Genetic Algorithm and Application in training Multilayer Perceptron Model

Tuan Dung Lai
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Hawthorn, Victoria 3122
Email: tuandunglai@gmail.com

***Abstract*—In today's world, an intelligent and optimal problem solving approaches are required in every field. Machine are becoming more and more efficient, many applications have been made recently to solve complex problem. Genetic algorithm is a heuristic search technique in artificial intelligent to find the most optimized solution for a given problem based on crossover, mutation, selection and some other techniques inspired by Darwin's theory of evolution. This report demonstrate how GA approaches to a optimization problem in general. An implementation of GA on building an AI for a famous arcade game namely Flappy Bird is provided.**

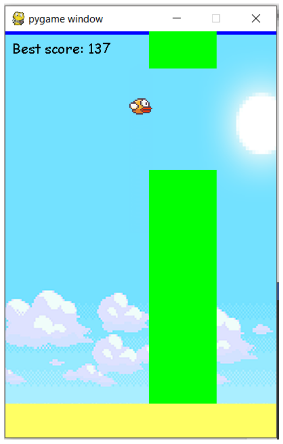
I. Introduction

GA has been successful in complex engineering applications that involved multiple objective, non well-defined optimization function. For example, the design of analog circuits [1] and optimization of space-born antennae [2] are developed using GA

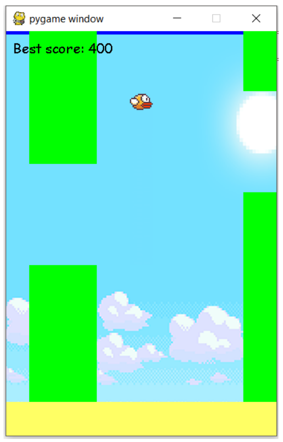
Optimization algorithms can roughly be divided into three classes: mathematical programming algorithms, heuristics, and metaheuristics. Mathematical programming algorithms have the most rigorous foundations, and it may be possible to prove that the algorithm actually converges, check that the proposed solution is close to at least a local optimum, and to estimate the rate of convergence. Metaheuristics algorithm use metaphors, usually come from unrelated field to define selection algorithm. Simulated annealing, ant colony, bee swarm, harmony search are examples of algorithm that are derived from different fields rather than computer science

GA is an artificial intelligence metaheuristic search technique that is derived from the Darwin's theory of biological organism evolution. As for many other metaheuristic methods, controversial reviews have been made around GA due to the lack of core and concrete mathematical foundation. In other words, many people consider it as a complete black box.

In the next part, a brief introduction to GA will be demonstrated as well as its application in training a neural network that can play Flappy Bird game.



(a) Generation = 300, BestScore = 137



(b) Generation = 350, BestScore = 400

Fig. 1: Experiment result

II. Genetic Algorithm Methodology

A. Optimization Problem

Most problem in real life don't have formula and technique to calculate the exact result because of the vast generic complexity. GA works on a population of possible solutions and evolve them using method inspired by Darwin's theory in biology. The rest of the report will discuss the different between GA and other method and also perform some experiment to estimate the effectiveness of GA.

Each problem using GA requires a **fitness function** which measures the quality of the solution toward an optimization problem.

B. Terminology

In GA, a **population** of candidate solutions (also called phenotypes) is evolved toward better solutions of an optimization problem. Each candidate solution is represented by a **chromosome** which is a set of **genes** which can be alter **mutate** and **crossover**. A new set of chromosome, also known as **generation** is formed using **selection**.

1) Population, chromosome and genome:
Population: The number of individuals present with same length of chromosome. In other words, they are a set of possible solution to a optimization problem.
Genome: A part of a chromosome. The value of each gene has an effect on the quality of solution.
Chromosome: A set of genomes. Chromosome is the solution in form of genes

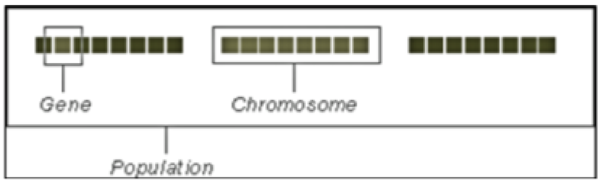


Fig. 2: Population, Chromosomes and Genes

2) Selection, mutation and crossover:
Selection: Next generation's population will keep some of the best solution in the previous generation so that best traits is retained.
Mutation: Alter genome in chromosome.
Crossover: Mixing two individual to produce a new pair of offspring that have the trait of both

III. Genetic Operators

1) Encoding and Decoding: Encoding technique depends heavily on the problem. Generally, encoding is the order of every genes that have an effects on the solution. Decoding is translate chromosome to solution of optimization problem. Encoding is a method to clean data before putting it in genetic operators.

Encoding Method	Chromosome
Binary	1011000010010
Value	1 2 6 7 1 8 0 1 5
Value	ABDOMNABCDAC
Value	(back), (back), (right), (forward)

2) Selection: Selection process is mainly responsible for assuring survival of the best-fit individuals. Best solution will be retained in the next generation.

2.1) Roulette wheel selection method Fitness proportionate selection, also known as roulette wheel selection, is a genetic operator used in GA for selecting potentially useful solutions for recombination.
In this method, each gene have a probability of being selected for next generation. This probability is defined by:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{1}$$

p_i: Probability of individual with index i being selected.
f: Fitness function
f_i: Fitness value of individual with index i

2.2) Best fitness selection method
In this selection method, best individuals with highest fitness is being selected. This method is a trade off between diversity of population and average fitness of population where diversity decreases and average fitness increases in comparison with Roulette wheel selection method.

3) Mutation:
Mutation is used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation.

The purpose of mutation in GA is preserving and introducing diversity. Mutation should allow the algorithm to escape local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. Mutation can be done with a formula or randomly.

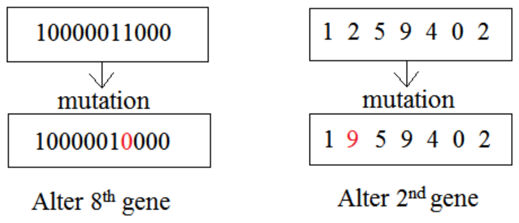


Fig. 3: Example of mutation on binary encoding and value encoding

4) Crossover:
The crossover splits up the parent individuals and recombines them. Crossover point can be chosen randomly to increase diversity of new population.

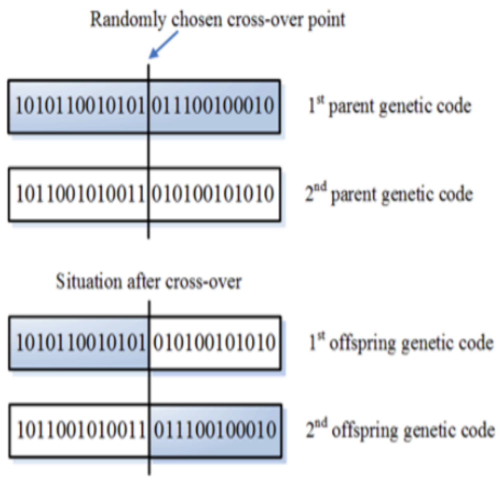


Fig. 4: Genetic Code of the parents and offspring before and after the crossover [4]

Multi-point crossovers are simply crossovers with more than one position where crossover will occur.

IV. Implementation

1) Implementation overview: The algorithm can be done by continuously create new set of possible solution using GA to evolve every generation overtime until an acceptable solution is found.

2) Flowchart:

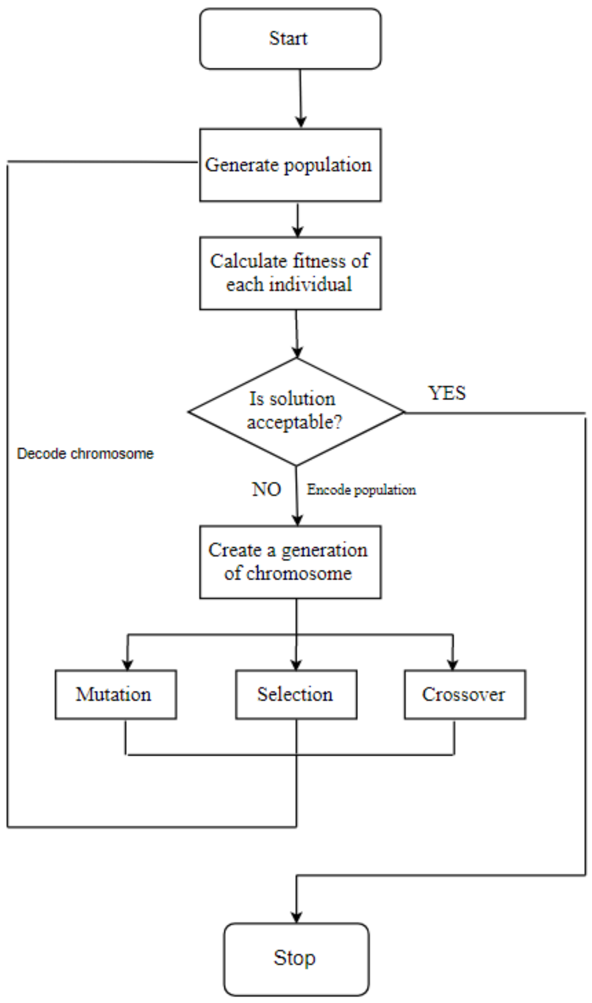


Fig. 5: GA Flowchart

3) Pseudo Algorithm:

```
START

Initializing population.
Calculate fitness of each individual.

DO UNTIL BEST SOLUTION IS FOUND

  Encoding each individual to produce chromosome.
  Perform GA operators on exist population
  Decode new population and kill old population

LOOP

END
```

V. Application in building AI for Flappy Bird Game

A. Gameplay overview

Flappy Bird is a mobile game developed by a Vietnamese developer Dong Nguyen. The objective was to direct a flying bird, named ”Faby”, who moves continuously to the right, between sets of Mario-like pipes. If the player touches the pipes, they lose. Faby briefly flaps upward each time that the player taps the screen; if the screen is not tapped, Faby falls because of gravity; each pair of pipes that he navigates between earns the player a single point.

B. Training model

In this section, multilayer perceptron model will be discussed and an implementation on Flappy Bird game will also be provided.

1) Neural Network Overview:

1.1) Layer: They are a set of neuron (a circle that contain a number). Beside input layers and output layers, one Multilayer Perceptron can have one or more hidden layer

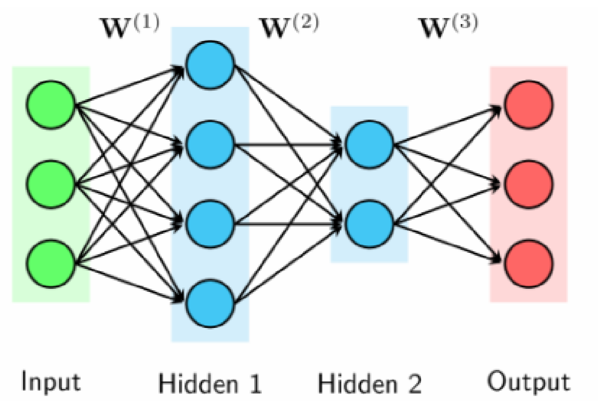


Fig. 6: Example of multilayer perceptron network with 2 hidden layers

1.2) Unit: One node (the circle in Fig 6) is called one unit. Input of each unit is symbolised as z and output of each unit is symbolised as a (a stands for activation, input unit in next layer)

1.3) Weights and Biases: In fig 7, the number that in the line which connects 2 nodes in 2 layer is called Weights, they determine how much affect a node could have on the next input unit in the next layer, biases is the node x_0 in fig 7, the value is normally constant at 1. They add the flexibility to the network.

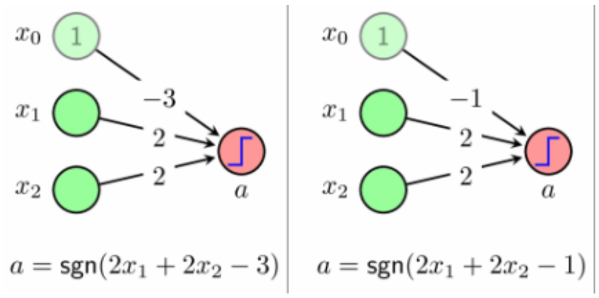

x_0
1
x_1
x_2
-3
2
2
a
$a = \mathrm{sgn}(2x_1 + 2x_2 - 3)$
x_0
1
x_1
x_2
-1
2
2
a
$a = \mathrm{sgn}(2x_1 + 2x_2 - 1)$

Fig. 7: Feed Forward Process with Sigmoid activator

1.4) Activator, activation function: When talking about activator, they mean the function that apply on a nodes to produce the output unit. The purpose of activation function is to squeeze the value after multiplying nodes value and weights to produce a number within a defined range.

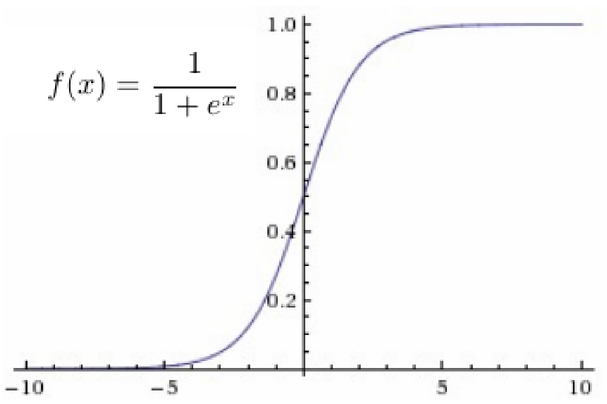


Fig. 8: Sigmoid function squeeze number to value of 0 when s goes to -infinity and 1 when x goes to infinity

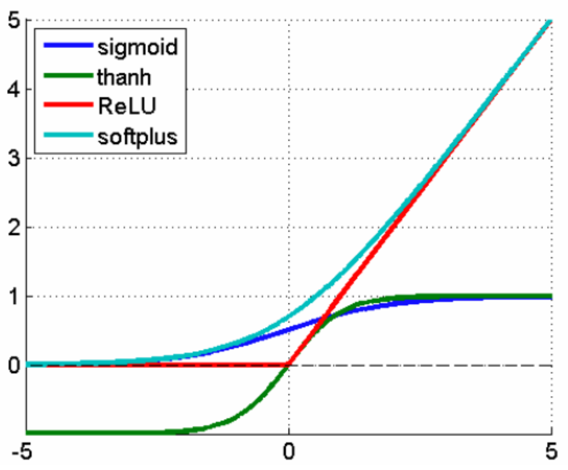


Fig. 9: Different activation functions

2) *Neural Network for Flappy Bird AI:* In this experiment, a simple multilayer perceptron model [5] is chosen. The network includes one hidden layer with 6 nodes, one output layer and one input layer. One bias node is added in input layer and hidden layer. These bias nodes ensure constant variable can have an effect on the solution. Hidden layer and output layer use sigmoid function as activator. Output of neural network is a number in range 0 and 1. Threshold is set to 0.5. If $output > 0.5$ then the bird will flap.
Source code can be found at:
$https://github.com/DungLai/AI-FlappyBird$

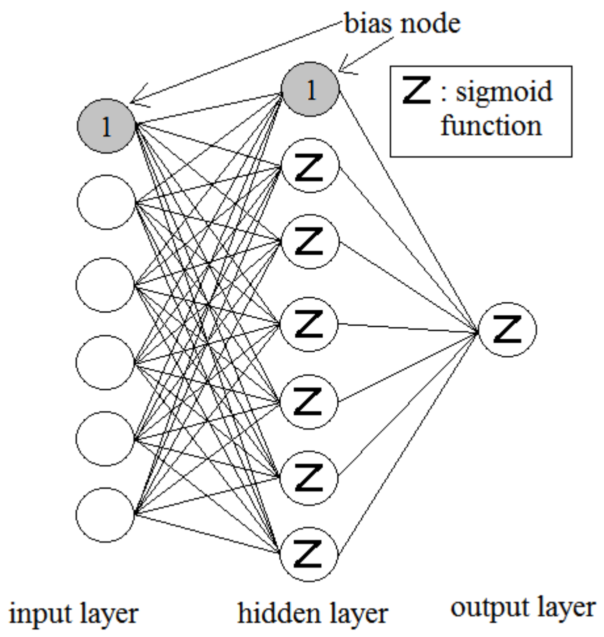
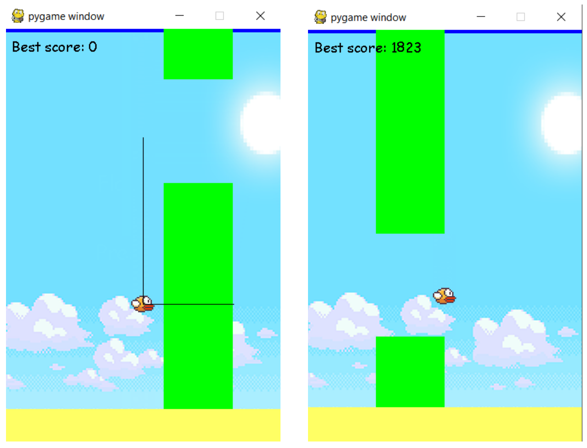


Fig. 10: Neural Network Architecture

C. *Input for training model*

In neural network, every input should be related to the solution. There are 5 inputs as described in figure 10



(a) Two sensors of a bird horizontal line, vertical line
(b) Best performance 1823 scores

Fig. 11: Experiment result

Input 1: Horizontal distance between bird and Tube (Horizontal line in figure 12).

Input 2: Vertical distance between bird and the middle of two tubes (Vertical line in figure 12).
Input 3: Width of bird.
Input 4: Height of bird.
Input 5: Width of tube.

D. *Implementation of GA*

1) Encode: There are 49 weights in neural network in figure 10. All of these weights are added into an float array of size 49. This array will carry all the information of the whole network. As a result, each array is considered as an chromosome with value encoding method (described in III-1). Each weight is now a gene in a chromosome.

2) Applying genetic operator: Selection, mutation and crossover is used in this experiment.
Selection: Best fitness selection method (III-2.2)
Mutation: Modify a weight by randomly assign a new number to it. (III-4)
Crossover: Exchange two weight by a fixed possibility (III-3)

Parameter	Value
POPULATION	50
SELECTION RATE	0.1
MUTATION RATE	0.7
CROSSOVER RATE	0.6
MUTATION PERCENTAGE	0.6
CROSSOVER PERCENTAGE	0.1

3) Fitness function: In this experiment, fitness function is simply the survival time of a bird, the unit is the number of frames being refreshed after the bird dies.

VI. Experiment Statistical Result

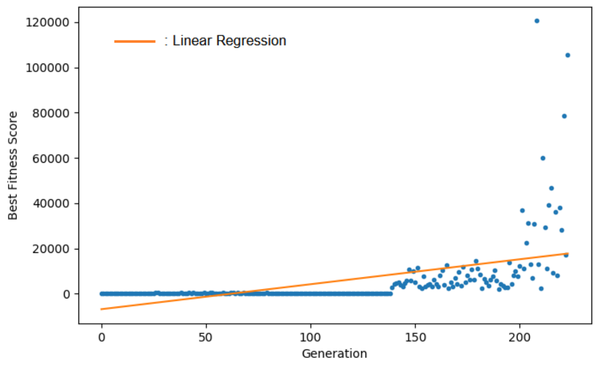


Fig. 12: Experimental result 1

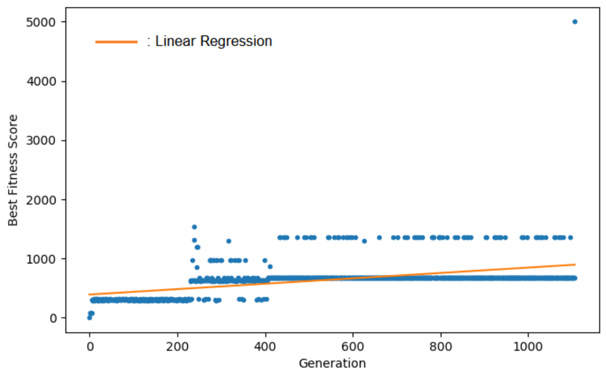


Fig. 13: Experimental result 2

The two above graphs describe the correlation between number of generation and best fitness score to figure out if GA actually helps the population evolve over time or not. In both experiment, an impressive solution is found. However, the first experiment took 230 generations while the second experiment took over 1000 generations in order to find the best solution. A correlation coefficient (Pearson) is performed to test whether there is a correlation between number of generation and best fitness score.

Experiment	pearson-r	p-value
Experiment 1	0.5	0.001
Experiment 2	0.5	0.001

Both experiment share the same result, there is a strong, positive linear relationship between number of generation and best fitness score and Pearson's correlation shows that this relationship is significant, $r = .5$, $p < .001$.

We can conclude that GA do improve the optimization overtime. However, time complexity cannot be measured exactly due to the high arbitrary aspect.

VII. Further Research

As there are many randomness in GA, the experiment result could be biased. Further research can be done by controlling the arbitrary factors.

VIII. Discussion

It can clearly be seen that there is not much concrete mathematical foundation to GA. The algorithm itself is heavily derived from biology. The performance of GA is also not being calculated or estimated, this is the reason why many people tend to have negative opinions on it. However, I strongly believe that one day, researcher will find a solid foundation and make improvement GA. Neural network is an example, the idea is derived from an unrelated field, namely neuroscience, nodes describe the connection between millions of neurons in the brain of human. Overtime, researchers and developers improve the performance of neural network, some fantastic training technique such as backpropagation are introduced and ensure the convergence of the optimization problem, many variations have been made to neural network too, such as convolutional neural network (CNN), recurrent neural network (RNN) and long short-term memory (LSTM) architecture. Again, I strongly believe that the same boom in research will happen to GA one day.

Acknowledgment

The author would like to thank Luke Gavin, my tutor for this project. He constantly encourages me and also recommend many good pieces of advice as well as learning materials. Also thanks to Dr. Christopher McCarthy for a great serie of lecture on Object Oriented Programming (OOP), this project is part of the unit, without solid understanding of OOP, the implementation of GA as well as Flappy Bird game will be very challenging.

References

[1] Ali Jafari, Maryam Zekri, Saeed Sadri, Alireza Mallahzade *"Design of Analog Integrated Circuits by Using Genetic Algorithm"*. [Online] Available at: http://ieeexplore.ieee.org/document/5445763/

[2] Haihong Tao, Guisheng Liao, Ling Wang *Space-borne antenna adaptive side-lobe nulling algorithm based on gradient-genetic algorithm.* [Online] Available at: http://ieeexplore.ieee.org/document/1321992/

[3] E. Eiben (1994). "Genetic algorithms with multi-parent recombination". PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: 78–87. ISBN 3-540-58484-6

[4] E. Schultz, J. Mellander, C. Endorf (2008), "Intrusion Detection and Prevention - A basic genetic algorithm", [image online], http://my.opera.com/blu3c4t/blog/show.dml/26 36486.

[5] Andrew, Ng. , 'Machine learning', Standford University Online, lecture notes week 4, [online]: https://www.coursera.org/learn/machine-learning