**Graph Partitioning Assignment**

**Artificial Intelligence**

**COMP3008L**

Batch : **Computer Science Batch 06(CS 06)**

Group Name : **Code Chefs**

|  |  |
| --- | --- |
| Group Members : |  |
| Tharindu Perera | **16211200** |
| Ravindu Athapattu | **16211138** |
| L W Kawmali | **16211217** |
| E M D Oshada Vishal Ekanayaka | **16211150** |
| L V O Fernando | **16211238** |
| A K J S Athukorala | **16211133** |

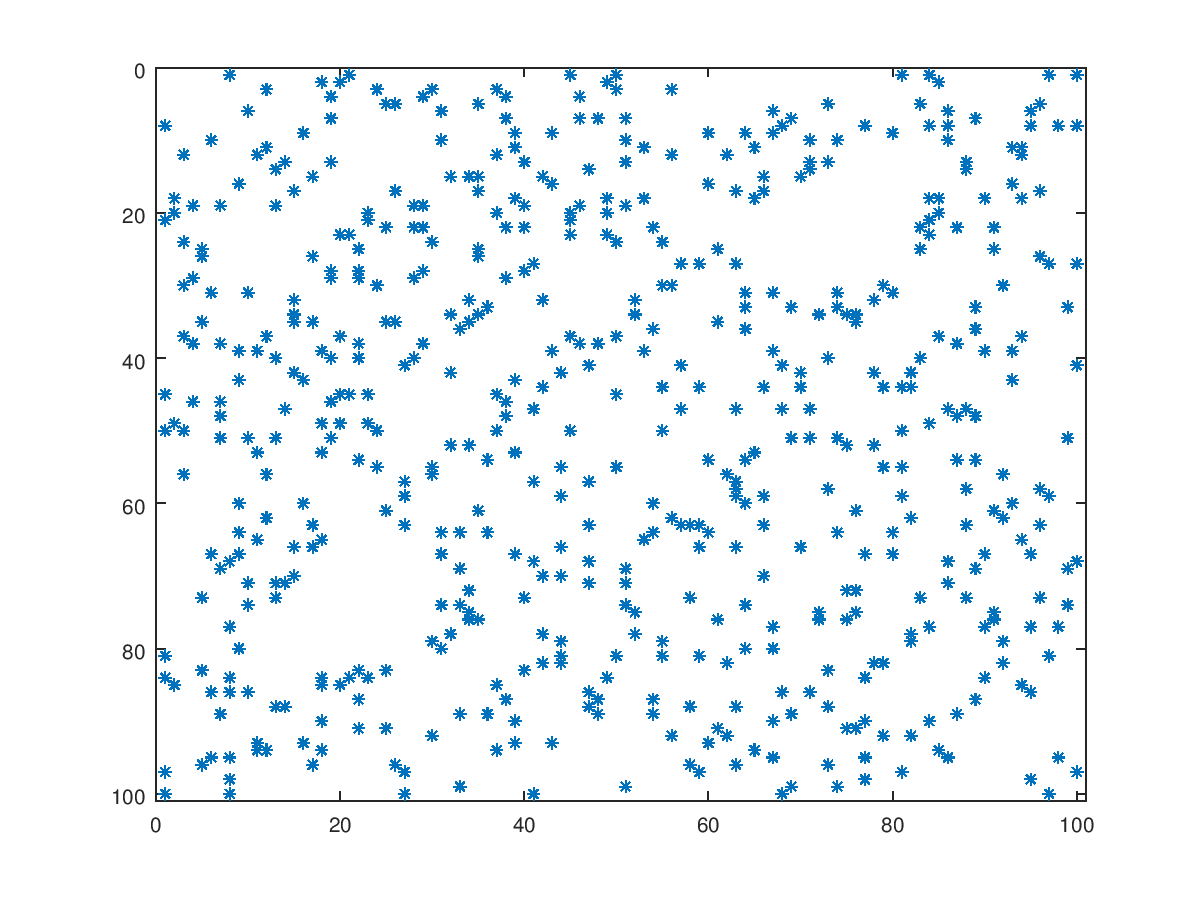
1. Simulated Annealing

* Step1 –

Select a graph of size N (ex: N = 100)

Populate Compressed Column Sparse matrix (A) by using the graph size N (rows = 100, cols = 100). (Ex: [A,t,x,y] = makePlanarSparse(N); )

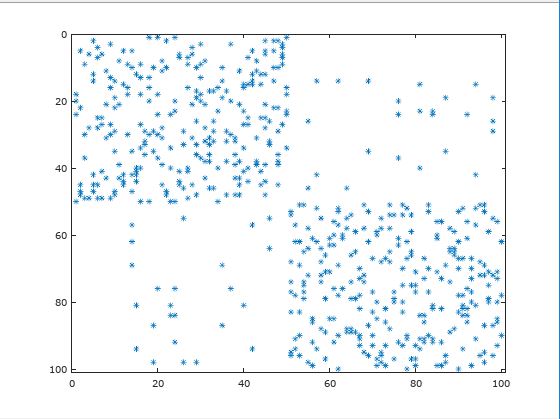
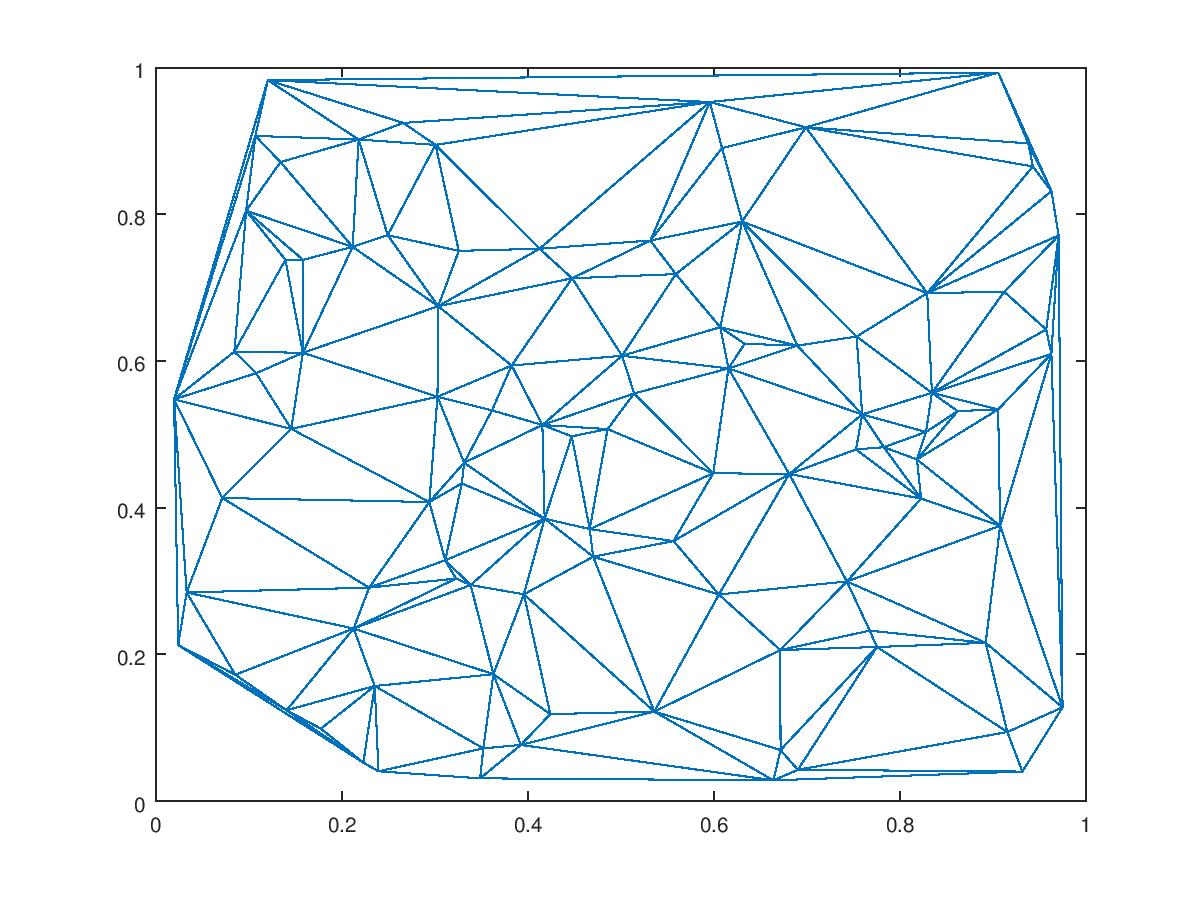
Plot the sparsity pattern by using spy(A). Population has spread scattered over the graph.



2-dimesional plot of attribute A and attribute [x,y] can be generated through gplot(A,[x,y]);

* Step 2 –

Graph is partitioned into two clusters using the inbuilt simulated annealing algorithm.

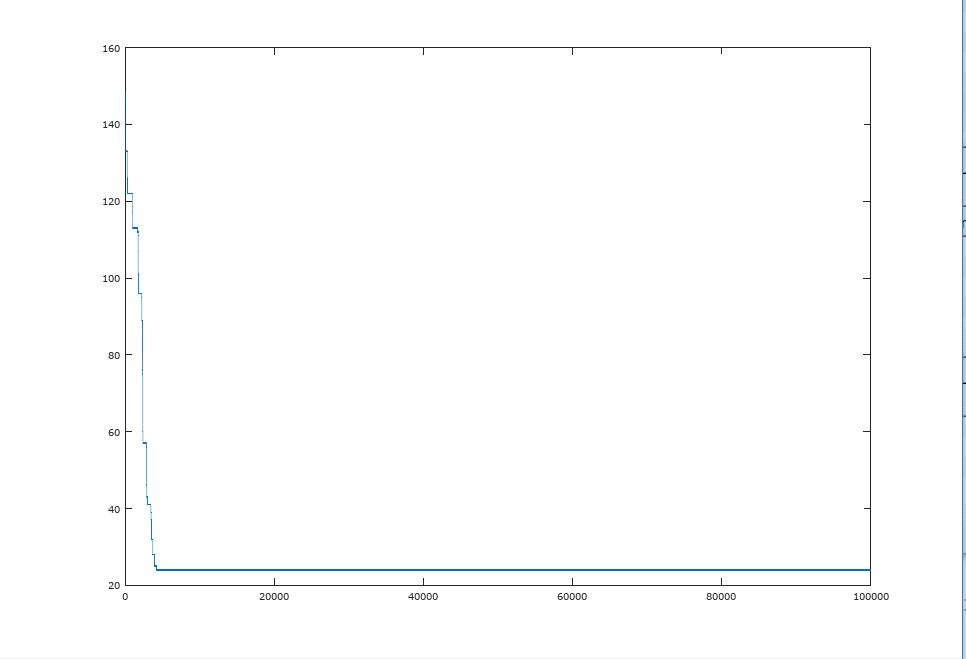


* 1. Examining the impact of the Parameters
     1. Number of Generations

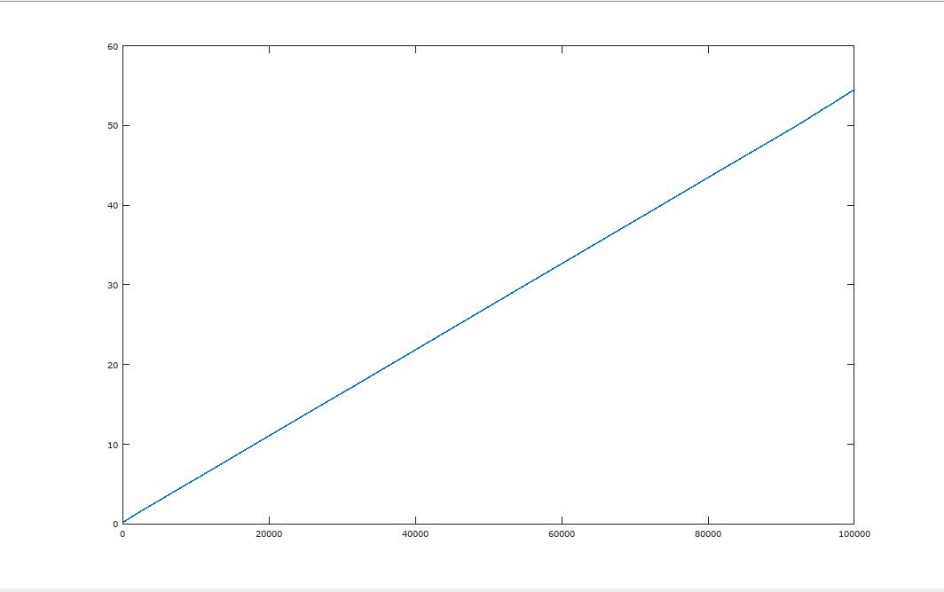
Simulated annealing is written to be run total 100000 generations and quality in each generation is written to text file.

Data in the text file will be loaded as a single numeric matrix with the name of the variable get from the name of the file.

Below plot shows, the different quality obtained after each generation.

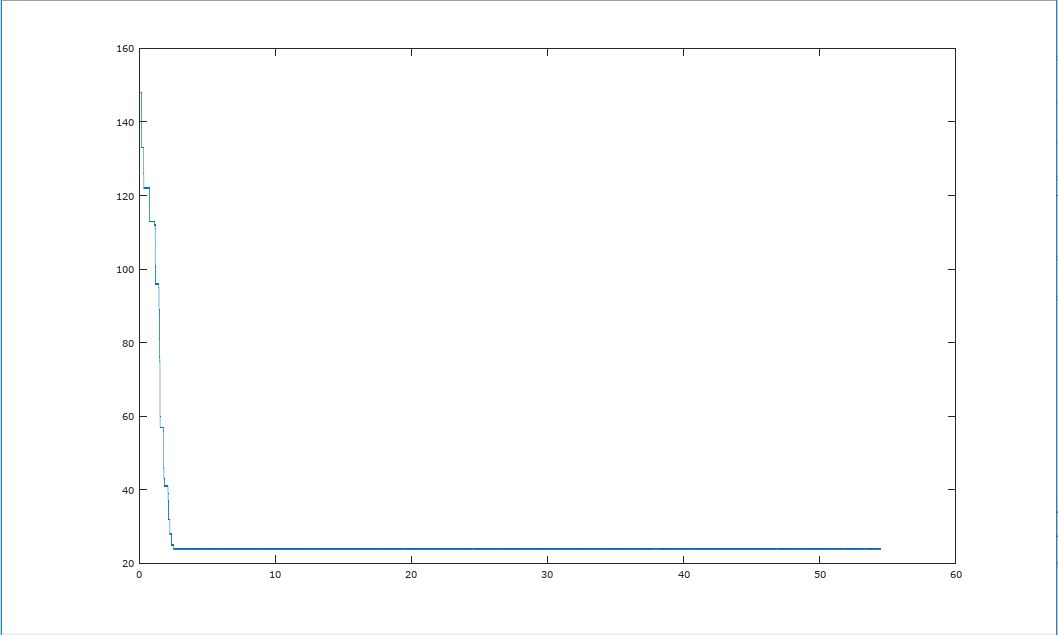


This plot shows the time that elapsed by the end of each generation.



This plot shows the quality against the time.

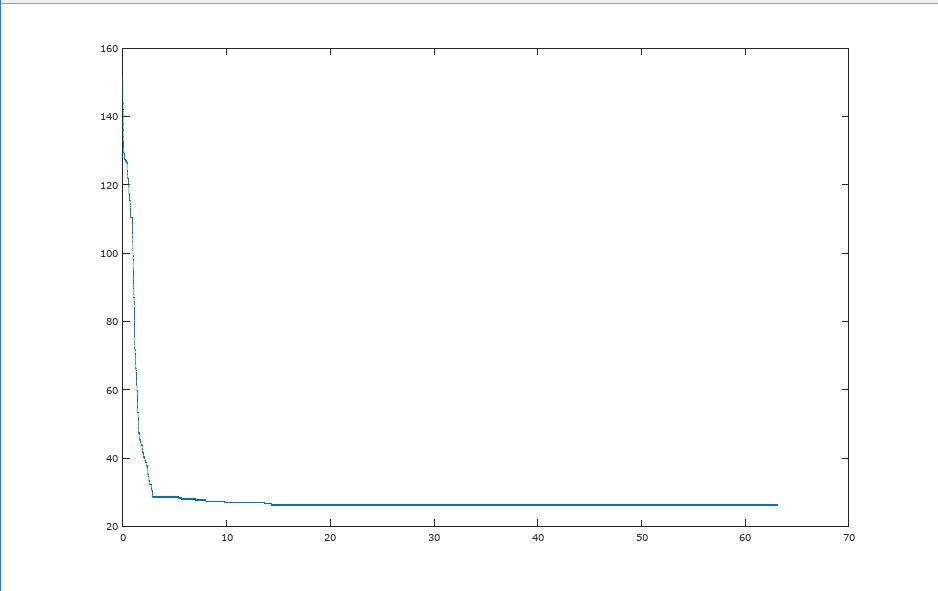
Here the quality is dropped in the first 10seconds and then it becomes nearly stabled.



* 1. Dealing with randomness

We can get a smooth graph when we get the average of the above two plots

(When more iterations are executed to get the average more, we can get smooth graphs)

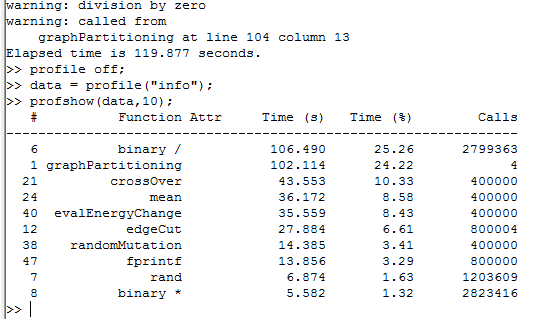
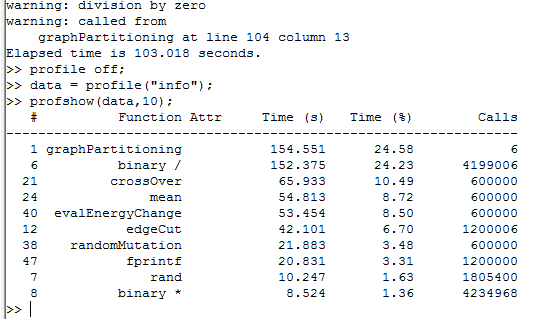


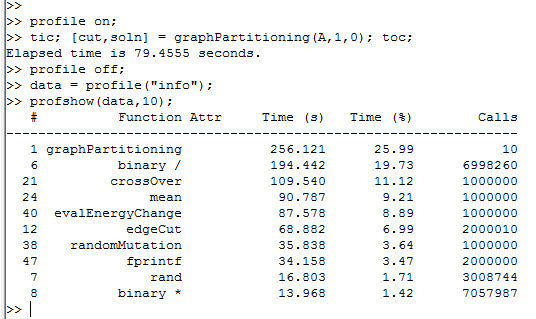
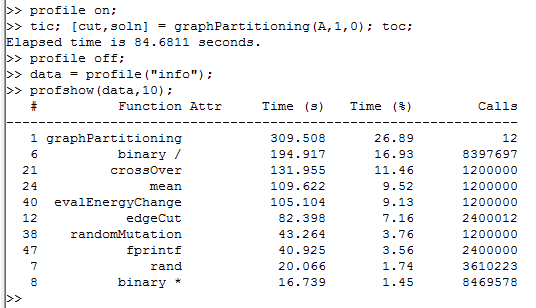
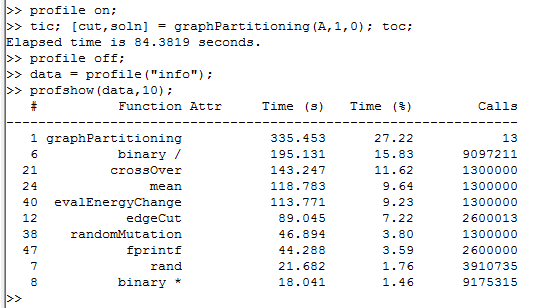
* + 1. Effect of *α*

We can use a profiler to exam the time is taken to execute in changing the value of α.

Using a profiler, we can get information about where this time is spent.

Time elapsed results after changing α values in range of 0.1 – 0.9 randomly

* α = 0.1
  + Time elapsed = 119.877
* α = 0.3
  + Time elapsed = 103.018

* α = 0.6
  + Time elapsed = 79.4555
* α = 0.8
  + Time elapsed = 84.6811
* α = 0.9
  + Time elapsed = 84.3819

α Effect Summarization

|  |  |
| --- | --- |
| α Value | Process Time (in Sec :) |
| 0.1 | 119.877 |
| 0.3 | 103.018 |
| 0.5 | 102.248 |
| 0.6 | 79.4555 |
| 0.8 | 84.6811 |
| 0.9 | 84.3819 |

According to above result we recommend 0.6 as a suitable α value.

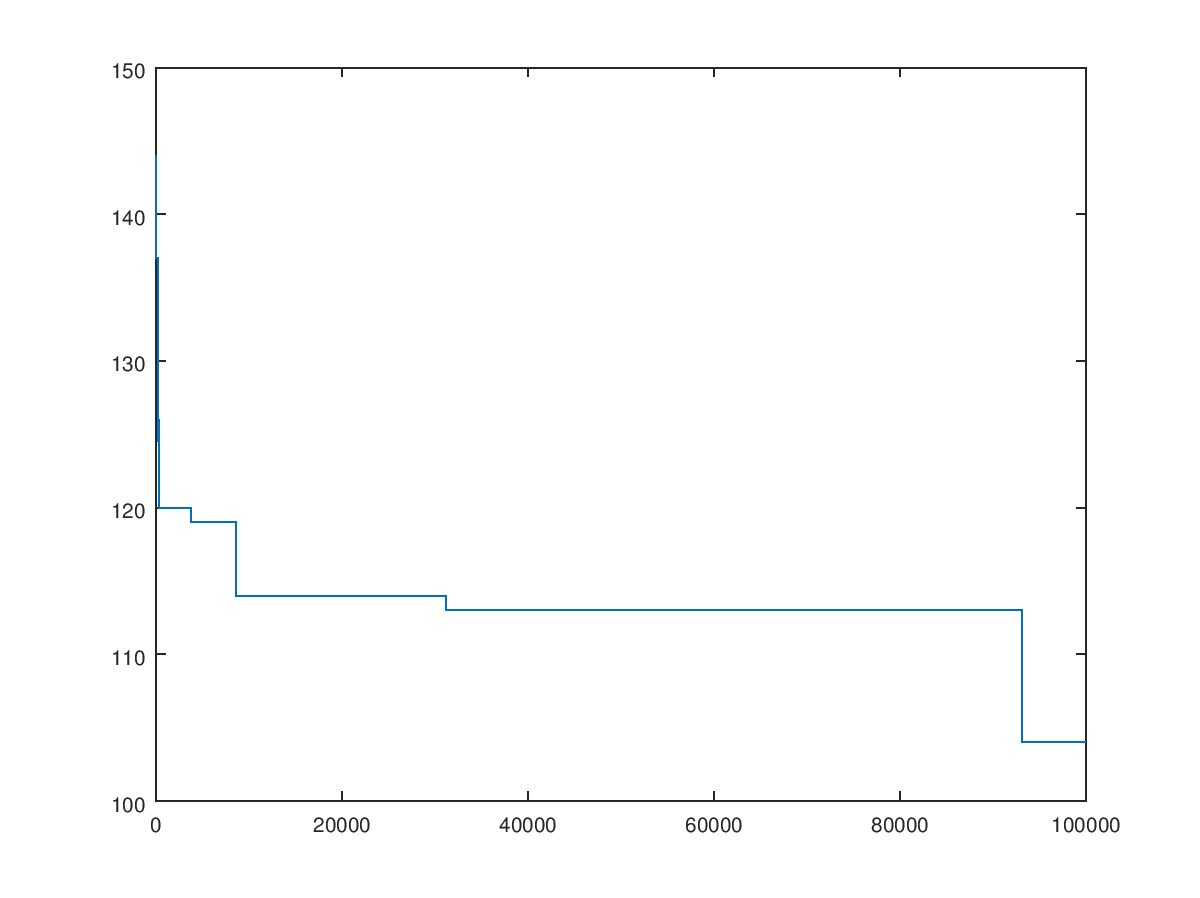
Yes

1. α < 0.5

It takes a longer time to execute

1. . α > 0.99

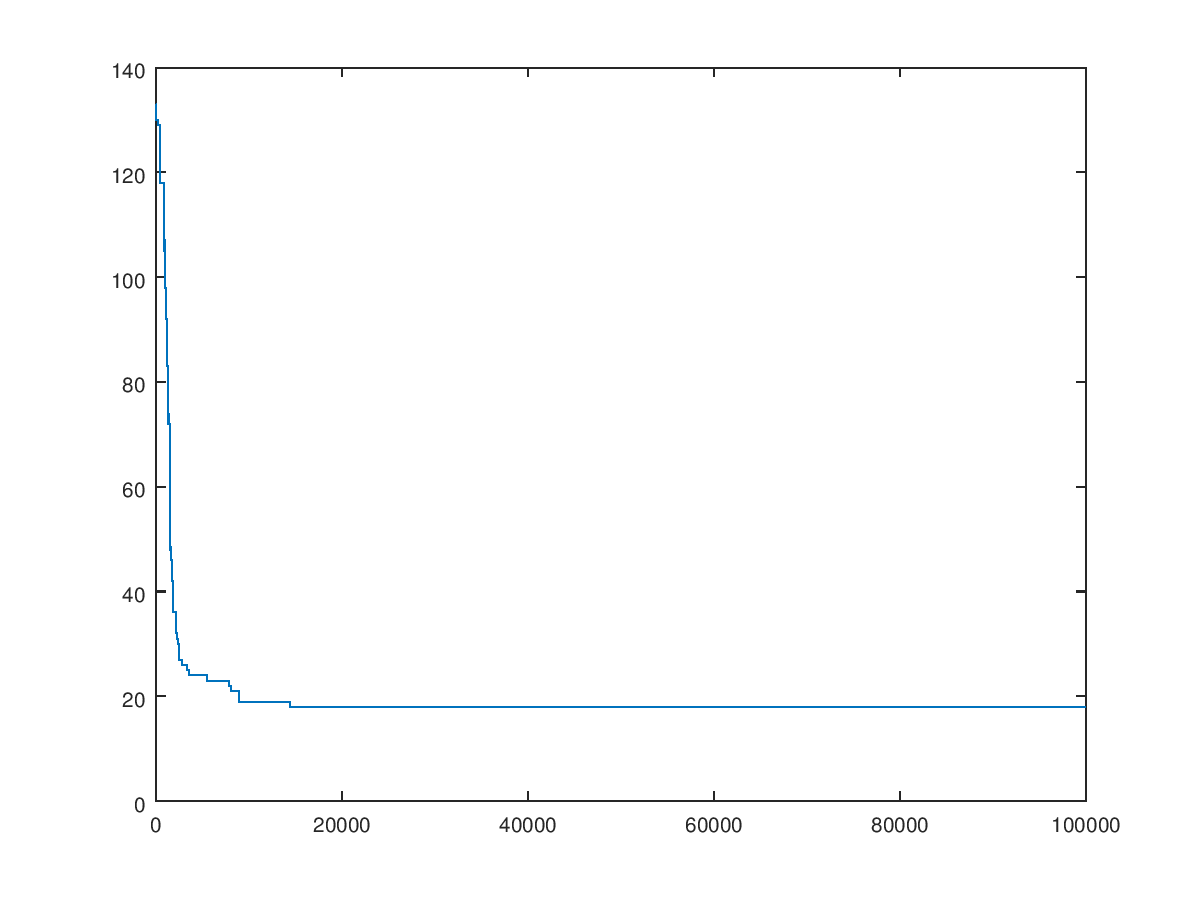
Density and the smoothness of the graph getting low. Curves are not accurate than alpha =0.9 (graph shows α = 1.0)



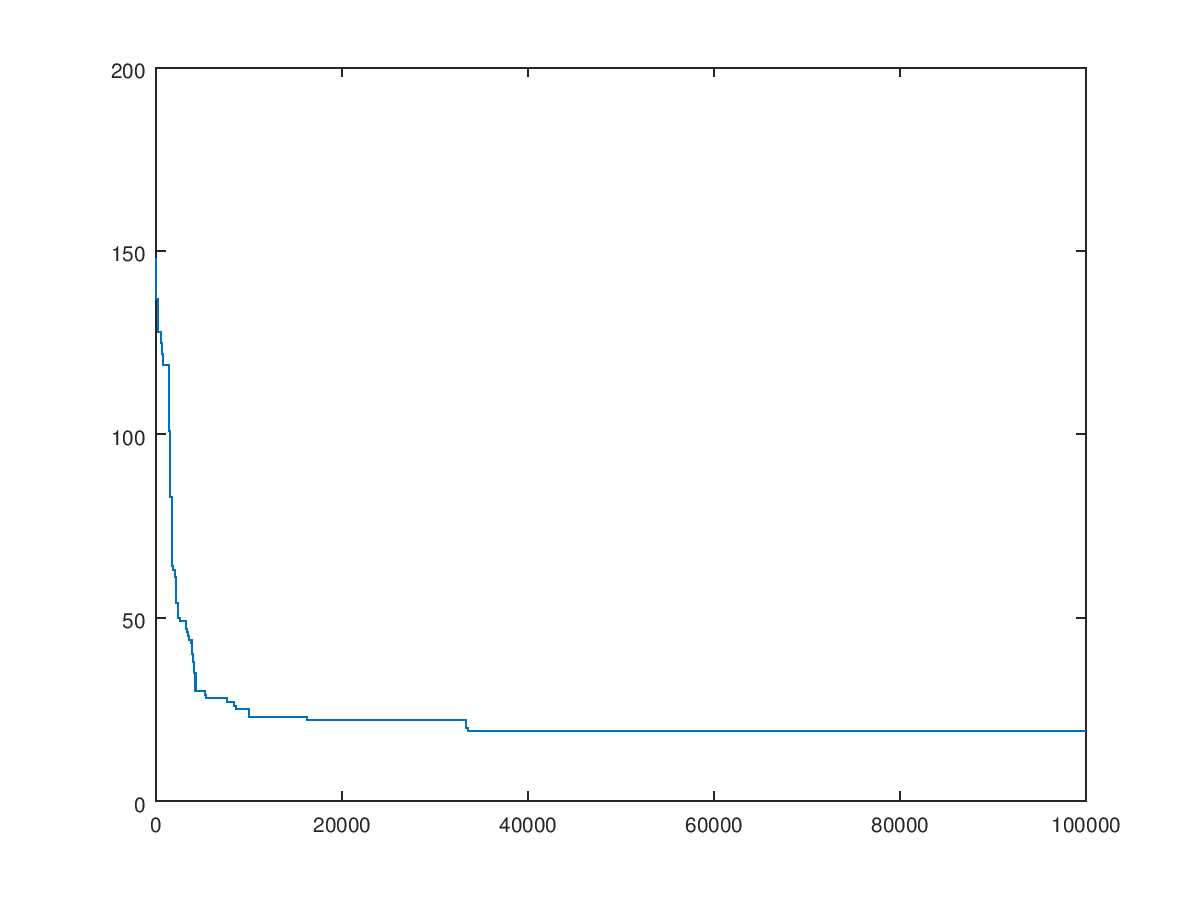
+

* + 1. Effect of

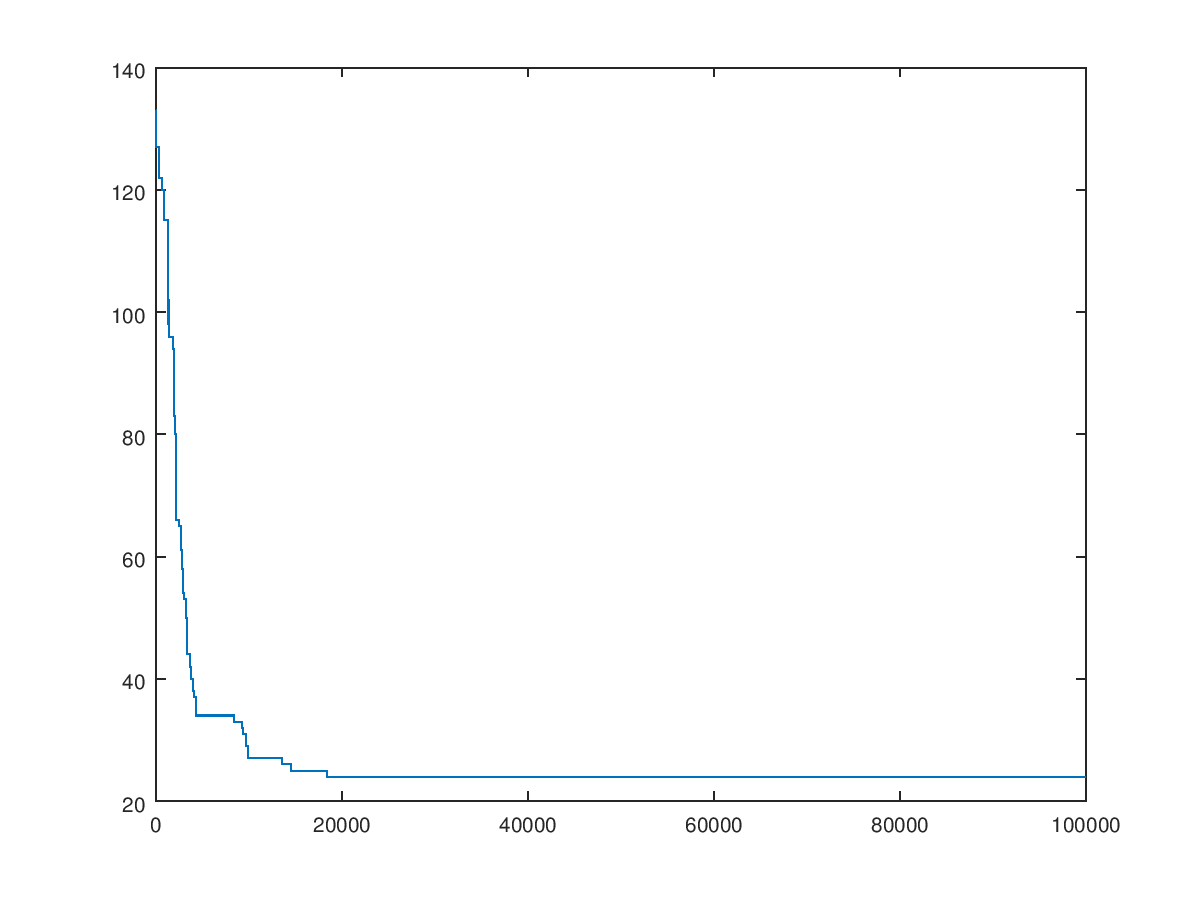
*MarkovChainLength*

Time taken to be executed in Different proportion values and Quality plot according to each value:

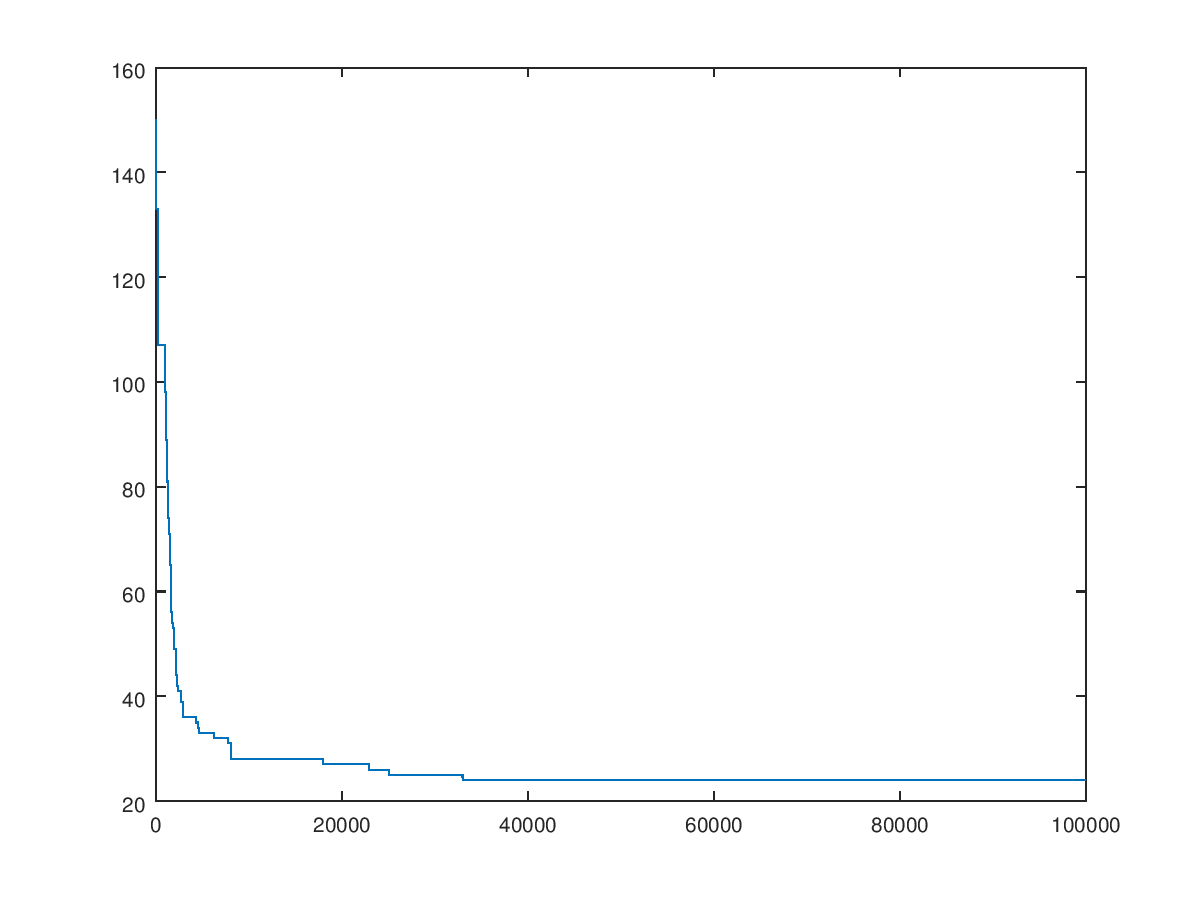
* 0.05 - 80.9564



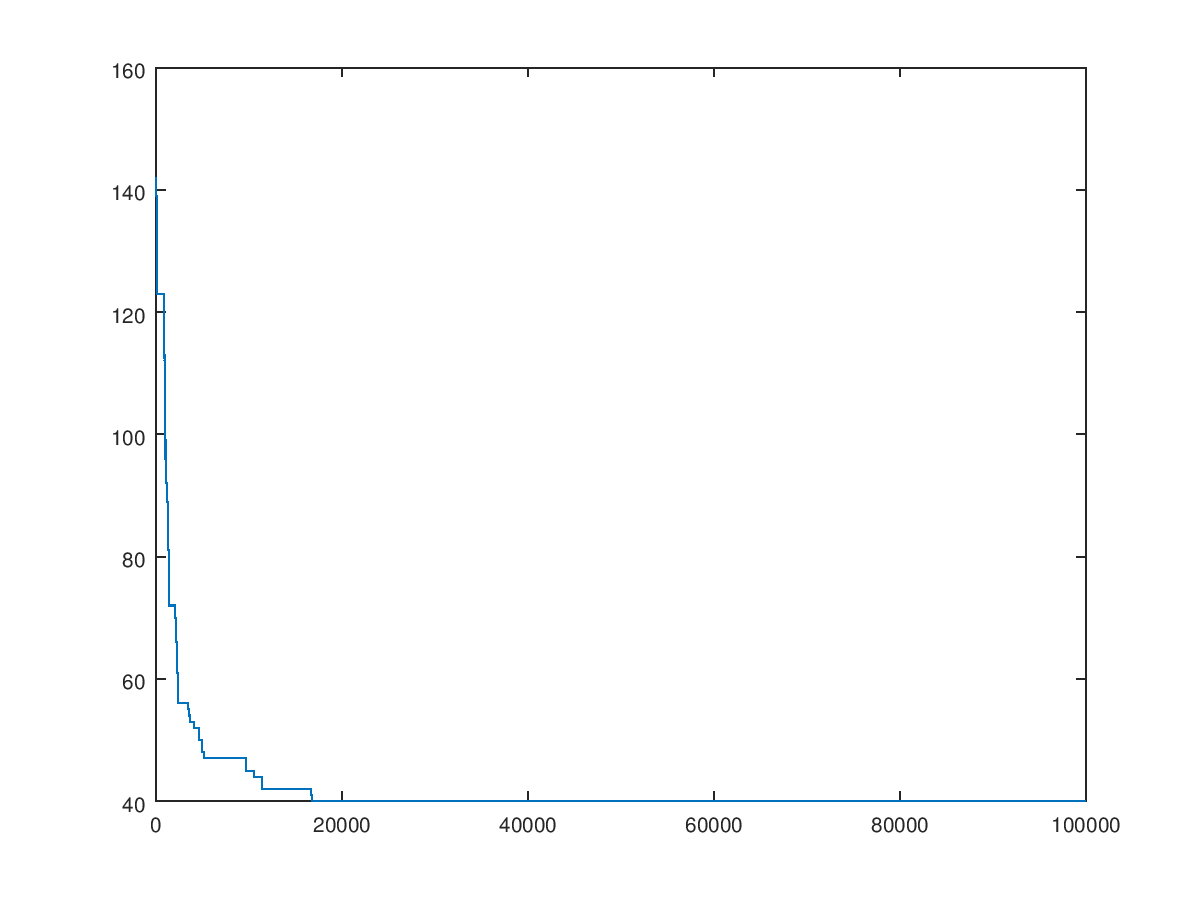
* 0.1 - 72.1247



* 0.2 – 70.1305



* 0.3 – 71.0568



* 0.4 – 70.8743

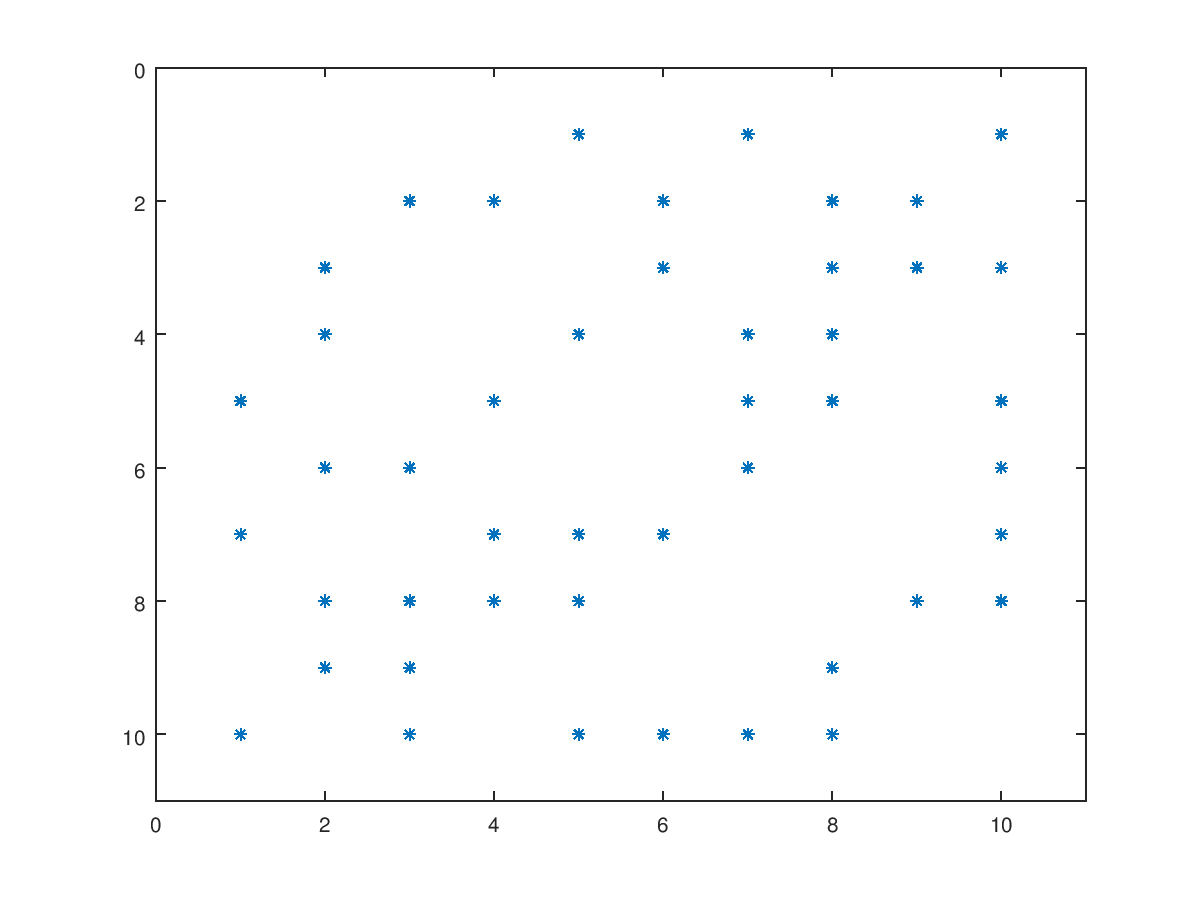
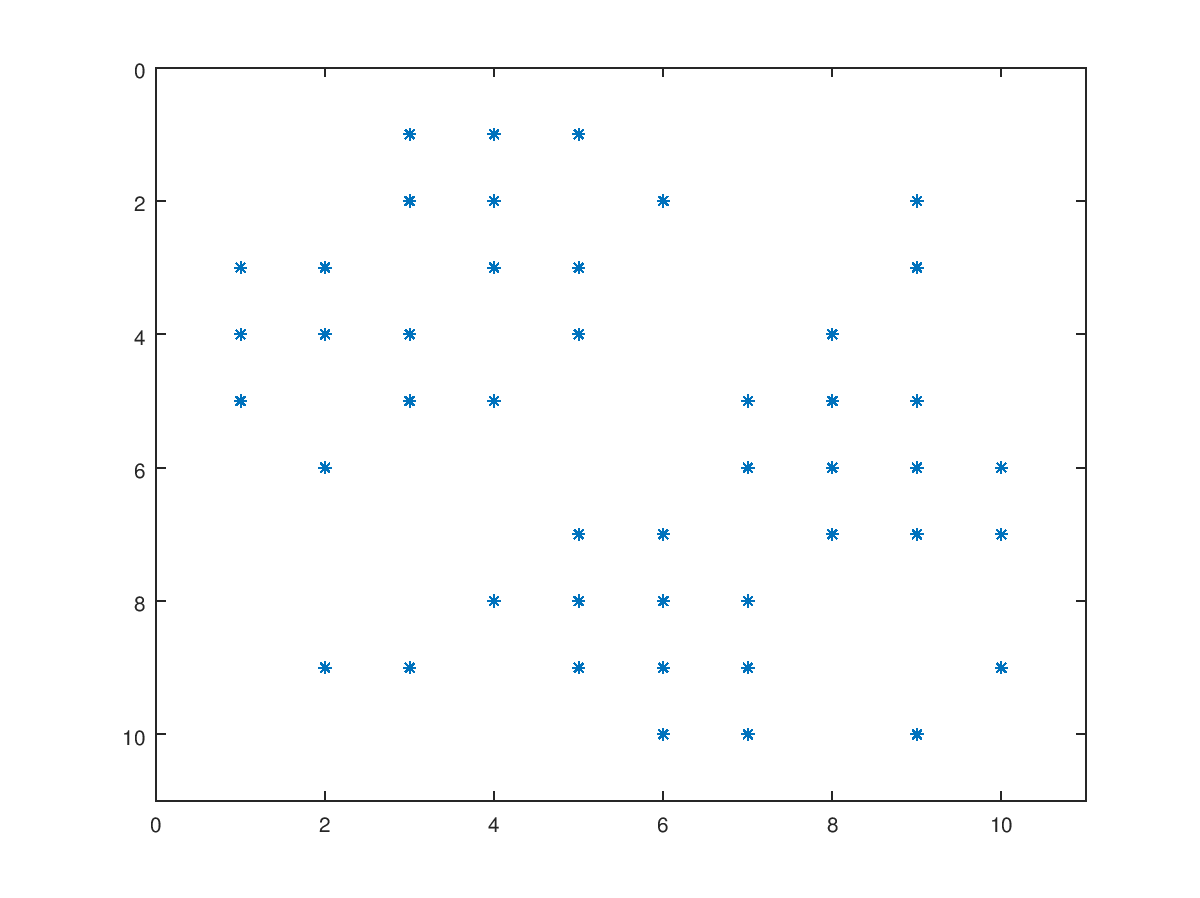
1. Best proportion for the Markov Chain length 0.2

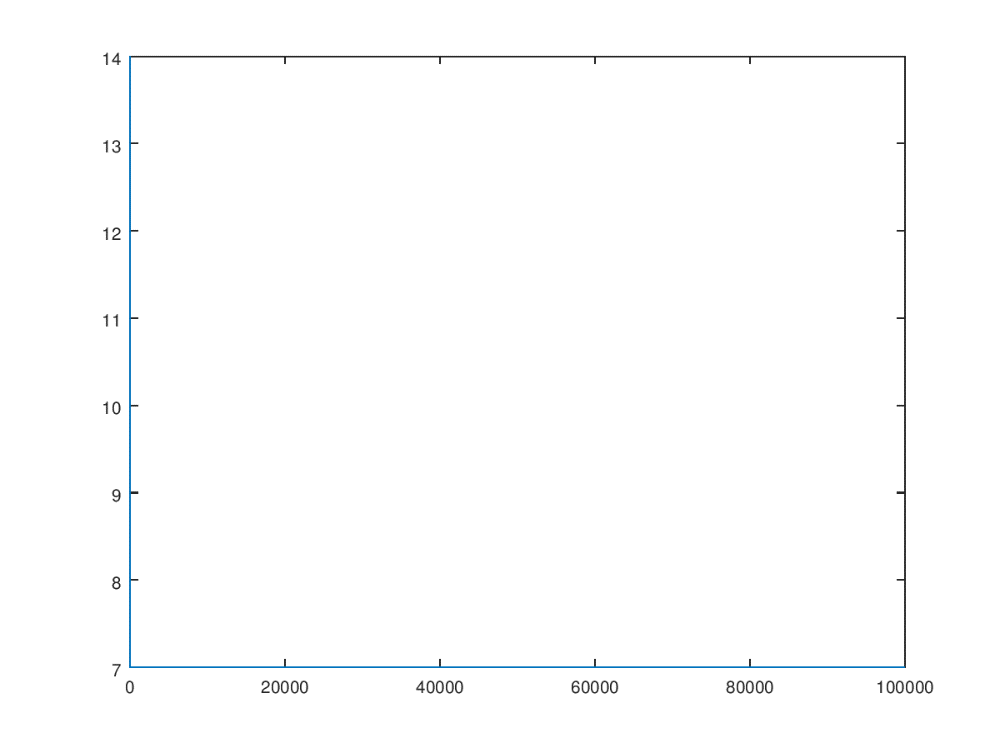
* + 1. Effect of *N*

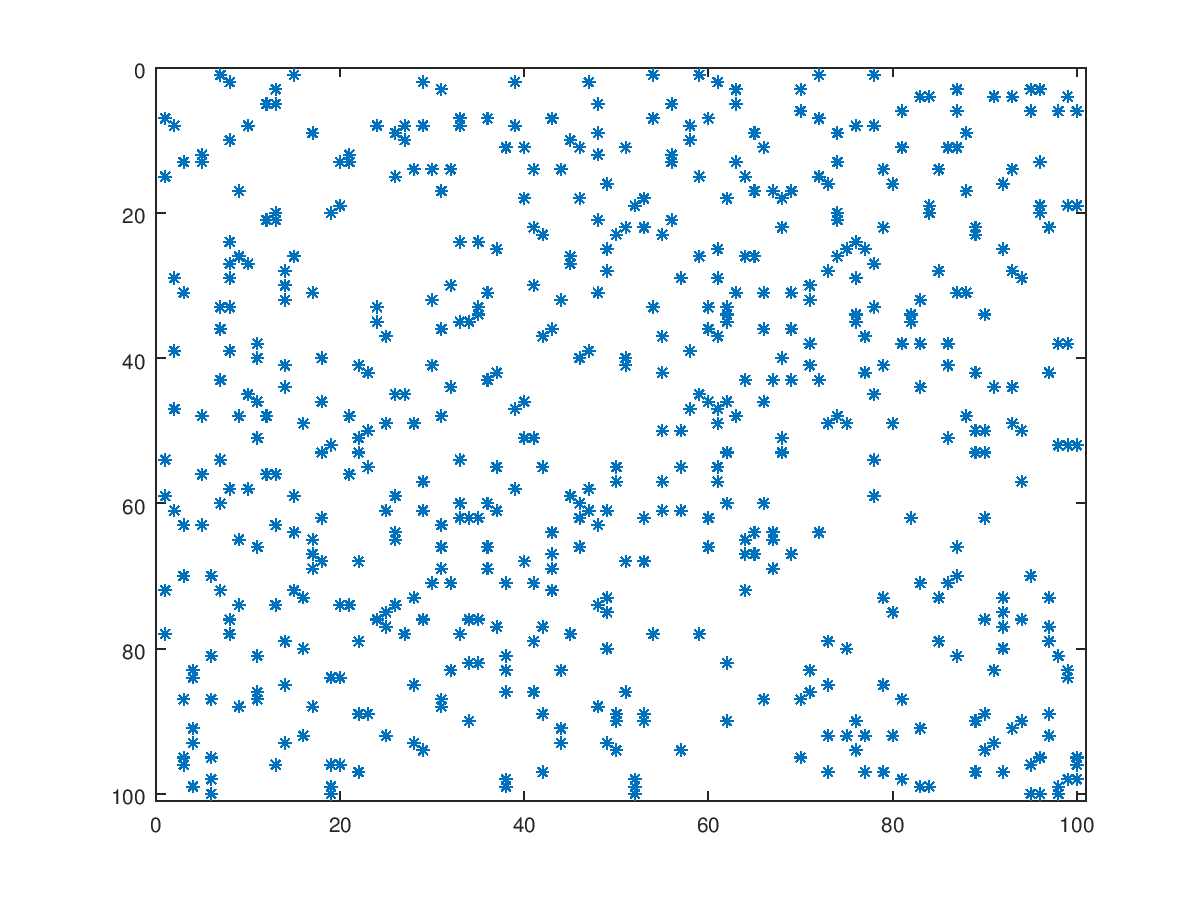
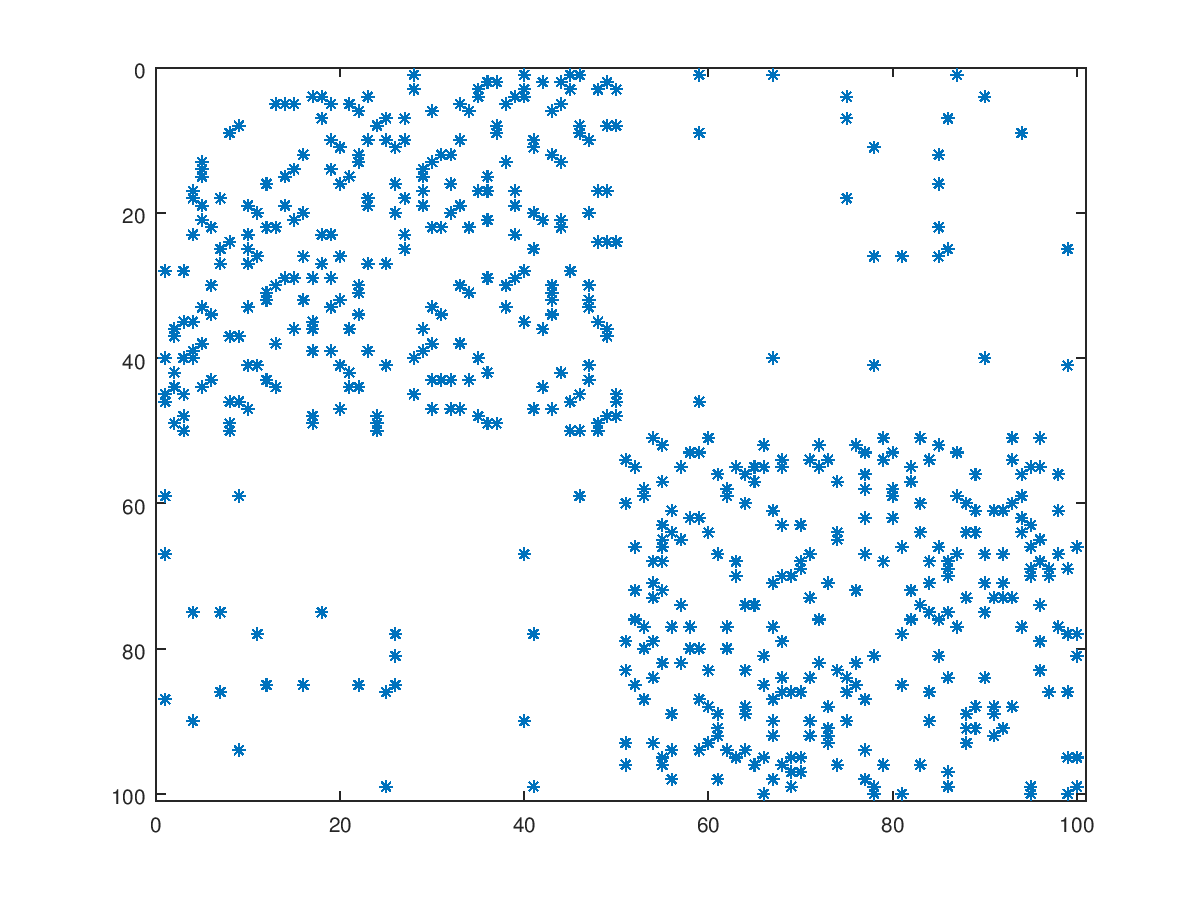
N depends on the speed and memory of the computer. When N is getting larger the speed of the process gets low.

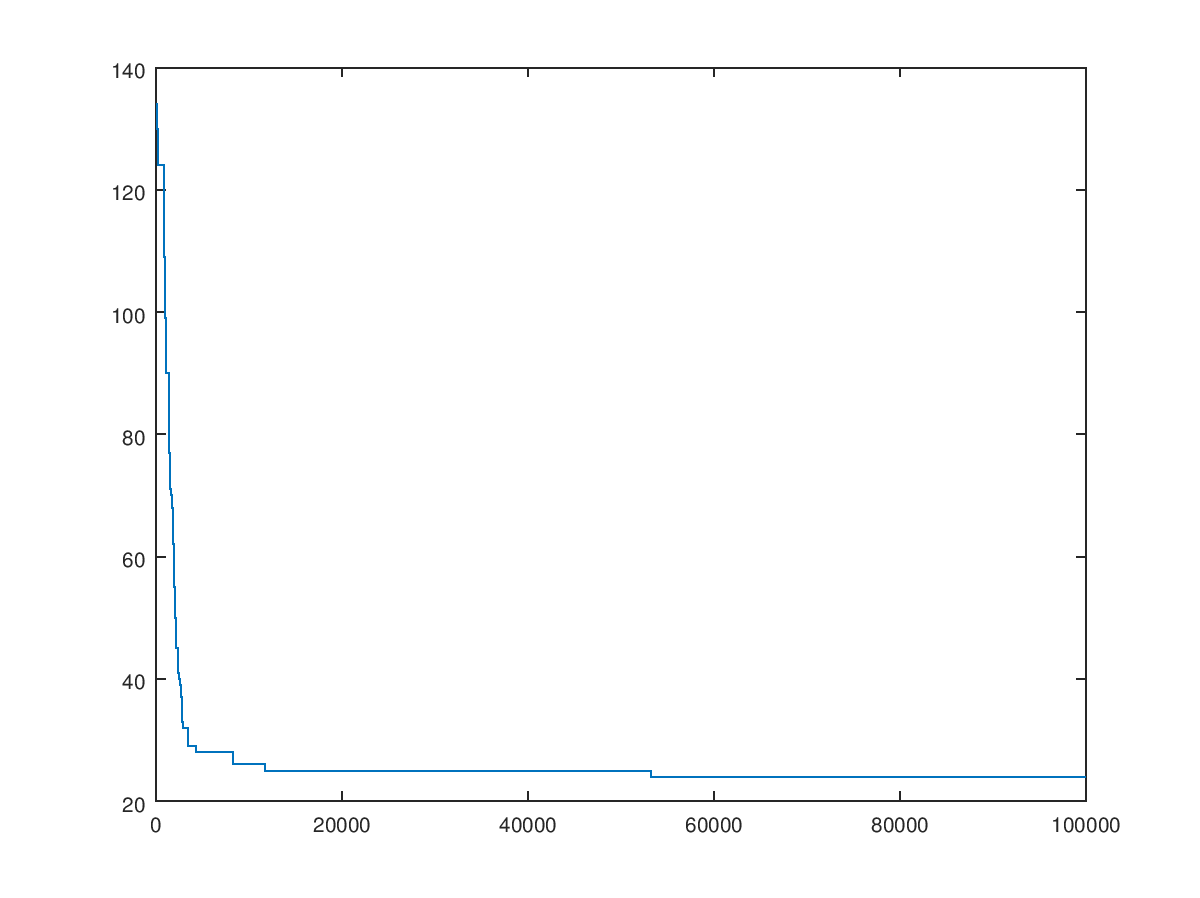
We consider N in 5 different values: {10, 100, 1000, 10000}

* + - N = 10;
* Time = 63.597Sec

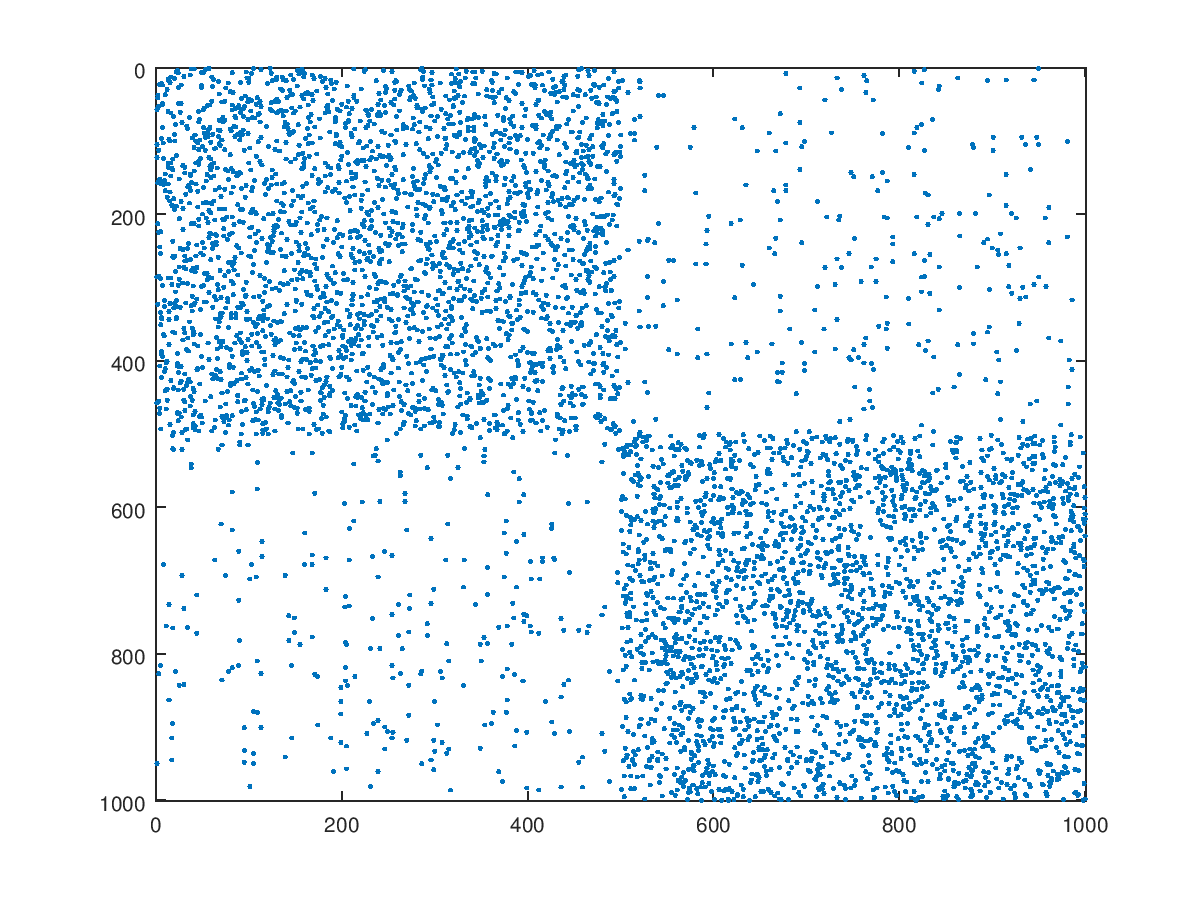


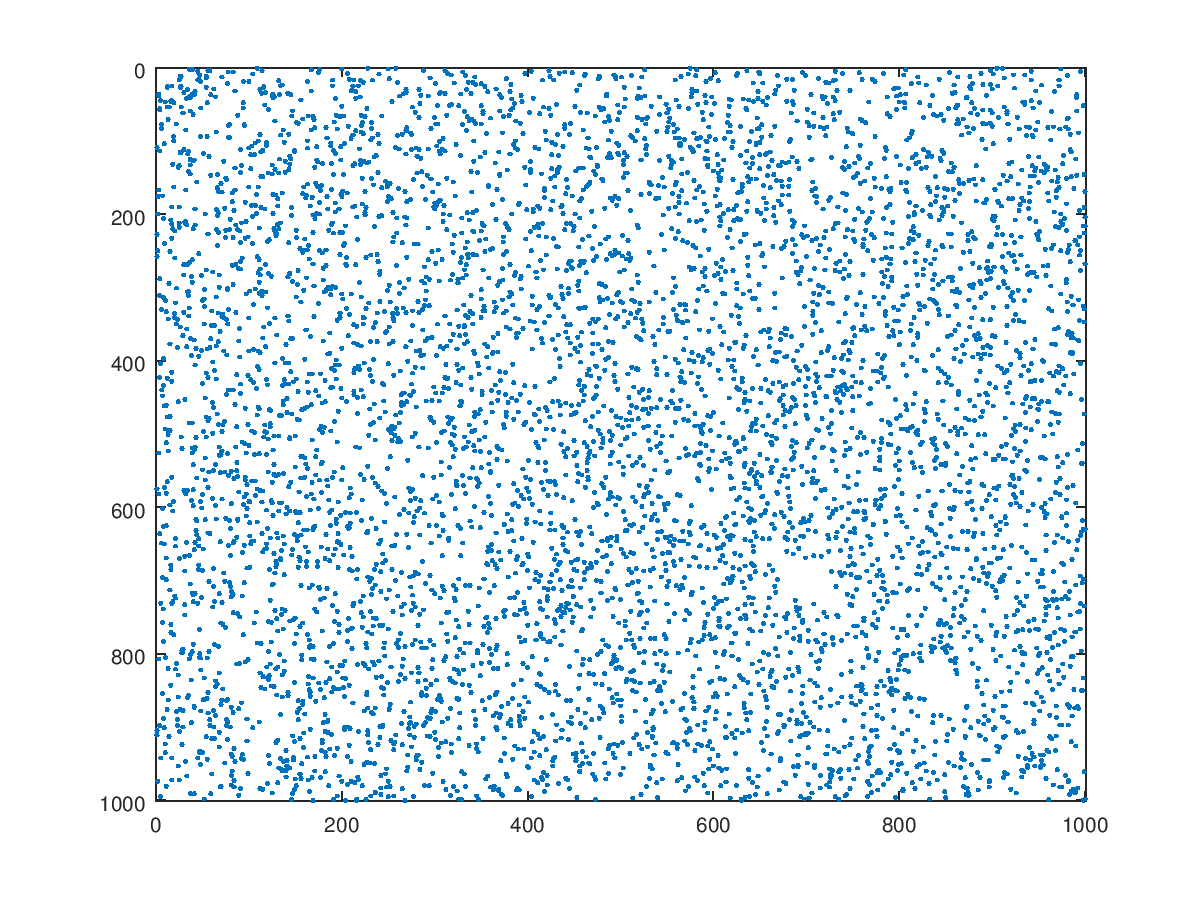


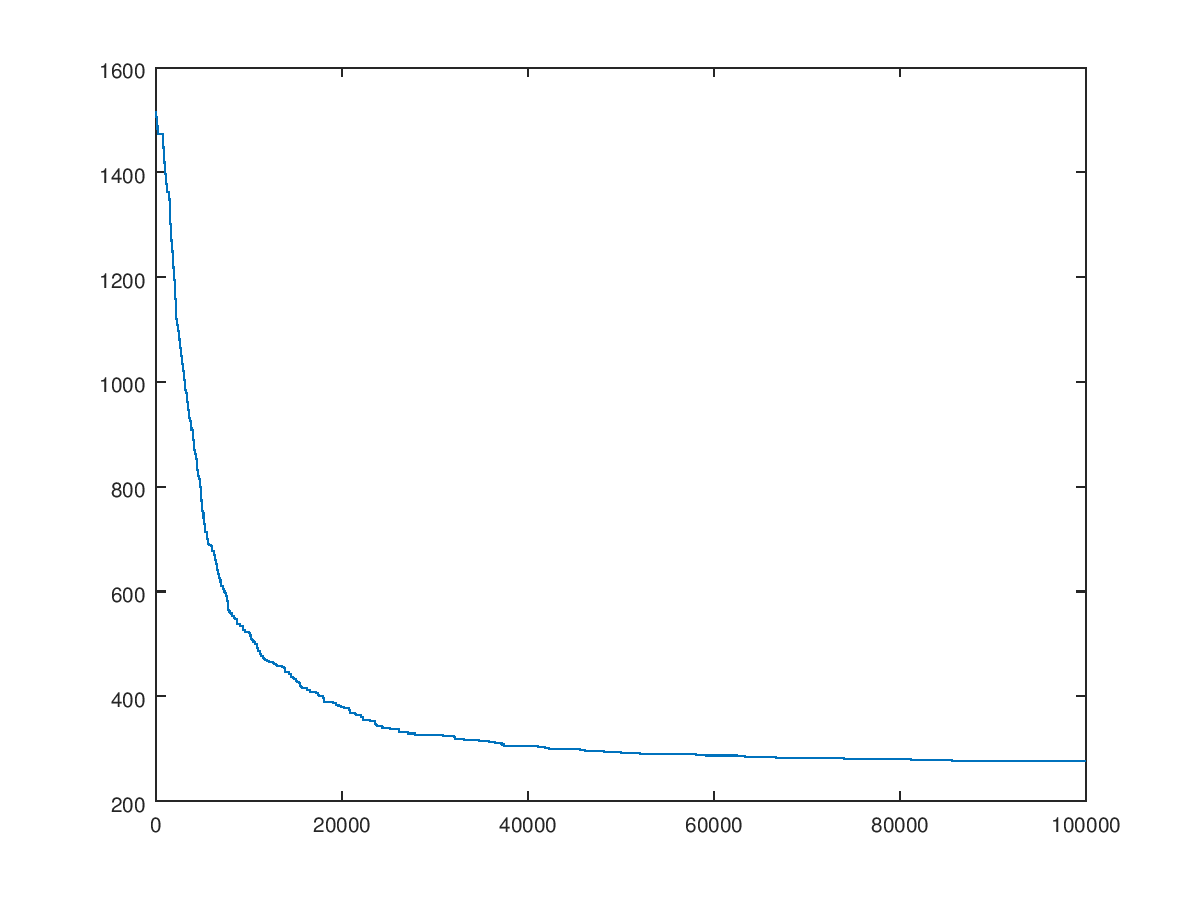
* N=100;
* Time =71.523 sec



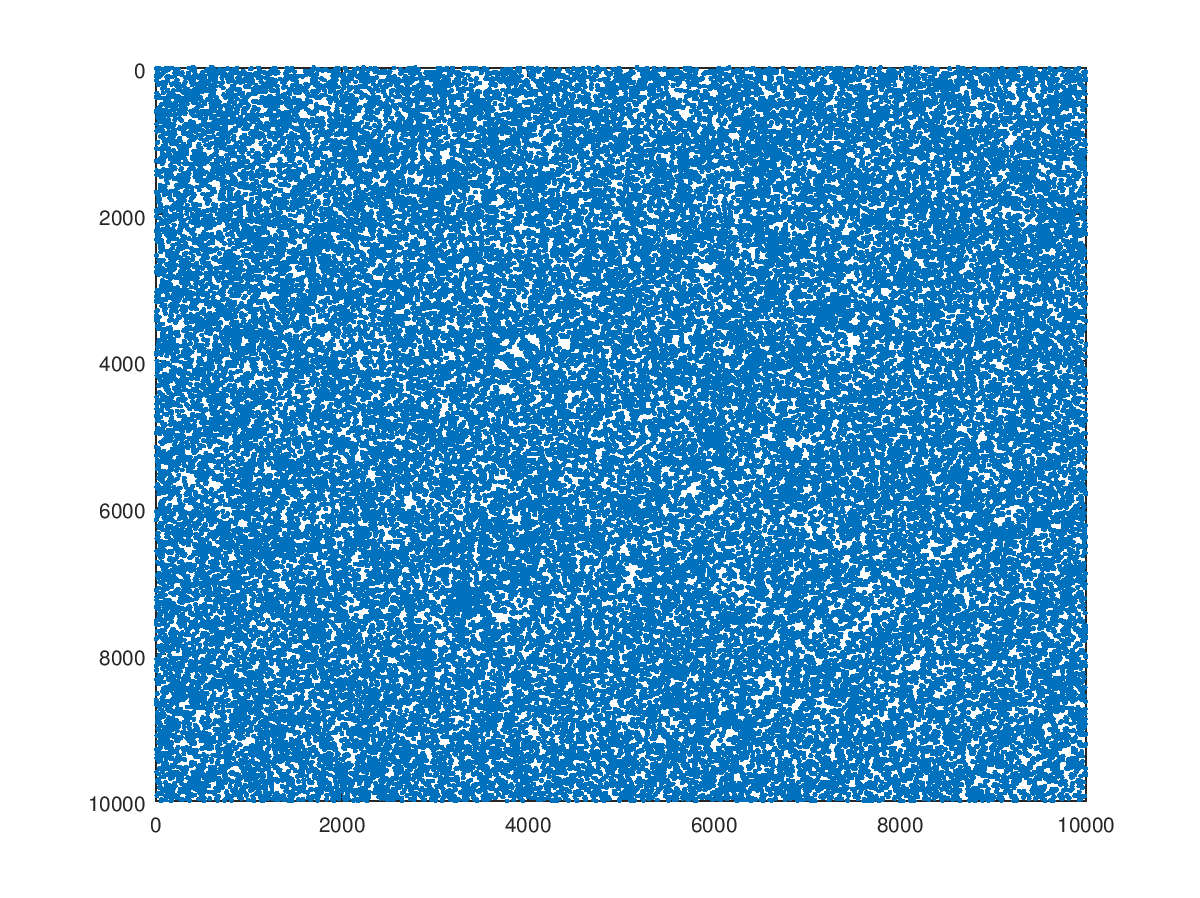
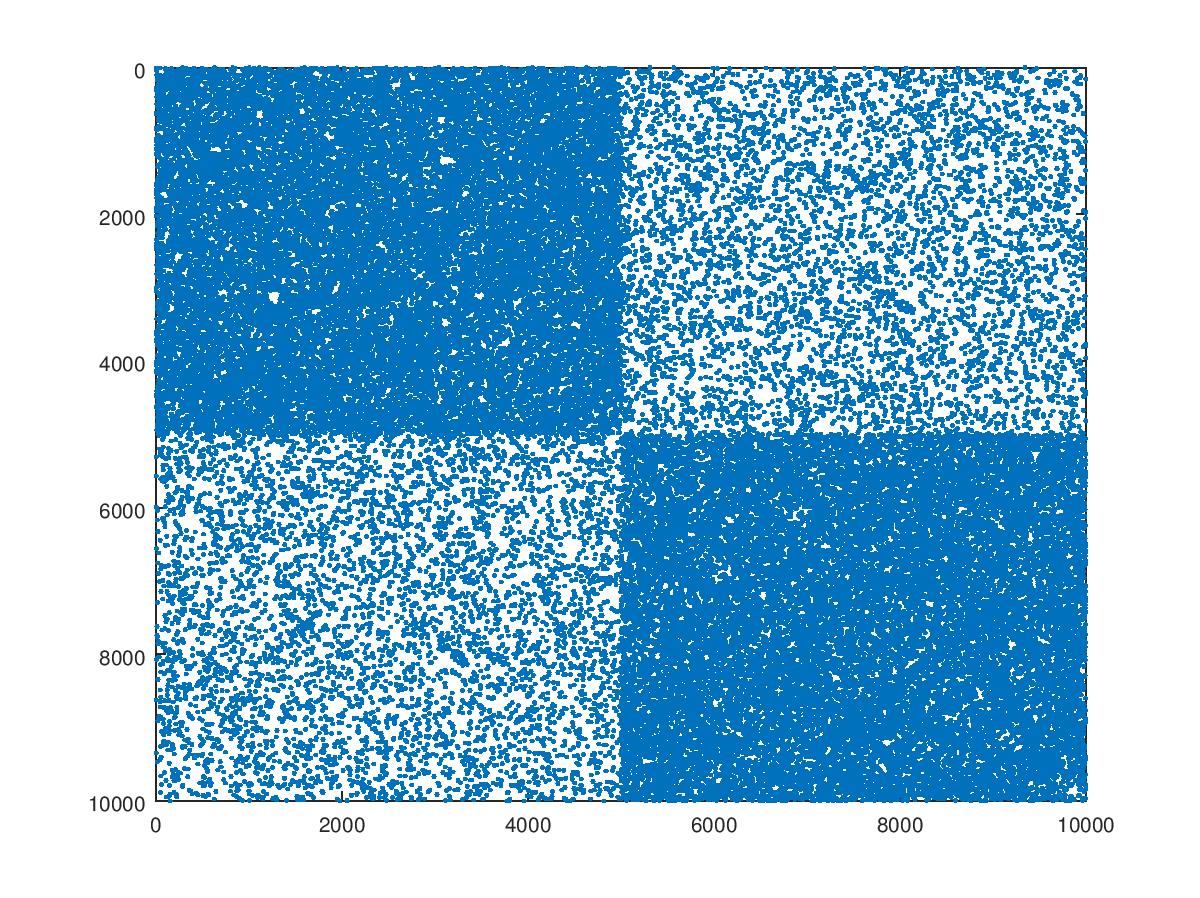
* N = 1000;

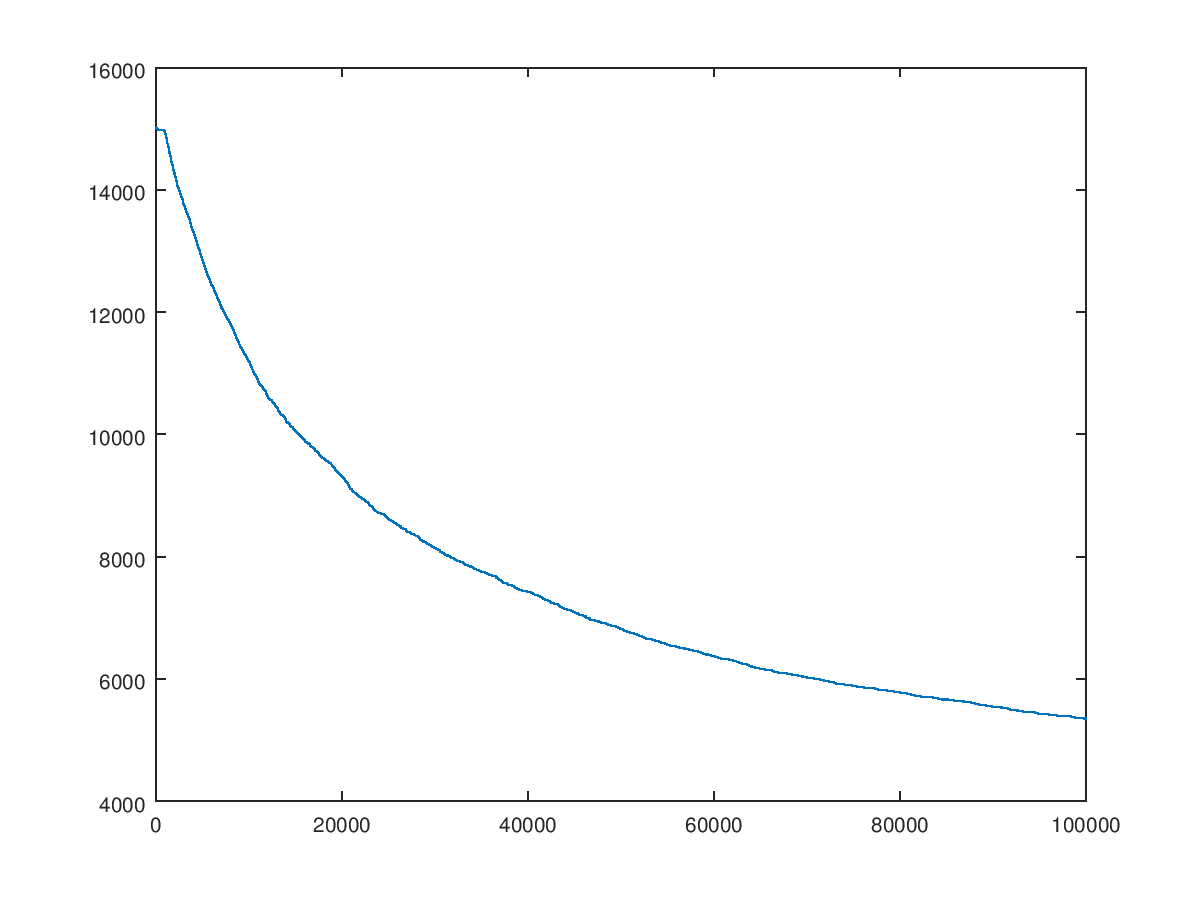
 Time = 87.4626 sec





* N = 10000;
* Time = 251.77 Sec



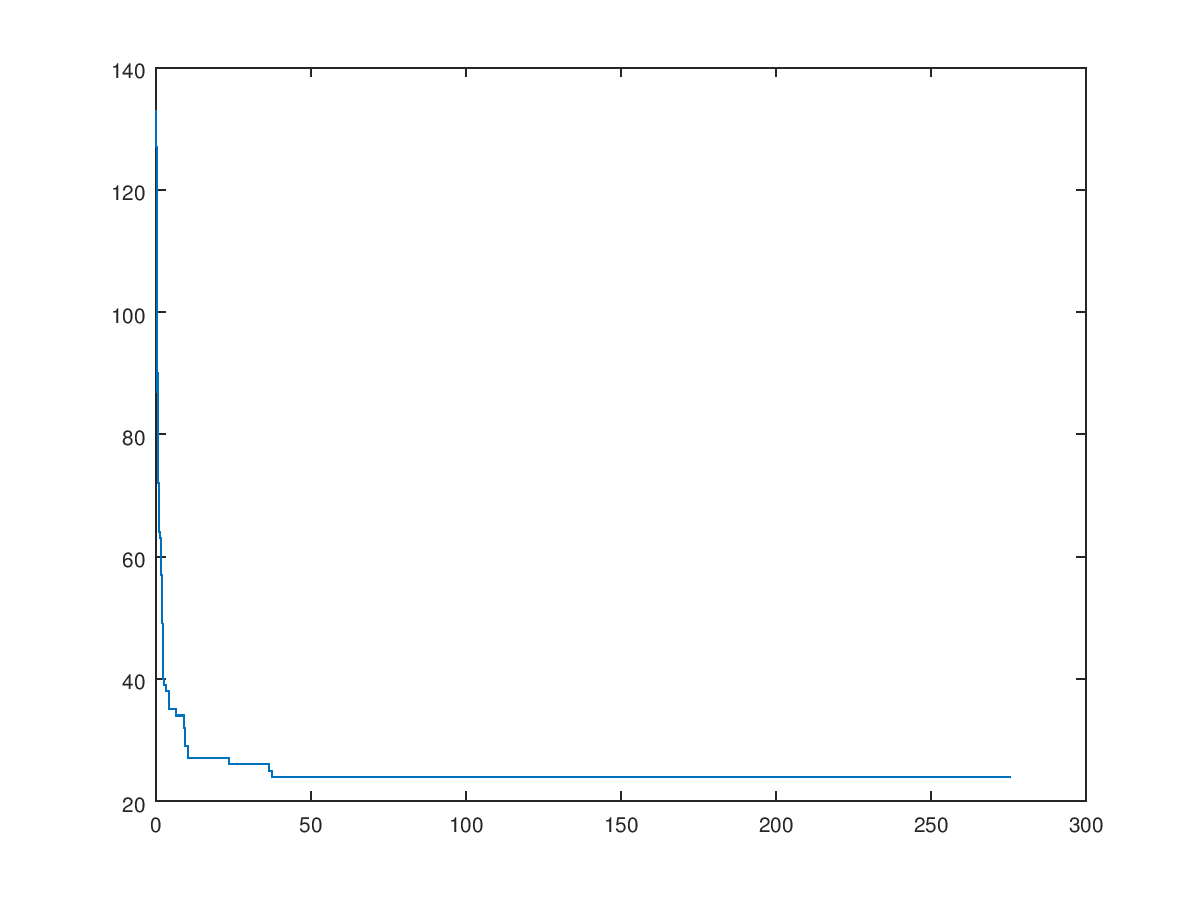


1. Genetic Algorithm
   1. Crossover and Mutation

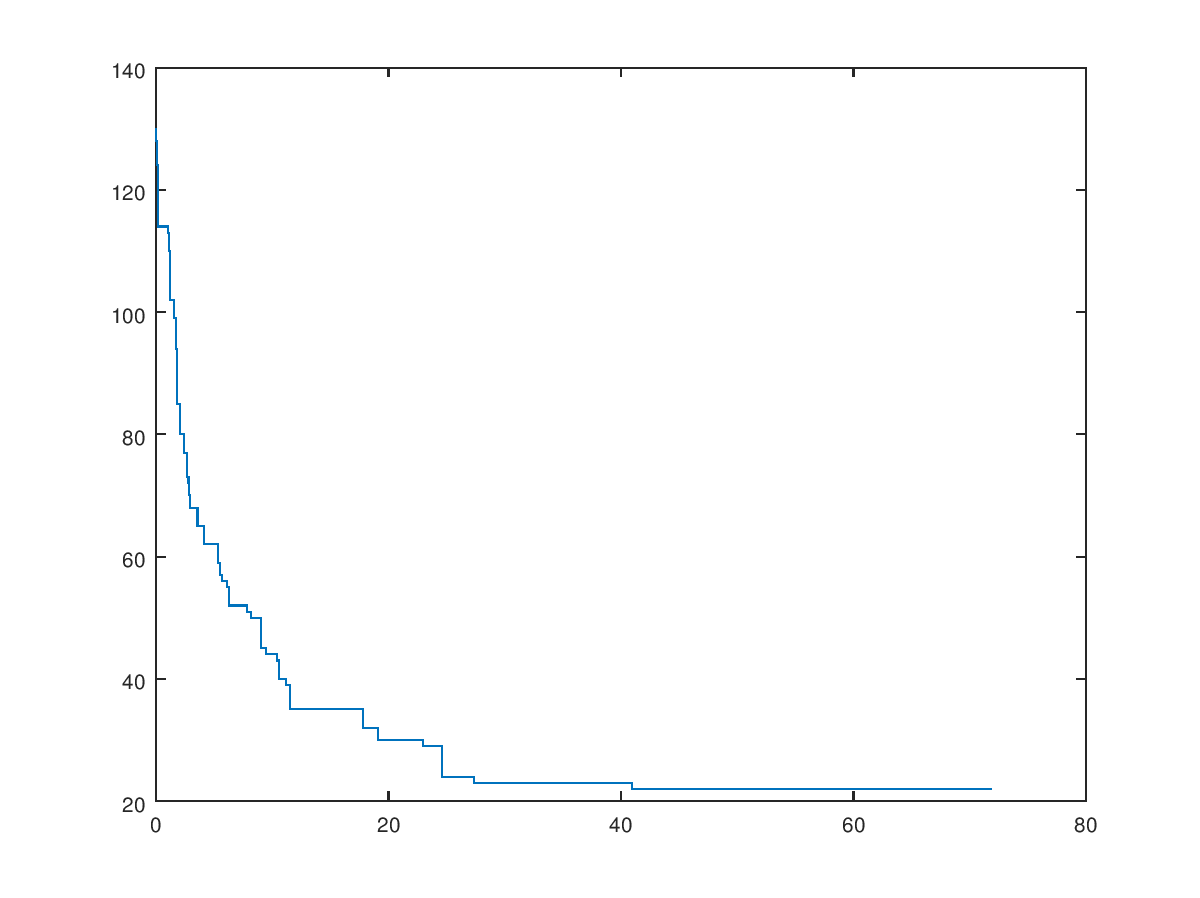
Selected Population Size (P) = 10

We considered 10 combinations of mutation rate and crossover rate and plotted the quality obtained and the time. (x = time, y = quality)

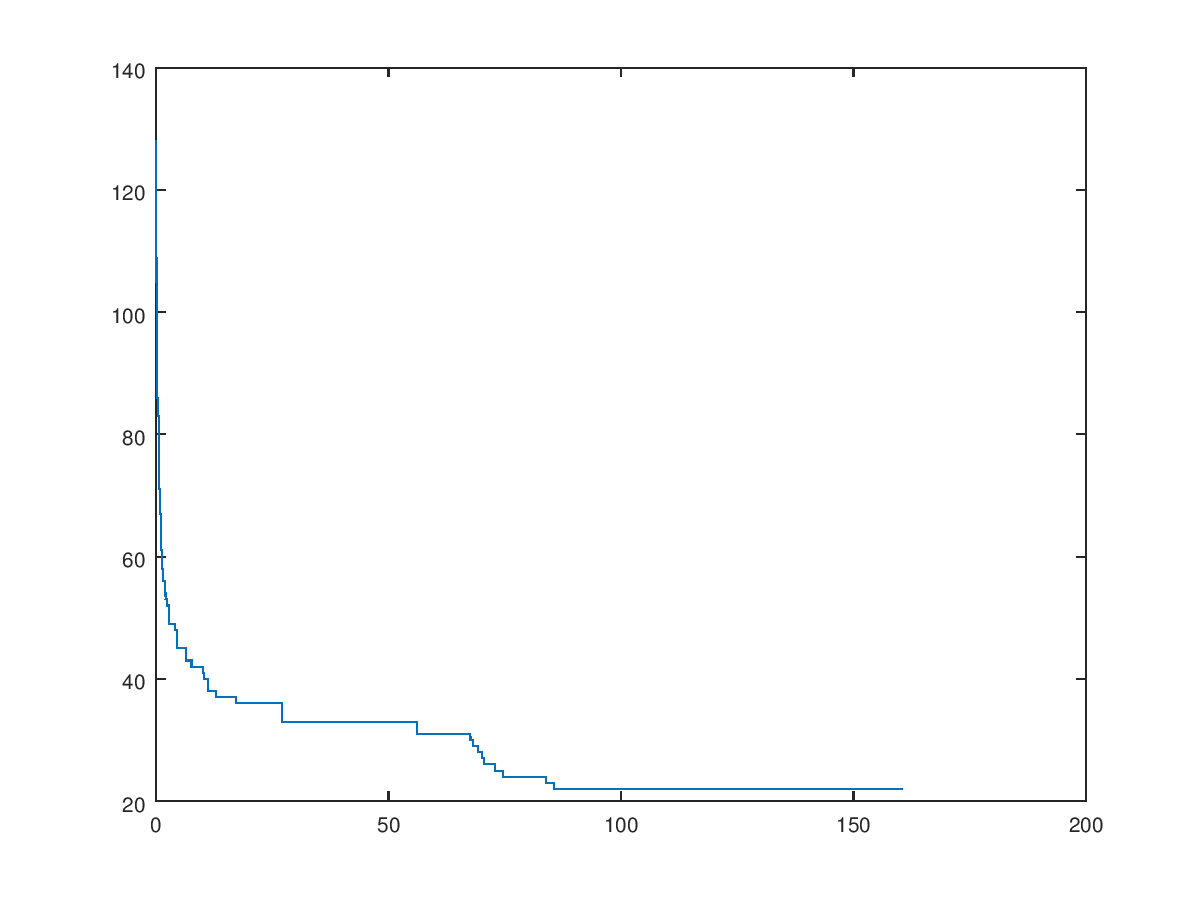
* Mutation Rate – 0.0, Crossover Rate – 0.0
* Time = 278.612 Sec



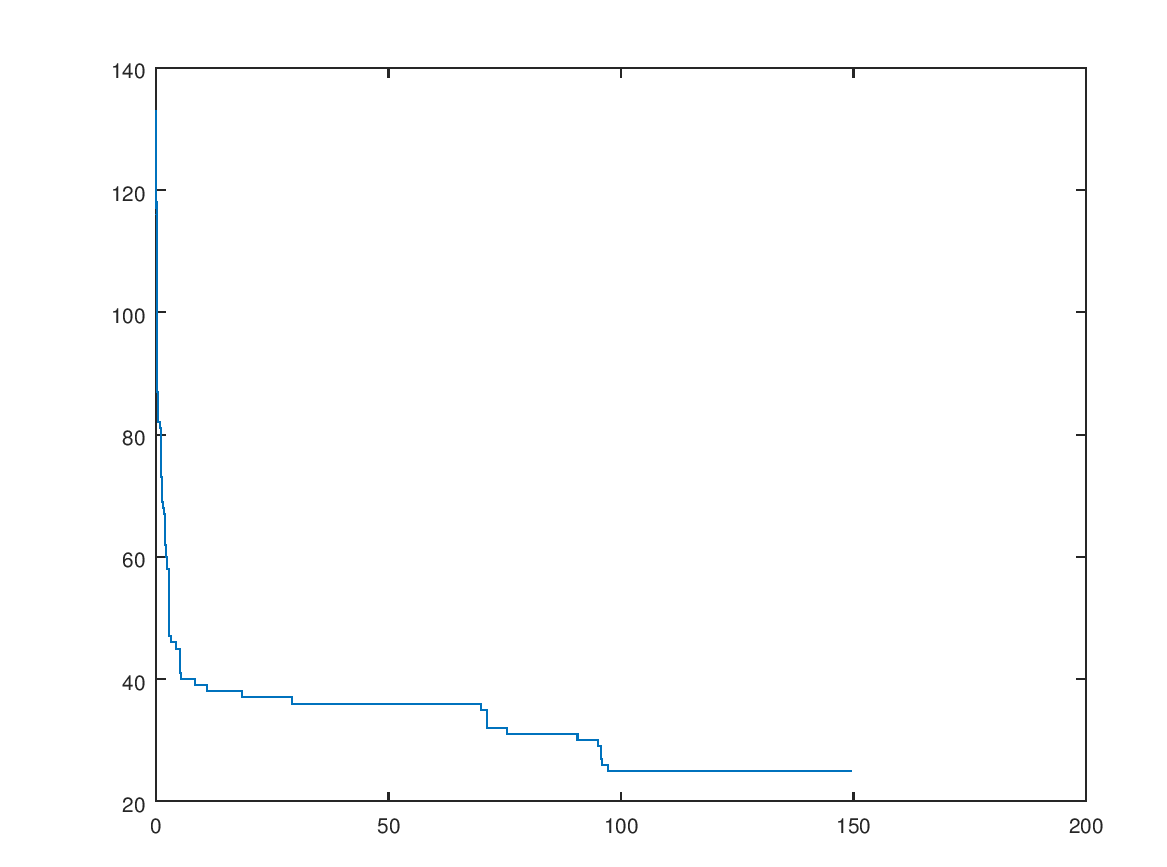
* Mutation Rate – 0.1, Crossover Rate – 0.0
* Time = 72.547 Sec

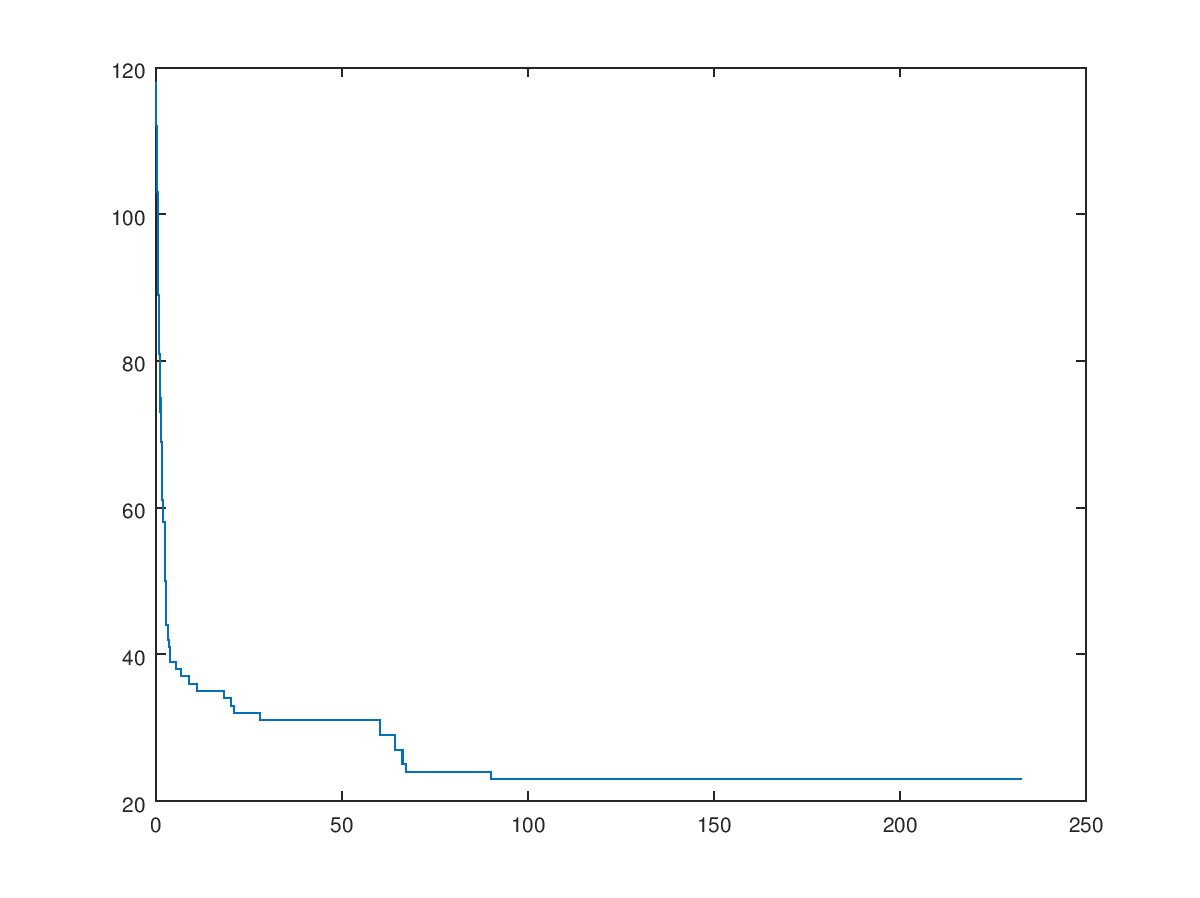


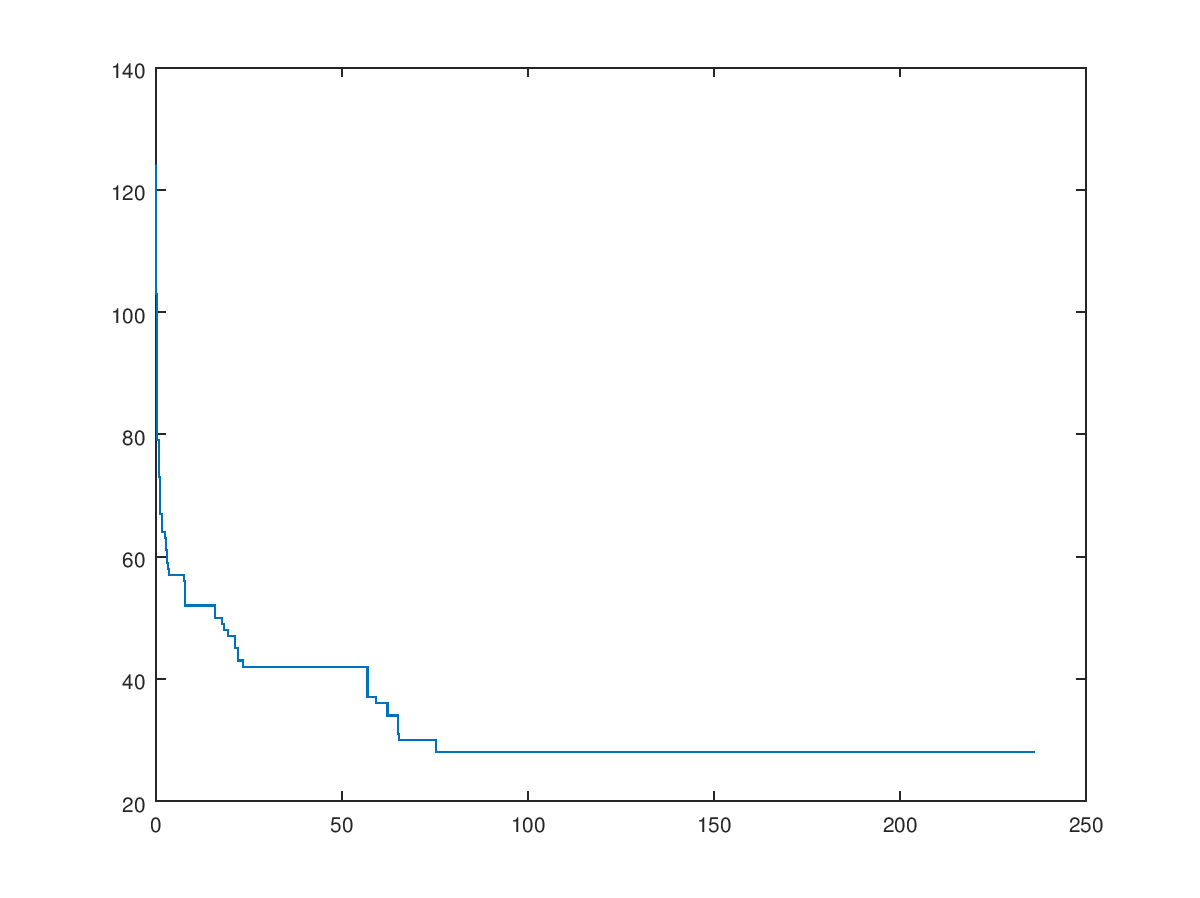
* Mutation Rate – 0.0, Crossover Rate – 0.2
* Time = 157.586 Sec



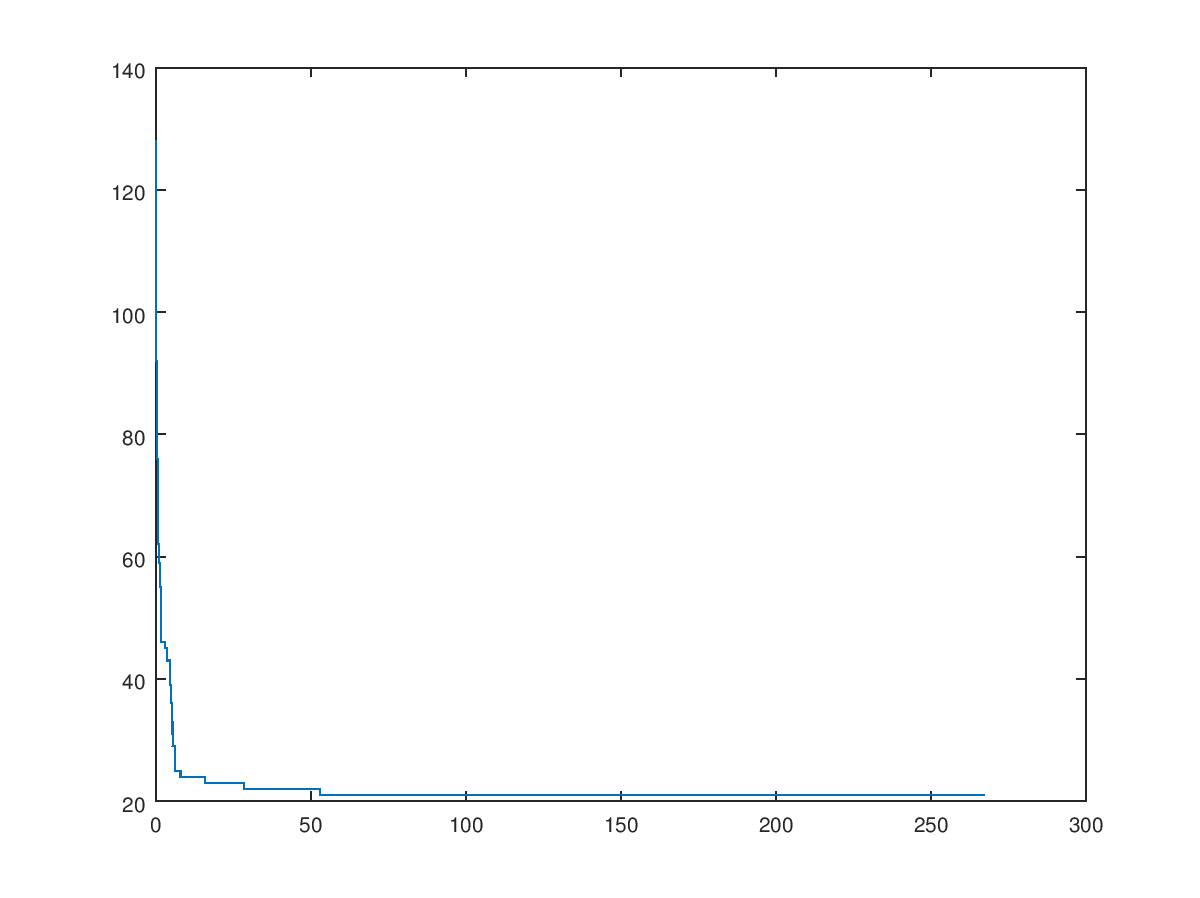
* Mutation Rate – 0.1, Crossover Rate – 0.2
* Time = 148.546 Sec



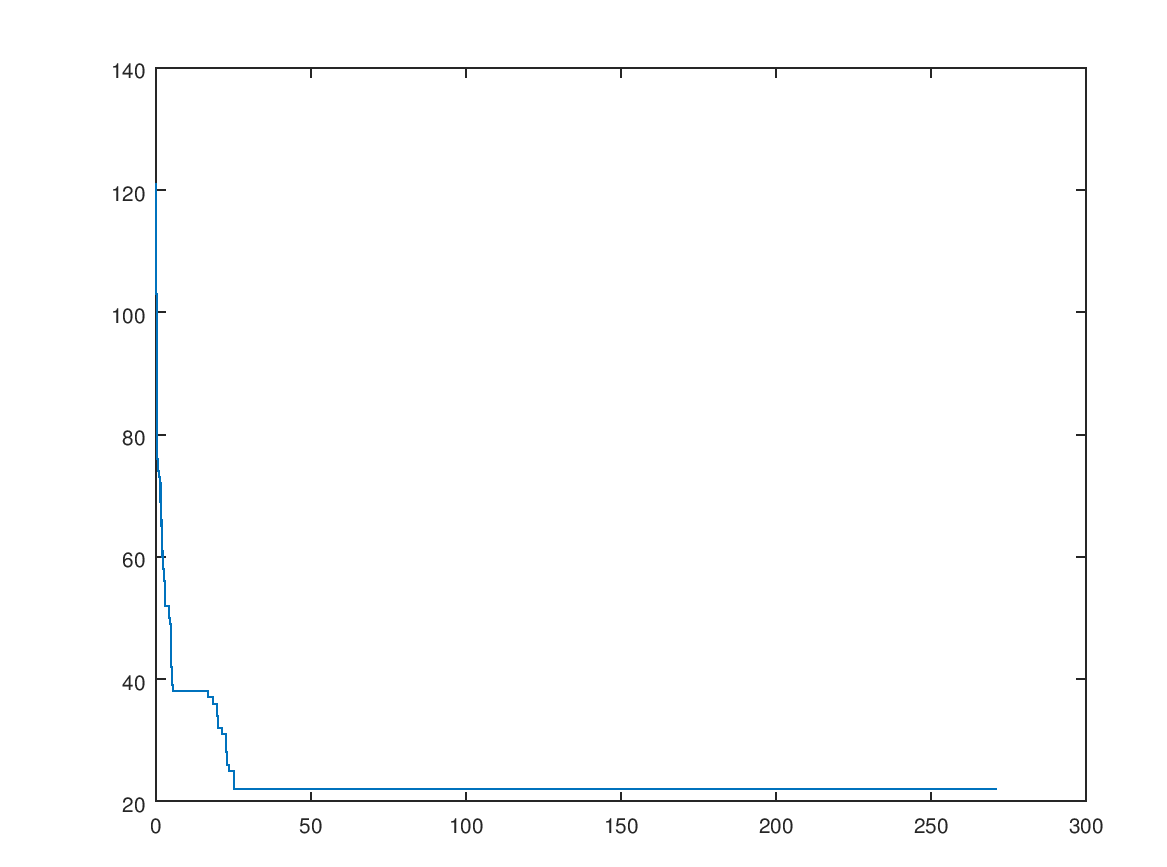
* Mutation Rate – 0.0, Crossover Rate – 0.4
* Time = 235.63 Sec
* Mutation Rate – 0.1, Crossover Rate – 0.4
* Time = 234.393 Sec



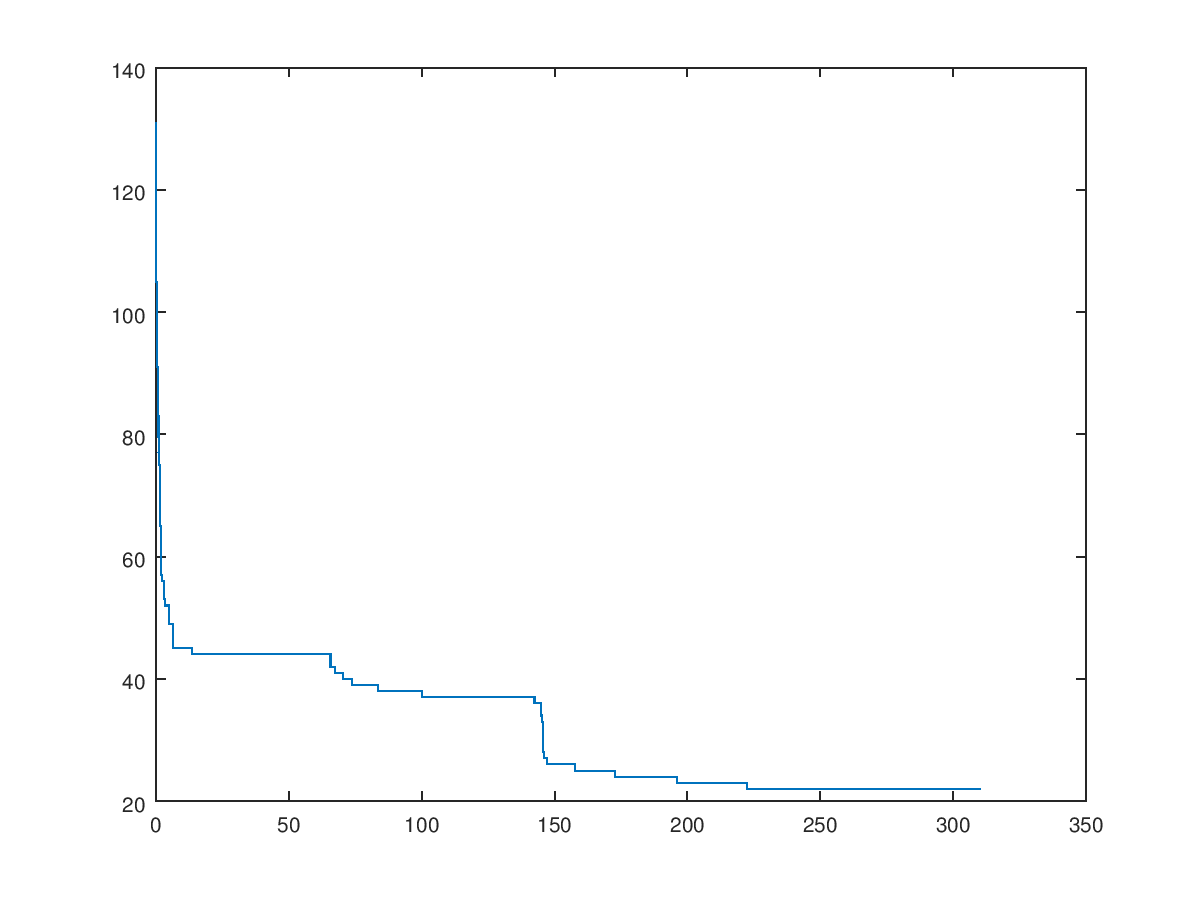
* Mutation Rate – 0.0, Crossover Rate – 0.5
* Time = 265.53 Sec



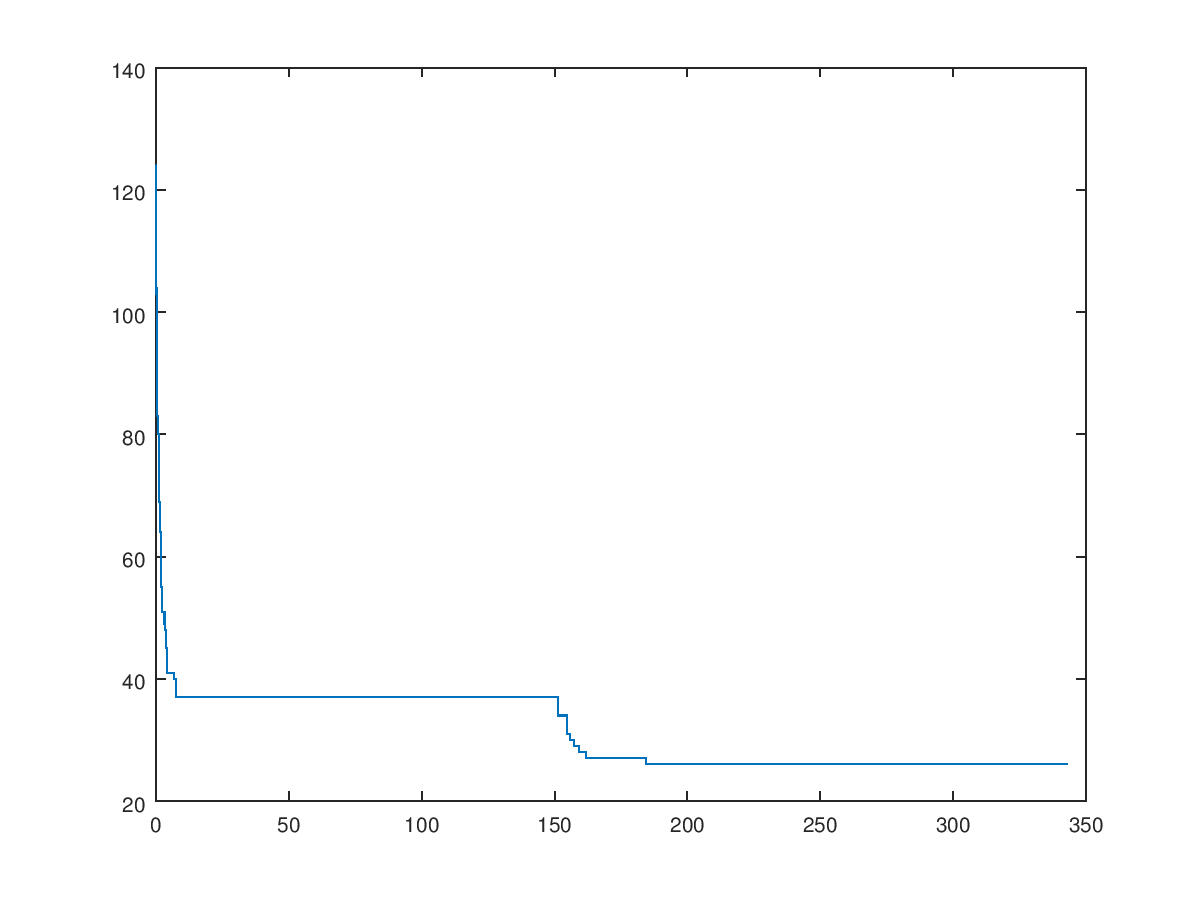
* Mutation Rate – 0.1, Crossover Rate – 0.5
* Time = 275.216 Sec



* Mutation Rate – 0.0, Crossover Rate – 0.6
* Time = 308.357 Sec



* Mutation Rate – 0.1, Crossover Rate – 0.6
* Time = 341.995 Sec



Crossover and Mutation Summarization

Selected Population Size (P) = 10

|  |  |  |
| --- | --- | --- |
| Crossover Rate | Mutation Rate | Process Time |
| 0.0 | 0.0 | 278.612 Sec |
| 0.1 | 72.547 Sec |
| 0.2 | 0.0 | 157.586 Sec |
| 0.1 | 148.546 Sec |
| 0.4 | 0.0 | 235.63 Sec |
| 0.1 | 234.393 Sec |
| 0.5 | 0.0 | 265.53 Sec |
| 0.1 | 275.216 Sec |
| 0.6 | 0.0 | 308.357 Sec |
| 0.1 | 341.995 Sec |

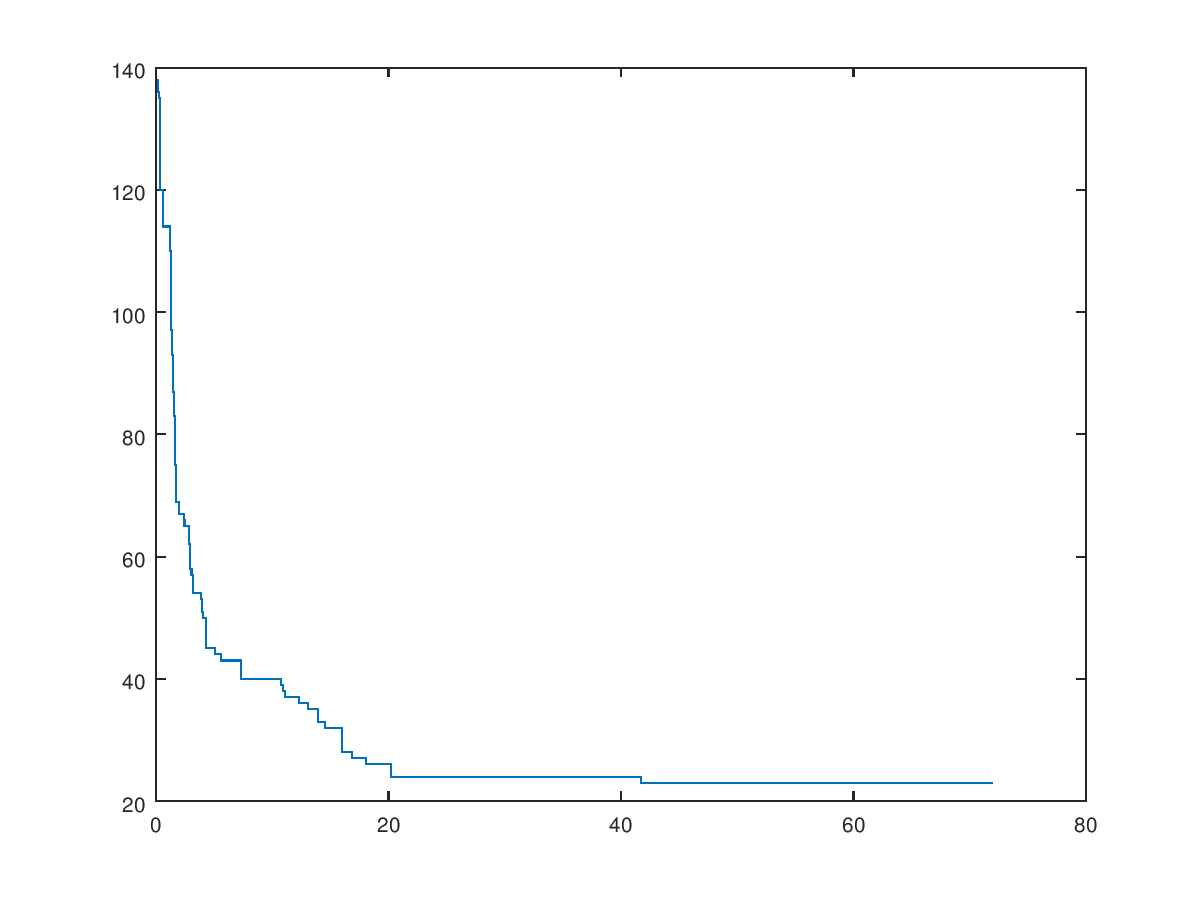
Mutation Rate – 0.1, Crossover Rate – 0.0

Because with the execution time = 71.867 Sec

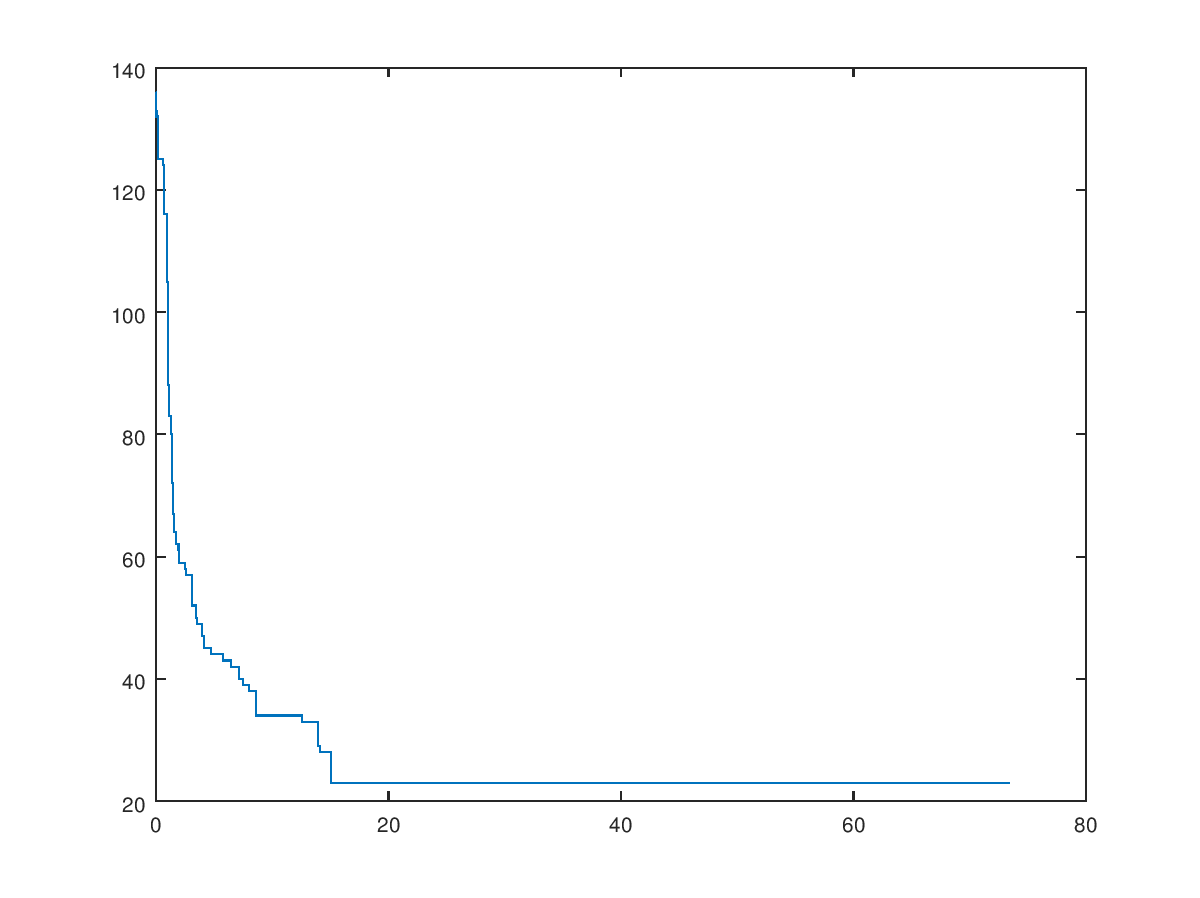
* 1. Population Size

Population Size (P) = 2

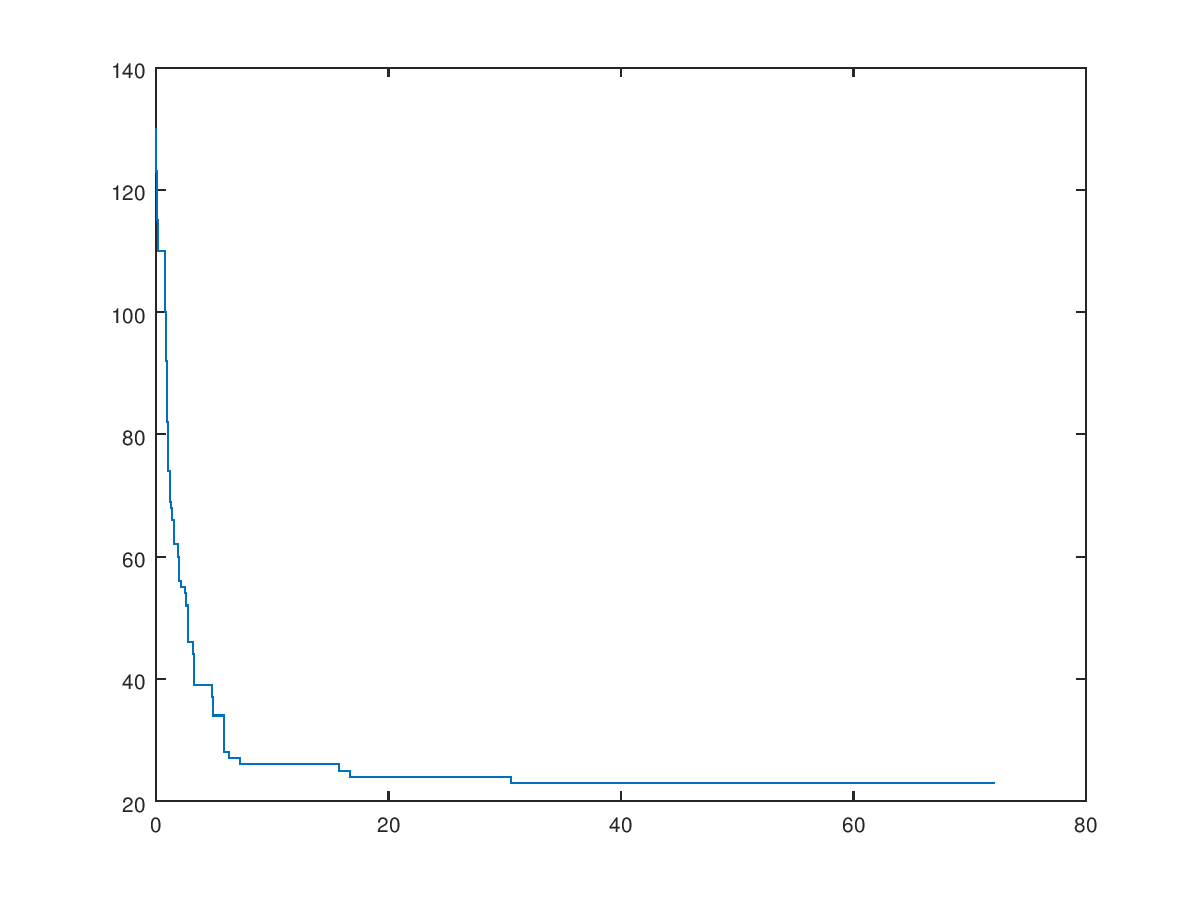
* Mutation Rate – 0.0, Crossover Rate – 0.0
* Time = 73.5541 Sec

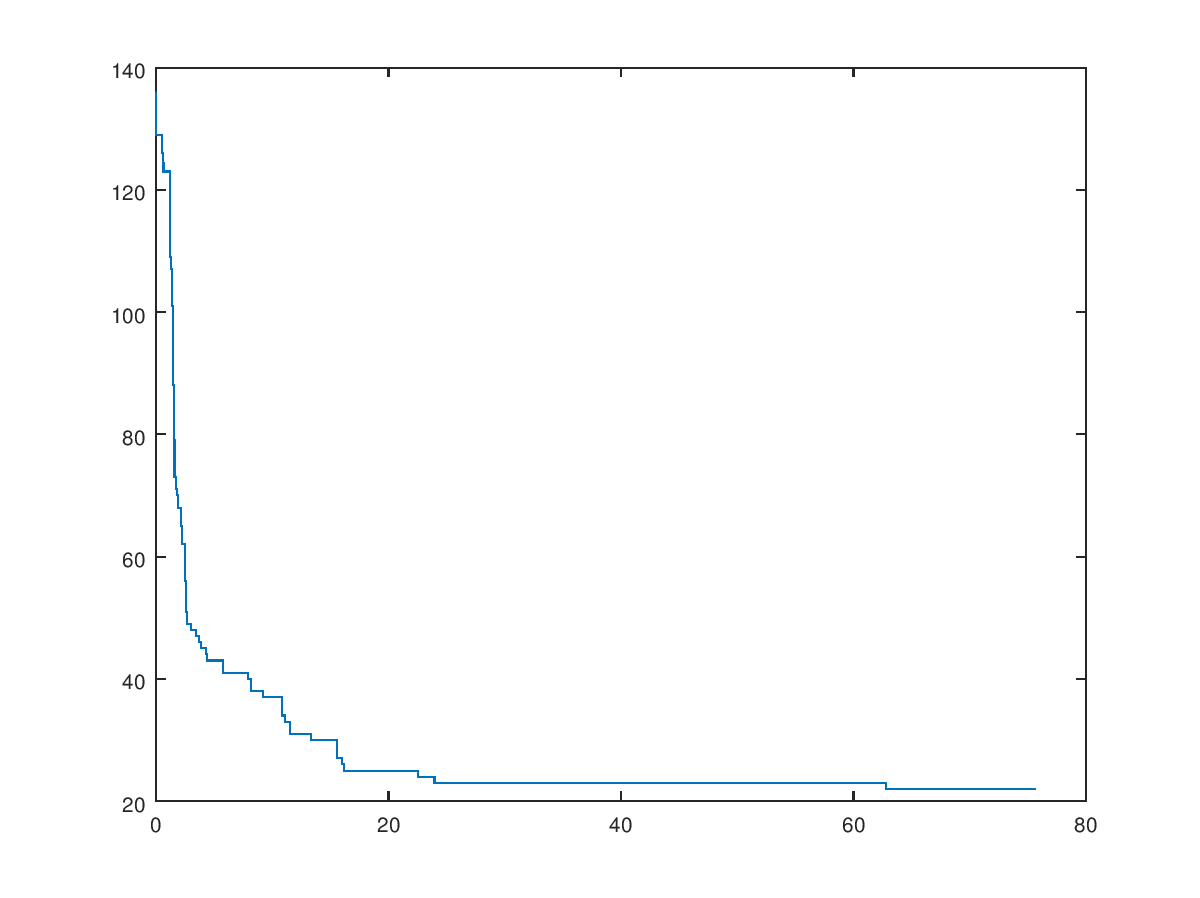
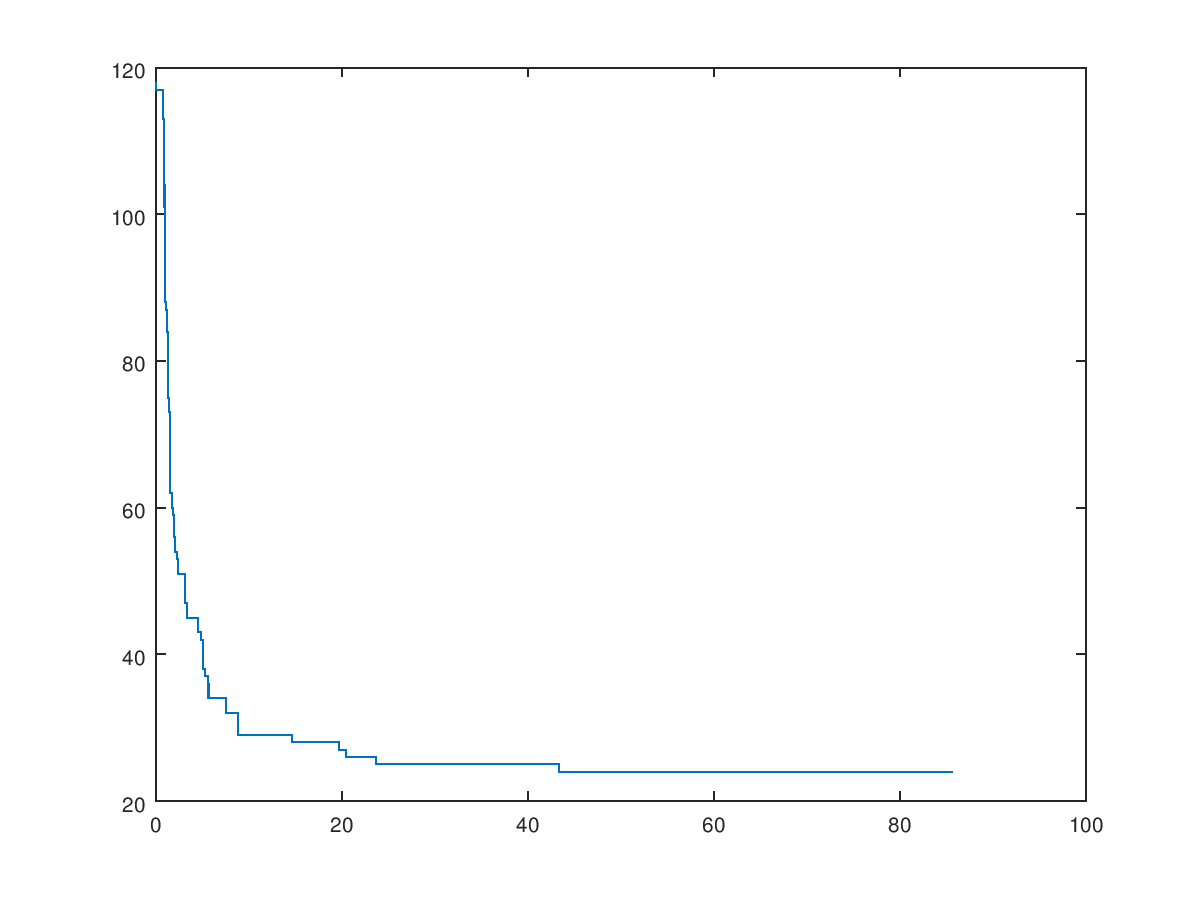


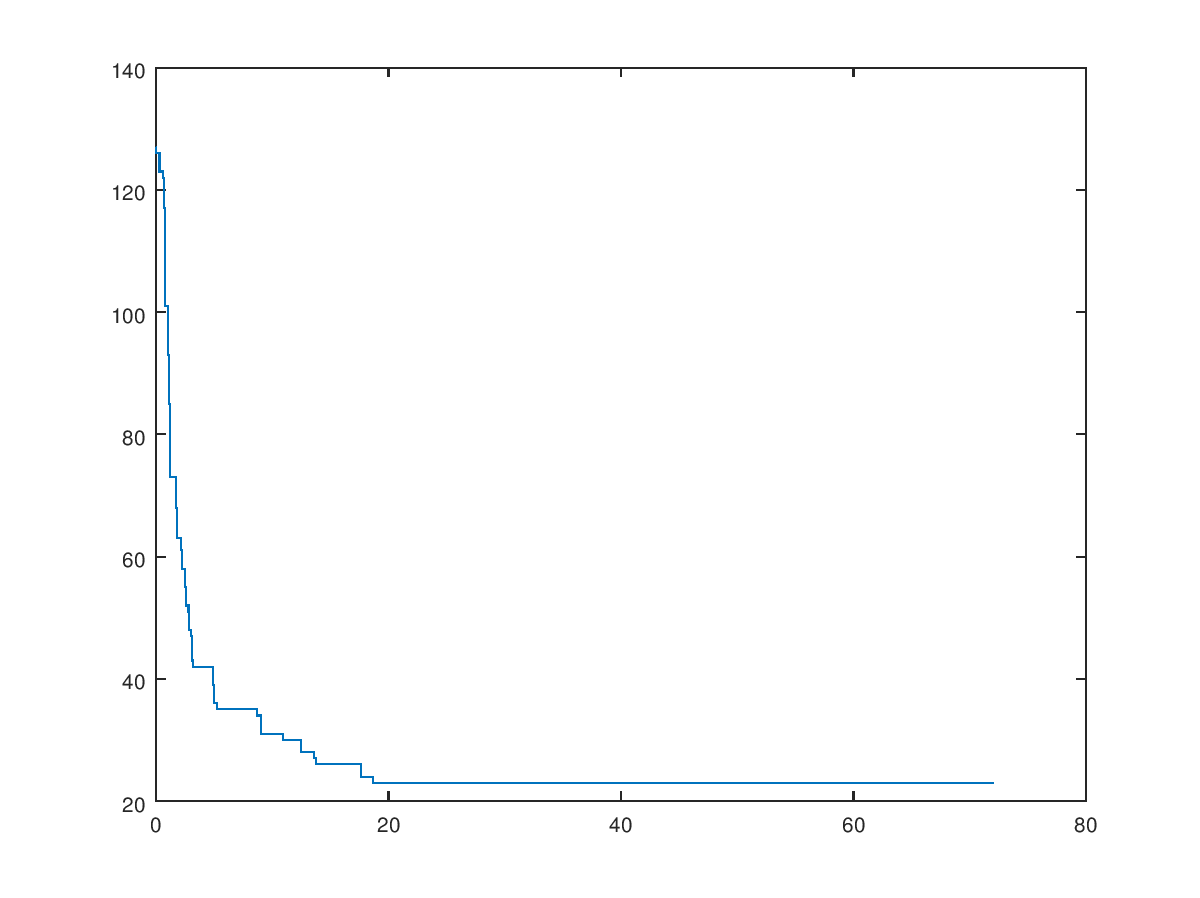
* Mutation Rate – 0.1, Crossover Rate – 0.0
* Time = 74.2121 Sec



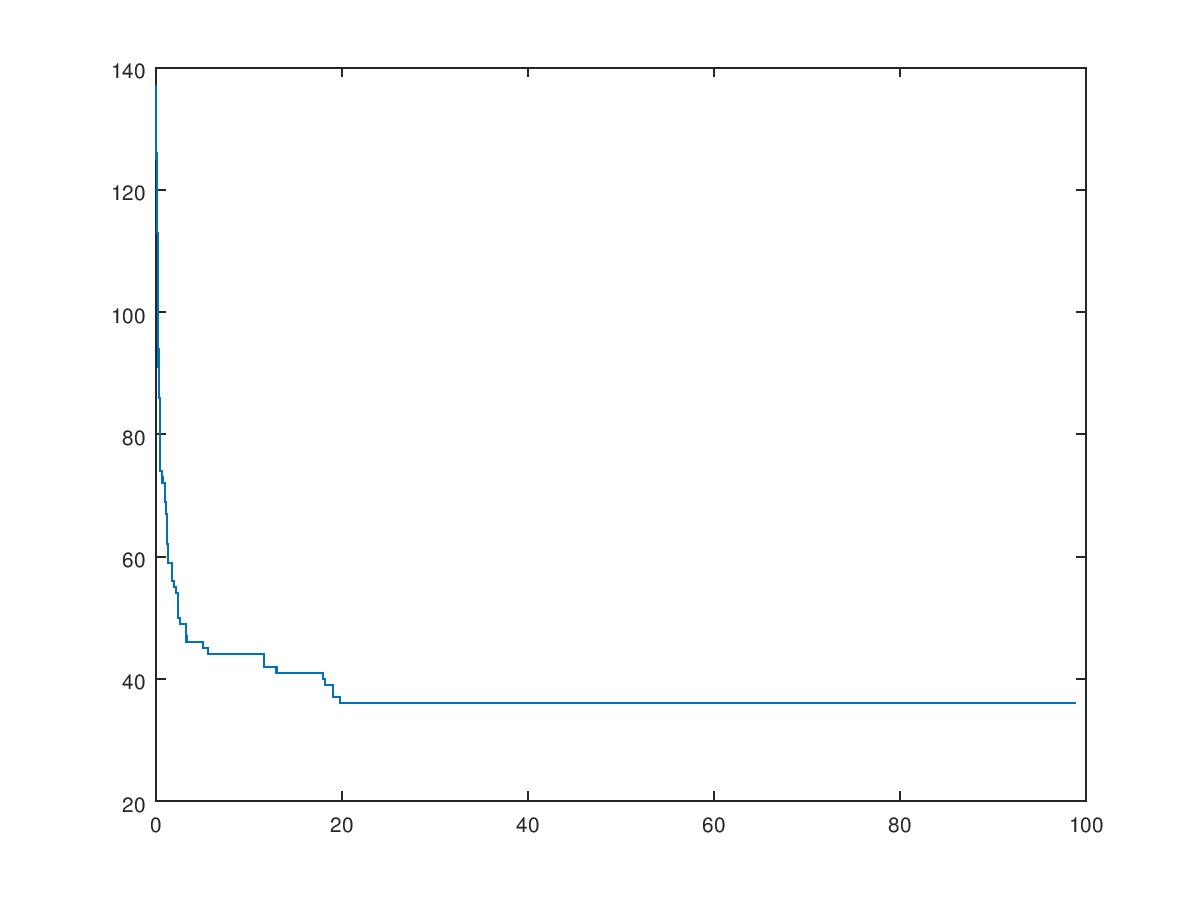
* Mutation Rate – 0.0, Crossover Rate – 0.2
* Time = 73.1842 Sec



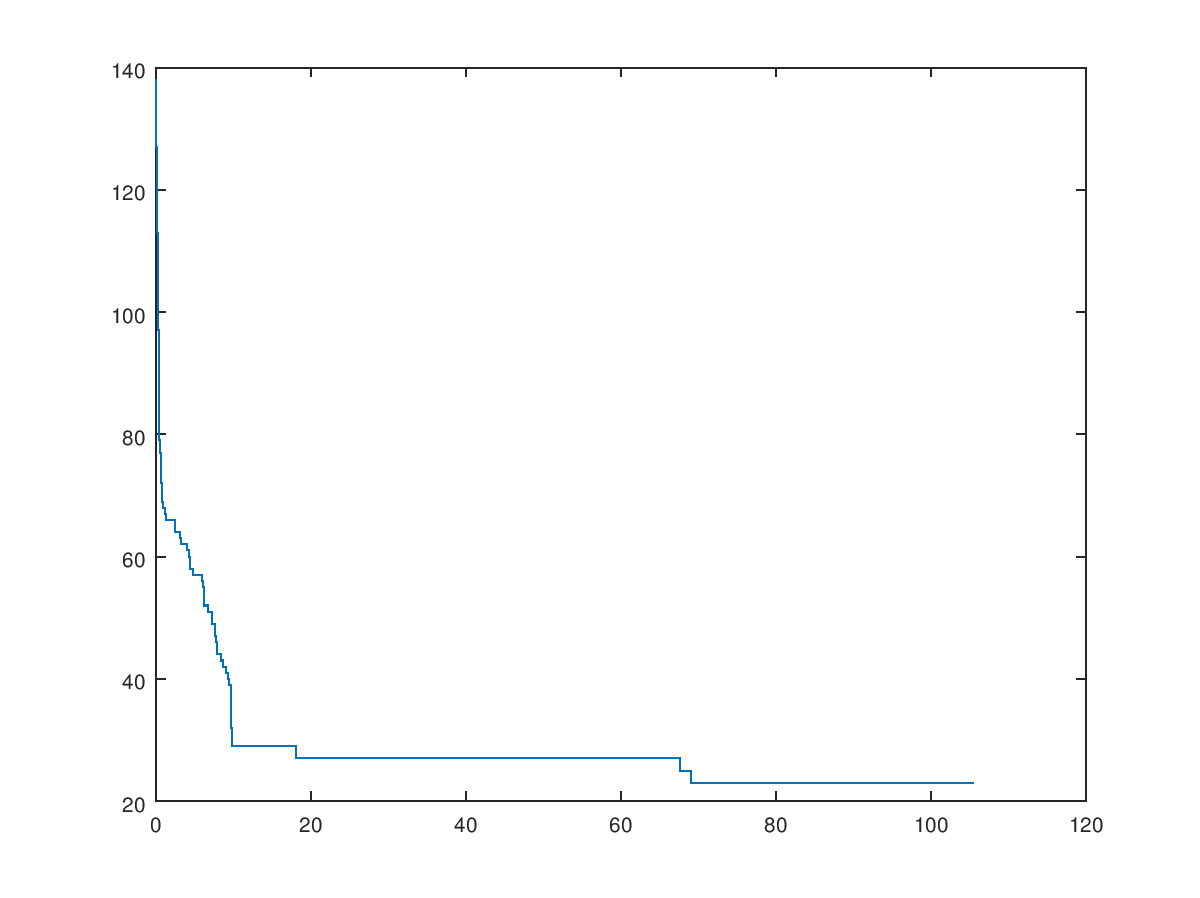
* Mutation Rate – 0.1, Crossover Rate – 0.2
* Time = 76.6227 Sec
* Mutation Rate – 0.0, Crossover Rate – 0.4
* Time = 84.6488 Sec
* Mutation Rate – 0.1, Crossover Rate – 0.4
* Time = 73.7882 Sec



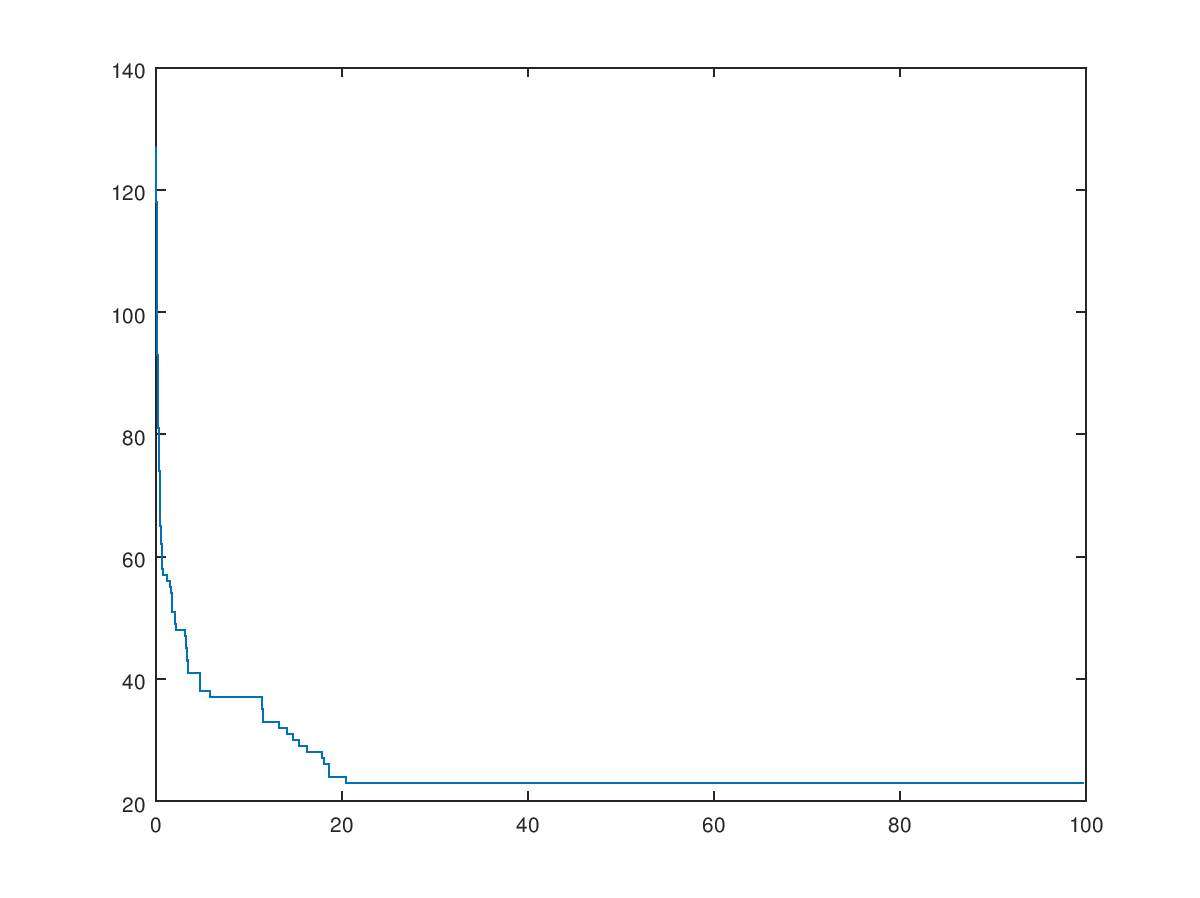
* Mutation Rate – 0.0, Crossover Rate – 0.5
* Time = 97.9122 Sec



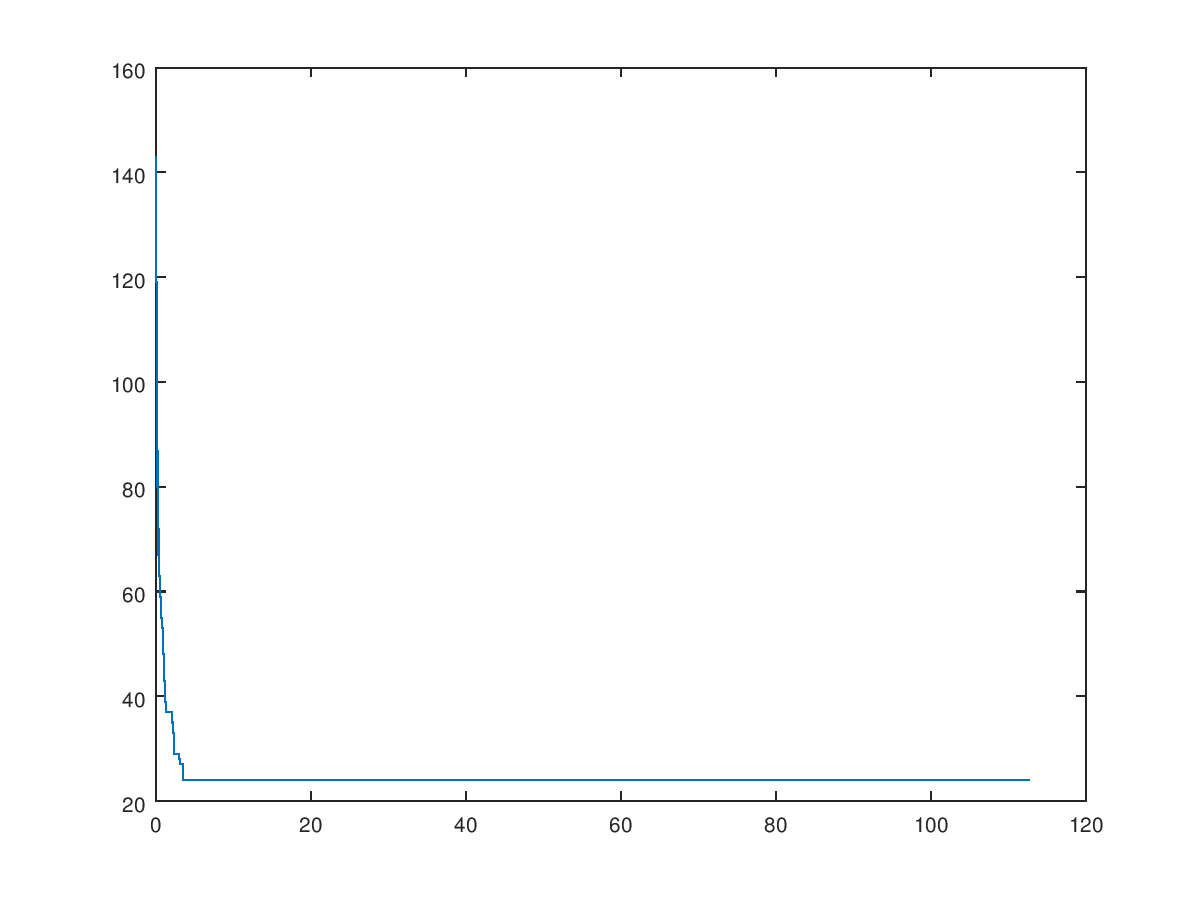
* Mutation Rate – 0.1, Crossover Rate – 0.5
* Time = 104.529 Sec



* Mutation Rate – 0.0, Crossover Rate – 0.6
* Time = 100.7238 Sec

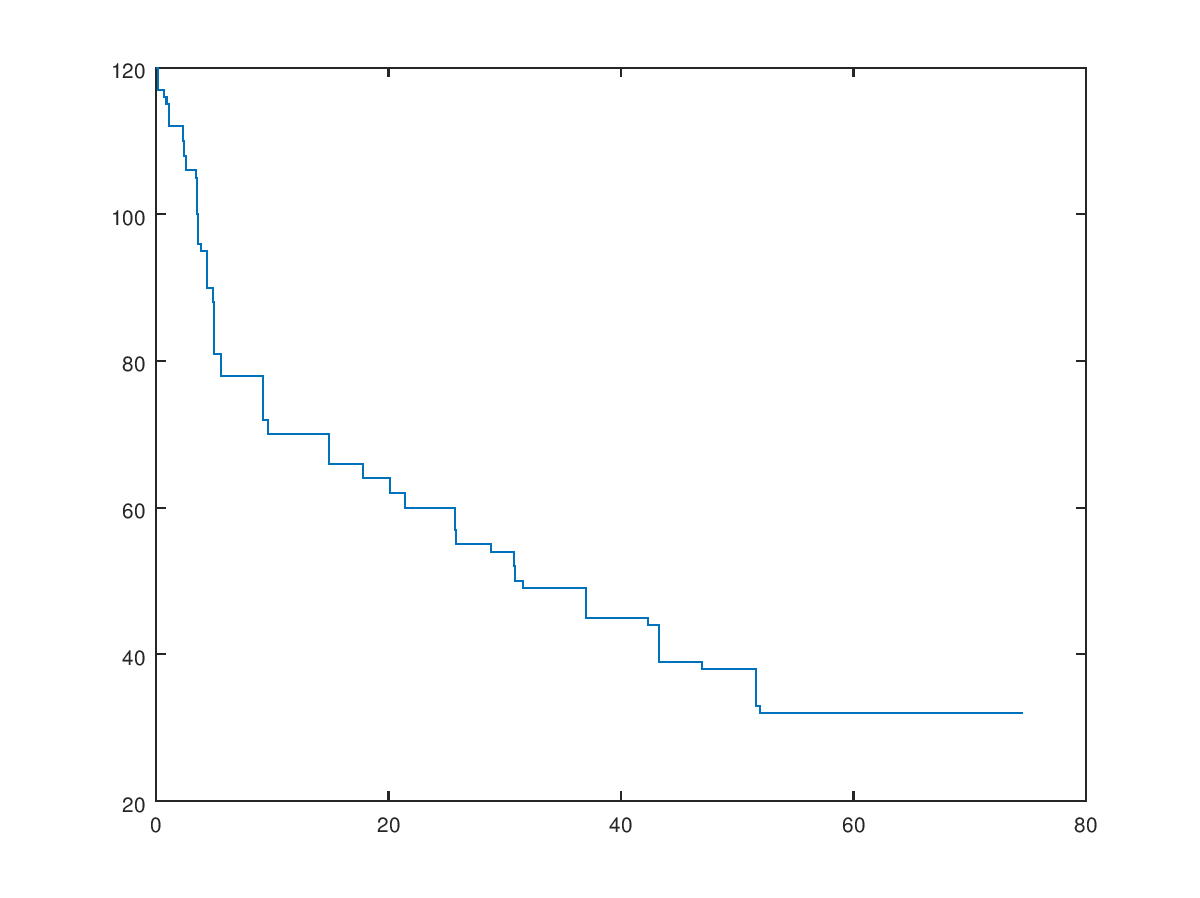


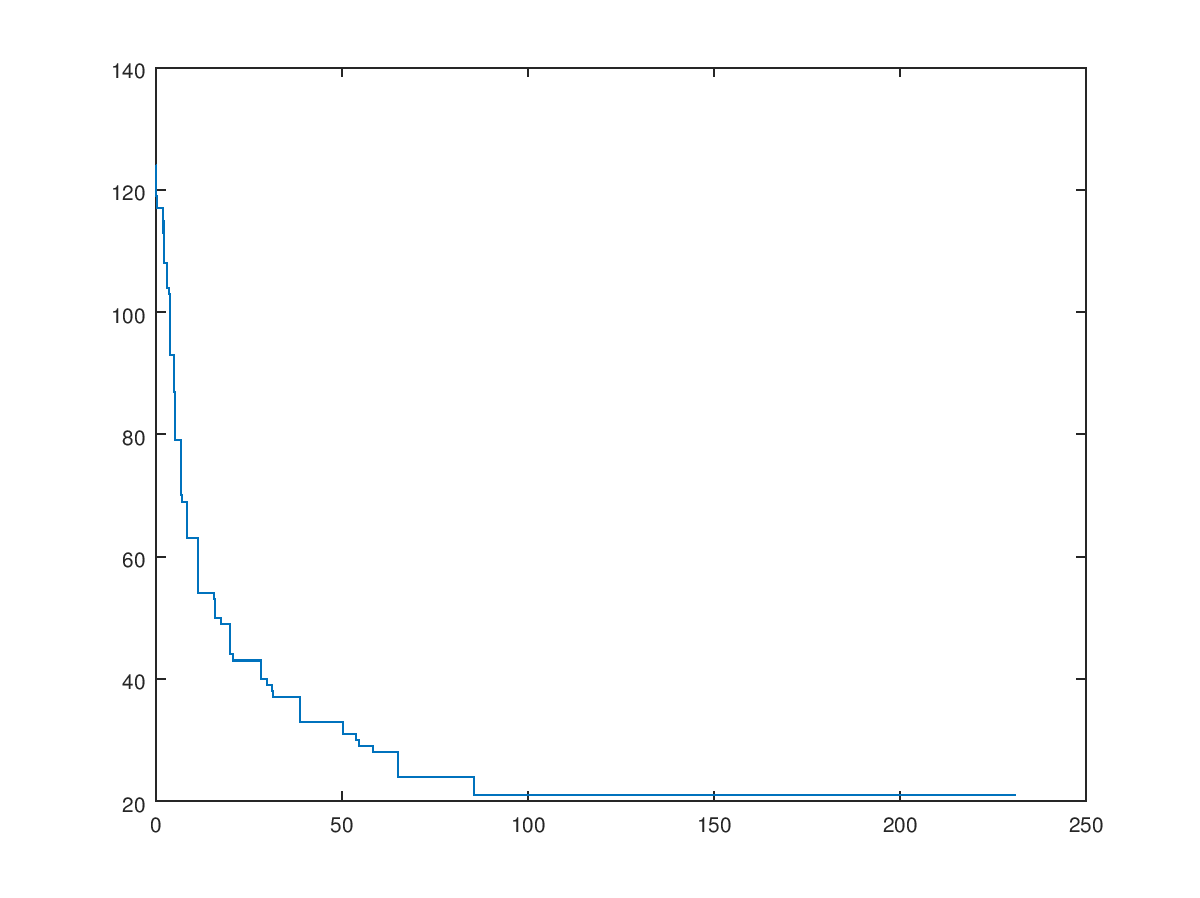
* Mutation Rate – 0.1, Crossover Rate – 0.6
* Time = 110.758 Sec

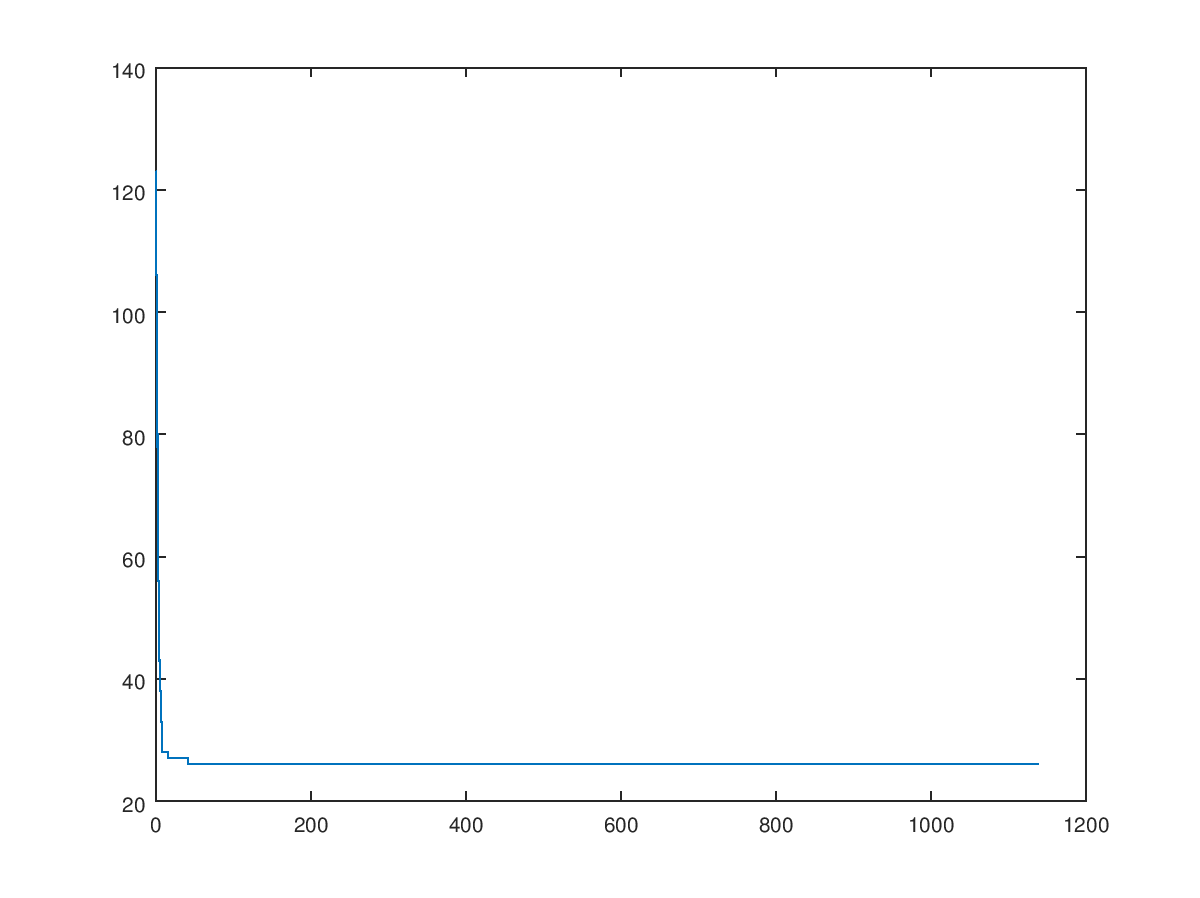


Population Size (P) = 50

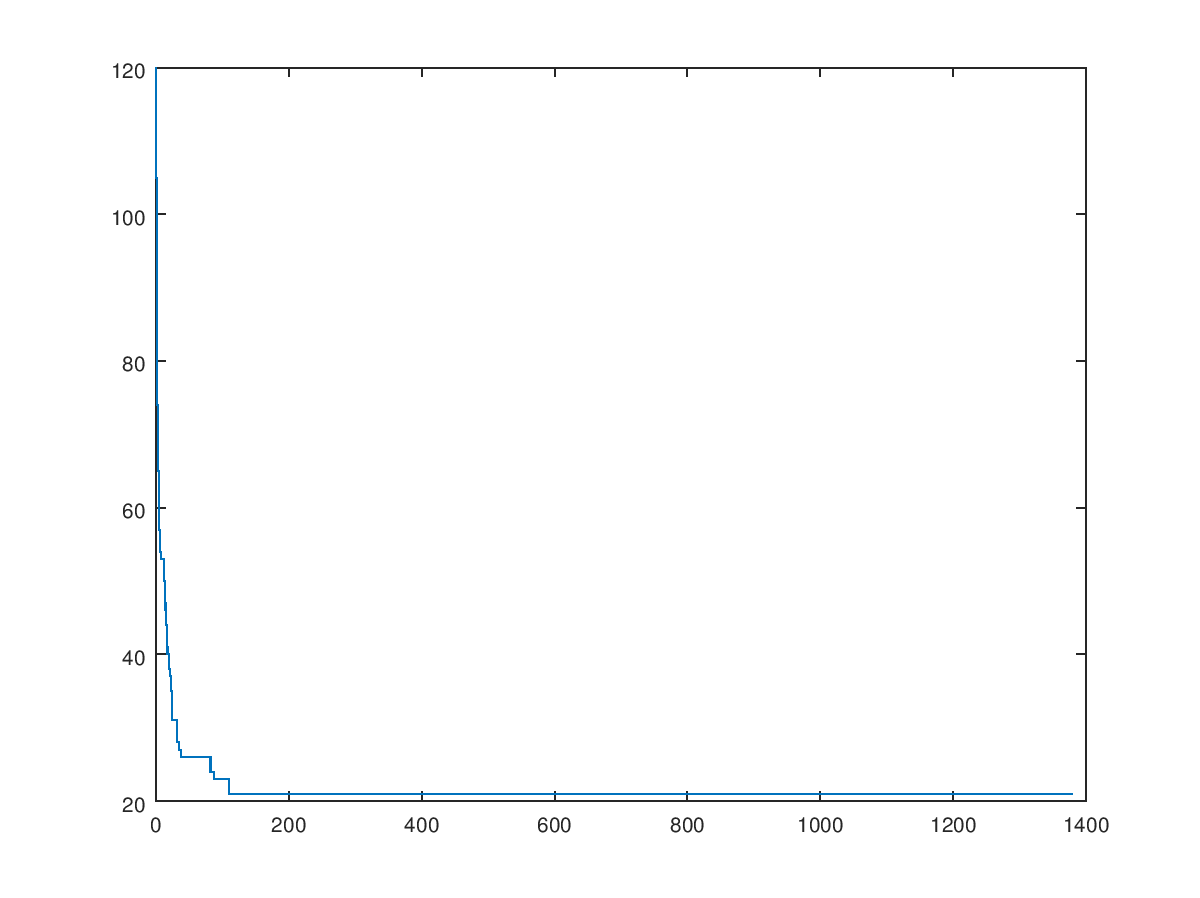
* Mutation Rate – 0.0, Crossover Rate – 0.0
* Time = 74.52 Sec

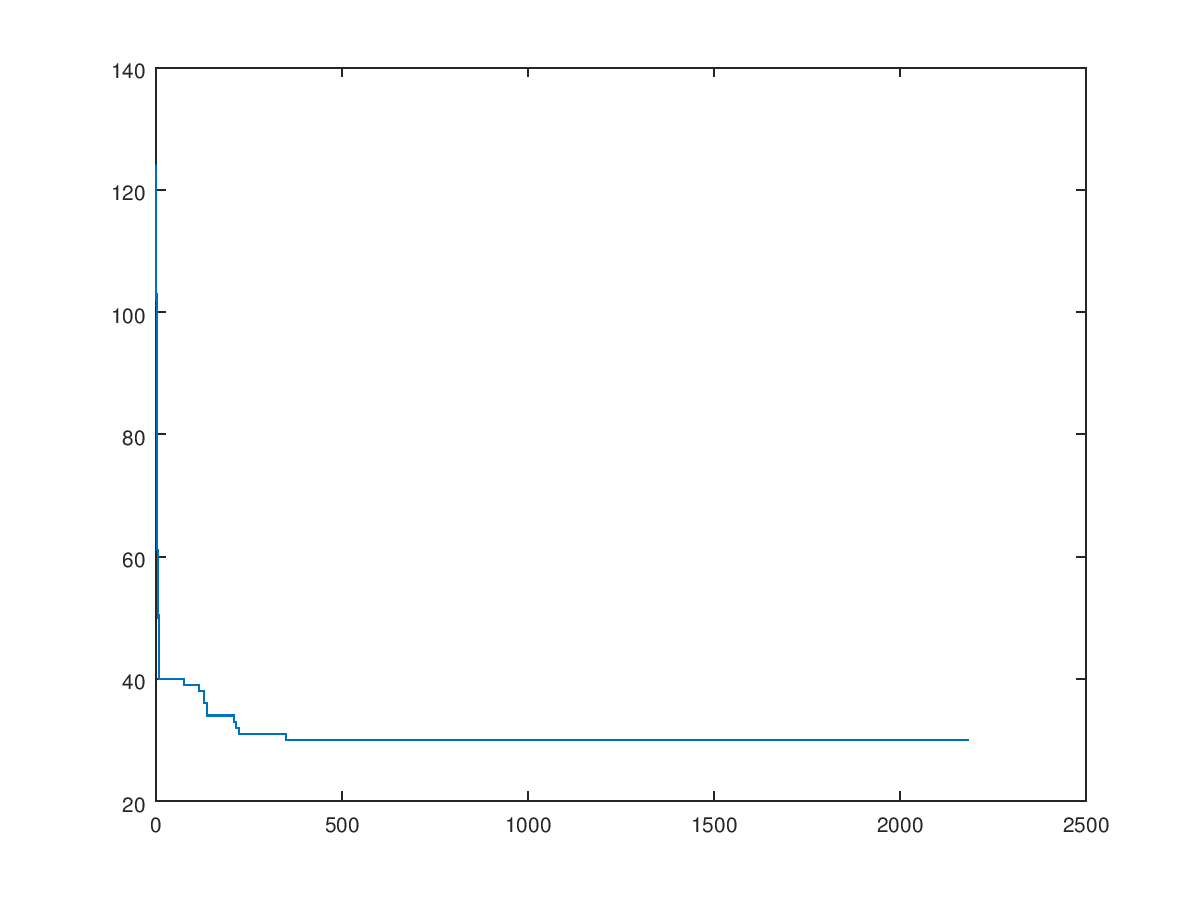
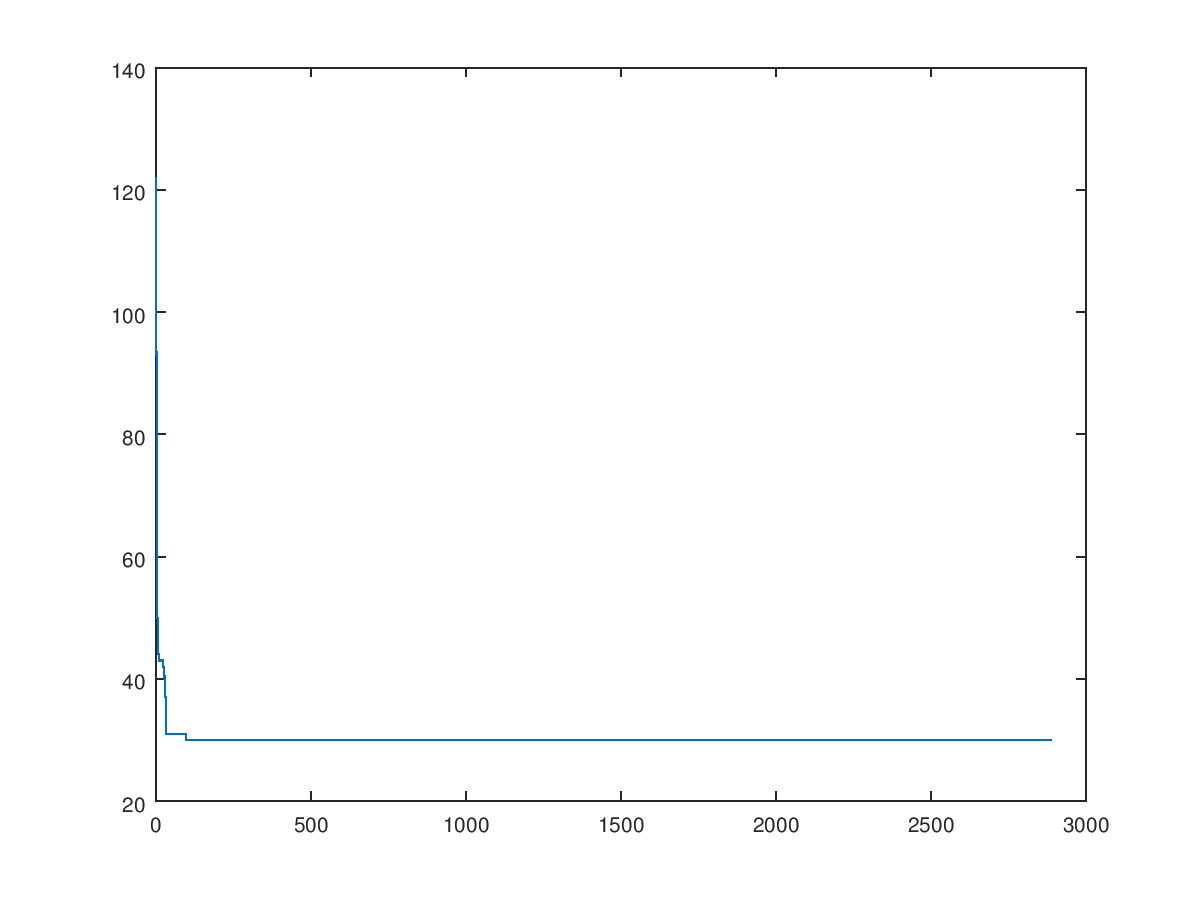
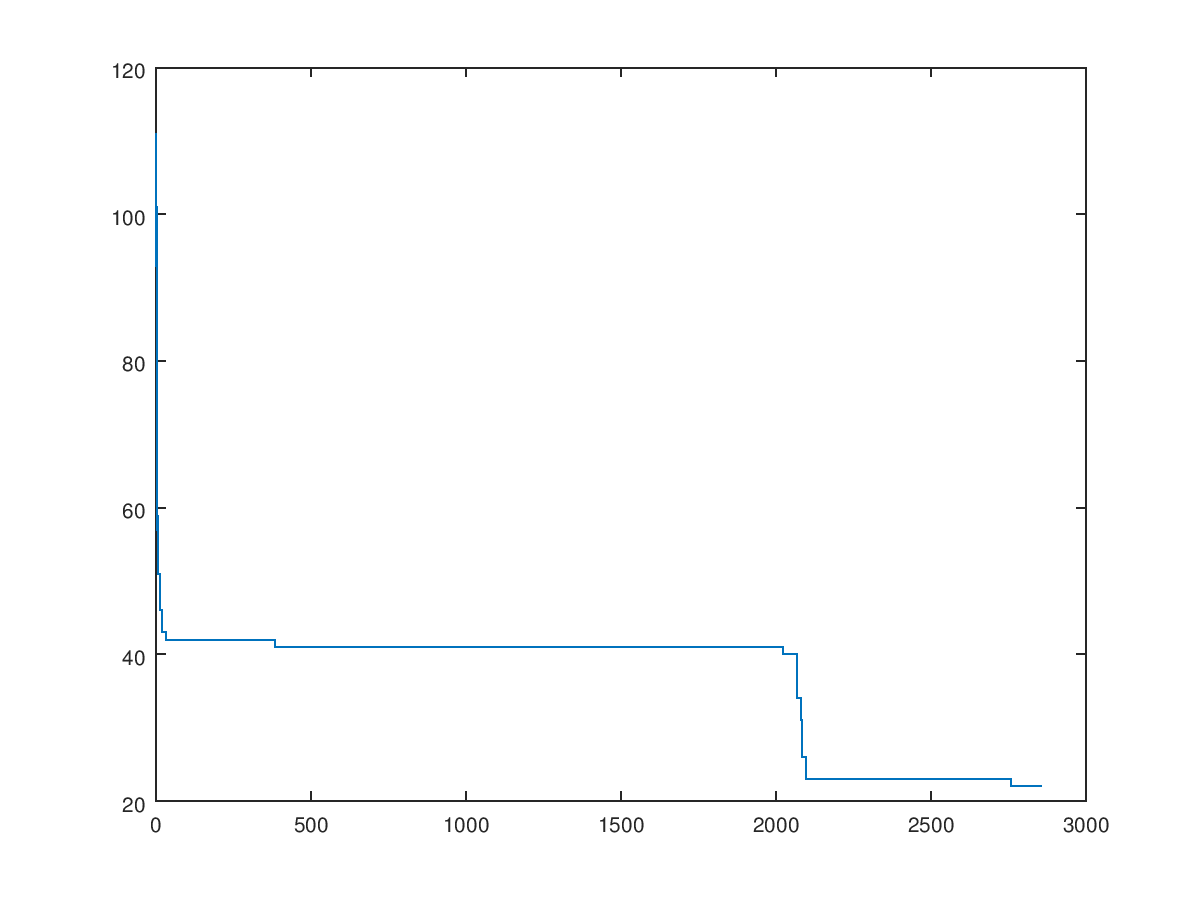


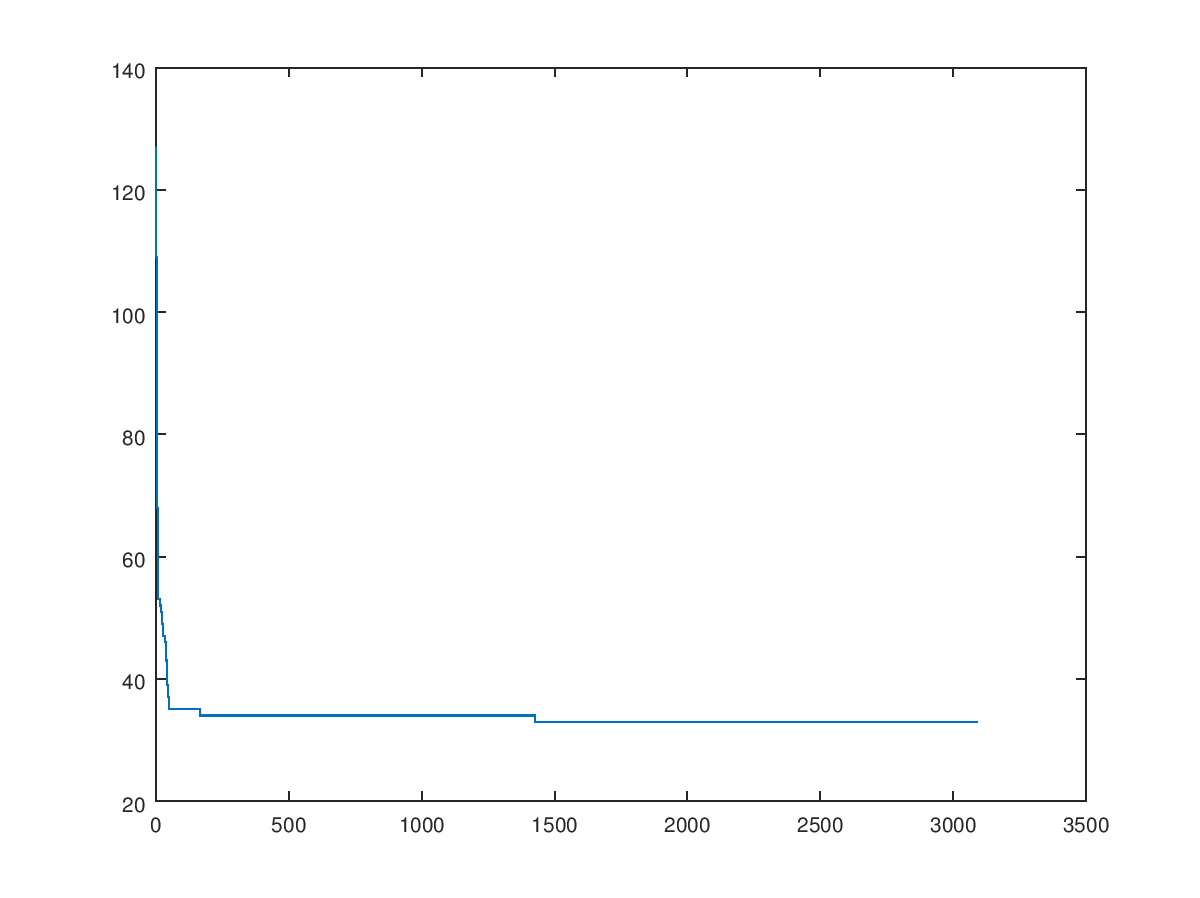
* Mutation Rate – 0.1, Crossover Rate – 0.0
* Time = 230.206 Sec
* Mutation Rate – 0.0, Crossover Rate – 0.2
* Time = 1140.01 Sec



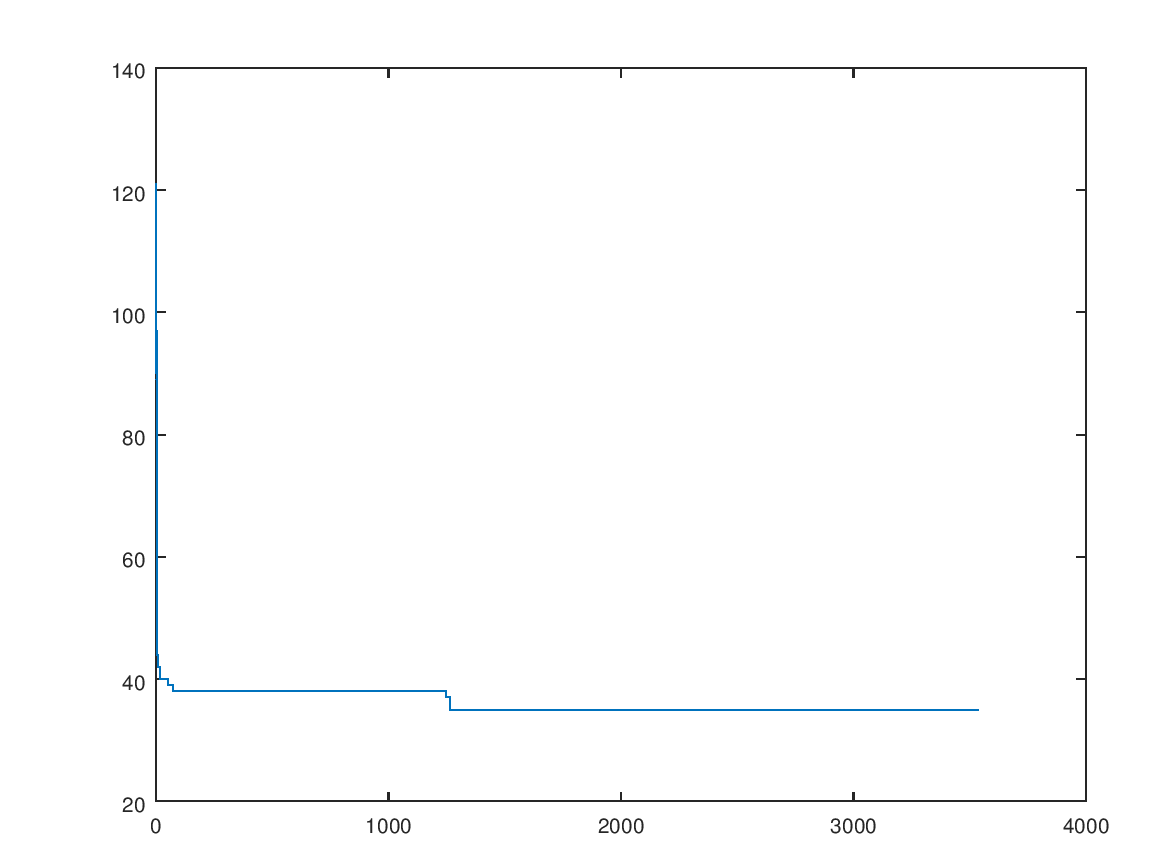
* Mutation Rate – 0.1, Crossover Rate – 0.2
* Time = 1382.08 Sec

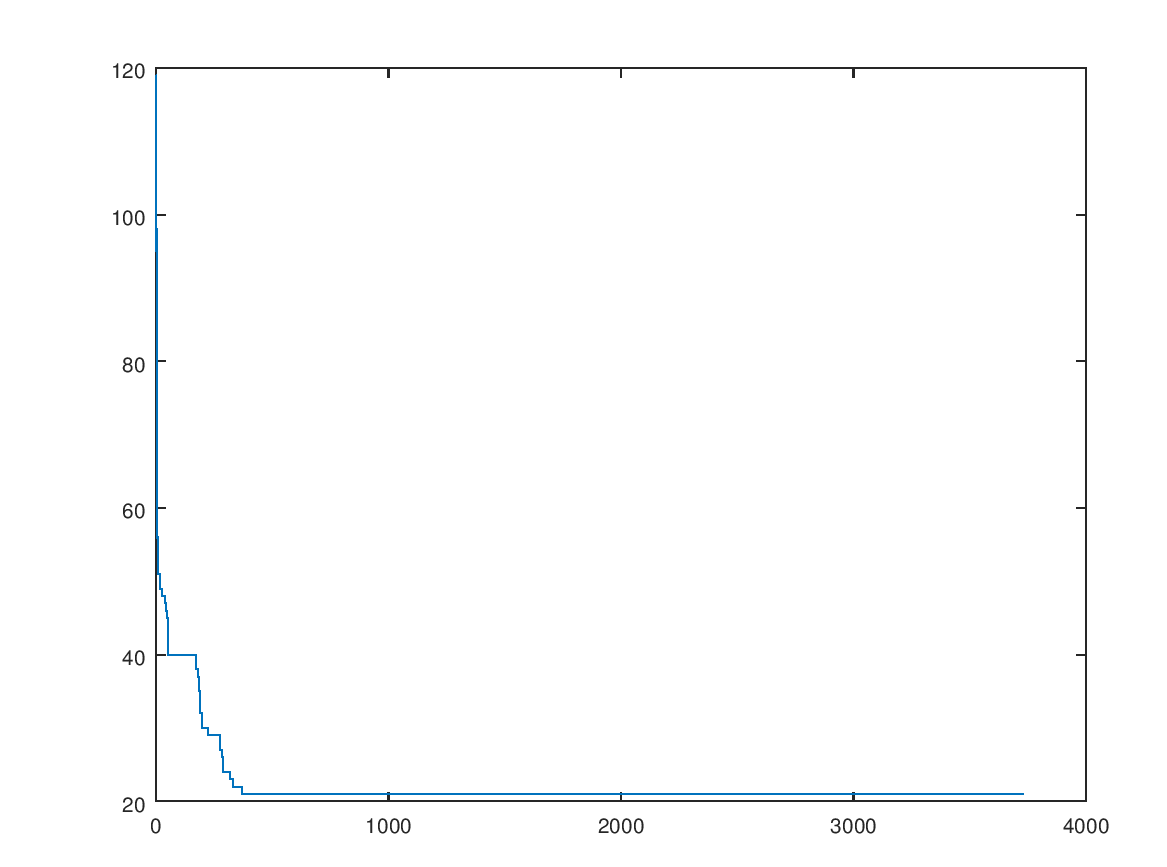


* Mutation Rate – 0.0, Crossover Rate – 0.4
* Time = 2185.12 Sec
* Mutation Rate – 0.1, Crossover Rate – 0.4
* Time = 2889.45 Sec
* Mutation Rate – 0.0, Crossover Rate – 0.5
* Time = 2857.5 Sec
* Mutation Rate – 0.1, Crossover Rate – 0.5
* Time = 3092.14 Sec



* Mutation Rate – 0.0, Crossover Rate – 0.6
* Time = 3539.55 Sec



* Mutation Rate – 0.1, Crossover Rate – 0.6
* Time = 3732.05 Sec

Population Size Data Summarization

|  |  |  |  |
| --- | --- | --- | --- |
| Population Size | Crossover Rate | Mutation Rate | Process Time |
| 2 | 0.0 | 0.0 | 73.5541 Sec |
| 0.1 | 74.2121 Sec |
| 0.2 | 0.0 | 73.1842 Sec |
| 0.1 | 76.6227 Sec |
| 0.4 | 0.0 | 84.6488 Sec |
| 0.1 | 73.7882 Sec |
| 0.5 | 0.0 | 97.9122 Sec |
| 0.1 | 104.529 Sec |
| 0.6 | 0.0 | 100.7238 Sec |
| 0.1 | 110.758 Sec |
|  |  |  |  |
| 50 | 0.0 | 0.0 | 74.52 Sec |
| 0.1 | 230.206 Sec |
| 0.2 | 0.0 | 1140.01 Sec |
| 0.1 | 1382.08 Sec |
| 0.4 | 0.0 | 2185.12 Sec |
| 0.1 | 2889.45 Sec |
| 0.5 | 0.0 | 2857.5 Sec |
| 0.1 | 3092.14 Sec |
| 0.6 | 0.0 | 3539.55 Sec |
| 0.1 | 3732.05 Sec |

For the population of 2:

* Crossover Rate – 0.5
* Mutation Rate – 0.0

For the population of 50:

* Crossover Rate – 0.0
* Mutation Rate – 0.0

Yes it does changes according to the population size.

It is better to have a small population, because when increase of population the time is also rapidly increase.

It doesn’t considered as a problem if the crossover rate and mutation rate is fixed to be zero.

* 1. Graph Size

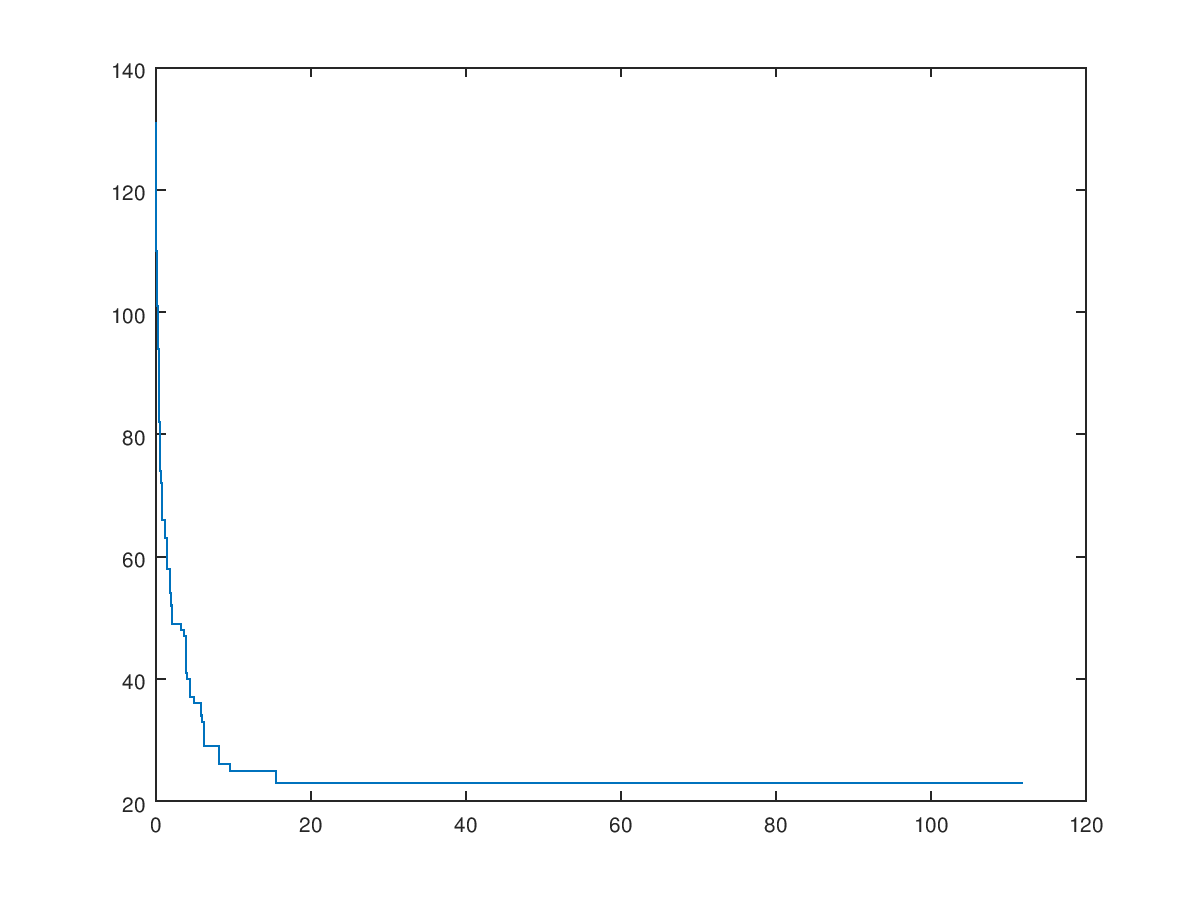
We have considered three different N values as N = {100, 1000, 10000}

Population Size = 2

Crossover Rate – 0.5, Mutation Rate – 0.0

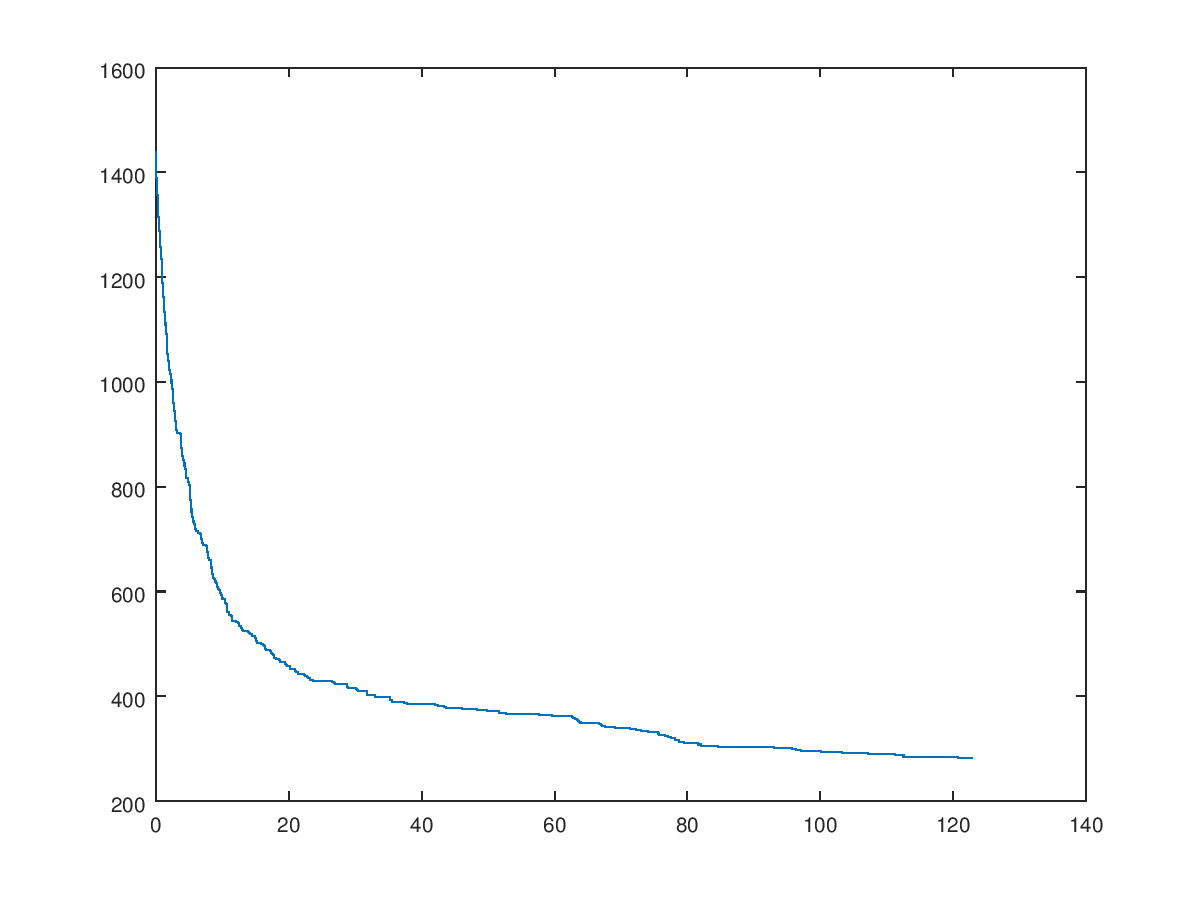
* + N = 100 (Time = 109.955Sec)

In the first 50sec the quality dip in nearly 85%



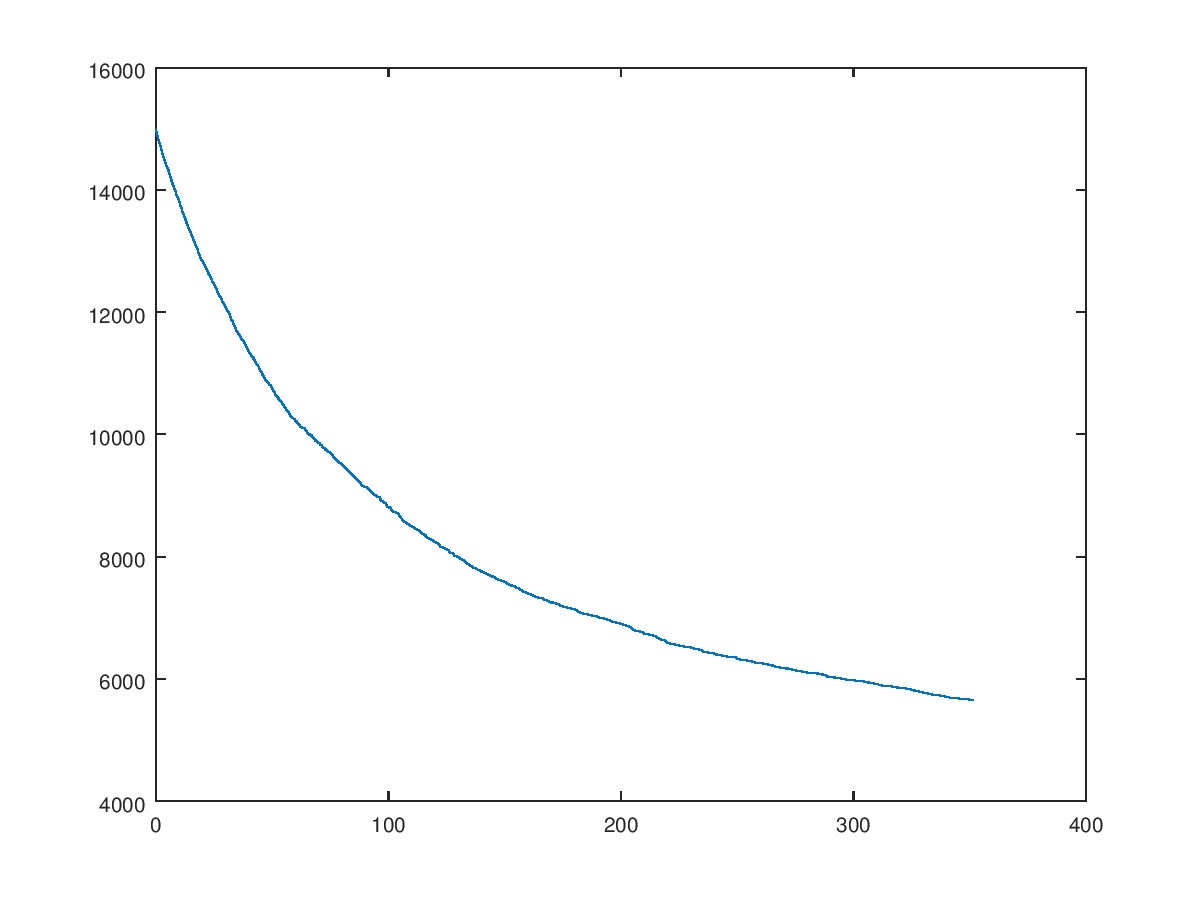
* + N = 1000 (Time = 122.517Sec)

In the first 50sec the quality dip in nearly 75%



* + N = 10000 (Time = 353.761Sec)

In the first 50sec the quality dip in nearly 35%

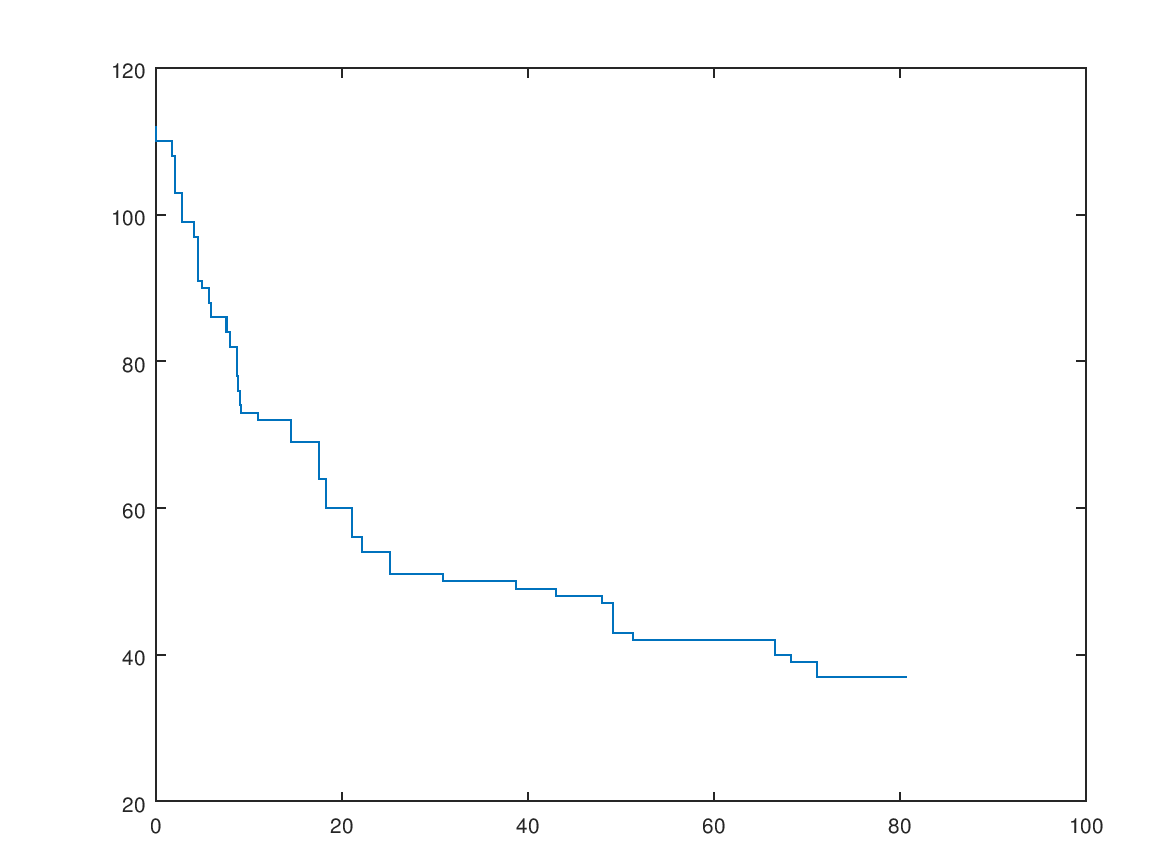


Population Size = 50

Crossover Rate – 0.0, Mutation Rate – 0.0

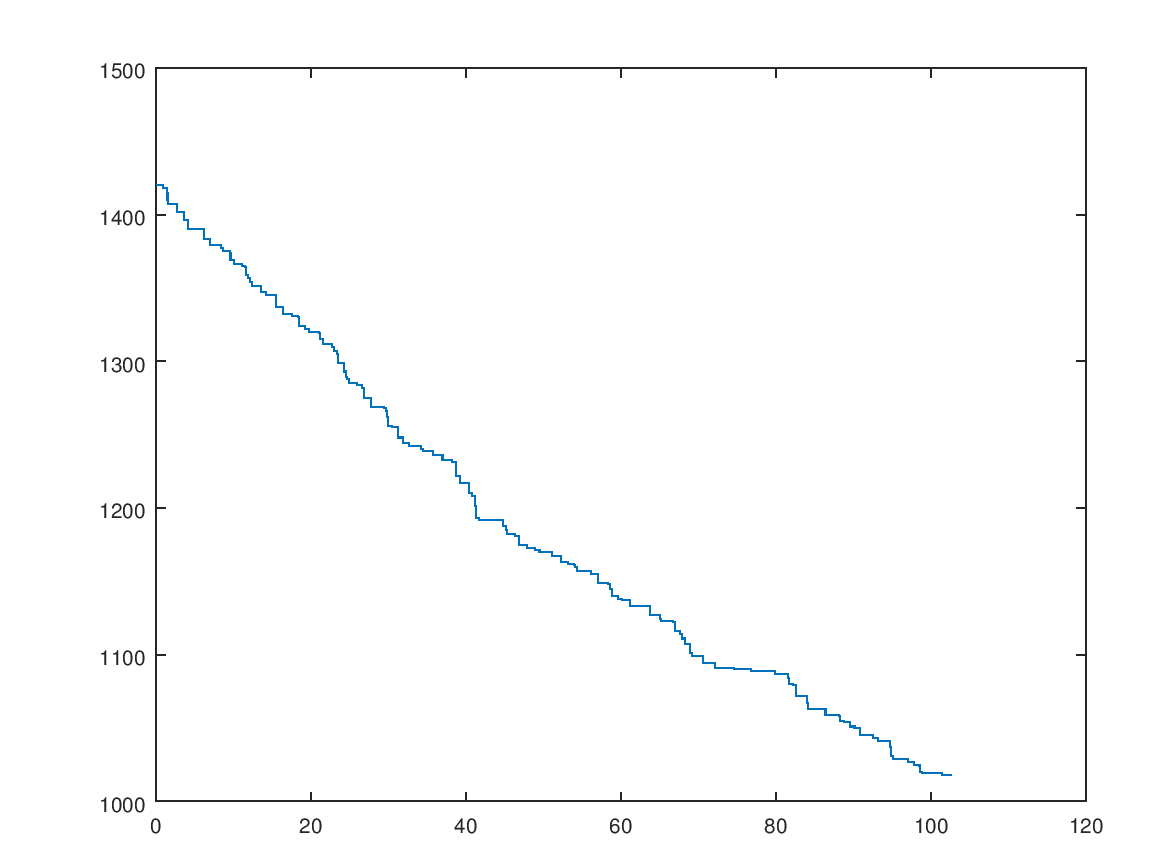
* + N = 100 (Time = 80.6758Sec)

In the first 50sec the quality dip in nearly 60%



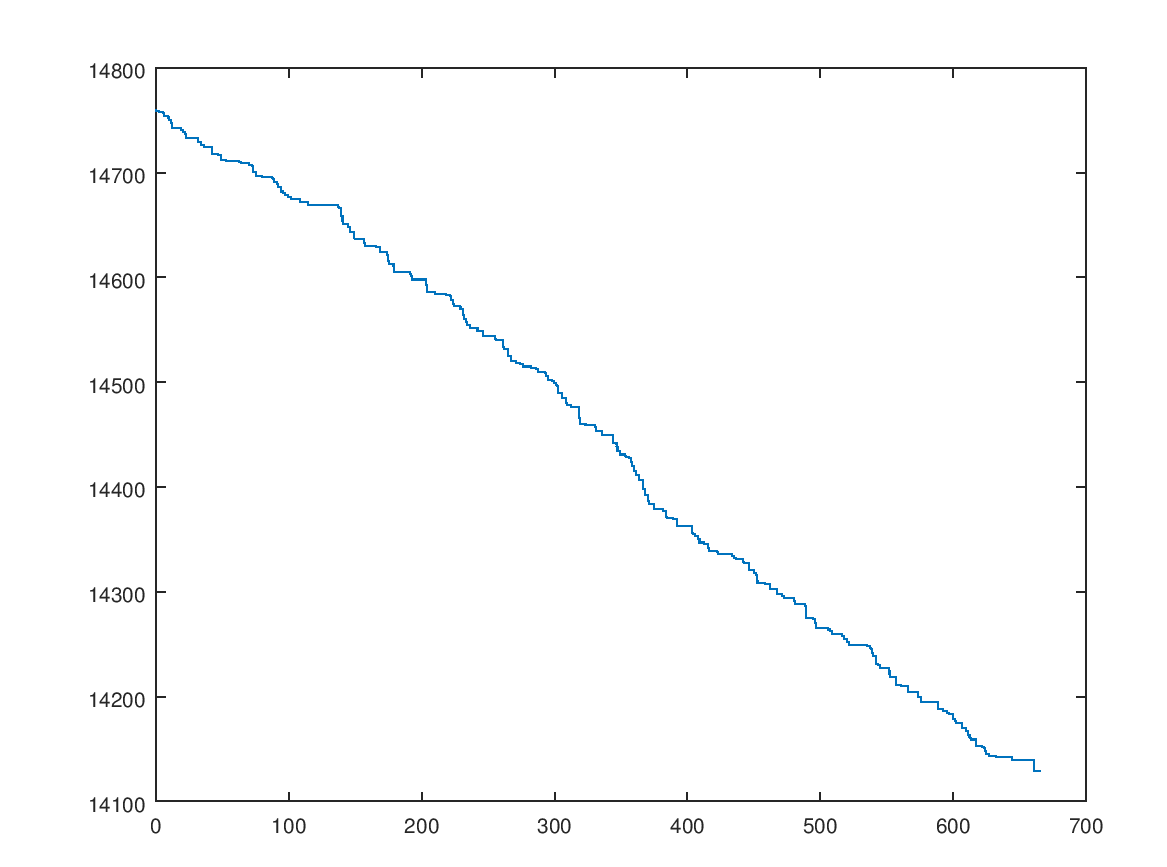
* + N = 1000 (Time = 110.66Sec)

In the first 50sec the quality dip in 9%



* + N = 10000 (Time = 686.71Sec)

In the first 50sec the quality dip in nearly 1.5%



1. Does the best population size P depend on the size of the problem N.?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P | 2 | | | 50 | | |
| N | 100 | 1000 | 10000 | 100 | 1000 | 10000 |
| Time | ≈110 | ≈122 | ≈353 | ≈80 | ≈110 | ≈668 |
| Dip in quality with the first 5Osec | 85% | 75% | 35% | 60% | 9% | 1.5% |

As the best population, we consider the small value of population so 2 is considered.

Yes, the population size depends on the N value. When N is larger, the time is increase.