Route Optimization

G R Navaneesh Kumar
Data Scientist

Problem Statement

Spring-Cleaning

Residential and Commercial Cleaning

Time – Resource – Money

What I have

How I will tackle the problem

Optimize the Time

Travel Time

Save Time - Save Resource - Gain Profit

Execution Time

Data - Available

JobExecutionTime.csv

- Stop ID
- Execution Time (Minutes)

TravelTime.csv

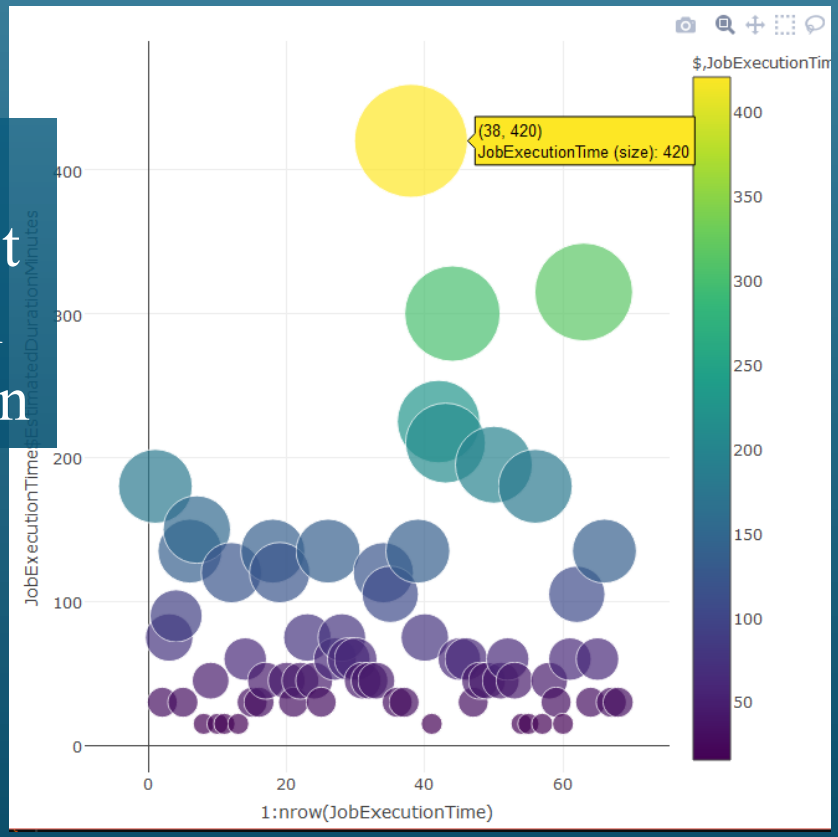
- Stop ID 1
- Stop ID 2
- Travel Time (Minutes)

Data Analysis-I

```
> summary(JobExecutionTime)
      StopId    EstimatedDurationMinutes
 12308520: 1   Min.    : 15.00
 12332932: 1   1st Qu.: 30.00
 12337737: 1   Median : 45.00
 12338526: 1   Mean    : 80.29
 12339545: 1   3rd Qu.:108.75
 12339675: 1   Max.    :420.00
 (Other) :62
>
```

Observations

1. 68 Locations Excluding Depot
2. Max execution Time :420Min
3. 75% Execution time <108 Min

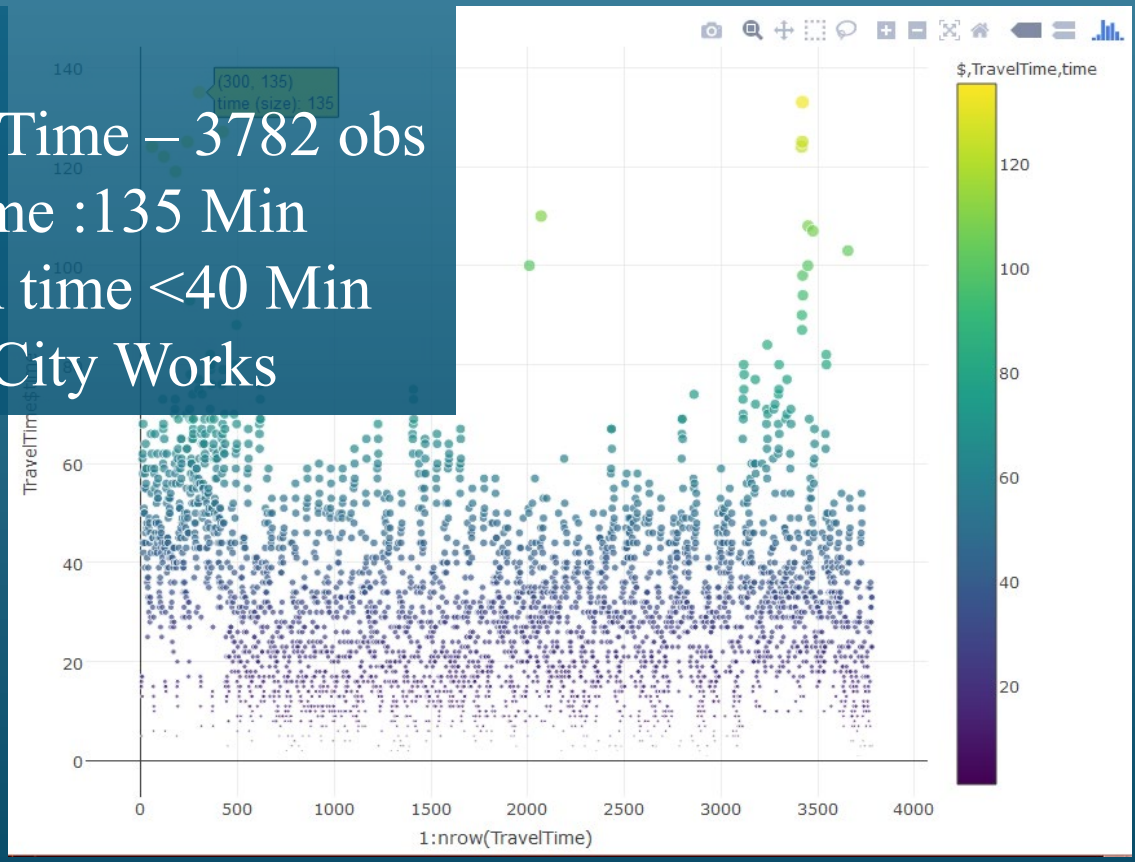


Data Analysis-II

```
> summary(TravelTime)
     stopId1          stopId2           time
 12308520:  61   12308520:  61   Min.   :  1.00
 12332932:  61   12332932:  61   1st Qu.: 17.00
 12337737:  61   12337737:  61   Median : 29.00
 12338526:  61   12338526:  61   Mean   : 30.33
 12339545:  61   12339545:  61   3rd Qu.: 40.00
 12342376:  61   12342376:  61   Max.   :135.00
 (Other) :3416   (Other) :3416
```

Observations

1. 62 x 61 Travel Time – 3782 obs
2. Max Travel Time :135 Min
3. 75% Execution time <40 Min
4. Mostly Single City Works



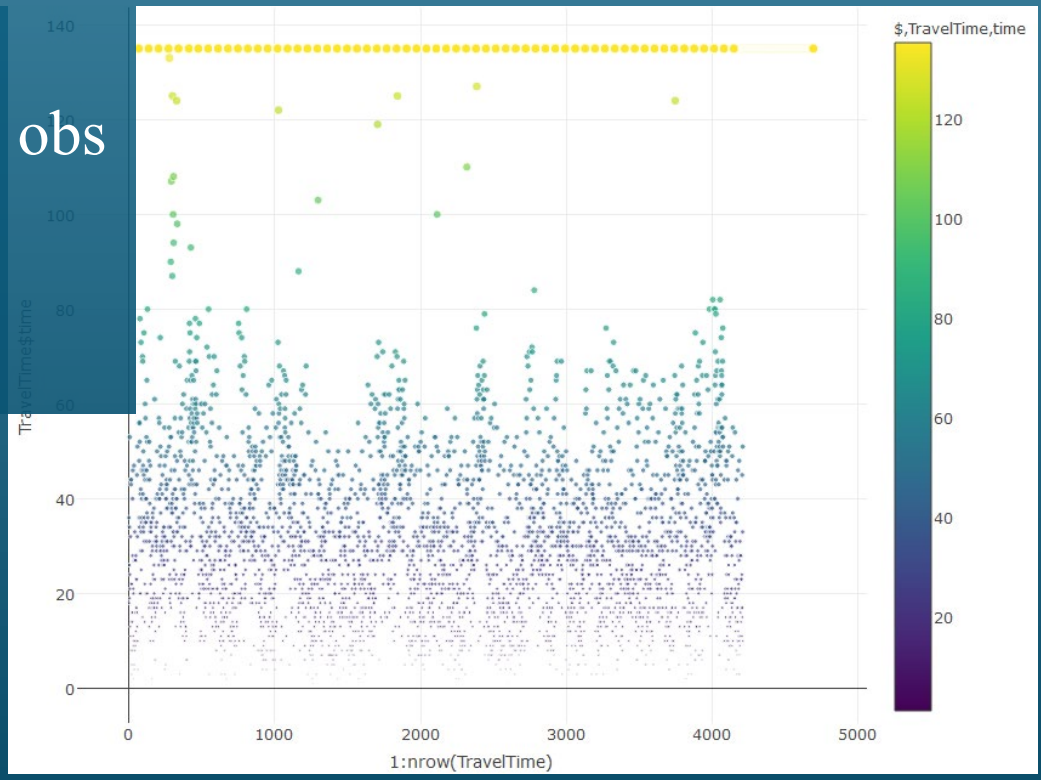
Permutation of Data

Data Analysis-III

```
> summary(TravelTime)
     stopId1          stopId2          time
 12308520:  68   12308520:  68   Min.   :  1.00
 12332932:  68   12332932:  68   1st Qu.: 20.00
 12337737:  68   12337737:  68   Median : 33.00
 12338526:  68   12338526:  68   Mean   : 50.63
 12339545:  68   12339545:  68   3rd Qu.: 57.00
 12342376:  68   12342376:  68   Max.   :135.00
 (Other) :4284   (Other) :4284
```

Observations

1. 69 x 68 Travel Time – 4692 obs
2. Max Travel Time :135 Min
3. 50% Travel time <50 Min
4. Mostly Single City Works



Critical Business Conditions

Max Route Time < 720 min

Route Time = Travel Time + Execution Time

Max places vehicle can travel is 7

Excluding Depot it starts and stops

Locations it visits should be unique

No Location repeats in the travel

Challenge :
Looks Linear

D:X_1:D - 68
D:X_1:X_2:D - 4556
D:X_1:X_2:X_3:D - 3,00,696
D:X_1:X_2:X_3:X_4:D - 1,95,45,240

Complex Solution Space

.
.
.

No Use of Linear Programming

Space and Time complexity issues

D:X_1:X_2…….X_{67}:X_{68}:D - 2.480035542 E+96

248003554243683059960099041
856917158104739920135536767 2
371710738018221445712183296 0
00000000000000

Business should not stop for our model

Genetic Algorithm:

Heuristic Method

Proved in Business, Science and Engineering

Best for searching for new solutions

Large population

Selection should be done

Survival of the fittest

Evolutionary Biology

Basics of GA

Encoding principles (Gene, Chromosome)

Gene - Stop ID – Ex: Depot, 12350484, 12350757

Chromosome - Route - Ex: Depot : 12350484:Depot

Initialization procedure (Creation)

Creation - Initial Population

Ex: Depot : 12350484:Depot

Depot : 12350484: 12350757 :Depot

Depot : 12350484: 12350757 : 12350740 :Depot

Selection of parents (Reproduction)

Here we use Business Constraints :

Ex: Stop ID (Gene) should not exceed 7
Each Stop ID should be visited only once
Total Time (Chromosome) =

Execution Time + Travel Time < 720 Min

Genetic operators (Mutation, Crossover)

Crossover - Consider Two Chromosomes (Same Length)

Route 1: Depot : 12350444 :12350222 : 12350555 : 12350666 :Depot
Route 2: Depot : 12350888: 12350333 : 12350999 : 12350777 :Depot

12350444 :12350222 12350555 : 12350666

12350888: 12350333 12350999 : 12350777

Route 1: Depot : 12350999 :12350777 : 12350555 : 12350666 :Depot
Route 2: Depot : 12350888: 12350333 : 12350444 :12350222 :Depot

Mutation

Route 1: Depot : 12350444 :12350222 : 12350555 : 12350666 :Depot

Depot 12350444 12350222 12350555 12350666 Depot

Combine

Route 1: Depot : 12350666 :12350222 : 12350555 : 12350444 :Depot

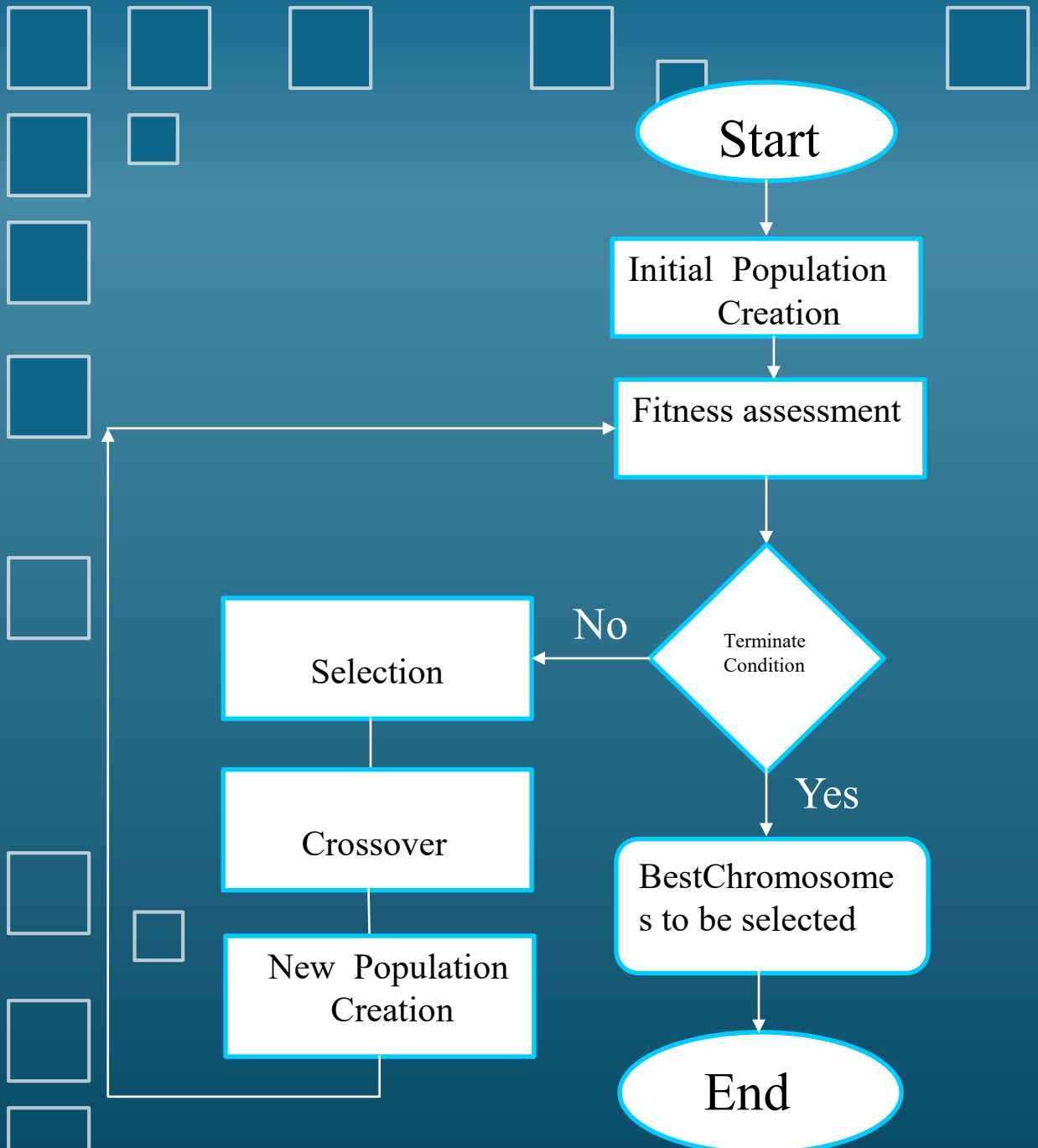
Evaluation function (Fitness)

Fitness = Fitness + Total Time

Total Time =Travel Time + Execution Time

Termination condition

Below 75% of Data after Fitness Evaluation will be removed


Start
Initial Population Creation
Fitness assessment
Terminate Condition
No
Selection
Crossover
New Population Creation
Yes
BestChromosomes to be selected
End

ADVANTAGES OF GENETIC ALGORITHMS

A fastest search technique
GAs will produce "close" to optimal results in a "reasonable" amount of time
Suitable for parallel processing
Fairly simple to develop
Makes no assumptions about the problem space

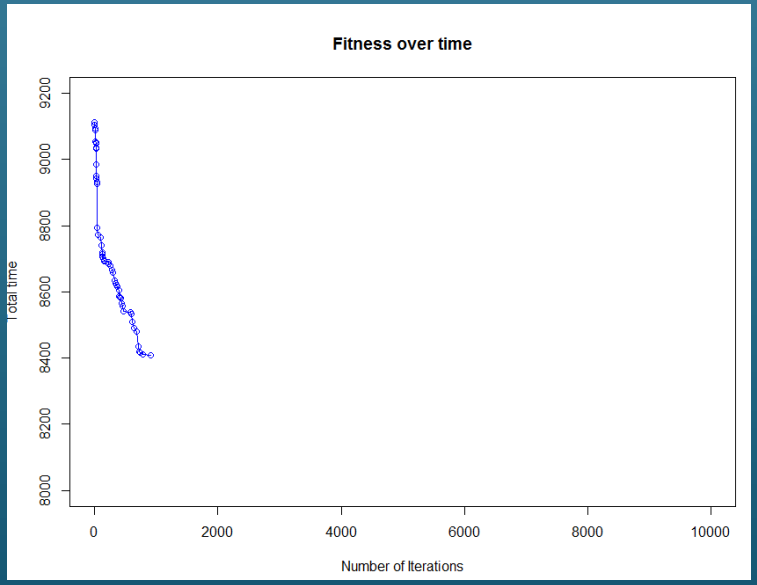
Initial Population

```
> InitialPopulation
 [1] "Depot:12351196:12348709:12345328:12348896:12350492:12350687:Depot"
 [2] "Depot:12349989:12351049:12350484:12350818:Depot"
 [3] "Depot:12347928:12341082:12351769:12347871:12351217:Depot"
 [4] "Depot:12350840:12351272:12350757:12349786:12348335:12349516:12342376:Depot"
 [5] "Depot:12351488:12347260:12350406:12348803:Depot"
 [6] "Depot:12349005:12343415:12349651:12308520:12337737:Depot"
 [7] "Depot:12332932:12348694:12350656:12351082:12346603:12349706:12349283:Depot"
 [8] "Depot:12348665:12350265:12350468:12348444:12350842:12350358:Depot"
 [9] "Depot:12339545:12348388:12350103:12349074:Depot"
[10] "Depot:12349842:12349553:Depot"
[11] "Depot:12348192:12346752:12345229:12350684:12349009:Depot"
[12] "Depot:12350501:12351024:12350627:12345730:Depot"
[13] "Depot:12350758:12339675:12349968:12351315:Depot"
[14] "Depot:12349057:12350135:12338526:Depot"
[15] "Depot:12350822:12351426:Depot"
```

Total Time ,Unique Ids

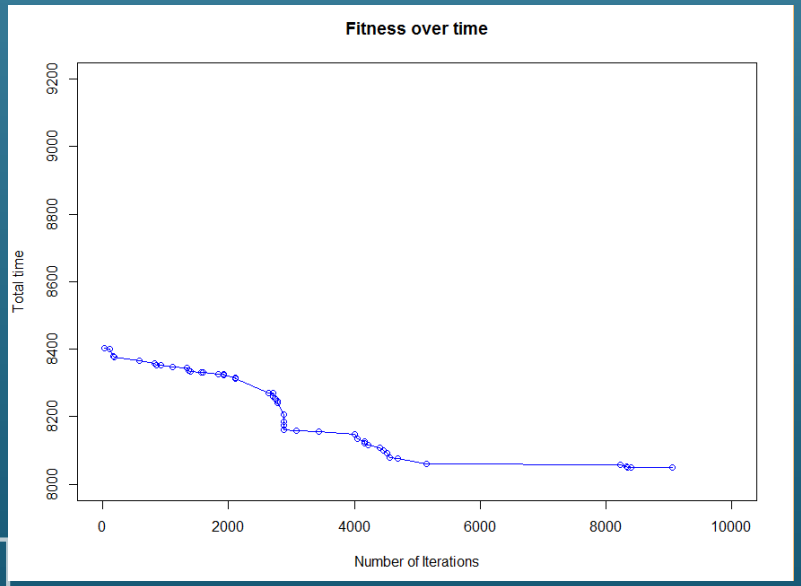
```
> totaltime(InitialPopulation)
[1] 9120
> StopIdsUnique(InitialPopulation)
[1] 68
>
```

First 1000 Iterations




```
> InitialPopulation = GA(1000,0.75,0.2)
Iteration :   1    | New Population fitness -  9113
Iteration :   3    | New Population fitness -  9105
Iteration :   4    | New Population fitness -  9094
Iteration :   6    | New Population fitness -  9088
Iteration :   7    | New Population fitness -  9055
Iteration :   19    | New Population fitness -  9051
Iteration :   20    | New Population fitness -  9050
Iteration :   22    | New Population fitness -  9037
Iteration :   27    | New Population fitness -  9033
Iteration :   28    | New Population fitness -  8984
Iteration :   29    | New Population fitness -  8951
Iteration :   30    | New Population fitness -  8943
Iteration :   41    | New Population fitness -  8933
Iteration :   42    | New Population fitness -  8927
Iteration :   45    | New Population fitness -  8795
Iteration :   52    | New Population fitness -  8772
Iteration :   96    | New Population fitness -  8765
Iteration :   117    | New Population fitness -  8740
Iteration :   118    | New Population fitness -  8720
Iteration :   122    | New Population fitness -  8715
Iteration :   128    | New Population fitness -  8705
Iteration :   133    | New Population fitness -  8700
Iteration :   147    | New Population fitness -  8696
Iteration :   159    | New Population fitness -  8694
Iteration :   229    | New Population fitness -  8691
Iteration :   230    | New Population fitness -  8684
Iteration :   251    | New Population fitness -  8680
Iteration :   281    | New Population fitness -  8665
Iteration :   297    | New Population fitness -  8657
Iteration :   328    | New Population fitness -  8633
Iteration :   343    | New Population fitness -  8627
Iteration :   349    | New Population fitness -  8621
Iteration :   364    | New Population fitness -  8616
Iteration :   391    | New Population fitness -  8605
Iteration :   401    | New Population fitness -  8586
Iteration :   403    | New Population fitness -  8583
Iteration :   428    | New Population fitness -  8581
Iteration :   437    | New Population fitness -  8565
Iteration :   448    | New Population fitness -  8556
Iteration :   461    | New Population fitness -  8540
Iteration :   578    | New Population fitness -  8538
Iteration :   601    | New Population fitness -  8534
Iteration :   615    | New Population fitness -  8509
Iteration :   636    | New Population fitness -  8491
Iteration :   681    | New Population fitness -  8481
Iteration :   716    | New Population fitness -  8434
Iteration :   726    | New Population fitness -  8418
Iteration :   736    | New Population fitness -  8415
Iteration :   779    | New Population fitness -  8411
Iteration :   908    | New Population fitness -  8407
>
```

Next 10000 Iterations




```
> InitialPopulation = GA(10000,0.75,0.2)
Iteration :  23   | New Population fitness -  8402
Iteration :  115   | New Population fitness -  8401
Iteration :  165   | New Population fitness -  8378
Iteration :  180   | New Population fitness -  8377
Iteration :  585   | New Population fitness -  8366
Iteration :  820   | New Population fitness -  8359
Iteration :  854   | New Population fitness -  8353
Iteration :  927   | New Population fitness -  8352
Iteration :  1109   | New Population fitness -  8346
Iteration :  1339   | New Population fitness -  8345
Iteration :  1363   | New Population fitness -  8336
Iteration :  1399   | New Population fitness -  8335
Iteration :  1568   | New Population fitness -  8332
Iteration :  1588   | New Population fitness -  8330
Iteration :  1833   | New Population fitness -  8326
Iteration :  1923   | New Population fitness -  8325
Iteration :  1926   | New Population fitness -  8324
Iteration :  2110   | New Population fitness -  8314
Iteration :  2112   | New Population fitness -  8312
Iteration :  2638   | New Population fitness -  8270
Iteration :  2708   | New Population fitness -  8269
Iteration :  2714   | New Population fitness -  8260
Iteration :  2730   | New Population fitness -  8254
Iteration :  2753   | New Population fitness -  8251
Iteration :  2776   | New Population fitness -  8247
Iteration :  2784   | New Population fitness -  8240
Iteration :  2873   | New Population fitness -  8207
Iteration :  2874   | New Population fitness -  8186
Iteration :  2880   | New Population fitness -  8174
Iteration :  2885   | New Population fitness -  8160
Iteration :  3084   | New Population fitness -  8158
Iteration :  3435   | New Population fitness -  8156
Iteration :  4001   | New Population fitness -  8148
Iteration :  4051   | New Population fitness -  8134
Iteration :  4162   | New Population fitness -  8126
Iteration :  4167   | New Population fitness -  8120
Iteration :  4215   | New Population fitness -  8117
Iteration :  4411   | New Population fitness -  8108
Iteration :  4456   | New Population fitness -  8099
Iteration :  4523   | New Population fitness -  8092
Iteration :  4568   | New Population fitness -  8079
Iteration :  4691   | New Population fitness -  8076
Iteration :  5149   | New Population fitness -  8060
Iteration :  8227   | New Population fitness -  8058
Iteration :  8329   | New Population fitness -  8053
Iteration :  8337   | New Population fitness -  8050
Iteration :  8398   | New Population fitness -  8049
Iteration :  9062   | New Population fitness -  8048
>
```

What I Achieved

Initial Time : 9113
Current Time :8048
Saved Time :1065

Nearly :17Hrs 45Min
A Nearly Two Working Days

Future Scope

- Courier Services
- Food and Groceries Delivery App
- Container Shipping Industry
- And More

Q and A

Hasta la vista

-- Navaneesh Gangala