Genetic Algorithm for Bee Pollination

Allison Solano, Luis Mariano Ramírez
Algorithm Analysis - IC3002
Instituto Tecnológico de Costa Rica
Project Repository: https://github.com/Lumanter/Genetic-Bees

Abstract*— *In this work a genetic algorithm is presented to simulate pollination interactions inside a field of flowers, to optimize the bees number of pollinated flowers and minimizing the distance traveled. Upon optimizing the algorithm parameters, a strong tendency of the flowers to cluster and be of the same color could be seen. And the bees showed an increasing tendency of pollinated flowers, and search distance, considered as a satisfactory result.

Keywords*— *genetic, evolution, optimization, bee, flower

I. Introduction

The aim of the project is to optimize two populations: the bees, for more pollinated flowers and less distance traveled in its search, and the flowers, for adapting to bee preferences. The interactions occur in a flower field matrix of 128 by 128, with a honeycomb on its center. The flowers have color and a given position inside the flower field, and passively wait to be pollinated. The bees travel an area of the flower field, determined by its genetic traits, searching specially for flowers of their favorite color. When a bee pollinates a flower it inserts in the flower the pollen of all the previous flowers that it has visited.

A bee is considered a better solution when it has collected more pollen and has traveled less search distance, bees that follow this characteristic are more likely to reproduce to create the next generation. Flowers are considered a better solution when it simply has more pollen. The flower population is expected to evolve following to some extent the bees' preferences. This project is implemented in python 3.

II. Related Work

A. Related Applications of Genetic Algorithms

Optimization. They are used in optimization problems in which it is necessary to maximize or minimize an evaluation function value under a given set of constraints.

Neural networks. They are mainly used to train neural networks, commonly recurrent neural networks (RNN)

Financial sector. They are used to find the values of combination of parameters in a negotiation rule and to be able to be incorporated in the models of artificial neural networks designed to select actions and identify operations.

Use of genetic algorithms for search and evaluation of strategies in HU-NLHE (poker variant). Where populations of players with parameterizable strategies evolve until they find an optimal strategy for different game situations. Where the effectiveness of genetic algorithms is found in the search for solutions and comparing said solutions and also for selection of the best players, regarding a fitness that is their games played divided between games won and a selection of Russian roulette [1].

B. Genetic Algorithm Techniques

A simple way of doing crossover is the single point crossover. In this crossover algorithm the genes of both parents are taken, of same length, a random point within the genes is chosen to cut the genes. The opposite resulting cut genes are mixed to produce a new pair of genes.

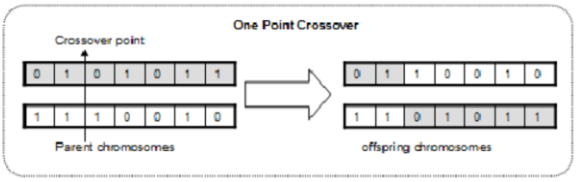

Fig 1. Single point crossover [2].

In the mutation side, a simple way of performing it is flip-bit mutation. In this algorithm a chance of flipping a bit exists, and the genes as a list of bits is traversed, applying the flipping chance to each bit.

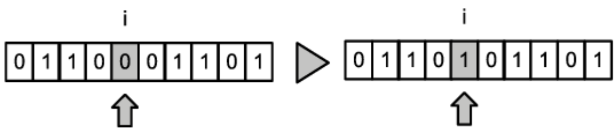

Fig 2. The flip-bit mutation [3].

III. Methods

A. Flower Genes

The genes of a flower are its field coordinates and its color.

- Its color is a form of binary rgb, with 3 bits and 8 colors.
- Since the flower field has side 128, the x and y values are in the [0, 127] range, with 7 bits.

B. Bee Genes

The bees have 6 genetically coded traits: favorite direction, favorite flower color, deviation angle, search radius, search strategy and if starts at honeycomb.

- Direction indicates the direction in which the bee searches for flowers from the center of the field, can be any of the 8 cardinal directions, so 3 bits.
- Favorite flower color contributes significantly in which flowers the bee pollinates, and is treated the same as the flower color gene with 3 bits.
- Deviation angle indicates how much the bee differs from its base search direction, has values in the [0, 31] range, with 5 bits.

- Search radius determines the flower search area, allows values in the [0, 90] range, with 7 bits.
- Search strategy indicates one of three strategies: depth, breadth or random. Allows values in the [0, 2] range, with 2 bits.
- If the bee starts at honeycomb indicates if the bee starts its search at the honeycomb or at a field border its search direction, with 1 bit.

Two cases of illegal values are adjusted. If the search radius happens to be above 90, a random value in the [38, 90] range is subtracted, ensuring a valid radius. If the search strategy is 3, due to the little range, a random value in the [0, 2] is chosen.

C. Genetic Algorithm

In the selection phase, the flowers that are ensured to be selected are those who have pollen. For the bees, a fitness is calculated with the next formula:

(pollinated flowers x constant) - search distance

Fig 3. Bee fitness formula.

The constant is tweaked to give more or less weight to the number of pollinated flowers. To select the best candidates for crossover the roulette wheel selection algorithm is applied, giving more selection probability to those with better fitness.

For the crossover, a single point crossover algorithm is applied, via choosing a random index within the genes. A bee needs another bee to reproduce, producing two children. The flower, on the other hand, chooses randomly one of its pollen to reproduce, producing a single child.

Mutation is applied using the flip-mutation algorithm.

The stopping condition to consider a generation to have reached the goal is the average fitness of the entire population reaching a given value. It was decided this way over stopping based on the fitness of an individual to handle a randomly generated high fitness when the rest of the population is not in a satisfactory state.

D. Flower Search

This is divided into two parts, the first is to determine the flowers that are in the area that is generated with the favorite direction and deviation of the bee, for this ray casting algorithm was used [4]. The second is the mere part of the search, this is carried out in a tree according to the type of search indicated by the bee's chromosome.

A ternary tree is used, its nodes are ordered with respect to the distance in which the flower is from the starting point of the bee, those closest to the root are the closest to the point. When the bee arrives at a node, it has the option to visit it or not, if this flower is its favorite color, it visits, otherwise a probability is applied. In this way it searches the entire tree, in the case of the random search the bee moves over the list generated by the ray casting with random indexes and applying the same conditions to visit the flowers.

IV. ANALYSIS OF RESULTS

It was decided to have 100 generations maximum and let the constant in the bee fitness formula be 100, giving more importance to pollinated flowers than search distance, as a point of reference for the other parameters.

In the bee population, the bigger the population, the bigger would be the expected amount of generations needed to converge upon flower preferences, so a number of 20 was chosen. For the flower population, a big number of flowers would make it difficult to read the tendencies, specially to determine when the flowers were scattered, and a little number would make the color converge too fast and difficult to determine when the flowers were clustered. Also, a big number of flowers would be taxing on execution time, making the process of testing and running slow. So, given the dimensions of the field, a value of 200 was accepted as suitable.

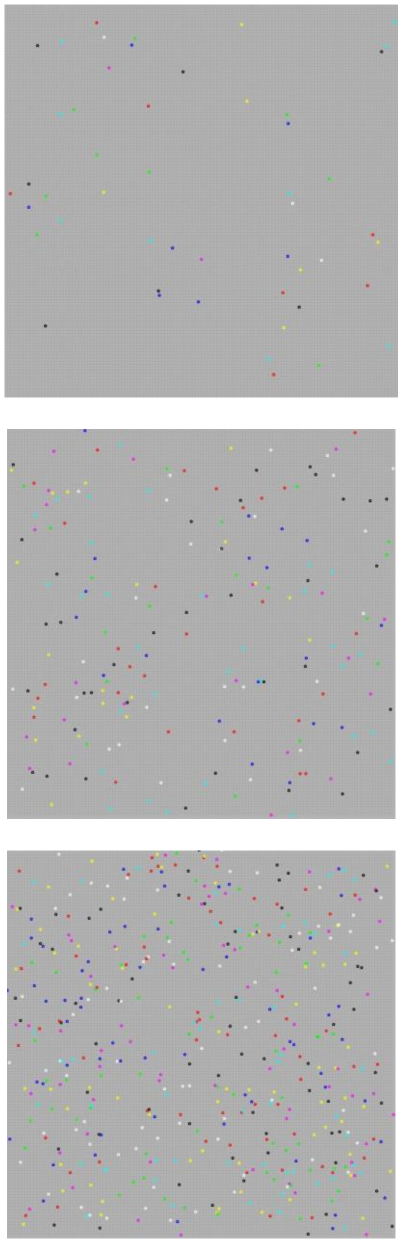

Fig 4. Random population of flowers: 50 vs 200 vs 500.

The next step was to choose a stopping value for the average fitness of the bee population. It was necessary to determine what was considered an acceptable solution. The wanted field would have a significant amount of flowers of the same color and to have most of its flowers clustered in the same field quadrant. A field appearance on the value 5500 was accepted as a good solution, given that the bees very rarely reached 7000.

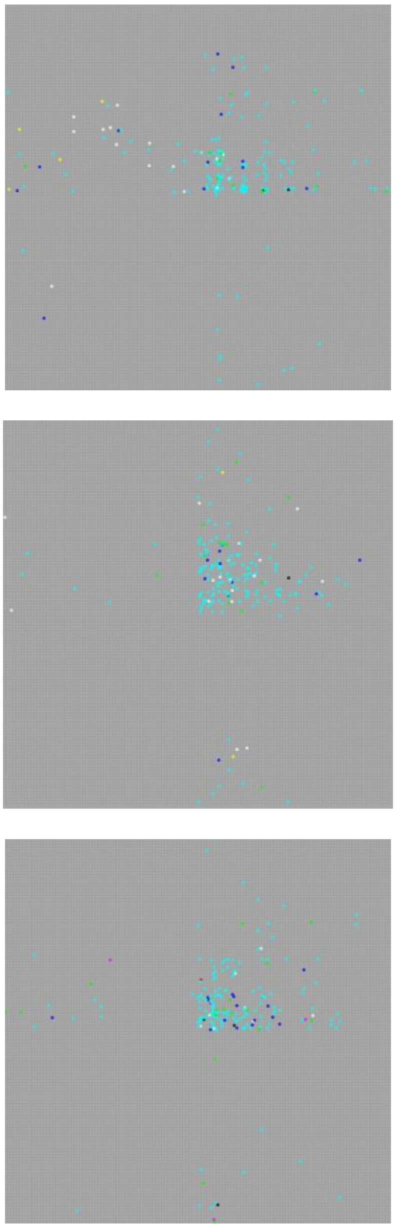

Fig 5. Average bee generation fitness: 4000 vs 5500 vs 7000.

For this point three values need tweaking: the probability of a bee visiting a flower that is not of his favorite color and the mutation probabilities. The result wanted is the simulation not converging to fast in a set of results or not converging never at all. So the values were adjusted looking for the goal to be reached mostly between the 40 and 100 generation. A low mutation chance means that the genes will converge fast, the same goes to a low chance of visiting a non-favorite flower, higher values will mean that the goal is not going to be reached within the generations established.

Ten executions were done to compare the stopping generation of probabilities 3%, 6% and 7%. It is important to note that a stopping generation of 100 is being taken as the execution fails to fulfill the fitness goal, so the expected value of the last percent has to be read with that in mind. The accepted value was 6%, with an expected value of 60.6.

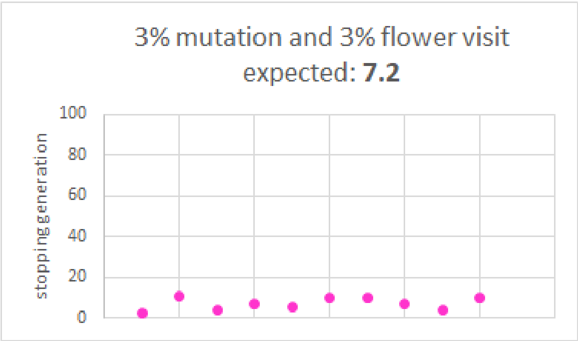

Fig 6. Stopping generation of probability 3%.

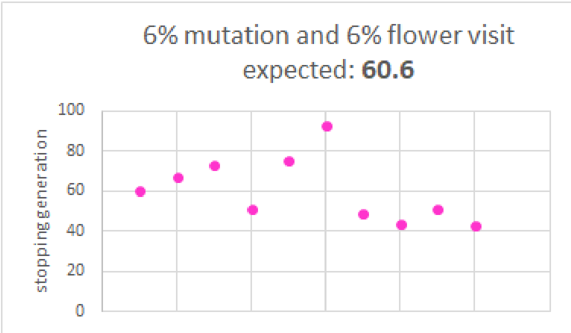

Fig 7. Stopping generation of probability 6%.

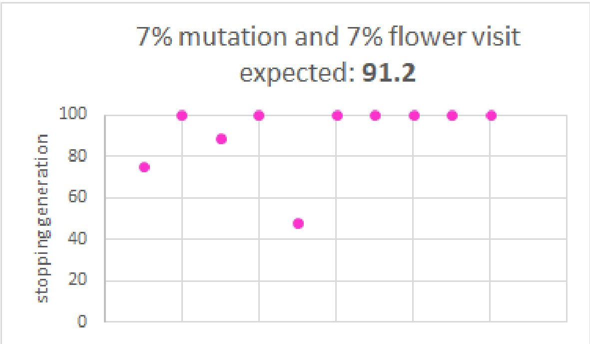

Fig 8. Stopping generation of probability 7%.

With all these parameters set, the tendencies of pollinated flowers and search distance can be seen. The following graphs were created by extracting 20 samples from an execution of 70 generations.

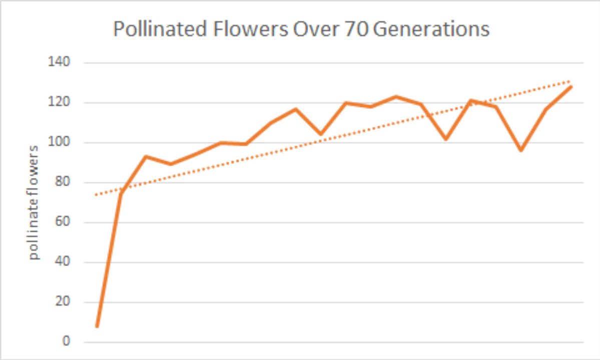

Fig 9. Pollinated flowers over 70 generations.

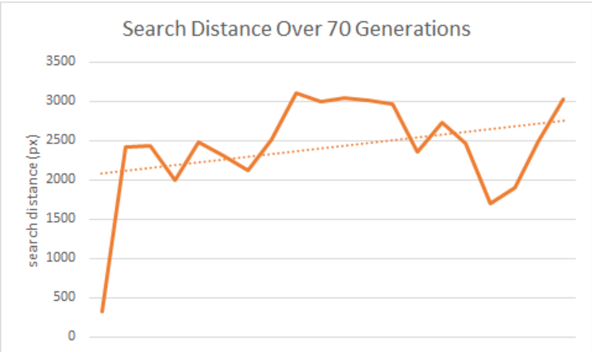

Fig 10. Search distance over 70 generations.

As desired, a clear increasing tendency of pollinated flowers can be seen, even though the search distance increases too. Nonetheless this was expected given that in the bee fitness formula the amount of flowers had plenty more importance due to the constant factor of 100, so the results of these tendencies are considered as satisfactory.

V. Conclusions

- The genetic algorithm is a suitable tool to simulate the real life interactions between bees and flowers.
- An increasing tendency of pollinated flowers was obtained, but with the same tendency for the search distance.
- Optimal values for the program are the following: 100 generations, 20 bees, 200 flowers, 5500 generation average fitness as goal and 0.06 probability for mutation and visiting a non-favorite color flower.
- The probability of mutation and visiting a non-favorite flower have a big impact on the average fitness of the generation of bees.
- If not implemented correctly, the genetic algorithm may never converge towards the optimal solution, especially in the crossover and repopulation phase.
- The simple flip-bit mutation and single point crossover techniques were effective tools to achieve acceptable solutions in this simulation.

VI. References

[1] Funke, I. (2015). Uso de algoritmos genéticos para búsqueda y evaluación de estrategias en HU-NLHE (doble grado). Universidad Complutense, Madrid.

[2] Kaya, Y. A Novel Crossover Operator for Genetic Algorithms: Ring Crossover, 2011. Accessed on: December 18, 2020.[Online]. Available: https://www.researchgate.net/figure/Single-point-crossover_fig1_220485962

[3] Rifki, O. A survey of computational approaches to portfolio optimization by genetic algorithms, 2012. Accessed on: December 18, 2020.[Online]. Available: https://www.researchgate.net/figure/The-flip-bit-mutation_fig3_305000230

[4] "Computer Geekery", Philliplemons.com, 2020. [Online]. Available: http://philliplemons.com/posts/ray-casting-algorithm. [Accessed: 13-Dec- 2020]