

TD 5: Optimisation de type stochastique

Exercice 1–

A la manière des algorithmes génétiques, la méthode DE recherche de manière stochastique le minimum global d'une fonction $J : \mathbb{R}^n \rightarrow \mathbb{R}$.

DE fait évoluer une population de N_{pop} éléments (ou individus) avec l'algorithme suivant (où $CR \in [0, 1]$ et $F \in [0, 2]$ sont deux paramètres) :

- (i) Initialisation aléatoire de N_{pop} éléments
- (ii) De la génération 1 à la génération N_{gen} :
- (iii) Pour chaque individu $x \in \mathbb{R}^n$:
 - Choisir aléatoirement trois éléments a , b et c dans la population, distincts entre eux et distincts de x .
 - Tirer i_0 indice aléatoire dans $\{1, \dots, n\}$ et calculer $y = (y_1, \dots, y_n)$ comme suit :

$$\forall i \in \{1, \dots, n\}, \quad y_i = a_i + F(b_i - c_i) \text{ si } (r_i < CR) \text{ ou } (i = i_0), \text{ sinon } y_i = x_i$$

où r_i est choisi aléatoirement dans $[0, 1]$.

- Si $J(y) < J(x)$, remplacer x par y dans la population.

- (iv) Fin d'une génération

1. Quels sont les principaux points communs et quelles sont les principales différences de l'algorithme DE par rapport à un algorithme génétique ?
2. Interpréter les paramètres CR et F pour l'algorithme. Quelles valeurs extrêmes peuvent-ils prendre ?
3. Le script suivant propose une implémentation de l'algorithme DE en Scilab :

```
function y=f(x)
    y=n+sum(x.^2-cos(2*pi*x));
endfunction
////////parametres//////////
Npop=40; Ngen=20; n=2; CR=0.2; F=0.8
```

```

////////////////////////////////////
A=zeros(Npop,n+1); // matrice de population
//
A=10*rand(Npop,n+1)-5*ones(Npop,n+1);
//
//evaluation
y=[];
for j=1:Npop
    y(j)=f(A(j,1:n));
end
A(:,n+1)=y;
////////////////////////////////////
for i=1:Ngen
//
[u,v]=gsort(A(:,n+1));
A=A(v,:); // rangement du plus mauvais au meilleur
<<<<<    disp('meilleur element'),disp(    )
<<<<<<    disp('meilleure valeur'),disp(A(    )
//
    for k=1:Npop
        i1=1;i2=1;i3=1;
        while (i1==i2)|(i1==i3)|(i2==i3)|(i1==k)|(i2==k)|(i3==k)
            i1=int(Npop*rand()+1);
            i2=int(Npop*rand()+1);
            i3=int(Npop*rand()+1);
        end
<<<<<<<    a=    ;b=    ;c=    ;x=    ;
<<<<<<<    j0=
        y=[];
        for j=1:n
            if (rand()<CR)|(j==j0)
<<<<<<    y(j)=
            else
<<<<<<    y(j)=
        end
        val1=f(y);val2=f(x);
        if (val1<val2) then
            A(k,1:n)=y';A(k,n+1)=val2;

```

```
        end
      end
    end
  end
```

Malheureusement, certaines lignes repérées par :

```
<<<<<<<
```

ont été effacées. Reconstituer les lignes correspondantes.

4. On souhaite tracer l'historique de décroissance de la meilleure valeur de f en fonction du nombre d'itérations. Rajouter les instructions manquantes pour cet affichage.

Exercice 2 –

On propose l'algorithme suivant pour la minimisation d'une fonction f :

```
x=-20+30*rand(); // point initial
Niter=2000;alpha=0.5;Ytot=[]
for i=1:Niter
    y1=f(x);
    xtilde=x+(-alpha+2*alpha*rand())
    y2=f(xtilde)
    p=exp(-(y2-y1)/(1/log(i+1)));
    if (rand())<p) then
        x=xtilde;
    end
end
disp('valeur finale obtenue pour x:')
disp(x)
```

1. Expliquer le fonctionnement global de ce programme ainsi que les instructions aux lignes 5, 7 et 8.
2. Que représente dans ce programme le paramètre α ?
3. Que représente le terme $1/\log(i+1)$ et pour quelle raison a-t-il été choisi ainsi ? Proposer un autre choix possible.

Exercice 3 –

Un algorithme génétique a pour opérateur de croisement la fonction suivante :

```
function Acrois=croisement(A,pc)
[Npop,n]=size(A)
Acrois=A;
for k=1:Npop/2
    n1=int(Npop*rand()+1);
    n2=int(Npop*rand()+1);
    alpha=rand();
    u1=A(n1,:);u2=A(n2,:);
    if(rand()<pc) then
        Acrois(2*k-1,:)=alpha*u1+(1-alpha)*u2;
        Acrois(2*k,:)=(1-alpha)*u1+alpha*u2;
    end
end
endfunction
```

1. Que représente la variable pc et quel effet a t-elle ?
2. Expliquer en quoi cet algorithme est de type aléatoire et à quel(s) niveau(x) intervient l'aspect aléatoire ?
3. On cherche à modifier l'opérateur de croisement afin de permettre une plus grande variété de solutions possibles en sortie. Quelle modification à l'algorithme précédent proposeriez vous ?

Exercice 4–

On cherche à comparer la façon dont intervient la fonction J qu'on cherche à minimiser, dans les quatre algorithmes suivants : recuit simulé, algorithme génétique, stratégie d'évolution et PSO.

1. Dans le recuit simulé, l'algorithme est-il modifié si la fonction J est remplacée par la fonction $4J$, respectivement $J + 3$? Que se passe t-il dans le cas d'un algorithme PSO ? Justifier la réponse.
2. Dans un algorithme génétique, est-il possible de garder à la génération suivante le plus mauvais individu ? Que se passe t-il dans le cas d'une stratégie d'évolution ?

3. La sélection par le rang dans un algorithme génétique a été souvent écrite ainsi :

```
//  
function Asel=selection(A)  
[Npop,L]=size(A)  
L=L-1;  
[s,p]=gsort(A(:,L+1));  
A=A(p,:);  
Asel=[];  
p=(1:Npop)/sum(1:Npop);ps=cumsum(p);  
for i=1:Npop  
    u=rand();isel=1;  
    while (u>ps(isel))  
        isel=isel+1;  
    end  
    Asel=[Asel;A(isel,:)];  
end  
endfunction
```

Où intervient la fonction J dans cette écriture et quel est le rôle dans cette fonction de la ligne $A=A(p, :)$?