

TD6: Optimisation globale

Ex 2) $\text{rand}()$: générateur aléatoire uniforme entre 0 et 1

```
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())
```

$\alpha \sim \mathcal{U}([-\alpha, \alpha])$
itération: i

$\alpha \sim \mathcal{U}([x-\alpha, x+\alpha])$

```
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:')
```

on reconnaît
 $p = \exp\left(-\frac{f(\tilde{x}) - f(x)}{1/\ln(i+1)}\right)$

$T_i \sim \mathcal{P}(1/\ln(i+1))$

tirage suivant une loi $\mathcal{B}(p)$

$\alpha = \tilde{\alpha}$
avec probabilité p

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.

2) α représente la taille du voisinage choisi (i , on la définit avec son rayon)

3) Le terme $\frac{1}{\ln(i+1)}$ représente la température à l'itération i .

Ce choix se justifie par la décroissance lente vers 0 de la suite T_i .

Autre choix possible:

* $T_i = \rho^i$ avec ρ proche de 1 (et < 1)

* ou $T_i = \rho \lfloor \frac{i}{k} \rfloor$ ($\lfloor x \rfloor$ = partie entière)

1) Il s'agit d'une méthode de recuit simulé.

Exercice 3.

Un algorithme génétique a pour opérateur de croisement la fonction suivante

```
1 function Acrois=croisement(A,pc)
2 [Npop,n]=size(A)
3 Acrois=A;
4 for k=1:Npop/2
5     n1=int(Npop*rand()+1);
6     n2=int(Npop*rand()+1);
7     alpha=rand();
8     if(rand()<pc) then
9         for j=1:n
10            Acrois(2*k-1,j)=alpha*A(n1,j)+(1-alpha)*A(n2,j);
11            Acrois(2*k,j)=(1-alpha)*A(n1,j)+alpha*A(n2,j);
12        end
13    end
14 end
15 endfunction
```

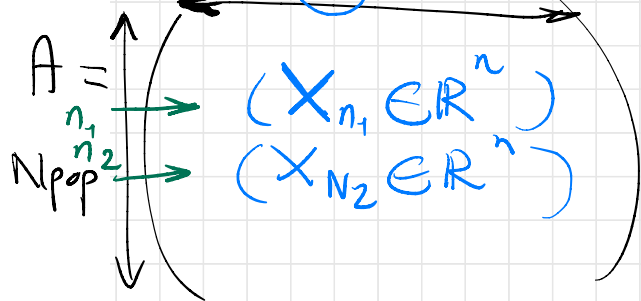
$\sim U([0,1])$
 $n_1, n_2 \sim U([1, N_{pop}])$

→ Opérateur de croisement pour un algorithme génétique

Entrée : A, pc

Sortie : $Acrois$

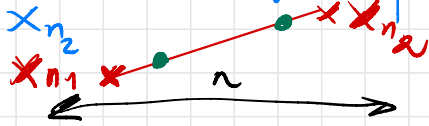
A : matrice taille $N_{pop} \times n$ / 2



L5 et 6 :

n_1 et n_2 prennent des valeurs entières entre 1 et N_{pop} , suivant une loi uniforme.

L8 : "avec une probabilité pc "
(pc : probabilité de croisement)
L10 et L11 : croisement barycentrique de X_{n_1} et X_{n_2}



$$Acrois = \begin{cases} \alpha X_{n_1} + (1-\alpha) X_{n_2} \\ (1-\alpha) X_{n_1} + \alpha X_{n_2} \end{cases}$$

1) f_c est une probabilité d'effectuer le croisement aux lignes $(k-1)$ et (k) de A_{cra} .

2) Effets aléatoires:

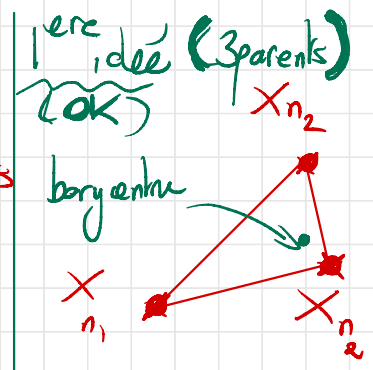
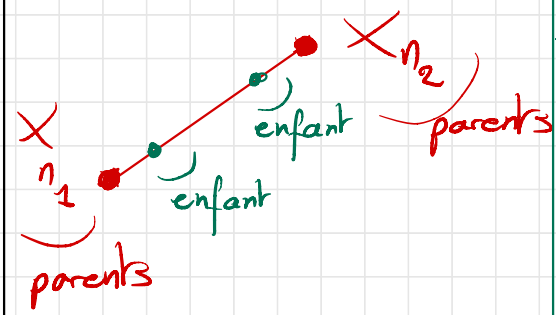
$(L5, L6) \rightarrow$ choix des 2 éléments à croiser ("parents")

$(L8) \rightarrow$ probabilité de "croiser" les deux parents choisis.

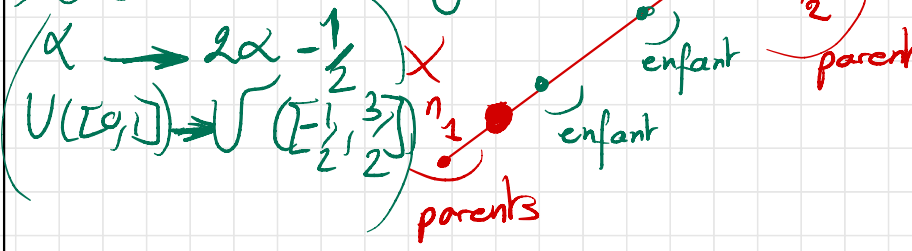
$(L10, L11) \rightarrow$ création des 2 nouveaux éléments ("enfants") à partir des 2 "parents".

(Parents \equiv parents)
 (Enfants \equiv offsprings (progéniture))

3) On cherche un croisement permettant de créer deux nouveaux éléments ("enfants") dans un ensemble plus vaste que le segment reliant les 2 parents:

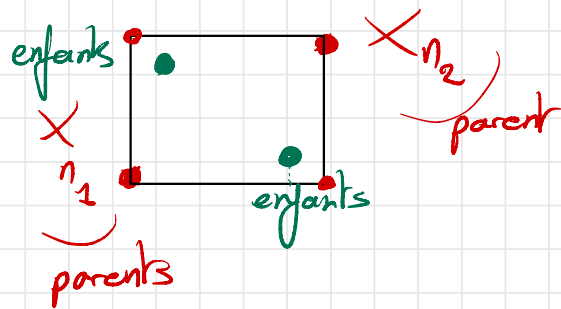


2ème idée ("sortir du segment")



$\alpha \rightarrow 2\alpha - \frac{1}{2}$
 $U([0, 1]) \rightarrow U\left(\left[-\frac{1}{2}, \frac{3}{2}\right]\right)$

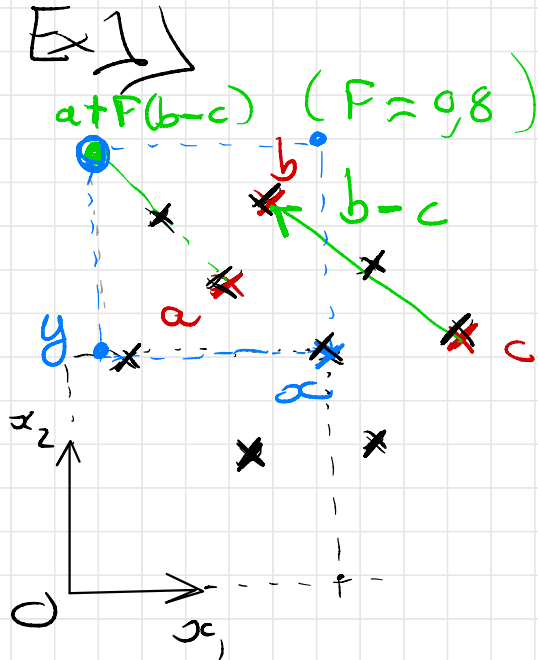
3^{ème} idée : remplacer le segment $[X_{n_1}, X_{n_2}]$
par l'hypercube de diagonale $[X_{n_1}, X_{n_2}]$



$$\Omega = \left\{ x \in \mathbb{R}^n \mid \begin{aligned} x_i &= \alpha_i (X_{n_1})_i + (1 - \alpha_i) (X_{n_2})_i \end{aligned} \right\}$$

Par cela, il suffit de mettre
la ligne 7 (tirage de α) après la
ligne 9 afin que α soit tiré
séparément pour chaque coordonnée.

- (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 : $r_i \in [0, 1]$ $r_i \in [0, 1]$
 $\forall i \in \{1, \dots, n\}, y_i = a_i + F(b_i - c_i)$ si $(r_i < CR)$ ou $(i = i_0)$, 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) Points communs par rapport à un AG : 5

- * Processus aléatoire de transformation de x en y (similaire à une mutation).
- * Evolution d'une population

Différences par rapport à un AG :

- * Pas d'opération de croisement (2 éléments en générant 2 autres).
 - * Sélection déterministe
- 2) CR : probabilité d'échange des coordonnées de x (hors i_0)
- $CR = 0$: on ne change que la coordonnée i_0 de x
 - ↘ $CR = 1$: on change toutes les coordonnées de x .
- F : Pas dans la direction $(b - c)$ partant de a

* F peut varier dans \mathbb{R} :

Cas particulier : $F=0$: on échange certaines coordonnées de a et x

Remarque : * il s'agit d'une méthode nommée Differential Evolution (DE) développée en parallèle des méthodes vues en cours.

* Il y a élitisme (car $J(y) < J(x)$)
(exemple Scilab d'implémentation)

Ex 4)

6

* Dans un recuit simulé :
si on remplace J par $4J$: la probabilité de garder un élément moins bon décrit fortement car $p' = \exp\left(-\frac{4J(\tilde{x}) - 4J(x)}{T_i}\right) \neq p$

* si on remplace J par $J+3$: il n'y a aucun changement car $p' = \exp\left(-\frac{(J(\tilde{x})+3) - (J(x)+3)}{T_i}\right) = p$

* Dans un algo PSO il n'y a aucun changement dans les 2 cas. En effet, J inter-

à travers la mémoire individuelle et collective du meilleur élément visité.

Cet ordre est inchangé si on remplace J par $4J$ ou $J+3$.

2) Dans le cas d'un AG : on peut retrouver le plus mauvais élément à la génération suivante car la sélection est aléatoire et chaque

élément a une probabilité non nulle d'être tiré au sort

Dans le cas d'un ES : on ne retrouve

^(*) jamais le plus mauvais élément car la sélection est déterministe (sur la famille "enfants" ou "parents + enfants")

(*) cas exceptionnel : si tous les enfants sont plus mauvais que le plus mauvais des parents et sélection "parent + enfant" 7

Ex 6

Différences AG/ES :

- sélection (AG : aléatoire, ES : déterministe)
- importance (voire présence) du croisement
voire nature
- plus d'enfants que de parents ($\lambda > \mu$)

Similitudes AG/ES :

- principes Darwiniens
- évolution aléatoire d'une population
- Initialisation


```

1   $I_j = j \forall j \in \{1, \dots, \lambda\}$ 
2  for  $i = 1$  to  $\lambda$  do
3    for  $j = 1$  to  $\lambda - 1$  do
4      sample  $u \in U(0, 1)$  (uniform random number generator)
5      if  $(\phi(\mathbf{x}_{I_j}) = \phi(\mathbf{x}_{I_{j+1}}) = 0)$  or  $(u < P_f)$  then
6        if  $f(\mathbf{x}_{I_j}) > f(\mathbf{x}_{I_{j+1}})$  then
7          swap( $I_j, I_{j+1}$ )
8        fi
9      else
10     if  $\phi(\mathbf{x}_{I_j}) > \phi(\mathbf{x}_{I_{j+1}})$  then
11       swap( $I_j, I_{j+1}$ )
12     fi
13   fi
14   od
15   if no swap done break fi
od

```

Fig. 2. Stochastic ranking procedure, $P_f = 0.45$.

* Initialisation : $\sigma_0 = Id$ (L1)

* Double boucle ($i = 1 \dots \lambda$; $j = 1, \dots, \lambda - 1$)

on compare $\underbrace{\sigma_k(j)}_{I_j}$ et $\underbrace{\sigma_k(j+1)}_{I_{j+1}}$

* L5 :
 si (I_j et I_{j+1} admissibles) ou ($X = 1$)
 al $X \sim \mathcal{B}(0, P_f)$
 → on classe suivant f (le meilleur au début)

* L9 :
 sinon :
 si (un des deux non admissible) et ($X = 0$)
 → on classe suivant ϕ (le meilleur au début)

On classe ainsi en acceptant de mieux classer certains éléments hors domaine par rapport à des éléments admissibles.

* si $P_f = 0$: plus de "ou"

on va classer d'abord tous les
 admissibles (suivant f) puis tous
 les non admissibles (suivant Φ).
 → classement déterministe donnant
 la primauté au caractère admissible.

« Si $\frac{P}{f} = 1$, on rentre toujours dans
 la 1^{ère} boucle "if": on ne classe que
 suivant f (plus de contrainte):
 → classement non adapté au problème
 (Exemple d'implémentation sur l'exemple
 de la cigarette)

TD 6: Optimisation globale

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 ?

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();
    if(rand()<pc) then
        for j=1:n
            Acrois(2*k-1,j)=alpha*A(n1,j)+(1-alpha)*A(n2,j);
            Acrois(2*k,j)=(1-alpha)*A(n1,j)+alpha*A(n2,j);
        end
    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 ?

Exercice 5.

La méthode du classement stochastique décrite ci-dessous permet de classer λ individus dans un problème d'optimisation avec contrainte.

```

1   $I_j = j \forall j \in \{1, \dots, \lambda\}$ 
2  for  $i = 1$  to  $\lambda$  do
3    for  $j = 1$  to  $\lambda - 1$  do
4      sample  $u \in U(0, 1)$  (uniform random number generator)
5      if  $(\phi(\mathbf{x}_{I_j}) = \phi(\mathbf{x}_{I_{j+1}}) = 0)$  or  $(u < P_f)$  then
6        if  $f(\mathbf{x}_{I_j}) > f(\mathbf{x}_{I_{j+1}})$  then
7          swap( $I_j, I_{j+1}$ )
8        fi
9      else
10       if  $\phi(\mathbf{x}_{I_j}) > \phi(\mathbf{x}_{I_{j+1}})$  then
11         swap( $I_j, I_{j+1}$ )
12       fi
13     fi
14   od
15   if no swap done break fi
od

```

Fig. 2. Stochastic ranking procedure, $P_f = 0.45$.

Dans cet algorithme, f est la fonction coût à minimiser et Φ la pénalisation et P_f est un paramètre.

1. Que se passe t-il quand $P_f = 0$, respectivement 1 ?

2. Ecrire une fonction Scilab de classement aléatoire de λ individus pour une fonction f et une pénalisation Φ données.

Exercice 6.

Donner deux différences et deux similitudes entre une stratégie d'évolution et un algorithme génétique.