Handling Uncertainty in Artificial Intelligence

David Hau, Computer Science

Mentor: David Johannsen

Kellogg Honors College Capstone Project

Introduction

Artificial Intelligence is defined as the ability of a machine to perceive, learn, and interact with its environment in a manner that best accomplishes its objectives. In other words, it receives input that is then used to train the machine certain behaviors that in turn generates output to achieve a certain goal. Depending on various factors such as problem domain, available data or information, application, and other factors, AI can achieve varying degrees of success in accomplishing objectives. Much of AI today depends heavily on training data used to teach the machine how to obtain the desired outputs. Give a problem similar to the training data used for the AI and it can find the solution with relatively minor difficulty. Yet what happens when the AI encounters a problem its training data never prepared it for, when information is uncertain and results are not guaranteed? How does Artificial Intelligence handle the uncertainty that comes with real-world applications?

Genetic Algorithm

Definition
A programming technique based on Darwinian concepts of evolution and natural selection, the Genetic Algorithm enables AI to generate solutions for search and optimization problems by representing possible solution sets as “chromosomes” for a given population. By selecting the best performing solution sets based on a “fitness” criteria, and then recombining the sets through genetic operators such as “crossover” and “mutation,” the Genetic Algorithm can generate solution sets that better suit the desired criteria.

Limitations
- ***Fitness Function:*** Complex problems require complex fitness functions to evaluate the Genetic Algorithm against. As more criteria is added to the fitness function, finding the optimal solution becomes increasingly difficult as more computational time is required.
- ***Complexity:*** As solution sets grow larger, the search time required to optimize those solutions grow exponentially large. Thus the algorithm tends to perform better when solution sets are broken down into their simplest representations. However this also creates the issue of getting those representations to work in conjunction of one another after finding their individual optimal solutions.
- ***Dynamic Data:*** Genomes tend to converge early when presented with Dynamic Data, resulting in solutions that may not hold valid for later data. Early convergence can result in a homogenous population that cannot generate new solutions. Increasing genetic diversity and preventing early convergence may assist the Genetic Algorithm in finding more optimal solutions.
- ***Local Optimum:*** Similar to hill climbing algorithms, the Genetic Algorithm has a tendency to find a local optima rather than the global optimum, meaning its search will often converge on the first optimal solutions it can find instead of the best. Maintaining a diversified population can help alleviate this issue by allowing more opportunities for the algorithm to converge on a global optimum.

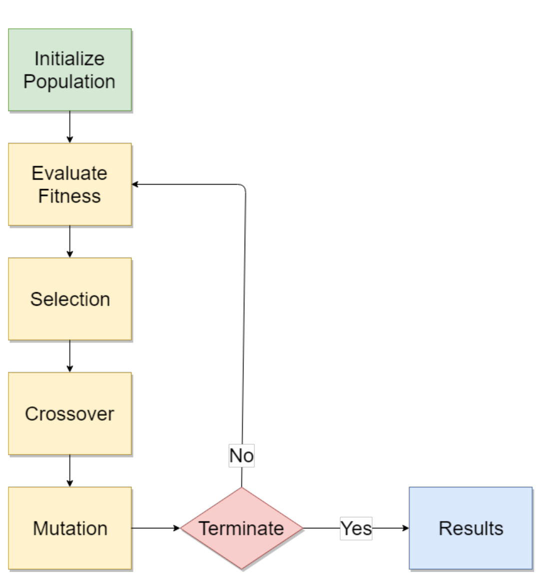

Fig. 1: Diagram of a *Genetic Algorithm*

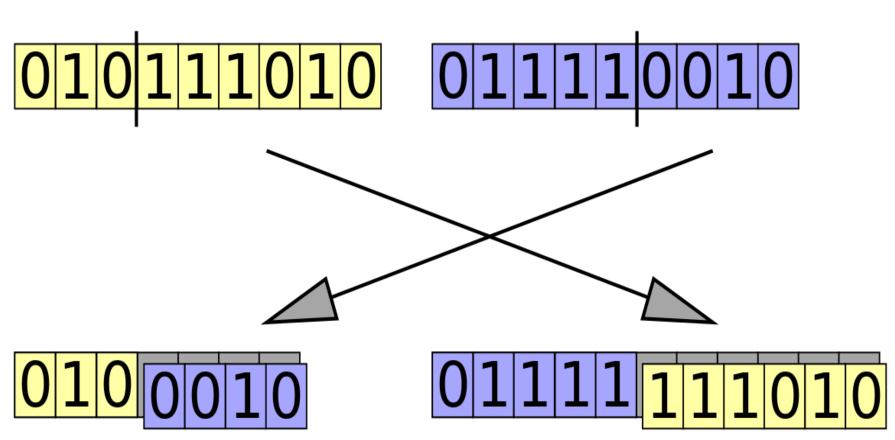

Fig. 2: *A crossover of two parents (Top) to generate two offspring (Bottom)*

Approach

The Genetic Algorithm’s ability to handle uncertainty in data will be tested against the environment of a simulated Chicken Farm. The Farmer, the Intelligent Agent used to test the Genetic Algorithm, will be tasked with maximizing revenue of the farm through raising and selling randomly-generated Chickens within an allotted time limit.

Chicken
The main resource of the Chicken Farm, the Chicken contains randomized attributes that either benefit or hinder the Farmer’s ability to generate revenue. Hens generate a passive varying income for the Farmer in the form of Unfertilized Eggs while Roosters can mate with Hens and fight other Roosters for dominance in the flock. Fully grown Chickens can be sold at the market for a price scaling to their current weight. Fertilized Eggs , created from mated Hens, may grow into Chicks that consume Feed with the other chickens. Failing to obtain feed within a certain time results in death for a Chicken.

Farmer
The Farmer is responsible for running the Chicken Farm in a manner that generates the greatest possible revenue in the allotted timeframe. The Farmer must determine how much feed to buy and allocate to the Chickens, as well as the Chickens to be sold for profit.

	Initial Characteristics Average	Final Characteristics Average	Percent Difference (%)
Growth	0.054	0.033	-38.7
Decay	0.155	0.096	-38.2
Weight	1502.63	2079.63	38.4
Feed	2.78	1.00	-64.0
Egg Count	2.03	1.00	-50.7
Egg Chance	52.93	25.02	-52.7
Aggression	49.24	23.26	-52.8

Fig. 3: Chicken population characteristics in a simulation

Fig. 4: A rooster, one of the Intelligent Agents modeled in the simulation

Results

Fitness Function
When the fitness function for the Genetic Algorithm only specifies the Farmer to maximize as much revenue as possible, a wide range of results occur within the Farmer population, with some opting to sell as much as possible in a short period without concern for early farm closure while others maintained the farm until the termination date earning a steady revenue stream instead, with both typically doubling the initial revenue of the first generation. When the fitness function to maintain the farm as long as possible was introduced, not only did the initial and final Farmer generations perform better compared to their early-terminating cousins, they also maintained their revenue gains all the way through the termination date.

Chromosome Complexity
When the Farmer is given an additional chromosome to specify the priority of which Chicken to feed first as opposed to randomly feeding Chickens indiscriminately, the performance of the Farmer drops dramatically. Not only does the final generation produce roughly half the revenue the original chromosome could accomplish, the population struggles to survive until the termination date.

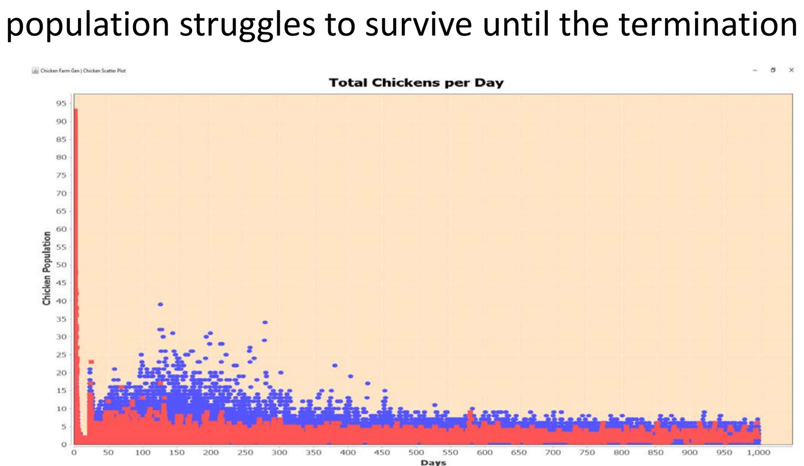

Fig. 5: Chicken Population – Revenue Focus Only

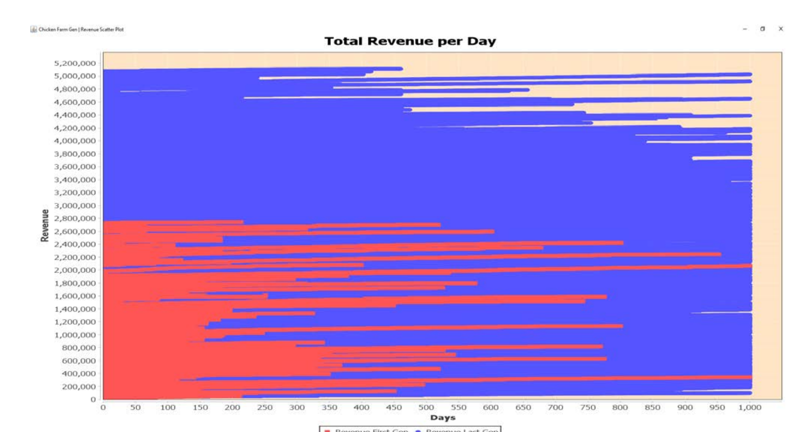

Fig. 6: Revenue Gain – Revenue Focus Only

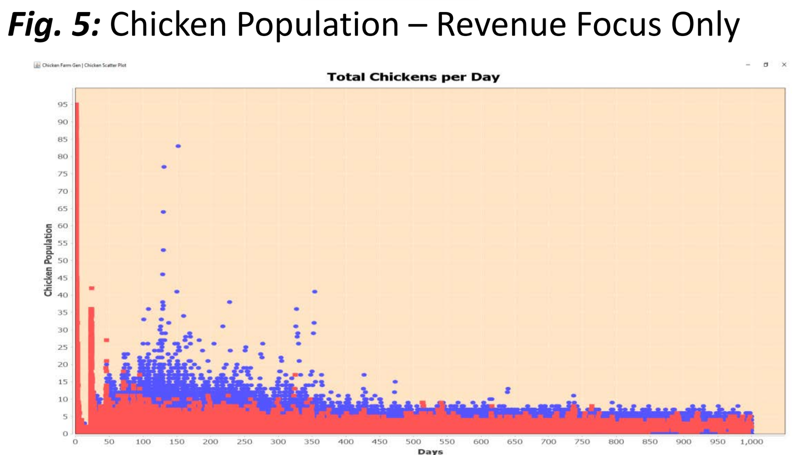

Fig. 7: Chicken Population – Revenue with Longer Life

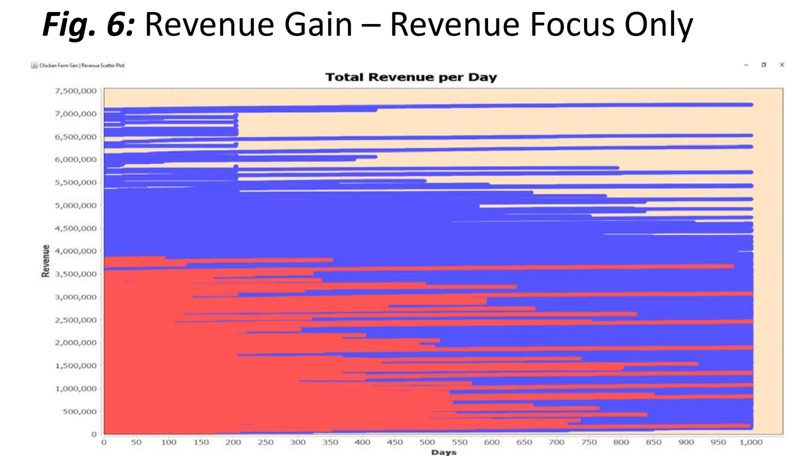

Fig. 8: Revenue Gain – Revenue with Longer Life

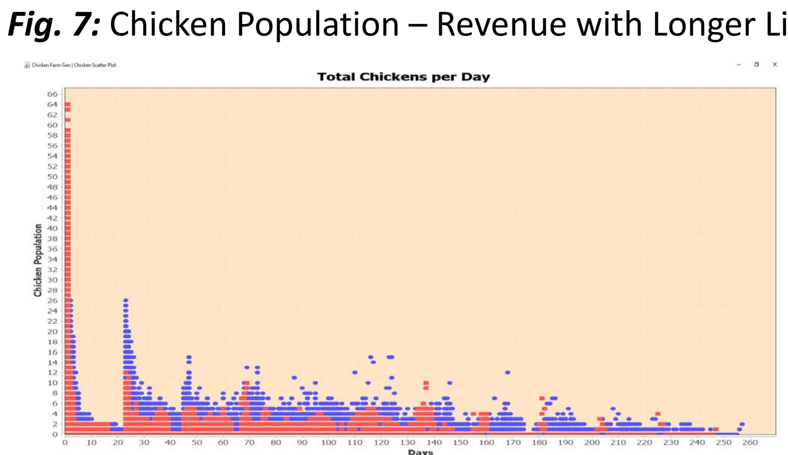

Fig. 9: Chicken Population – Feeding Priority

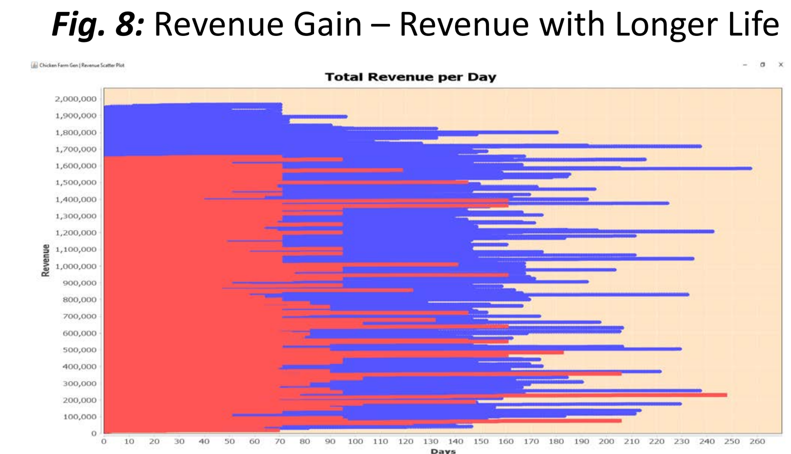

Fig. 10: Revenue Gain – Feeding Priority

Conclusion

When given narrowly-defined fitness functions and limited genetic instructions within its chromosomes, the Genetic Algorithm can perform adequately well in lieu of future uncertain conditions. Although it does not guarantee the absolute best result, it can provide insight to the performance of average optimization solutions in a given situation, as evidenced by overall growth in all test cases. However, this is as far as the Genetic Algorithm can accomplish given the uncertainty of its data. When data is basic and independent of one another, the Genetic Algorithm performs well in uncertain conditions, but when the data is complex and interrelated with other observed data, as in the case with Feed Priority, performance drops significantly as a result of self-inflicted increased uncertainty within its own system. Artificial Intelligence by design can handle uncertainty, but how much of that uncertainty is handled successfully without compromising the objective heavily depends on both the data it is given to work with and the system that is used to evaluate that data.