Compte rendu du TP 1 Logique Floue, Réseaux de neurones et Algorithmes génétiques

3 ing glsi 02 |Année universitaire 2014 - 2015

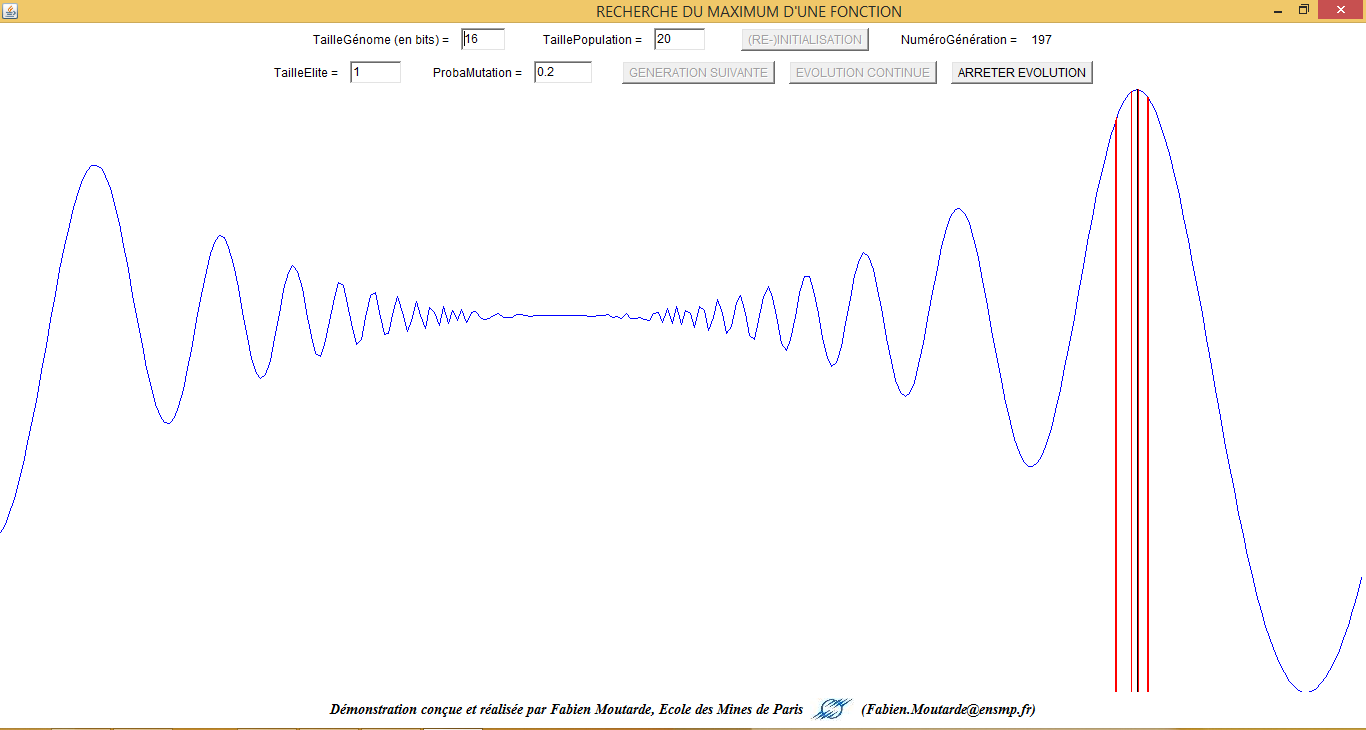
Réalisé par :

Barouni Sarra et Bennouri Ahmed

ISI

Partie1 :

# Exercice 1 : Recherche du maximum d’une fonction

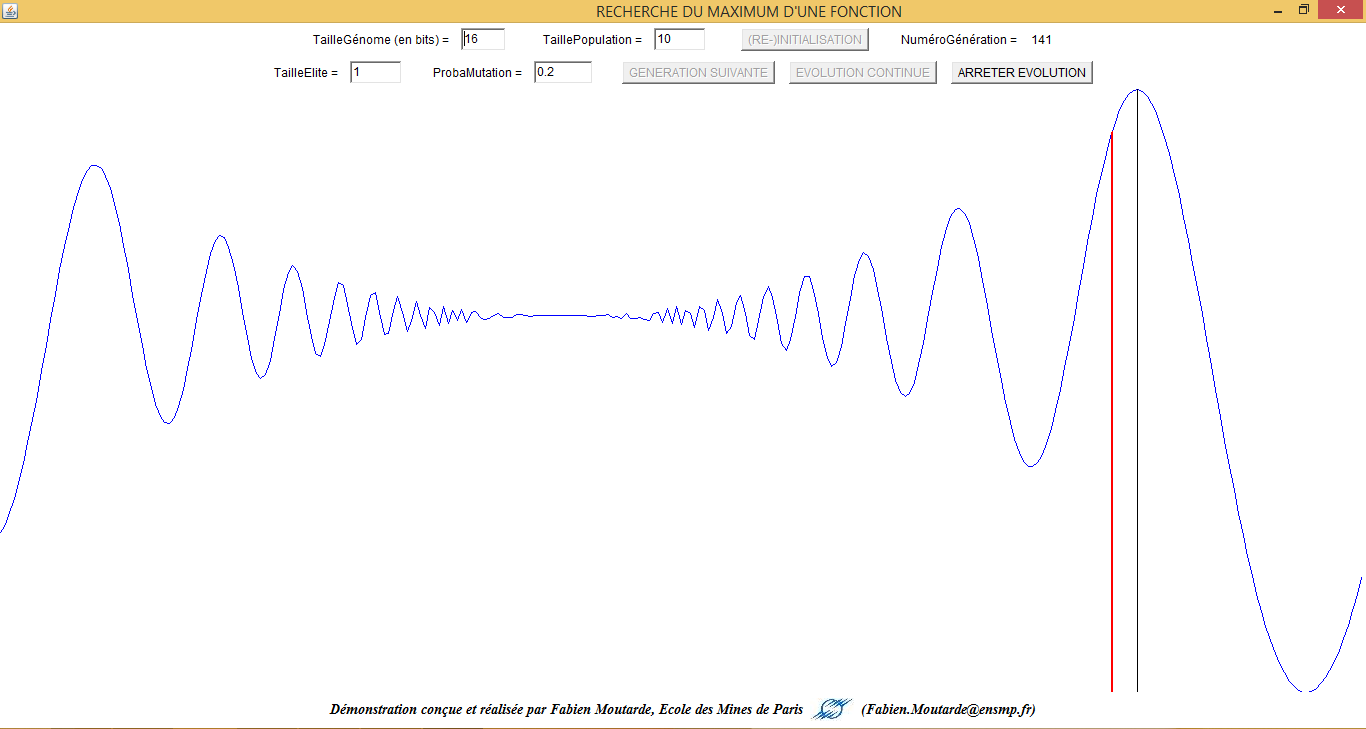


Taille de l’individu : 23 bits.

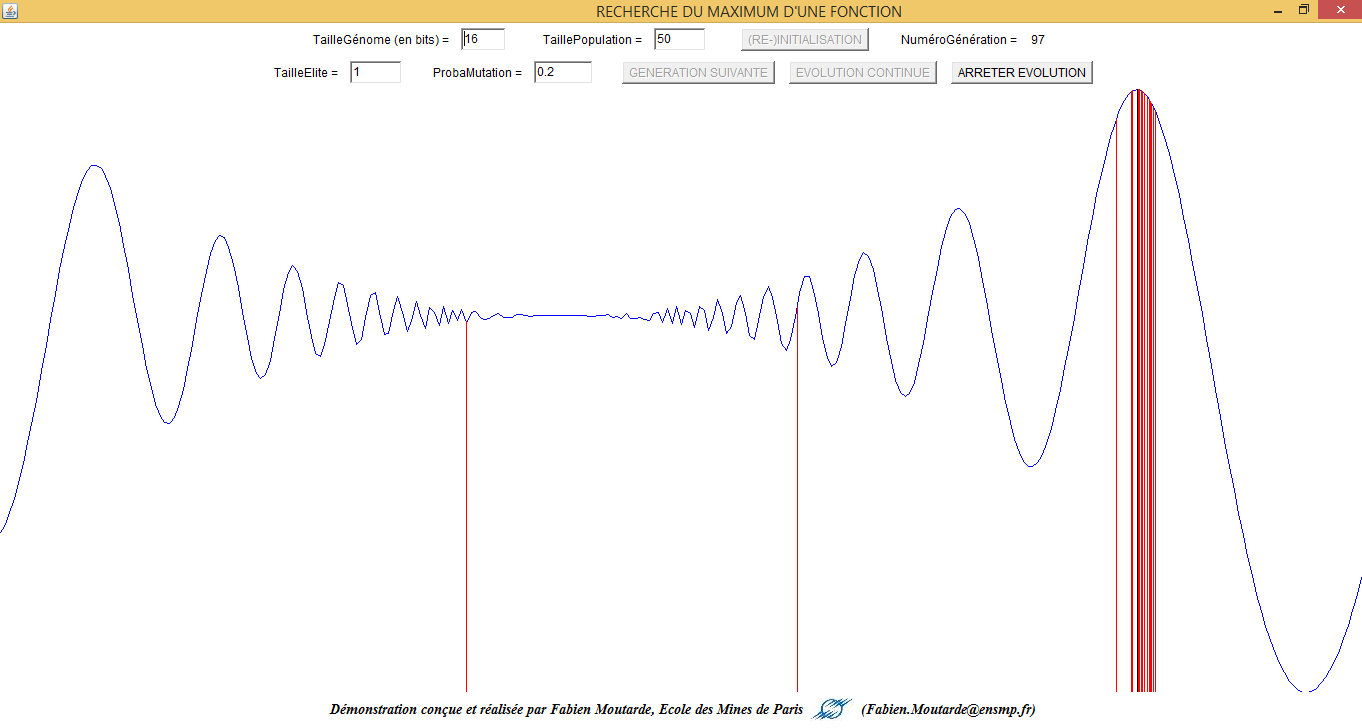
## 1er paramètre : Taille de population :

Taille de la population :

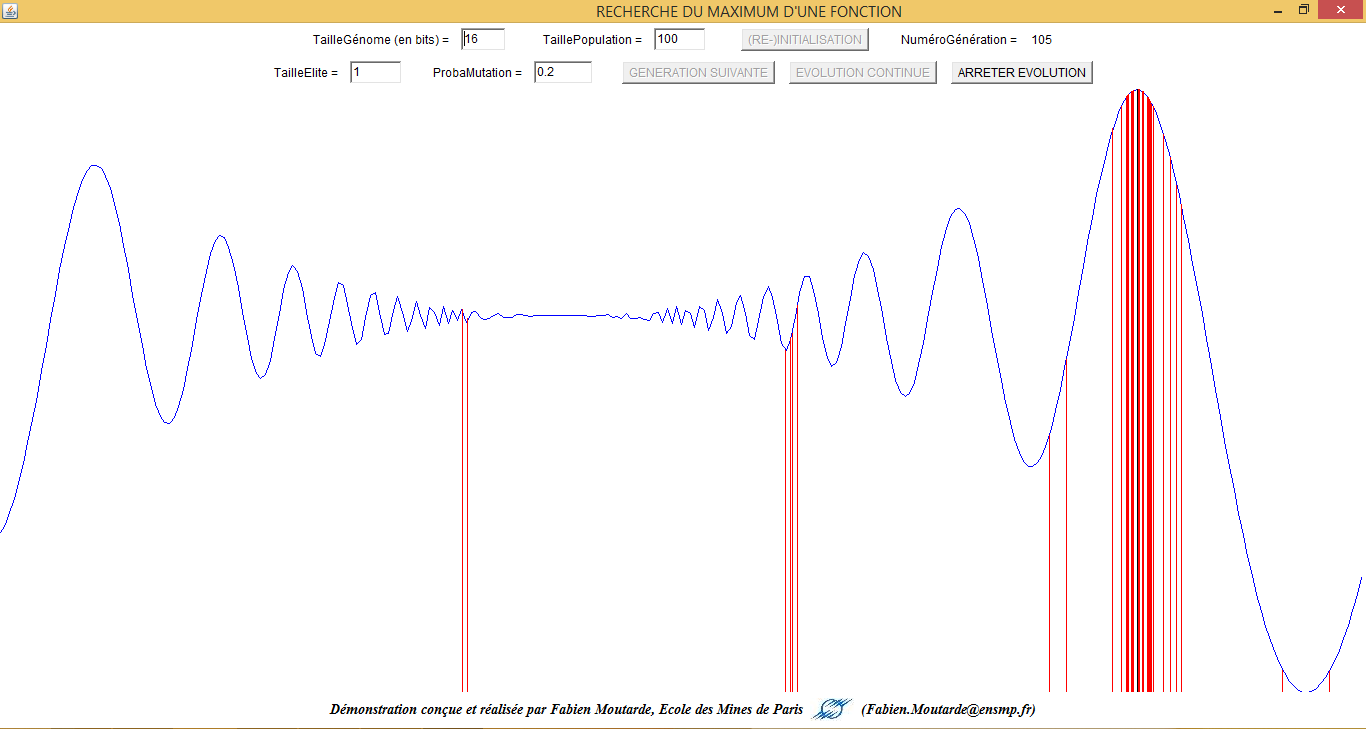
10 :



50 :



100 :

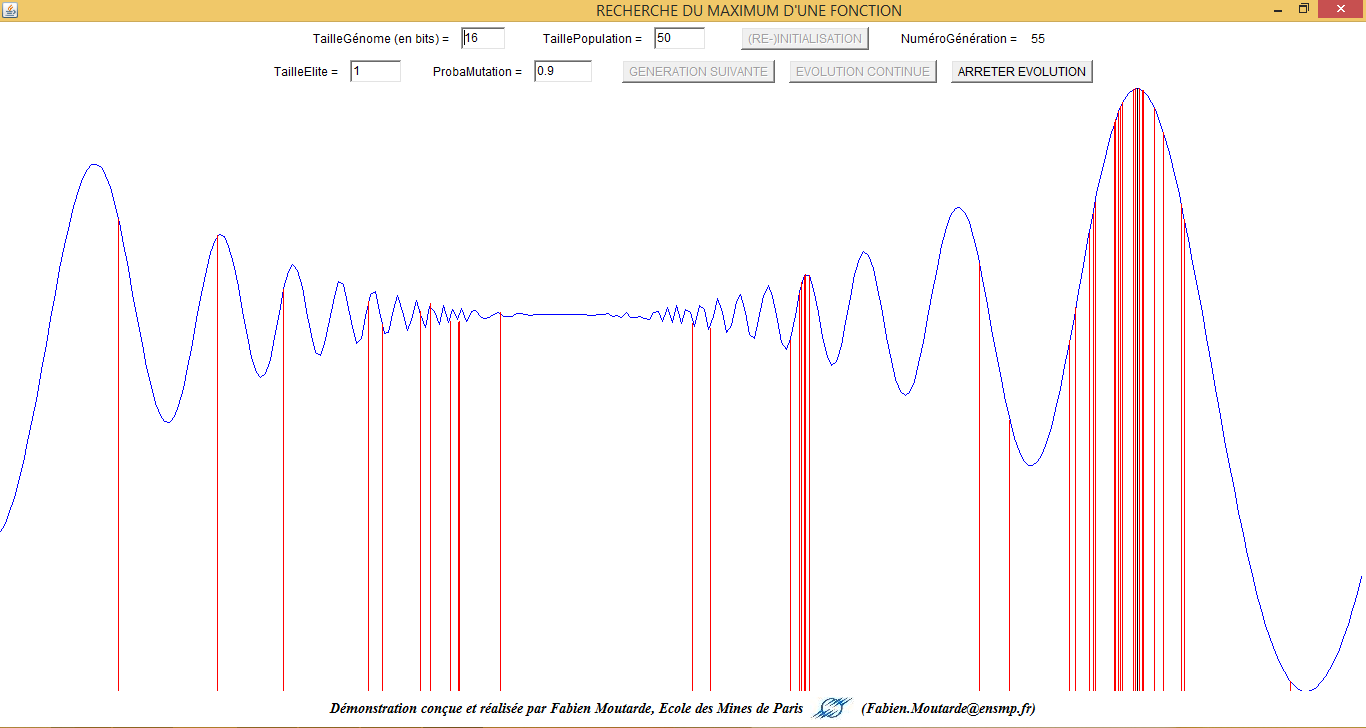


### Remarque :

Si on augmente la taille de population, le nombre de générations diminue et le temps d’exécution augmente. Ceci est due a

## 2ème paramètre : Taux de mutation :

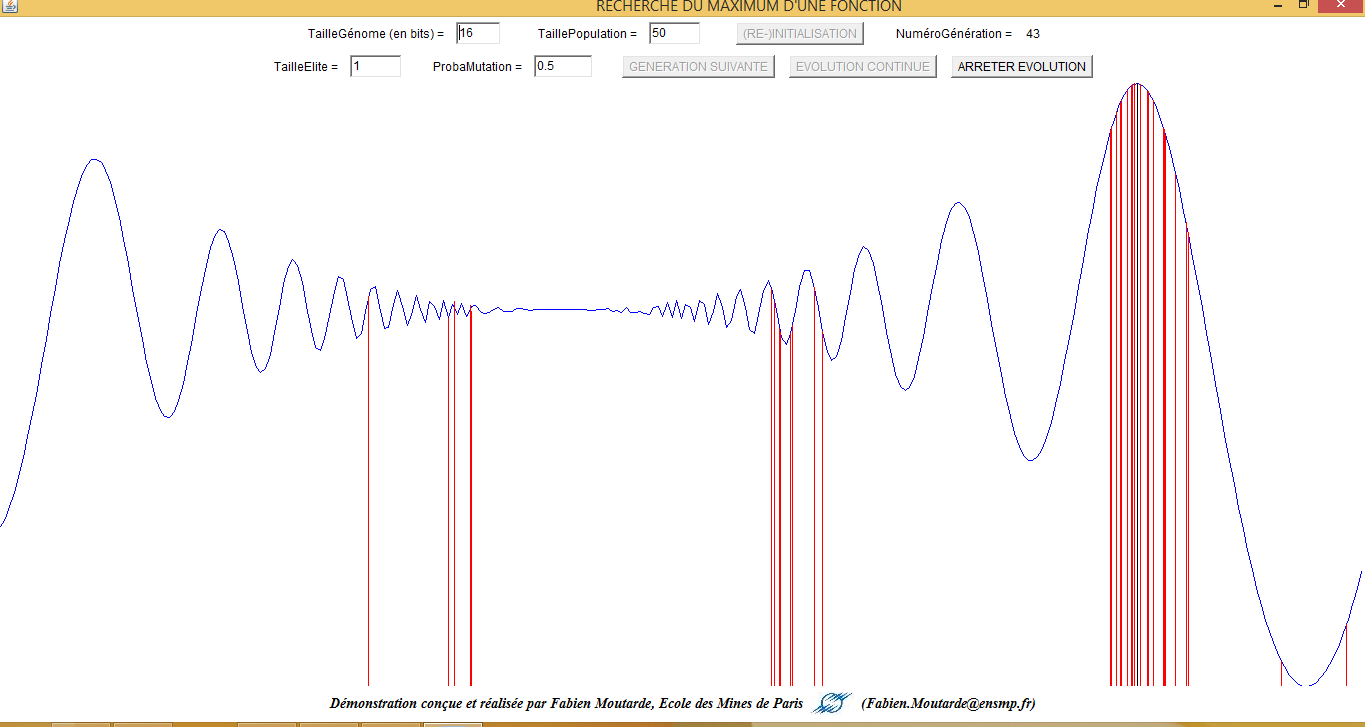
### Population 50, taux de mutation 0.9 :



A cause de la grande probabilité de mutation, on s’éloigne de la solution optimale, l’algorithme change presque tous les bits des individus.

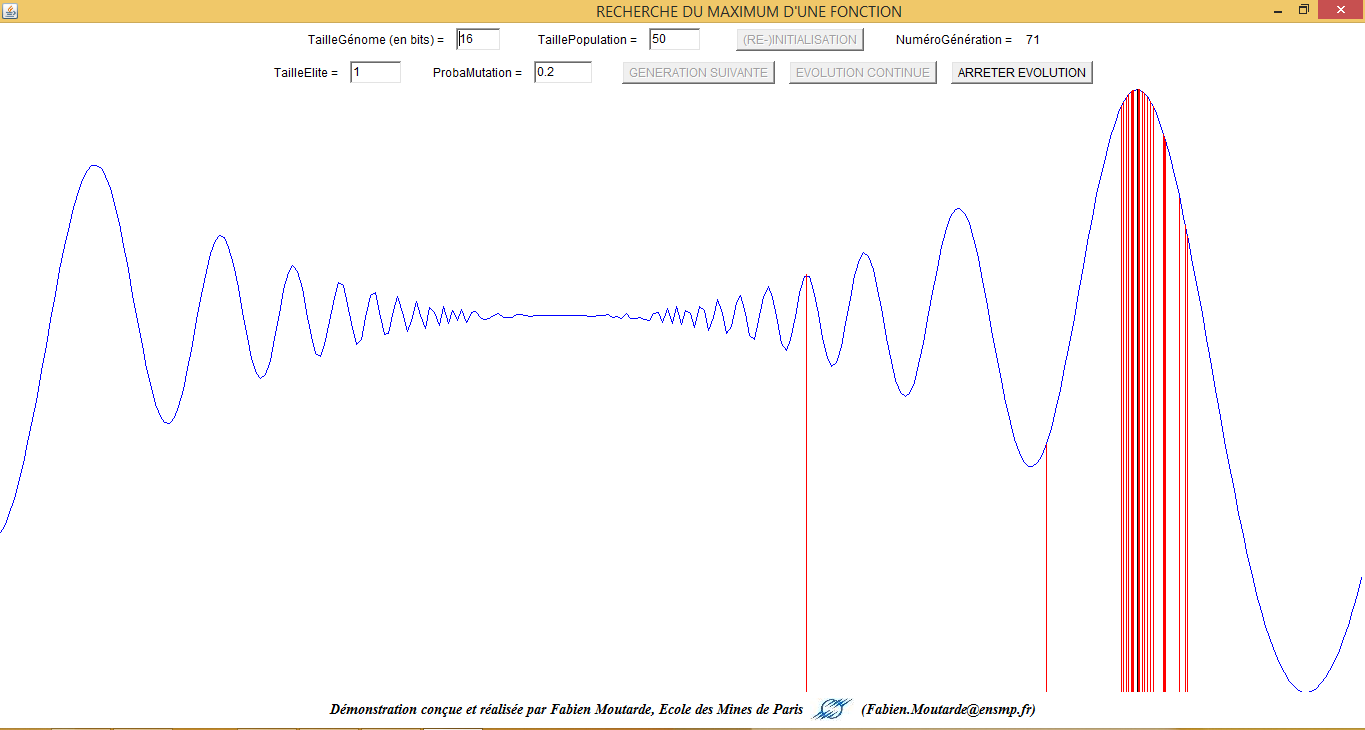
Résultat : on n’aura pas de convergence.

### Population 50, taux de mutation 0.5 :



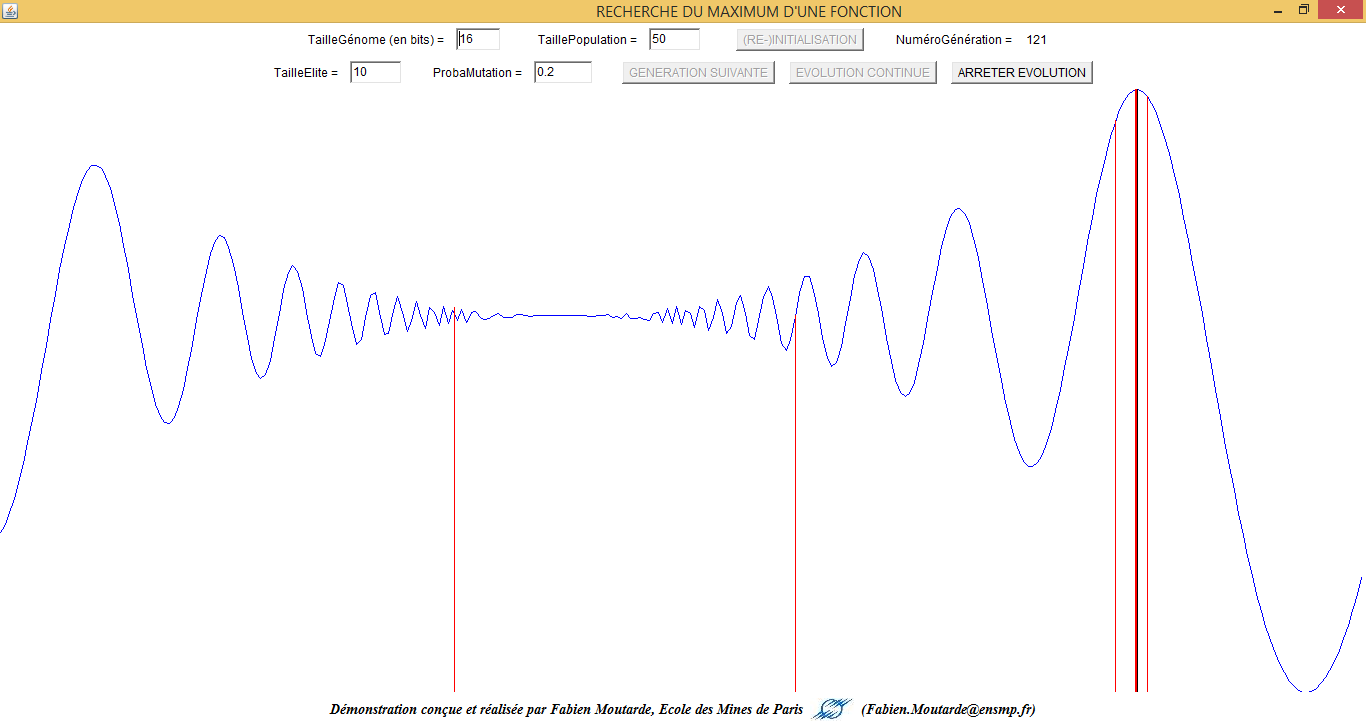
La moitié des bits de l’individu sera changée, le temps de convergence est plus rapide que dans le cas précédent.

### Population 50, taux de mutation 0.2 :



Un nombre limité de bits sera sujet de la mutation, dans notre cas c’est la meilleure solution car elle nous a permis de trouver le maximum global dans un temps minimum.

## 3ème paramètre : Critère de sélection (Sélection de l’élite) :

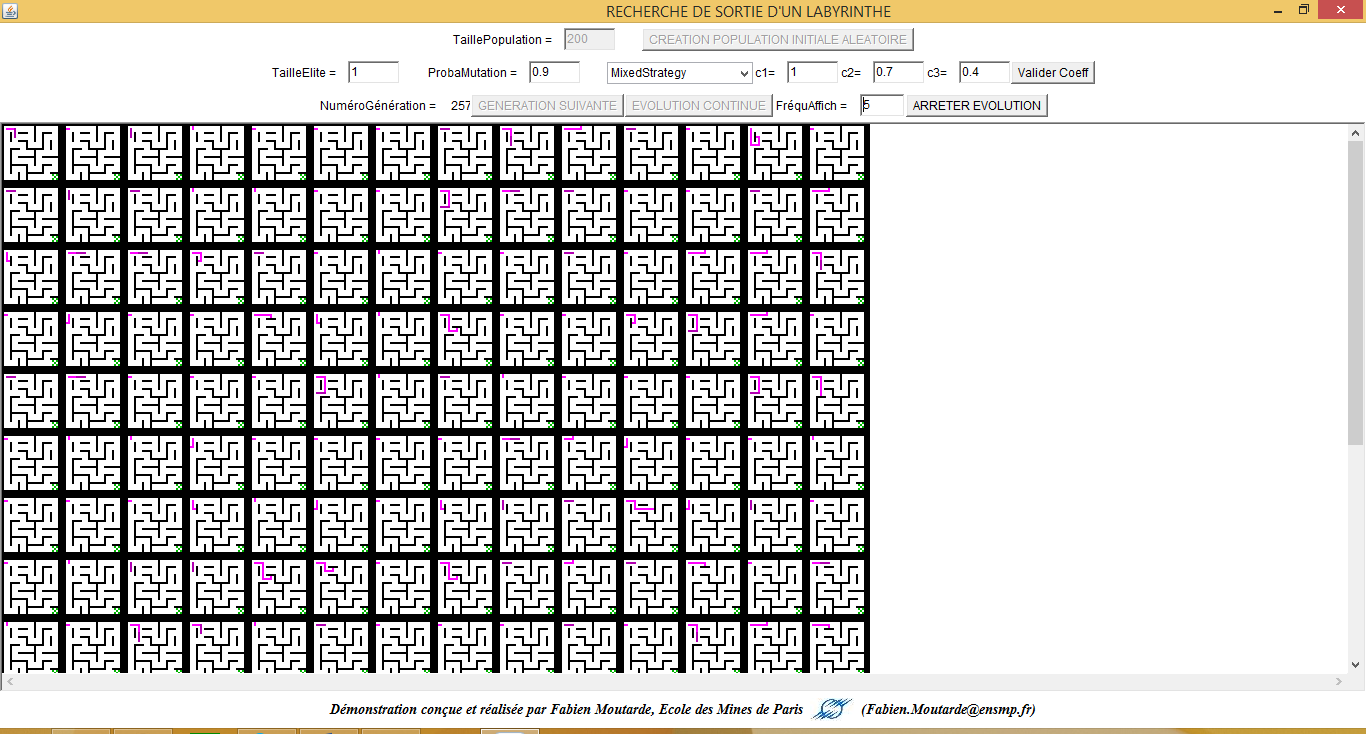


Si on augmente le nombre des élites on risque de tomber sur des maximums locaux.

Si nous augmentons le nombre des élites à garder dans la nouvelle génération. Si le nombre de population est grand, il ne faut pas dépasser 2 individus, sinon, il ne faut pas dépasser 1.

# Exercice 2 : Recherche de la sortie d’un labyrinthe

Individu : carreau



## Fonction d’évaluation  Exit Found :

Le résultat de la fonction d’évaluation est une variable booléenne, soit on trouve l’issue soit on ne la trouve pas, et ceux indépendamment du nombre des pas.

## Fonction d’évaluation Maximize Exit Proximity :

Cette fonction évalue à quel point le rebot est près de la sortie.

Dans ce type de fonction d’évaluation, il faut utiliser une population de taille importante.

L’élite est sélectionnée selon le score de proximité de la sortie.

## Fonction d’évaluation Maximize Cell Visited:

Cette fonction dépend du nombre de pas effectués par le robot indépendamment de la sortie. Cette fonction cause l’obtention des individus qui ont fait de longues boucles sans aboutir à un résultat.

## Fonction d’évaluation Mixed Strategy:

Cette stratégie combine les 3 critères :

1. La retrouvaille de la sortie.
2. La proximité de la sortie.
3. Les pas effectués.

Partie2 : Voyageur de commerce :

# Codage réel :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 2 | 1 | 5 | 4 |

%% Fonction d’initialisation :

Function [y]= Init (nbpop,nbv)

Y : zeros(nbpop,nbv) ;

For i=1 : nbpop

Y(i) randperm (nbv) ;

%% randperm est une fonction prédéfinie permettant de remplir un vecteur de n cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 2 | 1 | 4 | 5 |
| 2 | 1 | 3 | 5 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 2 | 1 | 5 | 4 |
| 2 | 1 | 3 | 4 | 5 |

End ;

End

# Opérateur de croisement :

Function[X1,Y1]= croiser(X,Y)

N=size(X) ;

K=round(rand()\*n) ;

I=k+1 ;

X1=x ;

Y1=y ;

For j=1 :n

Test=0 ;

For l=1 :k

If Y(j)==X(l)

Test=1 ;

End ;

End ;

If test ~=n

X1(i)=Y(j) ;

Y1(i)=X(j) ;

I=i+1 ;

End ;

# Opérateur de mutation :

Focntion d’évaluation :

Entrée : matrice de distance, d (i,i)=0.

D(i,i+1 mod n=1)

D(X(i) , X(i+1 mod (n))