

TD 4: Optimisation de type stochastique

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 les 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
  end
end

```

Malheureusement, certaines lignes repérées par:

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

ont été effacées. Reconstituer les lignes correspondantes.

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