Course on Optimization University of Mauritius, January 2018
Laurent DUMAS, Versailles University

TD2: stochastic methods

Exercise 1–

In genetic algorithms, the DE method stochastically explore the global minimum of a function $J : \mathbb{R}^n \to \mathbb{R}$.

DE actually evolves a population of N_{pop} elements (or individuals) with the following algorithm (where $CR \in [0, 1]$ and $F \in [0, 2]$ are two parameters):

(i) Random initialisation of the N_{pop} elements

(ii) From the generation 1 to the generation N_{gen}:

(iii) For each individual $x \in \mathbb{R}^n$:

- Randomly choose three elements a, b and c from the population, distinct from each other and distinct from x.
- Take the random index, i_0 from $\{1, ..., n\}$ and calculate $y = (y_1, ..., y_n)$ as follows:

  $$\forall i \in \{1, ..., n\}, \quad y_i = a_i + F(b_i - c_i) \text{ if } (r_i < CR) \text{ or } (i = i_0), \text{ else } y_i = x_i$$

  where r_i is chosen randomly from the interval $[0, 1]$.
- If $J(y) < J(x)$, substitute x by y in the population.

(iv) End of a generation

1. What are the main common points and what are the main differences of the DE algorithm compared to a genetic algorithm?
2. Interpret the parameters CR and F for the algorithm. What extreme values can they take?
3. The following script proposes an implementation of the DE algorithm in Scilab:

Unfortunately, some lines identified by:

<<<<<<<

have been erased. Reconstitute the corresponding lines.

4. We propose to plot the decreasing history of the best value of the function f according to the number of iterations. Add the missing instructions for this display.

Exercise 2

We propose the following algorithm for the minimization of a function 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('final value obtained for x:')
disp(x)
```

1. Explain the global function of this program and the instructions in lines 5, 7 and 8.
2. What does the parameter α represent in this program?
3. What does the term $1 = log(i + 1)$ represent and why has it been chosen? Propose another possible choice.

Exercise 3 –

A genetic algorithm has for crossing operator the following function:

```
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. What does the variable *pc* represent and what effect does it have?
2. Explain how this algorithm is stochastic and at what level (s) comes the random nature?
3. We are trying to modify the crossing operator to allow a wide range of possible solutions. What change to the previous algorithm would you suggest?