Athens Course
Genetic Algorithms: Report

Stefan Pante
KULeuven University

April 30, 2014

Contents

1 Introduction

Problem: A hypothetical town has buildings at 50 locations on a grid. The coordinates for these locations are given by (i, j) where $i = 1, 2, \cdots, 10$ and $j = 1, 2, \cdots, 5$. A river runs through the town on $j = 3.5$. A fire unit needs to be places somewhere inside the range of the buildings. The average response time should be optimal for all the buildings in the town. A location for the building needs to be calculated. A schematic diagram can be seen in Figure 1.

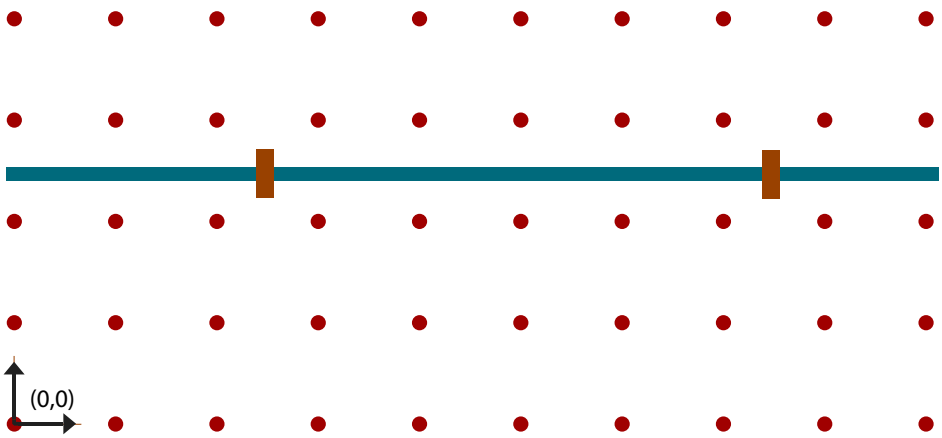

Figure 1: Diagram showing the layout of the town

This report described two approaches to solve this Optimization problem. The first one uses a Genetic Algorithm, the second one uses Particle Swarm Optimization. These solutions to the given problem are described in the following sections. In the Appendix, the output of the program is shown. The general Objective function is described in the section below.

2 Formulation of objective function

2.1 Explanation

Multiple choices where considered for the implementation of the objective function for this problem. One possible solution for the objective function could have been the **Manhattan distance**[1] between the possible location of the Fire Station and all the buildings in the city.

Instead, Since there is no information is available about roads and the size of the buildings, the **Euclidian distance** between the possible locations for the buildings was chosen instead. Some modification is required because of the river running through the town: When the cost between two locations on the same side of the river is calculated, the normal Euclidian distance is used:

$$P_1 = (x_1, y_1), \quad P_2 = (x_2, y_2)$$

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

When the two locations are on opposing sides of the river. The Euclidian distances between the first location and the two bridges and between the two bridges and the second location are calculated. The minimum of these two distance is chosen as the cost between the two locations. B_1, B_2 are the locations of the two bridges.

$$d(P_1, P_2) = \text{min} \begin{cases} d(P_1, B_1) + d(B_1, P_2) \\ d(P_1, B_2) + d(B_2, P_2) \end{cases}$$

[1] http://en.wikipedia.org/wiki/Taxicab_geometry

The total cost of a possible location Q for the fire station is the sum of the distances between the possible location and all the other locations.

$$\text{cost}(Q) = \sum_{P \in \text{locations}} d(P, Q)$$

The implementation of this objective function can be seen in section 2.3

2.2 Objective function graph

In Figure 2, the objective function is plotted for all possible locations of the fire station inside the town. It is obvious from the mesh and contourplot that there is a local maximum around the line of the river $j = 3.5$. The code to generate this graph is listed in section 2.3.

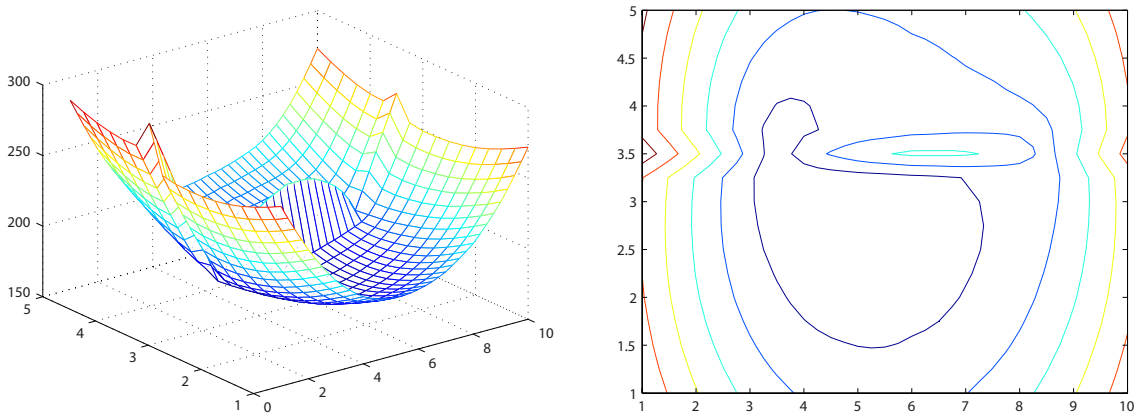

Figure 2: A mesh and a contourplot showing the objective function.

2.3 Matlab code

The following code is used to calculated all the possible locations inside the town

```
% Calculates the grid given by the assignment

function [ Grid ] = calculateGrid( )

% Preallocating vector is more efficient than appending values.
Grid = zeros(50,2);

k = 1;

for i=1:10
    for j=1:5
        Grid(k, :) = [i j];
        k = k + 1;
    end
end
end
```

Listing 1: Function to calculate all the possible locations

The following code is used to calculate the distance between two locations.

```
% Output is stored in D, two points are given, between which the minimal
% distance is calculated.
function [ D ] = euclidianDistance( P1, P2)
% the locations of the bridges
B1 = [3.5 3.5];
B2 = [8.5 3.5];

% the Y coordinate of the location of the river
R = 3.5;

if ( (P1(2) < R && P2(2) < R) || (P1(2) > R && P2(2) > R) )
    D = norm(P2 - P1);
else
    % The distance between the two points when the first bridge is used
    D1 = norm(B1 - P1) + norm(B1 - P2);

    % The distance between the two points when the second bridge is used
    D2 = norm(B2 - P1) + norm(B2 - P2);

    % We select the minimum of the two points.
    D = min(D1, D2);
end


end
```

Listing 2: Function to calculate the distance between two points

The following code calculates the total cost of a given position to all the other positions in the town.

```
% Calculates the cost of the location P to all the Sites on the map
% and stores it inside C.
function [ C ] = cost( P, Sites )
% Initialize the cost at zero.
C = 0;

% copy sites into locations.
L = Sites;

% Iterate over all the locations
for i = 1:length(L)
    % add the cost of each location to the total cost
    C = C + euclidianDistance(P, L(i ,:) ) ;
end

end
```

Listing 3: Function to calculate the cost of a particular point in the town

The following script was used to generate the plots shown in Figure 2.

```
% This script plots the cost function

Grid = calculateGrid();

[X,Y] = meshgrid(1:0.25:10,1:0.25:5);

[a,b] = size(X);
Costs = zeros(a,b);
for i= 1:a
    for j = 1:b
        selectedPoint = [X(i,j), Y(i,j)];
        Costs(i,j) = cost(selectedPoint, Grid);
    end
end

figure(1);
mesh(X, Y, Costs);
figure(2);
contour(X, Y, Costs);
```

Listing 4: Script to generate the plots

3 Specification of search space

The search space is a collection of discrete points given by:

$$(i,j) \quad i = i = 1, 2, \cdots, 10 \quad \text{and} j = 1, 2, \cdots, 5$$

4 Genetic Algorithm

4.1 Explanation

Since the search space is defined by a collection of discrete points, a two dimensional point is chosen as a chromosome in this genetic algorithm. The genes are the coordinates of the points. The algorithm described in section **??** functions as follows.

1. An initial population with `n` chromosomes is generated by generating random discrete points within the town. with $1 \leq i \leq 10$ and $1 \leq j \leq 5$.
2. The algorithm sorts the population.
3. the `nS` best results are selected as possible parents for the next generation.
4. `ChildSize` Children are generated via crossover by selecting two parents with a random exponential system.
5. There is a change `mut` that mutation occurs in the genes of the child.
6. the children are stored for the next generation.
7. The possible parents are also kept for the next generation.
8. Some of the remaining chromosomes are selected to fill the rest of the generation. These are the `Survivors`.
9. Random mutations can occur in the `Survivors` with a change of `mut`.
10. This process repeats itself for `it` iterations.

4.2 Matlab code

```
function [ ] = geneticAlgorithm(PopulationSize, PSize, ChildSize, mut, it )
% runs the genetic algorithm

% Preallocate for performance
Generation = zeros(PopulationSize,3);
% Get all the possible building sites
Sites = calculateGrid() ;

% initialize the population with n chromosomes
for i = 1:PopulationSize
    rndm1 = randi(10);
    rndm2 = randi(5);
    c = cost([rndm1 rndm2], Sites);
    Generation(i ,:) = [rndm1, rndm2, c];
end

for i= 1: it
    fprintf ('==========================================␣Iteration␣%d␣
        ==========================================␣\n', i);
    Generation = sortrows(Generation,3);
    disp('Current␣Generation:');
    disp(Generation');
    disp('Best␣of␣Current␣Generation:');
    disp(Generation(1,:))
    Parents = Generation(1:PSize, :);
    disp('Selected␣Possible␣Parents:');
    disp(Parents');

    kids = zeros(ChildSize, 3);

    % Create childSize children from the possible parents.
    for j=1:ChildSize
        % Used to select the two parents;
        R1 = floor(mod( exprnd(PSize/4), PSize -1) +1);
        R2 = floor(mod( exprnd(PSize/4), PSize -1) +1);

        % Make sure that the parents are two different chromosomes
        while R1 == R2
            R2 = floor(mod( exprnd(PSize/4), PSize -1) +1);
        end

        % index of the two genes to use with both parents
        Index1 = randi(2);
        Index2 = 2 - Index1 + 1;

        % Calculate if an mutation occurs
        Mutation1 = mutation(mut);
        Mutation2 = mutation(mut);

        % The child is generated
        value1 = Parents(R1, Index1) + Mutation1;
        value2 = Parents(R2, Index2) + Mutation1;
        kids(j , Index1) = value1;
        kids(j , Index2) = value2;
        kids(j , 3) = cost([value1 value2], Sites);
    end

    disp('Created␣Children:');
    disp(kids') ;

    % Select all the Chromosomes except the possible parents
    Survivors = Generation(PSize + 1, :);

    % let some survive and mutate them.
    NumSurv = PopulationSize - PSize - ChildSize;

    %Preallocate for performance
    Surv = zeros(NumSurv, 3);

    for j = 1: NumSurv
        [x y ] = size(Survivors);
```

```
        Index = randi(x);
        Survivor = Survivors(Index, :);

        Mutation1 = mutation(mut);
        Mutation2 = mutation(mut);

        value1 = Survivor(1,1) + Mutation1;
        value2 = Survivor(1,2) + Mutation2;
        Survivor(1,1) = value1;
        Survivor(1,2) = value2;
        Survivor(1,3) = cost([value1 value2], Sites);

        Surv(j, :) = Survivor;
    end
    disp('Selected and mutated Survivors:');
    disp(Surv');
    Generation = [Parents ; kids ; Surv];
     fprintf ('
          ==========================================================================================
          \n\n', i)

end




end
```

Listing 5: The function for the genetic algorithm

4.3 Parameters, Population Size and Convergence

After several executions of the genetic algorithm, the following parameters seemed well suited to find a solution quickly.

Population Size = 10 A population size of ten seemed to fit the purpose of the algorithm well. (`PopulationSize` in the genetic algorithm in the code).

Parent Size = 5 The number of possible parents to be selected and stored for the next generation.

Child Size = 3 The number of children generated by the possible parents.

Mutation Chance = 0.1 The change that a gene mutates on the transition of one generation to the next.

iterations = 15 The number of generations that needs to be generated before the algorithm stops executing.

These parameters caused convergence to appear rapidly. The Optimum often appeared in early generations. Almost the entire population is always converged to the optimum at the end of 15 iterations.

The output of this algorithm can be found in appendix A

5 Particle Swarm Optimization

A Genetic Algorithm: Matlab output

Listing 6: The output from matlab for the genetic algorithm

```
>> geneticAlgorithm(10,5,3,0.1,15)
========================================= Iteration 1 =========================================
Current Generation:
    5.0000    7.0000    4.0000    5.0000    5.0000    3.0000    9.0000    9.0000   10.0000    1.0000
    3.0000    3.0000    4.0000    4.0000    5.0000    1.0000    4.0000    1.0000    2.0000    3.0000
  168.5228  178.2436  179.1698  186.8445  200.3717  214.1816  216.0697  238.4096  254.4226  258.8336

Best of  Current Generation:
    5.0000    3.0000  168.5228

Selected Possible Parents:
    5.0000    7.0000    4.0000    5.0000    5.0000
    3.0000    3.0000    4.0000    4.0000    5.0000
  168.5228  178.2436  179.1698  186.8445  200.3717

Created Children:
    5.0000    5.0000    6.0000
    4.0000    4.0000    2.0000
  201.1111  186.8445  174.1430

Selected and mutated Survivors:
    3.0000    3.0000
    1.0000    1.0000
  214.1816  214.1816

=================================================================================================

========================================= Iteration 2 =========================================
Current Generation:
    5.0000    6.0000    7.0000    4.0000    5.0000    5.0000    5.0000    5.0000    3.0000    3.0000
    3.0000    2.0000    3.0000    4.0000    4.0000    4.0000    5.0000    4.0000    1.0000    1.0000
  168.5228  174.1430  178.2436  179.1698  186.8445  186.8445  200.3717  201.1111  214.1816  214.1816

Best of  Current Generation:
    5.0000    3.0000  168.5228

Selected Possible Parents:
    5.0000    6.0000    7.0000    4.0000    5.0000
    3.0000    2.0000    3.0000    4.0000    4.0000
  168.5228  174.1430  178.2436  179.1698  186.8445

Created Children:
    7.0000    4.0000    5.0000
    3.0000    3.0000    4.0000
  178.2436  190.3247  186.8445

Selected and mutated Survivors:
    5.0000    5.0000
    3.0000    4.0000
  168.5228  186.8445

=================================================================================================

========================================= Iteration 3 =========================================
Current Generation:
    5.0000    5.0000    6.0000    7.0000    7.0000    4.0000    5.0000    5.0000    5.0000    4.0000
    3.0000    3.0000    2.0000    3.0000    3.0000    4.0000    4.0000    4.0000    4.0000    3.0000
  168.5228  168.5228  174.1430  178.2436  178.2436  179.1698  186.8445  186.8445  186.8445  190.3247

Best of  Current Generation:
    5.0000    3.0000  168.5228

Selected Possible Parents:
    5.0000    5.0000    6.0000    7.0000    7.0000
    3.0000    3.0000    2.0000    3.0000    3.0000
  168.5228  168.5228  174.1430  178.2436  178.2436

Created Children:
    5.0000    5.0000    5.0000
    3.0000    3.0000    3.0000
  168.5228  214.7243  214.7243
```

Selected and mutated Survivors:
```
    4.0000    4.0000
    3.0000    4.0000
  167.3930  179.1698
```

==========

========== Iteration 4 ==========
Current Generation:
```
    4.0000    5.0000    5.0000    5.0000    6.0000    7.0000    7.0000    4.0000    5.0000    5.0000
    3.0000    3.0000    3.0000    3.0000    2.0000    3.0000    3.0000    4.0000    3.0000    3.0000
  167.3930  168.5228  168.5228  168.5228  174.1430  178.2436  178.2436  179.1698  214.7243  214.7243
```

Best of Current Generation:
```
    4.0000    3.0000  167.3930
```

Selected Possible Parents:
```
    4.0000    5.0000    5.0000    5.0000    6.0000
    3.0000    3.0000    3.0000    3.0000    2.0000
  167.3930  168.5228  168.5228  168.5228  174.1430
```

Created Children:
```
    5.0000    6.0000    5.0000
    3.0000    4.0000    3.0000
  214.7243  193.2794  214.7243
```

Selected and mutated Survivors:
```
    7.0000    7.0000
    3.0000    3.0000
  178.2436  178.2436
```

==========

========== Iteration 5 ==========
Current Generation:
```
    4.0000    5.0000    5.0000    5.0000    6.0000    7.0000    7.0000    6.0000    5.0000    5.0000
    3.0000    3.0000    3.0000    3.0000    2.0000    3.0000    3.0000    4.0000    3.0000    3.0000
  167.3930  168.5228  168.5228  168.5228  174.1430  178.2436  178.2436  193.2794  214.7243  214.7243
```

Best of Current Generation:
```
    4.0000    3.0000  167.3930
```

Selected Possible Parents:
```
    4.0000    5.0000    5.0000    5.0000    6.0000
    3.0000    3.0000    3.0000    3.0000    2.0000
  167.3930  168.5228  168.5228  168.5228  174.1430
```

Created Children:
```
    5.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  168.5228  167.3930  167.3930
```

Selected and mutated Survivors:
```
    7.0000    7.0000
    3.0000    3.0000
  178.2436  178.2436
```

==========

========== Iteration 6 ==========
Current Generation:
```
    4.0000    4.0000    4.0000    5.0000    5.0000    5.0000    5.0000    6.0000    7.0000    7.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    2.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  168.5228  168.5228  168.5228  168.5228  174.1430  178.2436  178.2436
```

Best of Current Generation:
```
    4.0000    3.0000  167.3930
```

Selected Possible Parents:
```
    4.0000    4.0000    4.0000    5.0000    5.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  168.5228  168.5228
```

```
Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  190.3247  167.3930  190.3247

Selected and mutated Survivors:
    6.0000    5.0000
    3.0000    3.0000
  172.5229  168.5228

==========================================================================================

========================================== Iteration 7 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    5.0000    5.0000    5.0000    6.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  168.5228  168.5228  168.5228  172.5229  190.3247  190.3247

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    5.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  168.5228

Created Children:
    3.0000    4.0000    4.0000
    2.0000    3.0000    3.0000
  216.8416  190.3247  190.3247

Selected and mutated Survivors:
    5.0000    6.0000
    3.0000    3.0000
  168.5228  172.5229

==========================================================================================

========================================== Iteration 8 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    5.0000    5.0000    6.0000    4.0000    4.0000    3.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    2.0000
  167.3930  167.3930  167.3930  167.3930  168.5228  168.5228  172.5229  190.3247  190.3247  216.8416

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    5.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  168.5228

Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930

Selected and mutated Survivors:
    5.0000    5.0000
    3.0000    3.0000
  168.5228  168.5228

==========================================================================================

========================================== Iteration 9 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    5.0000    5.0000    5.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  168.5228  168.5228  168.5228

Best of  Current Generation:
    4.0000    3.0000  167.3930
```

```
Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  167.3930

Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  167.3930  167.3930  190.3247

Selected and mutated Survivors:
    4.0000    4.0000
    3.0000    3.0000
  167.3930  167.3930

================================================================================================

========================================== Iteration 10 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  190.3247

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  167.3930

Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  190.3247  190.3247  167.3930

Selected and mutated Survivors:
    4.0000    5.0000
    3.0000    3.0000
  167.3930  168.5228

================================================================================================

========================================== Iteration 11 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    5.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  168.5228  190.3247  190.3247

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930  167.3930  167.3930  167.3930  167.3930

Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  190.3247  190.3247  190.3247

Selected and mutated Survivors:
    4.0000    4.0000
    3.0000    3.0000
  167.3930  167.3930

================================================================================================

========================================== Iteration 12 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
```

```
  167.3930 167.3930 167.3930 167.3930 167.3930 167.3930 167.3930 190.3247 190.3247 190.3247

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930 167.3930 167.3930 167.3930 167.3930

Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  190.3247 190.3247 167.3930

Selected and mutated Survivors:
    4.0000    4.0000
    3.0000    4.0000
  167.3930 179.1698

================================================================================================

========================================== Iteration 13 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    4.0000    3.0000    3.0000
  167.3930 167.3930 167.3930 167.3930 167.3930 167.3930 167.3930 179.1698 190.3247 190.3247

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930 167.3930 167.3930 167.3930 167.3930

Created Children:
    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000
  190.3247 190.3247 167.3930

Selected and mutated Survivors:
    4.0000    5.0000
    3.0000    3.0000
  167.3930 168.5228

================================================================================================

========================================== Iteration 14 ==========================================
Current Generation:
    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    5.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930 167.3930 167.3930 167.3930 167.3930 167.3930 167.3930 168.5228 190.3247 190.3247

Best of  Current Generation:
    4.0000    3.0000  167.3930

Selected Possible Parents:
    4.0000    4.0000    4.0000    4.0000    4.0000
    3.0000    3.0000    3.0000    3.0000    3.0000
  167.3930 167.3930 167.3930 167.3930 167.3930

Created Children:
    4.0000    5.0000    4.0000
    3.0000    4.0000    3.0000
  167.3930 186.8445 167.3930

Selected and mutated Survivors:
    4.0000    4.0000
    4.0000    3.0000
  179.1698 167.3930

================================================================================================
```

```
========================================== Iteration 15 ==========================================
Current Generation:
     4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    4.0000    5.0000
     3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    4.0000    4.0000
   167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  167.3930  179.1698  186.8445

Best of  Current Generation:
     4.0000    3.0000  167.3930

Selected Possible Parents:
     4.0000    4.0000    4.0000    4.0000    4.0000
     3.0000    3.0000    3.0000    3.0000    3.0000
   167.3930  167.3930  167.3930  167.3930  167.3930

Created Children:
     4.0000    4.0000    4.0000
     3.0000    3.0000    3.0000
   190.3247  190.3247  167.3930

Selected and mutated Survivors:
     4.0000    4.0000
     4.0000    3.0000
   179.1698  167.3930

=================================================================================================
```

B Particle Swarm Optimization: Matlab Output