**Name**: Gayathri Kumar

**Login**: [gk329@kent.ac.uk](mailto:gk329@kent.ac.uk)

**ID**: 24010288

**Module Code**: COMP8370

**Word Count**: 951

**Exploring Real-World Applications of Genetic Algorithms**

**I. Introduction**

Genetic Algorithms are computational optimization methods based on genetics and natural selection. These algorithms solve real world problems by simulation of the evolution process to improve population of solution sets. These solution sets are either binary encoded or another structure.[1]

Genetic Algorithms are very good at optimising solutions to problems finding the best solution from an abundant possibilities. Examples of such optimisation problems are mathematical functions, resource allocation, parameter fine tuning, and so on. These algorithms go through the solution space by evolving a set of possible candidate solutions with the use of genetic operators such as crossover and mutation.[1]

The essay focuses on 2 real world examples of applications of Genetic Algorithms from 2 different fields – Android Malware Detection and a simulation of the game Super Mario Bros called Infinite Mario Bross. [1][2]

**II. Real-World Application 1: Android Malware Detection**

**Introduction**

2 sets of APKs (Android Application Package) Malware and Goodware are reverse engineered from which features are extracted such as App permissions and statistics of app activity, content provider, etc. The obtained features are represented as vectors with binary values in CSV (Comma Separated Values) format. This CSV is provided as an input to a genetic algorithm for optimised feature selection, which is then used to train the SVM (Support Vector Machine) and neural network. [2]

**Individual Representation**

Features extracted are mapped into the feature vector into 2 categories: - 1) App Components; that count components such as Acitvity, Content Provider, Services, etc and 2) Permissions; that are mapped to a vector space such that if the app has the feature the dimension is 1, otherwise it is 0. These features represented in a binary form and are called chromosomes. The algorithm is initialised using these feature subsets and an initial population set of these chromosomes is generated randomly. [2]

**Fitness Function**

The algorithm keeps a set of chromosomes called population along with their fitness scores. Chromosomes with better fitness scores have a better chance to reproduce. The fitness function used here assigns a higher score to the chromosome with best accuracy after training in the ML (Machine Learning) classifier and features with lower accuracy are assigned lower scores. Chromosomes with the highest fitness scores are chosen to be the next generation parents. [2][3]

This is done through crossover and mutation. It is a technique that automatically improves characteristics to create the best offspring. Based on fitness requirements, even the worst generation can become the best. Based on features, if a generation is unmatched, it is excluded by the GA (Genetic Algorithm). This step is repeated until the best generation is found. Even if it cannot find a final solution to the problem, the GA can at least select the optimal features. [3]

**III. Real-World Application 2: Infinite Mario Bross AI**

**Introduction**

A finite state machine (FSM) is used along with a genetic algorithm to develop an AI agent for the game Super Mario Bros that can win levels. There are 4 states in this FSM – Run State, Jump State, Run Speed State and Run Jump Speed State. Transitions between the states are determined by the presence of enemies, obstacles or holes. A genetic algorithm is used to evolve this process. [4]

**Individual Representation**

The Genetic Algorithm is used as an adjustment system for finding the transitions that suit each state more. An individual chromosome, represented by 16 integer, is determined by the following rules: - 0 for an enemy, 1 for an obstacle, 2 for nothing, 3 for an enemy and a hole, 4 for an enemy and an obstacle and 5 for a hole and an obstacle. Each genotype depends on a transition that reacts based on the inputs provided. [4]

**Fitness Function**

A simple fitness function calculates the fitness of each individual chromosome based on this formula:- Fitness = distancePassedPhys \* sov.distance.

The variables are based on how much the agent has traversed in the level. The above fitness function means the more distance travelled by the agent, the greater the fitness value of that individual. This represents simple Mario behaviour and killing enemies or scoring coins is not mandatory to check the fitness value as long as the agent keeps travelling in the journey more. But, time is still important here as the level needs to be completed before the provided time limit. Without reaching the goal in time, Mario is said to have failed. [4]

**IV. Comparative Analysis of Applications**

Both applications are from diverse domains, but use genetic algorithms as a means of optimisation techniques, adapting to solutions over time and achieving a better performance. It also allows adapting and adjusting to different conditions over time, whether in evolving the scope of malware or facing challenges in the game. [2][4]

These applications have different ways of representing data types. One deals with file features, network traffic and so on while the other deals with data based on all the game states. The fitness functions used have different objectives; one is recognising whether an app has malware or is safe and secure to use while the other wants to maximise the success rate within a game. [2][4]

**V. Conclusion**

To summarise, genetic algorithms play vital roles in Malware detection and Game AI. They are tools that are versatile for optimisation and applied differently hence showing how adaptive these algorithms are especially in various domains it is applied in. A lot of advantages are solving complex to no mathematical problems, noise resilient and supports parallel processing and continuous learning. Hence these are signification in solving a myriad of real life problems across various real world domains and further research on genetic algorithms is to be encouraged. [2][4][5]

**VI. References**

[1] Vijay Kanade, Spiceworks. (2023, September 6th). What are Genetic Algorithms? Spiceworks. <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>

[2] A. Fatima, R. Maurya, M. K. Dutta, R. Burget and J. Masek, "Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning," 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 2019, pp. 220-223, doi: 10.1109/TSP.2019.8769039.

[3] Firdaus, A., Anuar, N.B., Karim, A. *et al.* Discovering optimal features using static analysis and a genetic search based method for Android malware detection. *Frontiers Inf Technol Electronic Eng* **19**, 712–736 (2018). <https://doi.org/10.1631/FITEE.1601491>

[4] N. C. Hou, N. S. Hong, C. K. On and J. Teo, "Infinite Mario Bross AI using Genetic Algorithm," 2011 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT), Semenyih, Malaysia, 2011, pp. 85-89, doi: 10.1109/STUDENT.2011.6089330.

# [5] Wirsansky, E, 2020. “Hands-On Genetic Algorithms with Python: Applying genetic algorithms to solve real-world deep learning and artificial intelligence problems” pp. 17