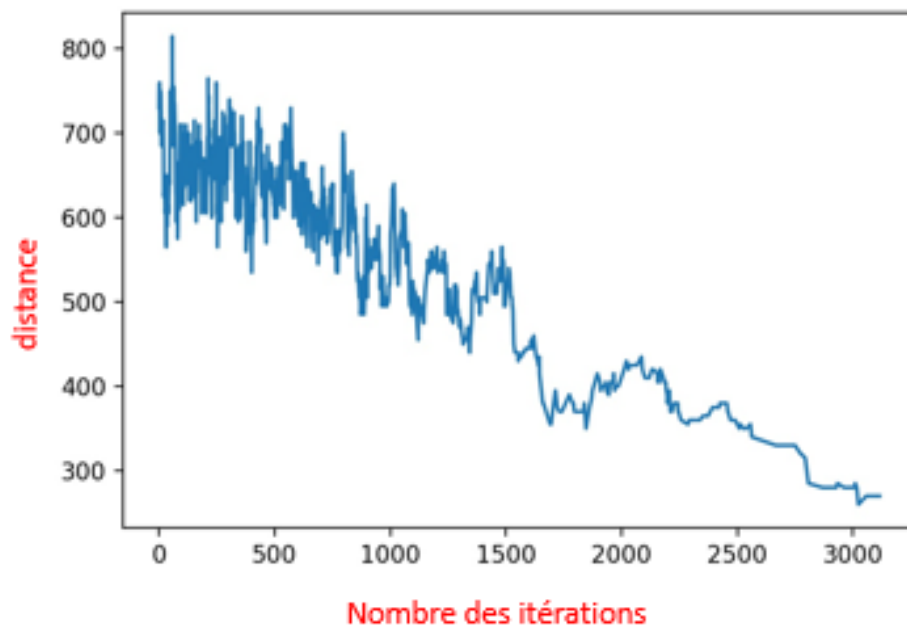
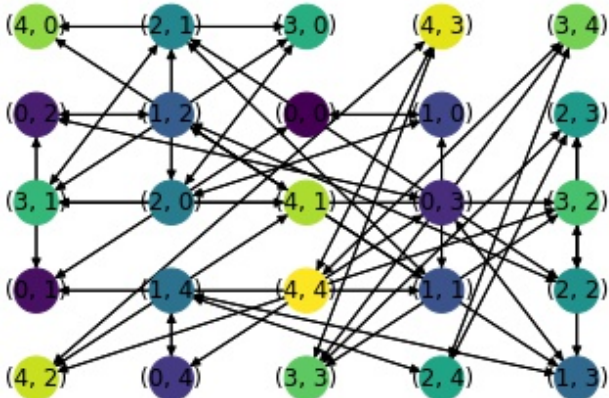
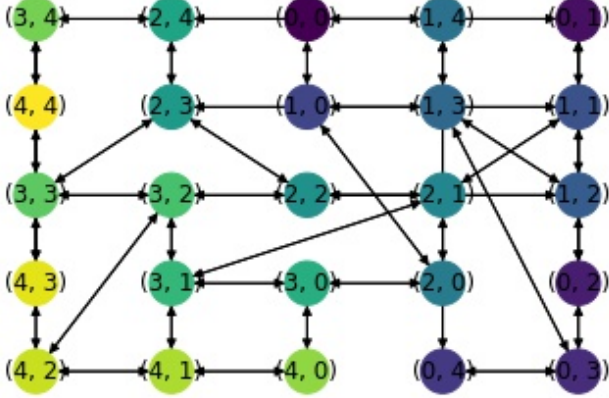



Figure 3: Evolution de la distance en fonction des itérations effectués

Essai 4:

Température initiale	500
Taux de refroidissement	0.5
Equilibre thermodynamique	<p>Il est atteint :</p> <ul style="list-style-type: none"> - $2*N$: mouvements acceptés - $50*N$ mouvements tentés <p>Avec $N=25$ nombres des composants</p>
Système figé	3 paliers de température successifs sans aucun mouvements

<p>Solution de départ (d=730)</p>	<p>Montage Départ</p> 
<p>Solution finale optimisé Après 0H :44M :29S 7448 permutations (distance=325)</p>	<p>Distance= 325 Permutations effectuées: 7448 Temps d'exécution: 0H:44M:29S Temperature: 0.12207</p>  <div data-bbox="1353 840 1465 1003">  piégé dans un minimum local </div>

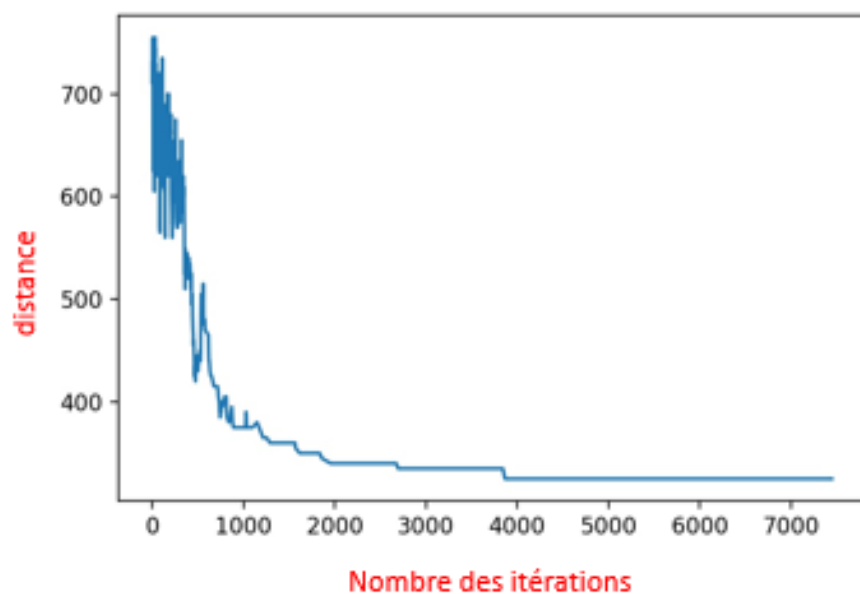


Figure 4:Figure 3:Evolution de la distance en fonction des itérations effectués

Interprétation :

On constate que dès que le taux de refroidissement est devenu de plus en plus grand, le recuit simulé est piégé dans un minimum local :

- Essai n°3 : minimum local $d=270$
- Essai n°4 : minimum local $d=325$

Quand la température T est très grande, la probabilité d'accepter une mauvaise configuration est proche de 1 d'où les mauvais mouvements seront acceptés avec une forte probabilité et il aura une forte chance de sortir d'un minimum local.

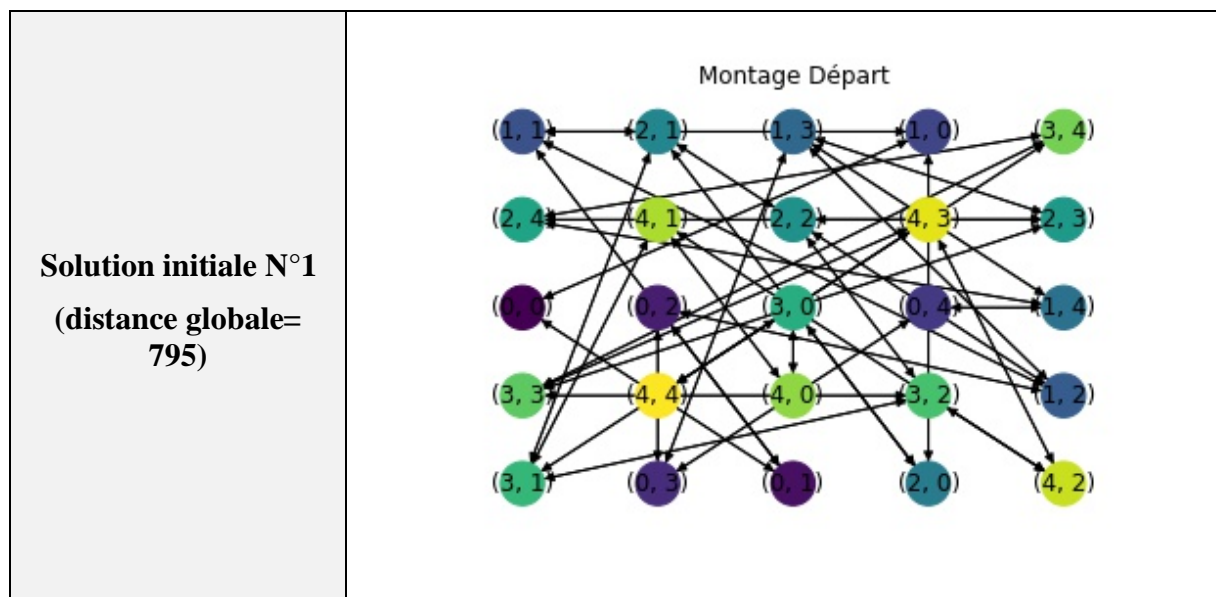
Alors, quand la température T est très faible, la probabilité d'accepter une mauvaise configuration est proche de 0 d'où les mauvais mouvements seront acceptés avec une très faible probabilité et il aura une faible chance de sortir d'un minimum local.

Après plusieurs essais, on constate que la configuration suivante de recuit simulé est la meilleure pour obtenir le minimum global.

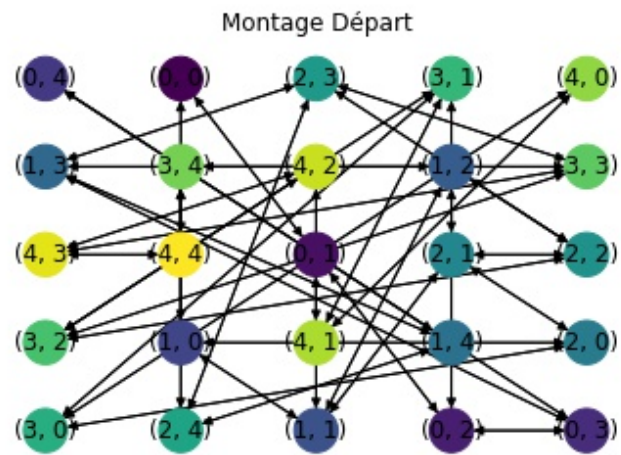
Température initiale	500
Taux de refroidissement	0.9
Equilibre thermodynamique	Il est atteint : <ul style="list-style-type: none">- $2*N$: mouvements acceptés- $50*N$ mouvements tentés Avec $N=25$ nombres des composants
Système figé	3 paliers de température successifs sans aucun mouvements

Pour valider cette décision, il suffit de tester cette configuration avec différentes solutions initiale et s'assurer que le minimum global est bien obtenu.

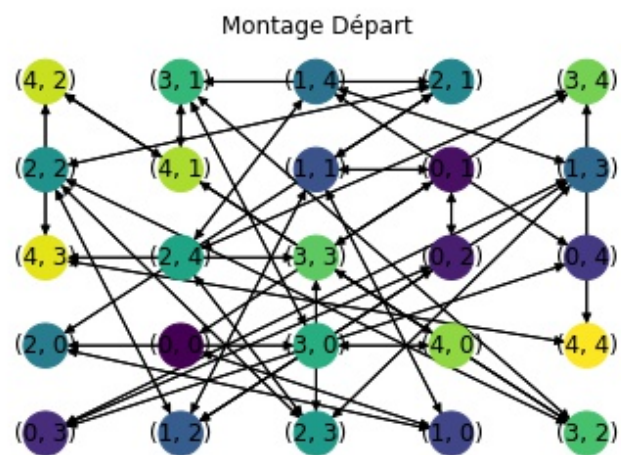
Les solutions initiales choisies sont les suivantes :



Solution initiale N°2
(distance globale=
745)

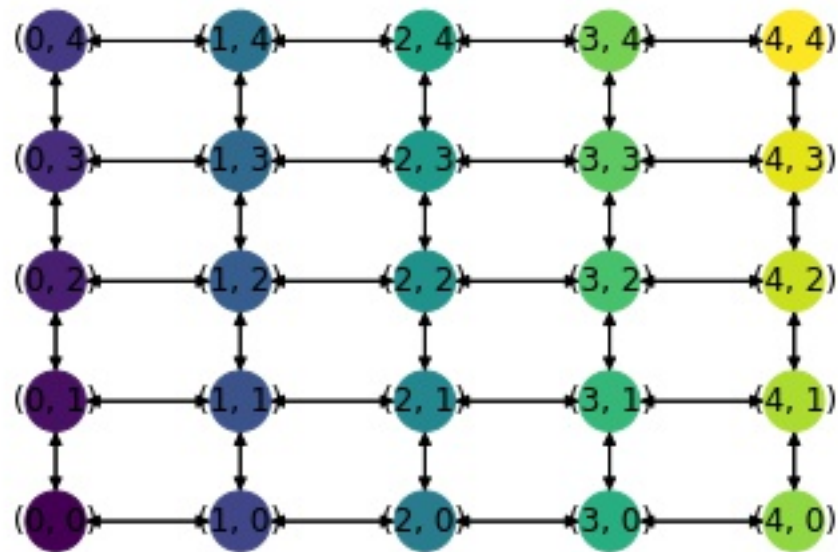


Solution initiale N°3
(distance globale=
765)

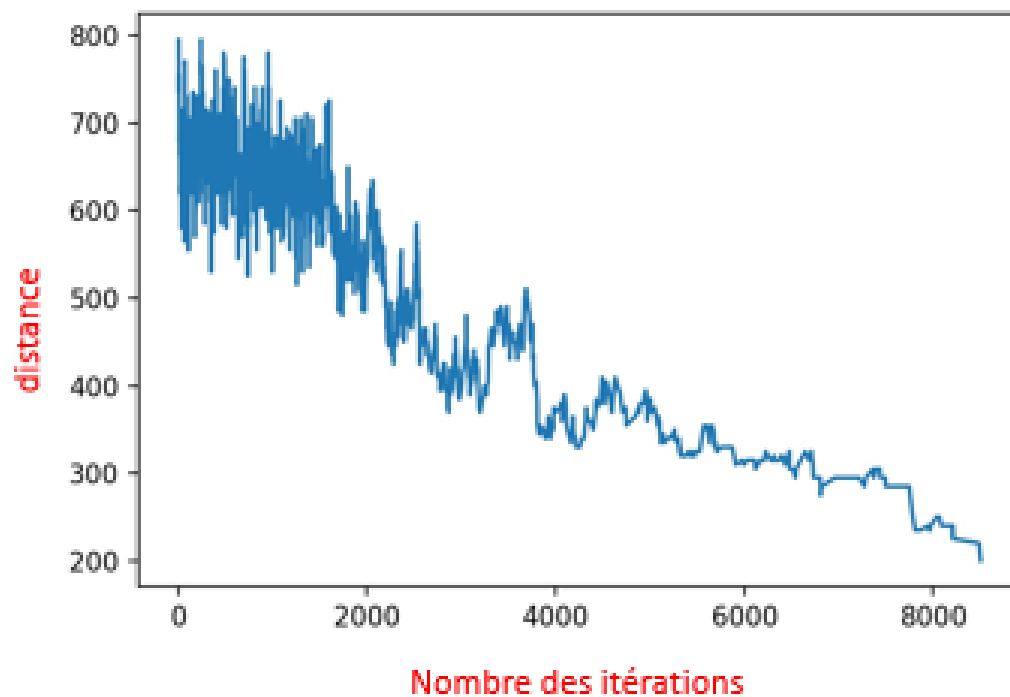


Le résultat de recuit simulé pour la solution initiale n°1 (distance globale= 795)

Distance= 200 Permutations effectuées: 8496
Temps d'exécution: 1H:19M:35 Temperature: 3.53483

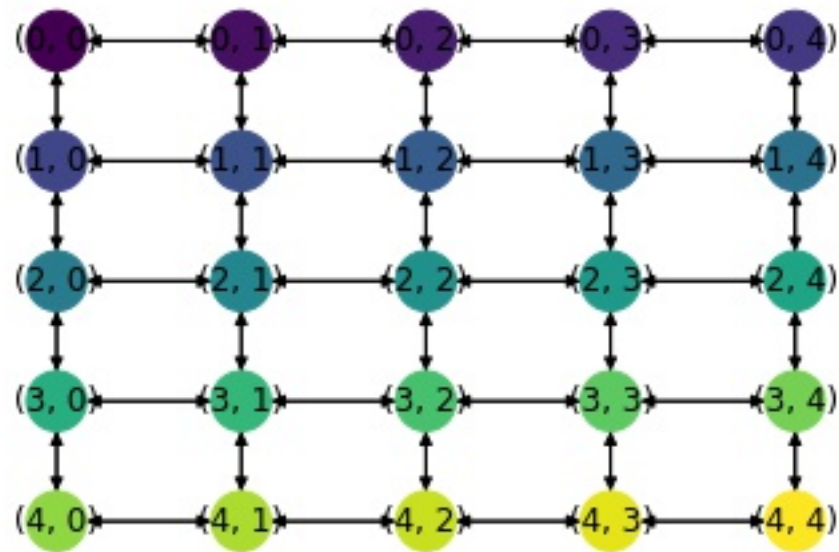


minimum global
obtenu

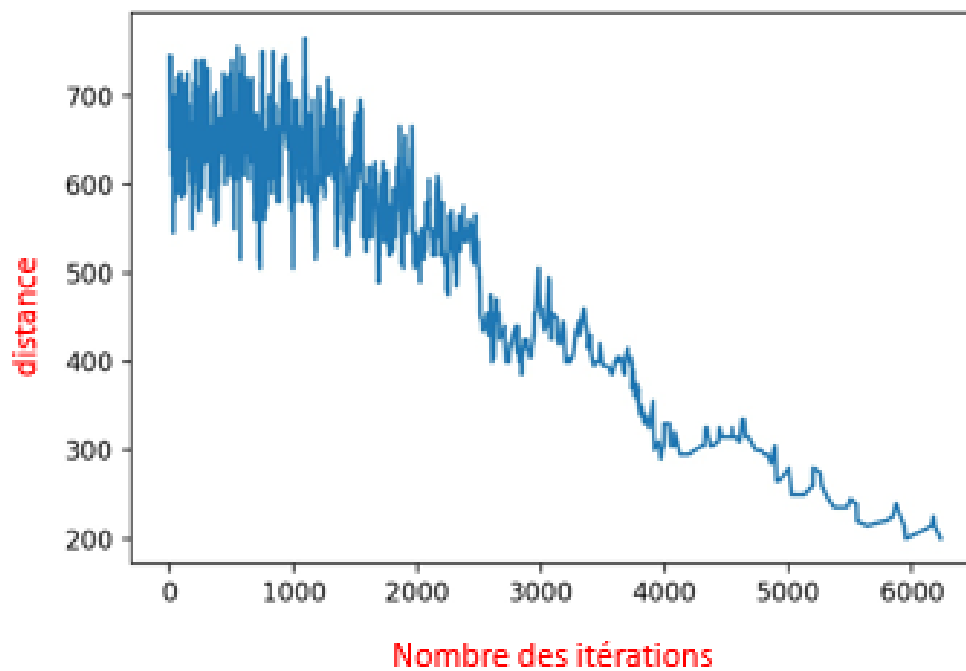


Le résultat de recuit simulé pour la solution initiale n°2 (distance globale= 745)

Distance= 200 Permutations effectuées: 6231
Temps d'exécution: 1H:19M:11S Temperature: 5.38763

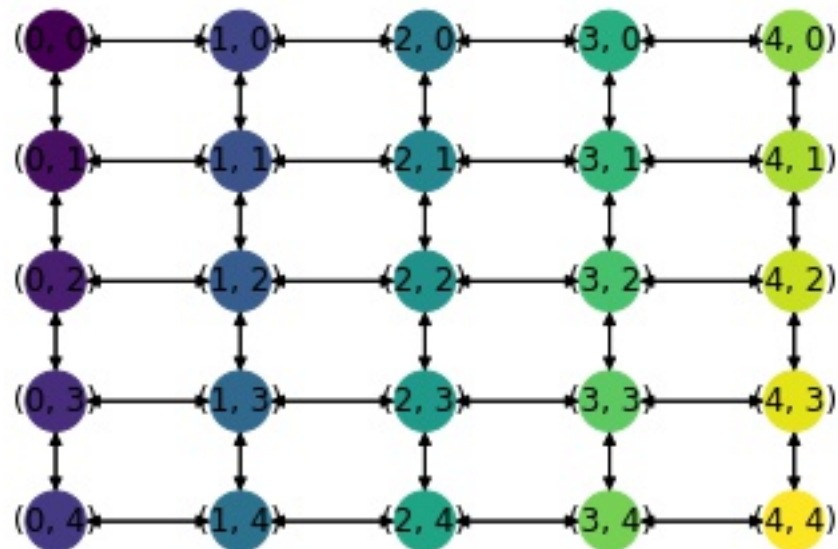


minimum global
obtenu

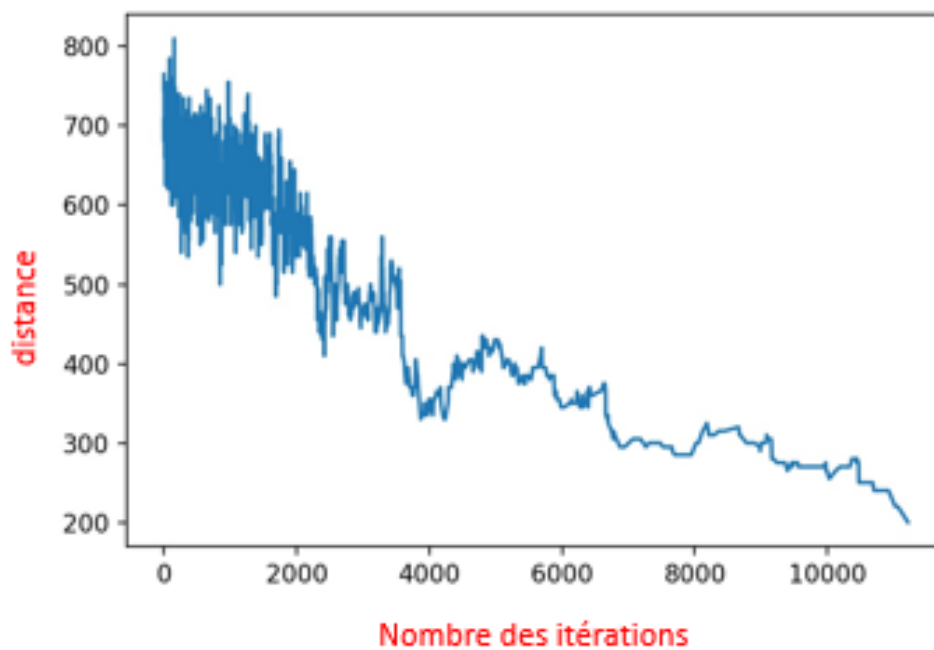


Le résultat de recuit simulé pour la solution initiale n° 3 (distance globale= 765)

Distance= 200 Permutations effectuées: 11215
Temps d'exécution: 1H:29M:8S Temperature: 2.57689



minimum global
obtenu



Interprétation :

En partant des différentes solutions initiales, le recuit simulé avec la configuration choisie a assuré une convergence vers le minimum global.

7. Conclusion générale

On conclut que le recuit simulé est un algorithme simple, adaptative et efficace pour résoudre les problèmes d'optimisation difficile. Il assure la convergence sans rester piéger dans un minimum local. Il est déjà utilisé industriellement et il a fait ses preuves sur des problèmes à grandes tailles.

Mais comme tout algorithme, le recuit simulé a des faiblesses. En effet, il dépend des plusieurs paramètres:

- le choix de la température initiale
- le critère de système figé ou non (critère d'arrêt)
- le critère de l'équilibre thermodynamique (choix des nombres des mouvements acceptés et mouvements tentés)
- le taux de refroidissements

et il a une forte sensibilité aux paramètres liées à la température.