Genetic Algorithm for Decision Making

Problem statement : Deciding combination of crops to be planted such that profit is maximum and risk is minimum.

Programming language - *Python3*
Algorithm - *Genetic Algorithm*
Library used - *Distributed Evolutionary Algorithms in Python (DEAP)*
Data Set - *Gudur Rythu Bazar 2017*

Objective Function :

$$\boxed{\mathbf{Max(}\ (w1 \times P - w2 \times R) / (w1 + w2)\ \mathbf{)}}$$

Where, P - Profit
R - Risk
$w1$ - Weightage of profit
$w2$ - Weightage of risk

Profit (P) **:** Profit earned on specific crop on a specific month in 1acre.

Risk (R) **:** There are several types of risks in mixed cropping, Few of which considered for this problem are Root System, Water Requirement, Market Volatility.

$$\boxed{R = (r1 \times R_{rs} + r2 \times R_{wr} + r3 \times R_{mv}) / (r1 + r2 + r3)}$$

Where, R_{rs} - Risk due to root system
R_{wr} - Risk due to water requirement
R_{mv} - Risk due to market volatility
r1, r2, r3 - Weights

Risk due to root system (R_{rs}) **:**

- No crop with similar root system should be planted side by side. Root system types, Shallow, Medium, Deep.
- Crop with similar root system as the crop in the previous cycle at a location should not be planted.

Risk due to water requirement (R_{wr}) :

- Cops which consume less water are better in places with water scarcity.
- In places with abundant water supply ‘r2’ is zero as it does not contribute to risk.

Risk due to market volatility (R_{mv}) :

- This accounts risk due to market fluctuation which is calculated using standard deviation.

Std(12 months**)** x **Sqrt(**12**)**

- According to market standards, Standard deviation is multiplied by Square root of 12 if we calculate for a year with monthly interval.

On Violation of any of the conditions a specific constant value is added which increases the risk value, So this helps the Objective function to converge.

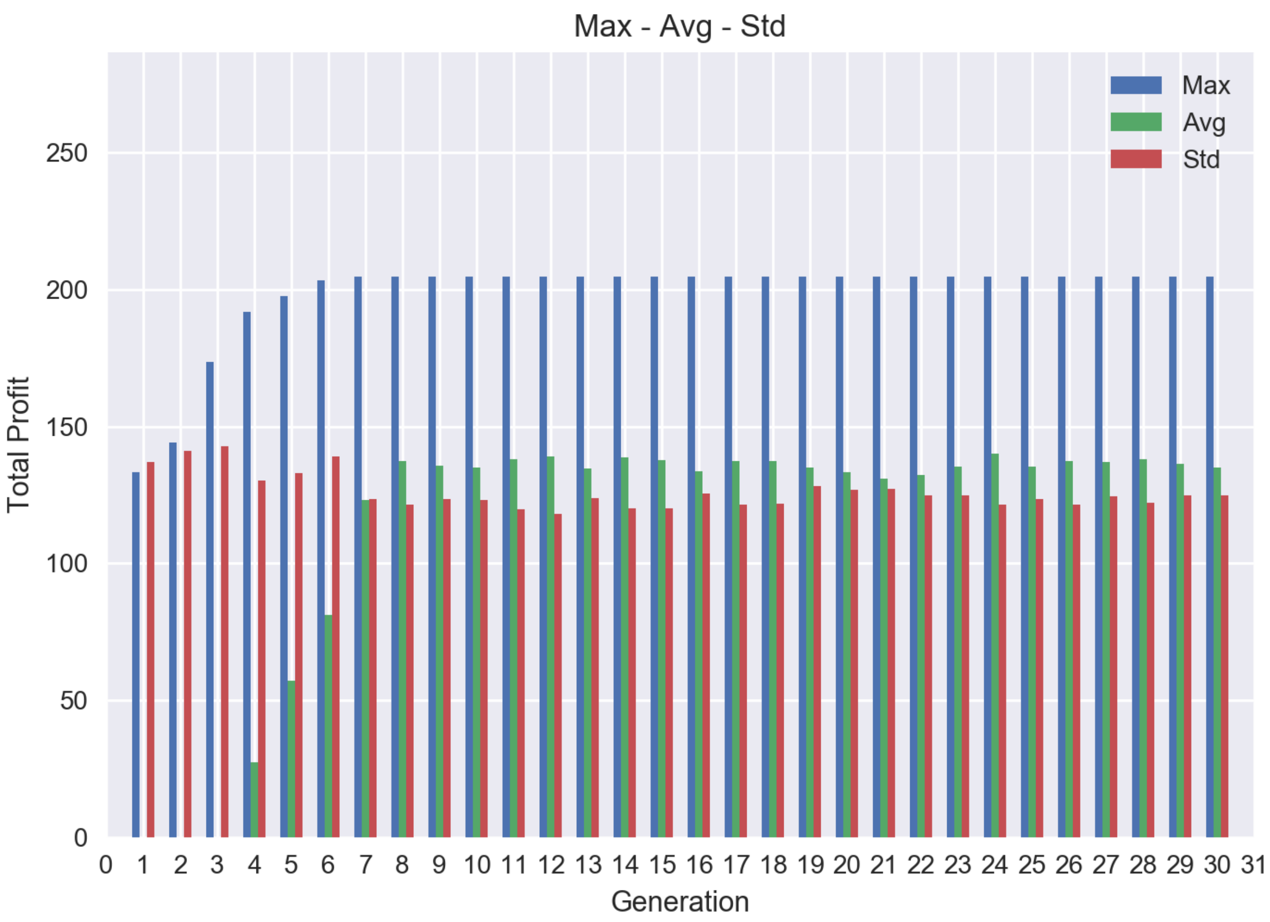
n - *no.of crops available*
m - *no.of crops to be planted in each crop cycle*
c - *no.of crop cycles*
Individual size : *c x m*

Custom Crossover and mutation functions are use to maintain the uniqueness of the individual which always outputs individual with unique gens in each cycles.

Test Solution :
n = 20
m = 5
c = 3
w1 = 0.5
w2 = 0.5
r1 = 0.7
r2 = 0
r3 =0.3
N = 3000 (no.of individuals in each generations)
Ngen = 30 (no.of generations)
CXPB = 0.7 (probability with which two individuals are crossed)
MUTPB = 0.4 (probability for mutating an individual)
INDPB = 0.2 (probability for mutating each gene of an individual)

Output for 3x5 individual :

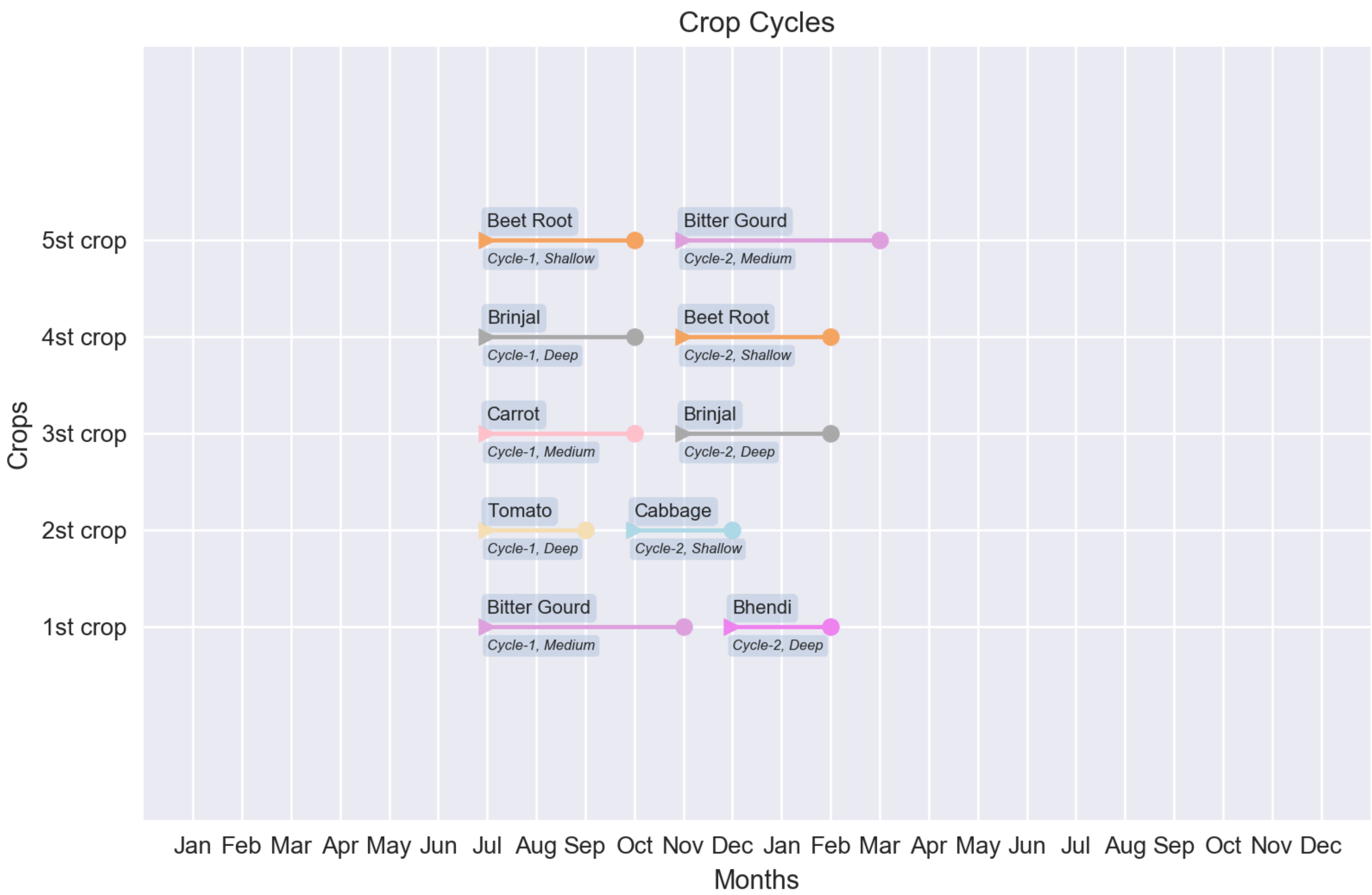
- Fitness value over generations, Optimised at 7th generation.



- Terminal Output

```
Profit          : 4300000.0
Risk            : -20.627219157906033
Combined_val    : 204.68639042104698
Risk_root       : [0, 0]
Risk_water      : [150, 190]
Volatility      : [40.6173882518157, 28.140008941204417]
Risk_list       : [0, 170.0, 68.75739719302011]
[[3, 20, 7, 5, 1], [2, 6, 5, 1, 3]]
204.68639042104698
+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
|     Cycles     |      Crop      | Planting Month | Harvest Month  |    Root Sys    |   Water Req    |     Profit     |
+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
|       1        |  Bitter Gourd  |      July      |    November    |     Medium     |       M        |     320000     |
|       1        |     Tomato     |      July      |   September    |      Deep      |       M        |     256000     |
|       1        |     Carrot     |      July      |    October     |     Medium     |       M        |     426000     |
|       1        |    Brinjal     |      July      |    October     |      Deep      |       M        |     835000     |
|       1        |   Beet Root    |      July      |    October     |    Shallow     |       M        |     520000     |
| -------------- | -------------- | -------------- | -------------- | -------------- | -------------- | -------------- |
|       2        |     Bhendi     |    December    |    February    |      Deep      |       H        |     275000     |
|       2        |    Cabbage     |    October     |    December    |    Shallow     |       H        |     459000     |
|       2        |    Brinjal     |    November    |    February    |      Deep      |       M        |     635000     |
|       2        |   Beet Root    |    November    |    February    |    Shallow     |       M        |     380000     |
|       2        |  Bitter Gourd  |    November    |     March      |     Medium     |       M        |     194000     |
| -------------- | -------------- | -------------- | -------------- | -------------- | -------------- | -------------- |
|    Total :     |       -        |       -        |       -        |       -        |       -        |    4300000     |
+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
```

Visual Representation :



Still to Improvements :

- Make it scalable for bigger individuals and bigger datasets.
- Improve crossover and mutation function to preserve uniqueness of the individuals
- Parellel operations.