PUCPR

Escola Politecnica

Lab Report

Differential Evolution Algorithms

Dimitrios Dimou

August 16, 2016

Chapter 1

Implementation of a Differential Evolution Algorithm

1.1 Code

```
%% Parameters %%
num_pop = 45; % Population Number
factor = 0.1; % Scaling Factor
cross_rate = 0.2; % Crossover Rate
max_iter = 1000; % Maximum Number of Iterations

Pop = zeros(num_pop, 4);

%% Initialization phase %%

for i = 1 : num_pop
    Pop(i, 1) = 10 + 70 * rand();
    Pop(i, 2) = 10 + 40 * rand();
    Pop(i, 3) = 0.9 + 4.1 * rand();
    Pop(i, 4) = 0.9 + 4.1 * rand();
end

%% Differential Evolution Algorithm %%
iterations = 0;
while iterations < max_iter
    NewPop = Pop;

    % For every member of the population
    for j = 1 : num_pop
        Parent = Pop(j,:);

        % We choose three random members
        X1 = randi([1, num_pop]);
        while (X1 == j)
            X1 = randi([1, num_pop]);
        end
        X2 = randi([1, num_pop]);
        while (X2 == X1 || X2 == j)
            X2 = randi([1, num_pop]);
        end
        X3 = randi([1, num_pop]);
        while (X3 == X1 || X3 == X2 || X3 == j)
            X3 = randi([1, num_pop]);
        end

        Xr1 = Pop(X1,:);
        Xr2 = Pop(X2,:);
        Xr3 = Pop(X3,:);

        % Compute the mutant member
        Vi = Xr1 + factor * (Xr2 - Xr3);
        Ui = zeros(1,4);
```

```
        % Replicate from parent
        for k = 1 : 4
            if rand() <= cross_rate
                if k == 1 || k == 2 && Vi(k) < 10
                    Vi(k) = 10;
                end
                if k == 1 && Vi(k) > 80
                    Vi(k) = 80;
                end
                if k == 2 && Vi(k) > 50
                    Vi(k) = 50;
                end
                if k == 3 || k == 4 && Vi(k) < 0.9
                    Vi(k) = 0.9;
                end
                if k == 3 || k == 4 && Vi(k) > 5
                    Vi(k) = 5;
                end
                Ui(k) = Vi(k);
            else
                Ui(k) = Parent(k);
            end
        end

        % Select parent or child for the new population
        if I_BEAM(Ui) < I_BEAM(Parent)
            Yi = Ui;
        else
            Yi = Parent;
        end

        NewPop(j,:) = Yi;
    end

    iterations = iterations + 1;
    Pop = NewPop;
end

%% Find Best Candidate %%
best = 1;
for i = 2 : num_pop
    if I_BEAM(Pop(best,:)) > I_BEAM(Pop(i,:))
        best = i;
    end
end

Best_Solution = Pop(best,:)
Best_Value = I_BEAM(Pop(best,:))
```

1.2 Results

We run the differential algorithm 3 times for 3 different parameter values and compare its results with the genetic algorithm.

1. Parameters:
   Population number = 25
   Scaling Factor = 0.5
   Crossover Rate = 0.5
   Maximum Iterations = 1000

   (a) 1st run
       Best solution: x = [70.48 36.94 1.00 0.90]
       Best value: z = 135.78

   (b) 2nd run
       Best solution: x = [75.80 35.50 1.11 0.90]
       Best value: z = 146.51

   (c) 3rd run
       Best solution: x = [68.65 34.35 1.77 0.90]
       Best value: z = 180.30

2. Parameters:
   Population number = 35
   Scaling Factor = 1.5
   Crossover Rate = 0.8
   Maximum Iterations = 1000
   (a) 1st run
       Best solution: x = [46.14 46.54 2.17 0.90]
       Best value: z = 180.27
   (b) 2nd run
       Best solution: x = [65.64 40.08 1.09 0.90]
       Best value: z = 142.27
   (c) 3rd run
       Best solution: x = [69.34 38.78 0.98 0.90]
       Best value: z = 136.63
3. Parameters:
   Population number = 45
   Scaling Factor = 0.1
   Crossover Rate = 0.2
   Maximum Iterations = 1000
   (a) 1st run
       Best solution: x = [70.08 35.46 1.13 0.93]
       Best value: z = 143.78
   (b) 2nd run
       Best solution: x = [40.97 28.54 0.98 2.27]
       Best value: z = 166.05
   (c) 3rd run
       Best solution: x = [71.61 29.88 0.91 1.24]
       Best value: z = 137.32
4. Genetic Algorithm
   (a) 1st run
       Best solution: x = [63.06 34.86 0.90 1.10]
       Best value: z = 131.83
   (b) 2nd run
       Best solution: x = [62.13 23.46 0.90 2.02]
       Best value: z = 147.29
   (c) 3rd run
       Best solution: x = [51.95 21.75 0.90 2.65]
       Best value: z = 157.35

And the fmincon Matlab function gave the following results:

1. Starting point: x = [10 10 0.9 0.9]
   Best solution: x = [78.39 34.93 0.90 0.90]
   Best value: z = 131.86
2. Starting point: x = [80 50 5 5]
   Best solution: x = [44.95 17.99 3.13 3.58]
   Best value: z = 247.45