

COMPARISON OF GENETIC ALGORITHM TRAINING METHODS
AS APPLIED TO Tic-Tac-Toe

A research paper submitted to the
Faculty of the Philippine Science High School –
Cordillera Administrative Region Campus
in fulfillment of the course requirements in
Science, Technology, Engineering and Mathematics Research 3

VASH PATRICK B. ANCHETA
DIEGO SULAYMAN R. PASCUA
RESH VNZI S. TOGUEÑO

18 May 2020

APPROVAL SHEET

In fulfillment of the requirements in Science, Technology, Engineering and Mathematics Research 3 (STR 3), this research entitled, “COMPARISON OF GENETIC ALGORITHM TRAINING METHODS AS APPLIED TO TIC-TAC-TOE” is submitted by Vash Patrick B. Ancheta, Diego Sulayman R. Pascua and Resh Vnzi S. Togueño on 18 May 2020

KAYE MELINA NATIVIDAD B. ALAMAG
Research Adviser

This research paper is hereby accepted by the Research Council.

CONRADO C. ROTOR, Jr., Ph.D.
Chair

MELBA C. PATACSIL
Co-chair

JAY JAY F. MANUEL
Member

MARITES P. RIVERA
Member

RICARIDO M. SATURAY, Jr.
Member

FREDA M. WONG
Member

ACKNOWLEDGEMENT

We are grateful for our friends and family for their continued support in our continuous lives. We are also thankful to our teachers, research teacher, and research adviser for their unwavering assistance in having the research completed. Without these people, this research would never be successful.

ABSTRACT

Vash Patrick B. Ancheta, Diego Sulayman R. Pascua and Resh Vnzi S. Togueño.
Philippine Science High School – Cordillera Administrative Region Campus, 18 May 2020.
“COMPARISON OF GENETIC ALGORITHM TRAINING METHODS AS APPLIED TO
TIC-TAC-TOE”

Adviser: **Kaye Melina Natividad B. Alamag**
According to George Mason University (2020),

An abstract is a 150- to 250-word paragraph that provides readers with a quick overview of your essay or report and its organization. It should express your thesis (or central idea) and your key points; it should also suggest any implications or applications of the research you discuss in the paper.

George Mason University (2020) also states that the common abstract is divided into such: 25% of space on the purpose and importance of the research (Introduction), 25% of space on what was done (Methods), 35% of space on what was found (Results), and 15% of space on the implications of the research.

TABLE OF CONTENTS

ABSTRACT	iii
List of Figures	v
List of Tables	1
CHAPTER I: INTRODUCTION	2
Background of the Study	2
Objectives of the Study	2
Significance of the Study	3
Scope and Limitations of the Study	3
DEFINITION OF TERMS	4
CHAPTER II: REVIEW OF RELATED LITERATURE	5
CHAPTER III: MATERIALS AND METHODS	11
Research Design	11
Locale of the Study	11
Materials and Research Instruments	11
Procedures	11
Treatment of Data	12
CHAPTER IV: RESULTS AND DISCUSSION	13
CHAPTER V: CONCLUSION AND RECOMMENDATIONS	16
LITERATURE CITED	17
APPENDICES	19
Appendix A: Project Plan	19
Appendix B: Raw Data	22
Appendix C: Statistical Tables	23
Appendix D: Documentation	24

LIST OF FIGURES

Figure	Title	Page
1	Game tree	6
2	Flowchart of GA training	8
3	Formula for the fitness score of algorithm x where n_{wins} is the number of games won	9
4	Sample bar graph	13
5	Network chart	19

LIST OF TABLES

Table	Title	Page
1	Example of Organisms Sorted Through a Classifier System	10
2	Characteristics of the Sample	14
3	Task Lists and Duration	19
4	Task Schedule Management and Personnel Assignment Plan	20
5	Material and Equipment Sourcing Plan	21
6	Risk Management Plan	21
7	Random Table	22
8	ANOVA Table	23

CHAPTER I INTRODUCTION

Background of the Study

Machine learning (ML) is vast—it is used in different situations such as spam detectors, web search engines, photo tagging applications and game development (Sharma, 2016). There have been researches that are aimed at improving the implementation of ML in various games. A category of games under investigation through ML is the set of m, n, k game games, comprised of games where there is an $m \times n$ grid and two players alternate turns trying to earn k pieces adjacent to each other horizontally, vertically or diagonally (Hayes & Loge, 2016). Among the most common examples of m, n, k games are Gō, Othello, and Chess. Tic-Tac-Toe, the game under investigation in this study, is an example of an m, n, k game. A Tic-Tac-Toe board is composed of three rows and three columns, and requires three adjacent pieces of the same player to render a win, thus it is considered to have a 3, 3, 3 configuration.

Improvements in ML have lead to the development of artificial intelligence (AI) players that can beat even the most competitive human players around the world. Machine learning methods (MLMs) are algorithms where machines are not explicitly programmed to do what is tasked. Rather, similar to its namesake, MLM-trained machines are capable of performing tasks given its own internal code without any human interference. In short, the machine *learns* (GeeksforGeeks, n.d.). An example of an MLM is the genetic algorithm (GA).

This study aims to develop multiple GAs with different elite preservation methods and compare their performance in Tic-Tac-Toe based on the possible situations.

Objectives of the Study

General Objective

- To compare the effectiveness of trained genetic algorithm (GA) organisms among each other as applied to Tic-Tac-Toe

Specific Objectives

1. To implement known heuristics into Python code
2. To train organisms of an implemented GA using different move generators (MGs)
3. To compare the development of the performance of trained GA organisms among each other within 500 generations

Significance of the Study

This study will contribute to the body of knowledge in ML. Through this study, more can be known about how information gained from one method of AI can be passed on to another mechanism of AI through training. This will shed light on how information from one AI player can be transmitted to an MLM such as GA. This, by extension, can improve the comprehension of how machines can learn strategies in games from one with greater skill.

Scope and Limitations of the Study

This study will focus only on Tic-Tac-Toe and not other games such as Chess or Gō because it is the simplest game to conduct the research on heuristics, namely the training of the GA under different MGs. The complexity of the board game is not relevant to the study because the focus of the research is to compare the effectiveness of trained GA organisms given an m, n, k game. Applying these heuristics on other m, n, k games however are beyond the time frame of the research. The study will only develop three GAs. The first is a Python implementation of the GA in the work of Bhatt et al. (2008). Using developed AI, the second and third are modified implementations of the same GA. The performance of each GA will be based on how many generations it will take for the GA to find a no-loss first player for Tic-Tac-Toe. This basis for comparison of performance, specifically using the skill of an organism as a first player, is due to the fact that Python is known for being slow. In line with this, indices are 0-based in this paper, as they are in Python.

DEFINITION OF TERMS

m, n, k game – a game where there is an $m \times n$ grid where two players alternate turns trying to earn k pieces adjacent to each other horizontally, vertically or diagonally

Artificial Intelligence (AI) – program that simulates human actions, can simulate a human player in a game.

Allele – the configuration value of a gene given a unique organism

Genetic Algorithm (GA) – algorithm that simulates natural selection and biological reproduction to produce solutions to a problem

Gene – representation of a unique and distinct situation in a game given the game rules

Genome – mapping table of genes with corresponding alleles

Machine Learning (ML) – a heuristic where a program learns rather than strictly follow a given instruction

Machine Learning (MLM) – algorithms used in machine learning

Organism – an algorithm represented by a genome

Probability Valuation (PV) – classical probability of a player winning at a given game state

Tic-Tac-Toe – m, n, k game configured as 3,3,3

CHAPTER II REVIEW OF RELATED LITERATURE

Effective Computer Algorithms on Tic-Tac-Toe

There have already been precedents in investigating the proper method of winning classic Tic-Tac-Toe. Examples such as prioritizing the center or placement of pieces in the opposite cell of the opponent's previous move are frequently cited as techniques to beat the opponent (Aycock, 2002). Multiple discussions have occurred on how many distinct games one can play in Tic-Tac-Toe. Schaefer (2002) argues that the common answer of $9!$ or 362,880 possible games is misleading for it ignores the symmetrical properties of the game. There are games that have exactly the same pieces placed on the board but are oriented differently in relation to the board. For example, a gametree turned 90° is considered a distinct game state in the $9!$ calculation. According to Schaefer (2002), it is of more concern to count game states that end when there are k adjacent pieces. Using this method of counting game states, Schaefer (2002) computed a total number of 765 distinct game states.

Numerous studies have been performed on the most effective algorithms for a computer program to win in Tic-Tac-Toe. In organizing the results of a computer algorithm, one of the most common methods of organization is the use of a game tree. A game tree is a collection of all possible game states arranged in chronological order. The root node represents the current state, its child node the set possible game states. The edges represent the moves and the terminal nodes represent game states that indicate a completed game. The game tree begins from the root node and branches out into nodes that have their own children (Adamchik, 2009). A diverse game tree is optimal for exploring the capabilities of the different machine learning methods to be used in the study.

A study by Cranenburgh et al. (2007) is concerned with the use and implementation of a heuristic known as depth-first search. Depth-first search is a machine learning method that utilizes *backtracking*. When the algorithm encounters a terminal node, it returns to the previous nodes to find other possible nodes, hence the term *backtracking* (HackerEarth,

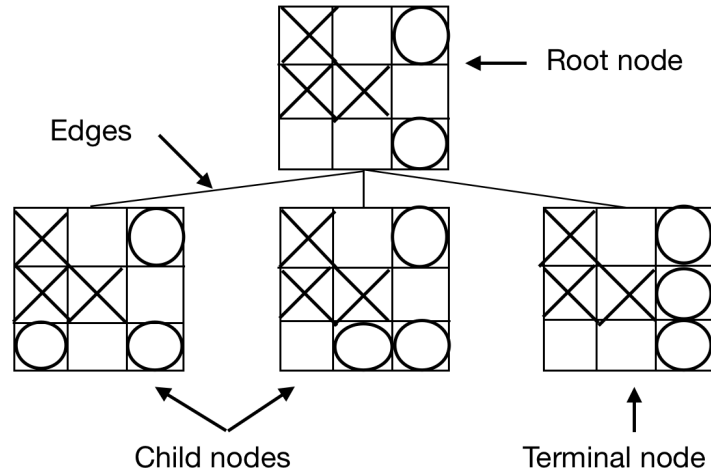


Figure 1. Game tree.

2019). According to Cranenburgh et al. (2007), a higher depth search with more game states, leads to a higher win rate or at least, leads the program to reach a draw better.

Researches that deal with the more complex Tic-Tac-Toe variant Ultimate/Super Tic-Tac-Toe seek to find the patterns and implementing these patterns into AI. Analogous to how classic Tic-Tac-Toe is symmetric, a study by George and Janoski (2016) specifies the rotational and reflectional symmetry of Ultimate/Super Tic-Tac-Toe. A study by Lifshitz and David (n.d.) deals with the use of a mixture of heuristics and algorithms such as MiniMax (an algorithm that traverses the whole game tree) and ExpectiMax (an algorithm that analyzes the expectations of the opponent's moves). Given a controlled set of parameters, ExpectiMax won against a random algorithm most of the time. When played against MiniMax however, ExpectiMax was more likely to lose. This is in agreement with the aforementioned conclusion that an increased depth produces better results. A caveat for this machine learning method is the drastic duration of time the algorithm requires with each increase in depth. The higher the depth of the depth-first search, the longer the time needed for the algorithm to provide the most optimal move.

Machine Learning Methods

Various methods of machine learning have been discovered prior. One of the earliest examples of machine learning methods is Hexapawn formulated by Martin Gardner. The game is composed of a 3×3 board with two sets of three pawns in a row on opposite sides of the board, and the players move alternately. The game is typically played by a human (who always goes first) and an AI player. A player wins the game by accomplishing one of three goals: move a pawn to the opposite edge, capture all enemy pieces, or leave the enemy with no moves. Because of the small scale of the game and its symmetry, all the possible game states can be represented in 24 cards. Each of these cards contains a move performed by the human player and arrows of different colors representing the possible moves the AI player can take (Ortiz, 2017). The AI player contains a matchbox for each card representing a game state. These matchboxes contain beads that correspond to the possible moves the AI player may make. During play, the beads are chosen at random and the AI player moves according to the bead taken. If the move made by the AI player leads to the loss of the AI, the bead corresponding to the move is taken away. Otherwise, the bead is returned to the matchbox. After 50 games, the AI player is practically unbeatable as game states with high probability of loss is reduced (Gardner, 1958).

Genetic Algorithm

Living organisms exhibit a level of problems solving that is almost impossible to be recreated by a computer scientist. Even more, the complexity that living organisms show is one that is incredibly enviable to a computer scientist wanting to achieve some problem solving program; problems that computer scientists have spent endless amounts of intellectual effort on have been solved by living organism without any thought, relying on the process of evolution.

Due to this, many researchers have began emulating the process of evolution on their algorithms, subjecting their own programs to a process of reproduction and natural selection.

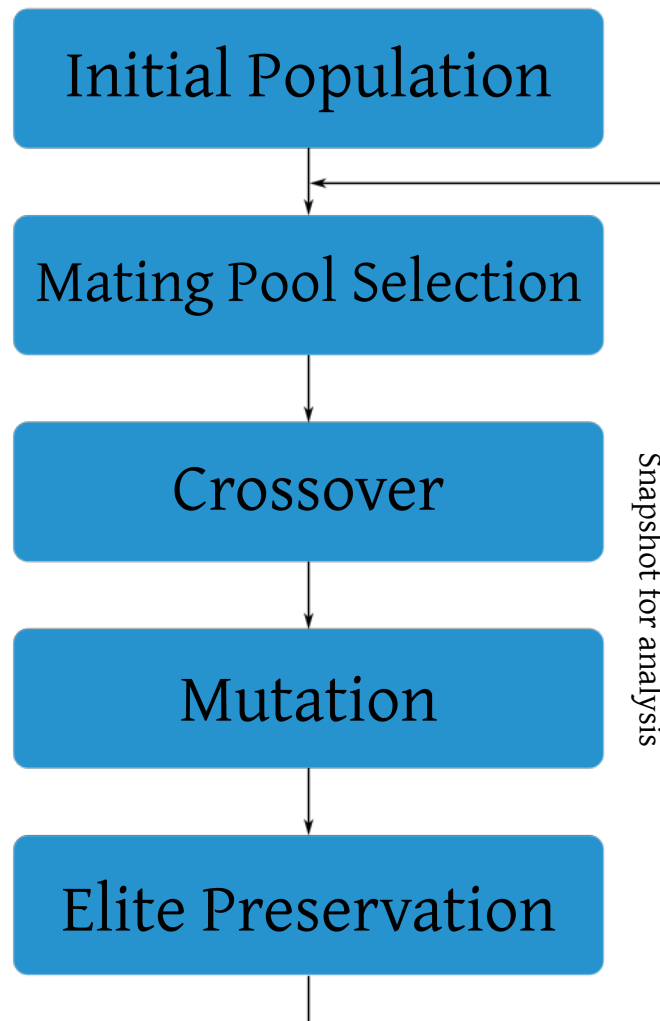


Figure 2. Flowchart of GA training.

This method is widely used in the realm of machine learning, since it eliminates a great hurdle in software engineering: programmers need not to specify the individual aspects of each program and the means that actions are carried out.

There are two aspects of the evolution process that computer scientists want to emulate: natural selection and reproduction. "Natural selection" is the process of choosing which solutions survive to make a new generation of algorithms, the survivors govern the characteristics of next generation. Reproduction is the recombination of genes from

$$f(x_i) = \sum n_{wins}$$

Figure 3. Formula for the fitness score of algorithm x where n_{wins} is the number of games won.

parent to offspring, making the "child" algorithm of an organism have similar characteristics to their two parent algorithms.

Programming natural selection is done by giving an algorithm a test of fitness, going through numerous iterations, then giving a fitness score based on that performance (Hochmuth, 2003). Fitness scores are usually calculated based on the ratio of games won, as shown in Figure 3. The higher the fitness score, the higher the chances of winning with the solution or the move done in the game. Algorithms with low fitness scores are discarded and those with high fitness scores move on to reproduction. The emulation of reproduction is far more complicated. Originally, it was based on random mutation of the algorithms, however this oftentimes produces algorithms that would not run, or are drastically different from the intended purpose of the program. Later development focuses on adding together characteristics from the two parents. This was also limited since this could only be done to characteristics that could be added meaningfully (Holland, n.d.).

Currently, reproduction is done by means of a classifier system. A classifier system is a system where conditions and actions are represented by strings of ones and zeroes corresponding to the presence or absence of that characteristic. For example, as shown in Table 1, since humans have eyes and opposable thumbs and require oxygen, but do not have wings or gills, humans can be recorded as [10011], while the only recorded characteristic of the bacterium that is present is that it requires oxygen, so it is recorded as [00001]. Reproduction can now be done on these "genes" by recombining different genes and making new offspring. In the computer science world, these aspects are usually very basic and primitive, but with strings with lengths reaching tens of thousands of bits long.

Aspect	Human	Fish	Bacterium
opposable thumbs	1	0	0
wings	0	0	0
gills	0	1	0
eyes	1	1	0
requires oxygen	1	1	1

Table 1. *Example of Organisms Sorted Through a Classifier System*

CHAPTER III MATERIALS AND METHODS

Research Design

This Developmental Research is composed of two components: Software Development and Data Collection. In Software Development the code for the move generators and genetic algorithms are initialized, and the efficiency of its implementation is optimized. In Data Collection the organisms will be trained for 500 generations, and the fitness data collected will be analyzed using *R*. The independent variable is the elite preservation method. The dependent variable is the number of generations the specific GA takes to find a no-loss solution. Extraneous variables such as the software specifications can be held constant by the use of the same software such as the operating system and the same Python version (3.8.3 64-bit). The study will not be affected by hardware specifications as it concerns the number of generations instead of the time taken on the system.

Locale of the Study

The software will be developed and data will be collected mainly at the Philippine Science High School – Cordillera Administrative Region Campus. The training and data collection will proceed at the aforementioned location on various personal computers.

Materials and Research Instruments

The software will be developed with Python 3.8.3 on KDE Neon 5.18 using Visual Studio Code 1.45.1 and hosted on GitHub. A link to the repository hosting the software is located in Appendix D. The developmental computer is equipped with 4 GB of Random Access Memory (RAM). Various personal computers will be used to perform the training.

Procedures

An elaborate and chronological description of the procedures to be taken is provided in this section. The proper reasoning for each step taken should be stated, as well as what

materials and research instruments are used in each step.

Treatment of Data

Statement of Hypotheses

$$H_0 : \mu_{\text{UMG}} = \mu_{\text{fitness}} = \mu_{\text{RMG}}$$

H_a : The means are not all equal

Analysis of Data

The data will be analyzed through the *R* with the use of ANOVA which gives the *p*-value to test the hypotheses at a given confidence interval. The study will utilize $\alpha = 0.10$. Should the alternative hypothesis be true, Tukey Honest Significant Differences (Tukey HSD) will be applied to locate the different mean.

CHAPTER IV

RESULTS AND DISCUSSION

“The Results section should set out your key experimental results, including any statistical analysis and whether or not the results of these are significant” (SkillsYouNeed.com, 2020).

Cite literature that support your analyses of the results. It is imperative to include relevant results, regardless of support of hypotheses. Brief descriptions of the results should be provided when clarification is needed.

Softening Points

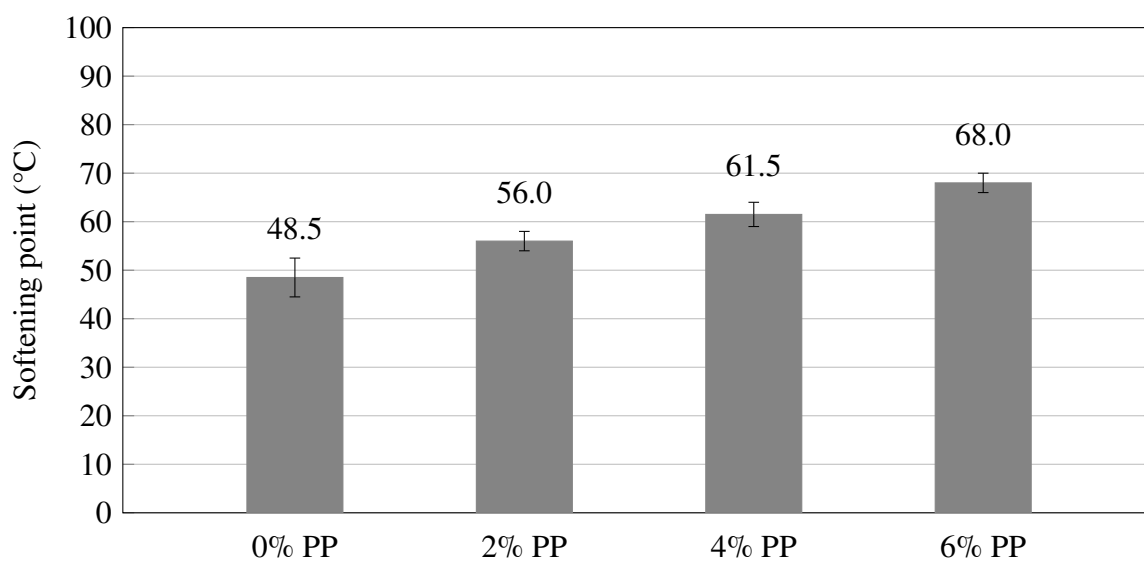


Figure 4. Sample bar graph of softening points.

Table 2. *Characteristics of the Sample*[†]

Variable	Control (n = 45)	Heat & moisture exchanger (n = 49)	Probability
Age (years) ¹	32.7 ± 3.5	36.3 ± 2.7	0.08
Height (m) ¹	1.72 ± 0.60	1.67 ± 0.80	NS
Weight (kg) ¹	76.6 ± 12.8	72.3 ± 16.2	NS
Gender (number of males) ²	21	26	NS
ASA Physical Status ³	2 ± 1	2 ± 1	NS
OR room temperature (°C) ¹	21.1 ± 3.6	20.6 ± 2.9	NS

¹ Data is expressed as mean ± one standard deviation. Probability determined using a two-tailed, unpaired Student's t-test.

² Data is expressed as number within the sample who possess the characteristic. Probability determined using Chi square (or Fisher's Exact test for 2 × 2 tables).

³ Data is expressed as median ± one interquartile range. Probability determined using a Mann-Whitney U test.

[†] Source: Dosch, M. P. (2009, June). *How to write the results and discussion*. Retrieved March 21, 2020, from <https://healthprofessions.udmercy.edu/academics/na/agm/htresult.htm>

Take advantage of `siunitx` package like so: $(5.67 \pm 0.12) \times 10^6$ A. Take advantage of citations with BibLaTeX like so: The question is posed as to whether or not writing systems influence the associations between phonological awareness, morphological awareness, and reading (Ruan et al., 2018, pp. 180–183). More examples are in the `biblatex-cheatsheet` on CTAN.

$$E^2 = (mc^2)^2 + (pc)^2 \quad (1)$$

$$x(t) = \int_{-B}^B X(f) e^{j2\pi ft} df \quad (2)$$

One may refer to figures within this section or in the appendix, similar to the following: One may also refer to appendices similar to the following: “Relevant documentation is included in Appendix D.”

Analysis/Discussion

“ This section has four purposes, it should:

- Interpret and explain your results
- Answer your research question
- Justify your approach
- Critically evaluate your study

The discussion section therefore needs to review your findings in the context of the literature and the existing knowledge about the subject.

You also need to demonstrate that you understand the limitations of your research and the implications of your findings for policy and practice. This section should be written in the present tense.

The Discussion section needs to follow from your results and relate back to your literature review. Make sure that everything you discuss is covered in the results section. ”
(SkillsYouNeed.com, 2020)

CHAPTER V

CONCLUSION AND RECOMMENDATIONS

“The Conclusions and Recommendations may be combined or, in long reports, presented in separate sections. If there are no recommendations to be made as a result of the project, just call this section Conclusions.

The Conclusions section sums up the key points of your discussion, the essential features of your design, or the significant outcomes of your investigation. As its function is to round off the story of your project, it should:

- be written to relate directly to the aims of the project as stated in the Introduction
- indicate the extent to which the aims have been achieved
- summarise the key findings, outcomes or information in your report
- acknowledge limitations and make recommendations for future work (where applicable)
- highlight the significance or usefulness of your work.

The conclusions should relate to the aims of the work[.]” (Monash University, 2020)

LITERATURE CITED

- Adamchik, V. S. (2009). *Game trees*. Retrieved October 19, 2019, from <https://www.cs.cmu.edu/~adamchik/15-121/lectures/Game%20Trees/Game%20Trees.html>
- Aycock, R. (2002). How to win at Tic-Tac-Toe, 14.
- Bhatt, A., Varshney, P., & Deb, K. (2008, July 12). In search of no-loss strategies for the game of tic-tac-toe using a customized genetic algorithm. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. <https://doi.org/10.1145/1389095.1389269>
- Cranenburgh, A. V., Samid, R., & van Someran, M. (2007). Tic-Tac-Toe.
- Dosch, M. P. (2009, June). *How to write the results and discussion*. Retrieved March 21, 2020, from <https://healthprofessions.udmercy.edu/academics/na/agm/htresult.htm>
- Gardner, M. (1958). Mathematical games. *Scientific American*, 232, 126.
- GeeksforGeeks. (n.d.). *Machine learning*. Retrieved October 22, 2019, from <https://www.geeksforgeeks.org/machine-learning/>
- George, W., & Janoski, J. E. (2016). Group actions on winning games of Super Tic-Tac-Toe. arXiv 1606.04779. Retrieved October 22, 2019, from <http://arxiv.org/abs/1606.04779>
- George Mason University. (2020). *Writing an abstract* [Library Catalog: writingcenter.gmu.edu]. Retrieved March 18, 2020, from <https://writingcenter.gmu.edu/guides/writing-an-abstract>
- HackerEarth. (2019). *Depth first search tutorials & notes*. Retrieved October 19, 2019, from <https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/tutorial/>
- Hayes, N., & Loge, T. (2016). Developing a memory efficient algorithm for playing m, n, k games. https://www.micsymposium.org/mics2016/Papers/MICS_2016_paper_28.pdf

- Hochmuth, G. (2003). On the genetic evolution of a perfect tic-tac-toe strategy. *Genetic Algorithms and Genetic Programming at Stanford*, 75–82.
- Holland, J. H. (n.d.). *Genetic algorithms*. Retrieved October 22, 2019, from <https://www2.econ.iastate.edu/tesfatsi/holland.gaintro.htm>
- Lifshitz, E., & David, T. (n.d.). AI approaches to Ultimate Tic-Tac-Toe, 5. <https://cs.huji.ac.il/~ai/projects/2013/UlitimateTic-Tac-Toe/files/report.pdf>
- Monash University. (2020, February). *Conclusions and recommendations* [Library Catalog: www.monash.edu]. Retrieved March 21, 2020, from <https://www.monash.edu/rlo/assignment-samples/engineering/eng-writing-technical-reports/conclusions-and-recommendations>
- Ortiz, A. (2017, July 10). *Machine learning with Hexapawn*. Retrieved October 19, 2019, from <http://ai.aeortiz.com/2017/07/10/machine-learning-with-hexapawn/>
- Ruan, Y., Georgiou, G. K., Song, S., Li, Y., & Shu, H. (2018). Does writing system influence the associations between phonological awareness, morphological awareness, and reading? A meta-analysis. *Journal of Educational Psychology*, 110(2), 180–202. <https://doi.org/10.1037/edu0000216>
- Schaefer, S. (2002, January). *How many games of Tic-Tac-Toe are there?* Retrieved May 3, 2020, from <http://www.mathrec.org/old/2002jan/solutions.html>
- Sharma, A. (2016, January 11). *Machine learning - Applications*. Retrieved October 22, 2019, from <https://www.geeksforgeeks.org/machine-learning-introduction/>
- SkillsYouNeed.com. (2020). *Dissertation writing: Results and discussion*. Retrieved March 21, 2020, from <https://www.skillsyouneed.com/learn/dissertation-results-discussion.html>

APPENDIX A PROJECT PLAN

Table 3. *Task Lists and Duration*

Task	Task Description	Preceding Tasks	Duration (in days)
A	Development of Tic-Tac-Toe Game Platform and Implementation of Algorithms	—	30
B	Testing, Refinement and Optimization of Implemented Programs	A	31
C	Data Collection	B	60
D	Data Analysis	C	60

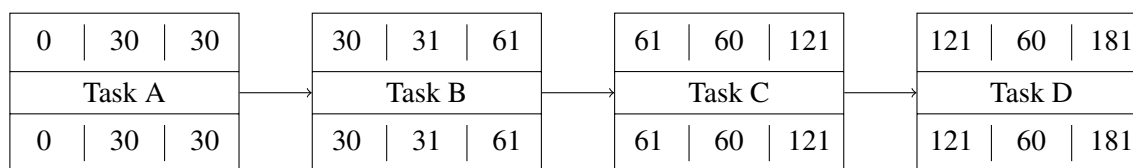


Figure 5. Network chart.

Table 4. *Task Schedule Management and Personnel Assignment Plan*

Task	Task Description	Personnel	Duration (in days)	EST	LST	ECT	LCT
A	Development of Tic-Tac-Toe Game Platform and Implementation of Algorithms	All Personnel	30	NOV 01 2019	NOV 30 2019	NOV 01 2019	NOV 30 2019
B	Testing, Refinement and Optimization of Implemented Programs	All Personnel	31	DEC 01 2019	DEC 31 2019	DEC 01 2019	DEC 31 2019
C	Data Collection	All Personnel	60	JAN 01 2019	FEB 29 2019	JAN 01 2019	FEB 29 2019
D	Data Analysis	All Personnel	61	MAR 01 2019	APR 31 2019	MAR 01 2019	APR 31 2019

Table 5. *Material and Equipment Sourcing Plan*

Protocol	Date/s Needed	Unit	Materials Needed	Potential Source	Remarks
Development of Tic-Tac-Toe Game Platform and Implementation of Algorithm	NOV-01 to 30	1	Laptop with Python	From Home	On Hand
Testing, Refinement and Optimization of Implemented Programs	DEC-01 to 31	1	Laptop with Python	From Home	On Hand
Data Collection and Analysis	JAN-01 to APR-31	1	Laptop with Python and R	From Home	On Hand

Table 6. *Risk Management Plan*

Risk	Safety Measure/Protocol
Development of Carpal Tunnel Syndrome	Frequent 5-minute breaks to relieve muscles
Electrocution	Proper usage of electronic devices
Loss of data	Upload of data into the cloud
Proprietary software trial expiry	Use of free and open-source software

APPENDIX B RAW DATA

Table 7. *Random Table*

Random	Bits
of	data

APPENDIX C STATISTICAL TABLES

Table 8. *ANOVA Table*

Source	SS	DF	MS	F
Treatments	SST	k-1	$SST/(k-1)$	MST/MSE
Error	SSE	N-k	$SSE/(N-k)$	
Total (corrected)	SS	N-1		

APPENDIX D DOCUMENTATION

GitHub Repository: <https://github.com/MasterToast10/paleo-str-g12>