Date: 28 February 2020

Letter of Acceptance for Abstract

Dear Authors: Nathanael Aldo Phillip (a*), Silvester Dian Handy Permana (a), Maya Cendana (b)

We are pleased to inform you that your abstract (ABS-81, Oral Presentation), entitled:

"Modification of Game Agent Using Genetic Algorithms in Game Card Battle"

has been reviewed and accepted to be presented at AASEC 2020 conference to be held on 21-22 April 2020 in Bandung Barat, Indonesia.

Please submit your full paper and make the payment for registration fee before the deadlines, visit our website for more information.

Thank You.

Best regards,

Prof. Dr. Ade Gafar Abdullah, M.Si.
AASEC 2020 Chairperson
CERTIFICATE

No. 3425/UN40.R4/DT/2020

This certificate is awarded to

N A Phillip, S D H Permana and M Cendana

as a Presenter of a paper entitled:

Modification of Game Agent using Genetic Algorithm in Card Battle game

in the 5th Annual Applied Science and Engineering Conference (AASEC) 2020 Universitas Pendidikan Indonesia “Green Technologies for Environmental Sustainability”, 20-21 April 2020.

Prof. Dr. Didi Sukyadi, MA.
Vice Rector for Research, Partnership, and Business Universitas Pendidikan Indonesia

Prof. Dr. Ade Gafar Abdullah, M.Si.
Conference Chair
Modification of Game Agent Using Genetic Algorithm in Card Battle Game

N A Phillip¹,², S D H Permana¹, M Cendana¹

¹ Informatics Universitas Trilogi, Jalan Kampus Trilogi No.1, Jakarta Selatan 12760, Indonesia

Abstract. Game Agent is currently being developed to be an opponent in the game, including games with the Card Games genre. Game Agents on traditional Card Games - such as poker, dominoes, or mahjong cards - have abilities that depend on the value of the cards, but the ability of these Game Agents will not be optimal if used in the Card Battle game. This is because Card Battle has many attributes that must be processed to become opponents. Therefore, this research modifies the Game Agent with Genetic Algorithm to optimize the playing ability of the Game Agent in Card Battle. The computational stages and fitness formula of the Genetic Algorithm are adjusted to the Card Battle rules to increase the computational speed of the Genetic Algorithm. The results of this study prove that Game Agent modification of Genetic Algorithm provides a more optimal playing ability than its predecessor algorithm. Game Agent that has been modified has several abilities that are not owned by the previous Game Agent, such as issuing cards to attack opponents directly and storing SP (Summon Points) they have.

1. Introduction

Game Agent (GA) has long played a role in the development of the game world and has been applied to a variety of games with different genres. GA has the ability to think and make decisions as Non-Player Character or computer-controlled characters in the game [8].

Card Games is a multi-player simulation game (can be played by many players) that uses cards as a medium for the game [3]. Traditional Card Games such as Poker, Domino, and Mahjong have Game Agents whose decision-making uses an algorithm that only processes one game attribute. The development of Card Games now leads to Card Battle which has many game attributes. The use of algorithms from traditional Card Games will not be optimal if used in Card Battle, but in the research of Norton, et al. 2017 [4] who examined the Card Battle Monsters of Darwin still uses the algorithm of the traditional Card Games namely MiniMax to determine the ability of the Game Agent. Surely, Card Battle which has many attributes to be processed will not be optimal when using this algorithm. This is because the MiniMax algorithm can only process one game attribute in its decision making, so the Game Agent is not optimal [6].

Each card in Card Battle has several attributes such as elements, card strength (Attack), and card endurance (Defense). On the other hand, the player itself also has several attributes such as life and energy. Each attribute has its own rules, so it needs to be processed by the algorithm used to produce the best solution. An algorithm that can do this is the Genetic Algorithm [5]. In the Genetic Algorithm, each game attribute will be processed as a parameter that will determine the value of each chromosome.
(a representation of the solution in the Genetic Algorithm). The fitness formula will be used to find the value of a chromosome by computing each parameter on the chromosome.

Modifying the Game Agent by using the Genetic Algorithm in the Card Battle game can optimize the processing flow of the Game Agent. Using arena conditions as parameters can produce new solutions that are better than solutions produced by previous Game Agents. The results of this study are expected to be used as a reference to improve the ability of Game Agents in Card Battle and become a challenging opponent for the players.

2. Method

2.1. Genetic Algorithm
Genetic Algorithm are metaheuristic search algorithms that are used to solve complex search problems [2]. Genetic Algorithm adapts the concepts of natural selection and evolution from Charles Darwin's theory. This concept is based on the term "Survival of the Fittest" which means a population that can survive will eliminate a population that cannot survive. The position of the population that cannot survive will be replaced by the mating result of the population that can survive [9].

![Figure 1. Genetic Algorithm Computational Flow](image)

Figure 1 above shows the computational flow of Genetic Algorithm in general. In the first stage, initialization (a) is made to make a population or chromosome that represents a decision. The second step is evaluating the fitness value of each chromosome using the fitness formula (b). The third stage is selection (c), where each chromosome will be selected based on its fitness value. Chromosomes with a low fitness value will be discarded. In the fourth step, Cross-Over (d), each chromosome that passes the selection will be mated with another chromosome to produce a chromosome with a new value. Finally, the mutation (e) stage. At this stage, some of the chromosomes chosen using the mutation formula will be changed to one of the values of the parameter. This stage makes the Genetic Algorithm suitable when used to solve problems that need to process many variables in their calculations, such as genetics in the marriage of living things and games that have many attributes, including Card Battle game.

2.2. The Proposed Method
Game Agent needs to be designed to be able to understand and process every attribute in the Card Battle game. The attributes are as follows:

- HP (1-20), ATK (1-10), dan Cost (1-7) from each card in hand.
- HP (1-20), ATK (1-10), Cost (1-7), and the position of each card in the player's arena.
- HP (1-20), ATK (1-10), and the position of each card in the enemy arena.
- SP (0-10) owned by player and enemy.
- LP (0-40) left by player and by enemy.
- Number of cards (0-7) in the opponent's hand.
Genetic Algorithm is used to determine the cards that will be summoned by processing all of the above attributes into parameters of the chromosomes. The elitism method is used to secure chromosomes with the highest fitness value so that these chromosomes can be carried to the next generation [1].

2.2.1. Initialization. The Initialization phase is used to make a set of first-generation chromosomes (initial population) with different allele values. Alleles are the components that make up the chromosome itself and the value of each allele on chromosomes in the initial population is randomly generated. The number of chromosomes made at the initialization stage of this study is determined by Equation (1) below.

\[
Total\ Chromosome = CH^2 \times 5
\]  

CH means Cards in Hand owned by NPC in its turn. The number of chromosomes has a maximum limit of 80 chromosomes. Each chromosome represents a solution with 5 different alleles, where alleles are the parameters used in determining the action of the solution. The alleles used on each chromosome in this study represent the position of the cards and the values of the alleles are the cards used.

2.2.2. Evaluation. This stage is used to determine the fitness value of each chromosome by using the fitness formula that has been specifically designed for this research. The fitness formula is designed based on the game system of a prototype Card Battle game. The fitness formula that will be used can be seen in Equation (2) below.

\[
Fitness = \frac{\left(\sum_{i=1}^{n} HP_i/2 + \text{damage}\{\text{position}\}_i\right)}{1 + \left(\text{total cost}/9\right) \text{diffSP}}
\]  

Explanation:
- \(HP_i\) is HP owned by card-\(i\)
- \(\text{damage}\{\text{position}\}_i\) is damage given to the opponent or received by the opponent
- \(SP\) is current SP owned by the NPC
- \(\text{total cost}\) is the total SP needed to summon all of the selected cards to be summoned
- \(\text{diffSP}\) is the difference between SP opponent and SP NPC

The total amount of HP and damage produced by each card summoned will be calculated and added up as a numerator of the fitness formula. Calculation of Damage[position] depends on the condition of the arena. If the opponent's turn in that position has been defeated (HP = 0), then Equation (3) is used. If in that position there is a card in the opponent's arena, then Equation (4) is used. If there are cards in the opposing arena and cards in the NPC arena are empty, then Equation (5) is used. If in that position there are no cards in the opposing arena, Equation (6) is used. The value of \(\text{diffSP}\) is determined from the comparison of SP NPCs and opponents. If the SP NPC is more or equal to the opponent SP, then Equation (7) is used. If SP value of NPC is less than the opposing SP, Equation (8) is used.

\[
damage\{\text{position}\} = ATK_i
\]  
\[
damage\{\text{position}\} = ATK_i - eATK_{\text{position}}
\]  
\[
damage\{\text{position}\} = -1.5\ eATK_{\text{position}}
\]  
\[
damage\{\text{position}\} = -1.5\ ATK_i
\]  
\[
\text{diffSP} = 1
\]  
\[
\text{diffSP} = 1 + \left(eSP - pSP\right)/5
\]
2.2.3. **Elitism Function.** Elitism is used to secure chromosomes with the highest fitness values. This is done by making a copy of the chromosome and securing the chromosome to be carried to the next generation. The chromosomes to be elitist are selected using Equation (9):

\[
\text{elite chromosome} = \text{MAX} (\text{chromosome}_0: \text{chromosome}_n)
\]  

(9)

2.2.4. **Termination.** Termination is the process of checking the fitness value of a chromosome that has passed elitism. If a chromosome that is secured using elitism meets the criteria, then the chromosome will be used as a solution for the movement of the Game Agent, but if it is the opposite, the original chromosome will undergo the process of making a new population by the method of selection, reproduction, and mutation. A copy of the elitism of the chromosomes will be added to the new population. An elite chromosome will be selected as the solution if it is the 10th generation or if its fitness value is equal to or more than 4.

2.2.5. **Selection (Tournament).** The selection stage is used to select several chromosomes for further processing. Selection uses the Tournament method which pairs one chromosome with another to be pitted. Chromosomes that have the highest fitness value will be selected for the reproductive stage. If the chromosomes have the same fitness value, the winner will be chosen randomly. The selection stage will only be carried out if the number of chromosomes is more than 10.

2.2.6. **Cross-Over.** The results of the selected chromosomes will then be reproduced using the Uniform Cross-Over method. The result of the reproduction of this method is in the form of two child chromosomes which have the parameters of the two-parent chromosomes (parent). The parameters that are owned by children depend on the percentage of each parent.

2.2.7. **Mutation.** The mutation stage is used to alter some of the allele values of a chromosome that has a duplicate value on the allele. This study utilizes the mutation stage to improve the value of alleles that are not in accordance with the rules of the game [7]. If there are no duplicates in the alleles, then the mutation will change the value of one of the alleles randomly.

3. **Results and Discussion**

Game Agent needs to be modified with Genetic Algorithm before being implemented in the Card Battle application. As explained in Section 2, Genetic Algorithm has 7 stages of computation to solve a game case. The Allele value of a chromosome is the n-th card in the hands of an NPC with a cost less than SP NPC. The number of chromosomes made in the first generation is determined by Equation (1). If the NPC has 6 CH (Cards in Hand), then the number of chromosomes in the first generation is 62 * 5 = 180, but according to the rules of the Initialization stage, if the number of chromosomes in the first generation exceeds 80, then they will be cut to 80.

Each chromosome will be determined its fitness value at the Evaluation stage using Equation (2). For example, Equation (2) is used to calculate the fitness value of chromosome for the followings case. The second allele has a value of 2 which refers to a card with 5 HP and 5 ATK. The enemy has a card (with 10 HP, 0 ATK) in 1st position and another card (3 HP, 1 ATK) in 3rd position. The enemy has 0 SP (Summon Point). Chromosome contains alleles with a value of [0] [2] [0] [0] [0] which means a card (5 HP, 5 ATK) summoned in the 2nd position. The total cost used by the chromosome in this case is 3 points. The formula used to calculate diffSP is taken from Equation (7) because the NPC has more SP than the opponent. The damage [position] formula used for calculations in positions 1 and 3 is taken from
Equation (5) because the chromosome does not plan to summon cards in position 1 and 3. The damage [position] formula used for the 2nd place calculation is taken from Equation (6) because the opponent does not have a card in 2nd position. Based on the things above, the calculation of the fitness formula from the chromosome is like in Equation (10).

\[
fitness = \frac{(damage[1] + \frac{HP}{2} + damage[2] + (damage[3]) + 2(SP - total cost))}{1 + \frac{total cost}{SP(diffSP)}} = 6.38 \tag{10}
\]

According to the allele value of the chromosome in the case above, it is more effective to summon the 2nd card from hand to reduce the opponent's LP (Life Point) by 5 points rather than holding the opponent's card attack on the 3rd position to the NPC LP by 1 point. The calculation from the chromosome above also states that the card in the 1st position in the opponent's arena does not pose a threat to the NPC because it has an ATK (Attack) of 0 points. Chromosomes with a total cost that exceeds the SP of NPC will be given a fitness value of -15 to avoid using these chromosomes. Chromosome with the highest fitness value in their generation will be secured by the Genetic Algorithm by making a copy of the chromosome. If there are other chromosomes with the same fitness value, Elitism will take the first chromosome as an elite chromosome. Take the case above as the current chromosome with the highest fitness value.

During the Termination stage, if the elite chromosome has a fitness value \( \geq 4 \), then the computation process of the Genetic Algorithm will be stopped and the NPC will summon a card based on the allele value of the elite chromosome. The current elite chromosome has a fitness value of 6.38, which means the elite chromosome passes the Termination stage and the NPC will summon the 2nd card in hand in the 2nd position. If the elite chromosome does not meet the requirements at the Termination stage, each chromosome in that generation will be passed on to Selection, Cross-Over, and Mutation stage for further processing as chromosomes in the next generation. Tournament is used as an approach from the Selection stage. Each chromosome will be divided into 2 groups for comparation. Chromosomes that have a higher fitness value will be reprocessed at a later stage, but chromosomes that have a lower fitness value will be removed from the population. For example, if chromosome 1 has 14 fitness and chromosome 2 has 10 fitness, then chromosome 1 will be processed to the Cross-Over stage and chromosome 2 will be removed from its population.

After passing the Selection stage, chromosomes that pass will be mated with other chromosomes in their generation. The approach used at this stage is Uniform Cross-Over. Each chromosome will be paired with another chromosome as a parent and will be mated to produce a pair of offspring chromosomes. 40% alleles from parent 1 will be combined with 60% alleles from parent 2 to produce offspring 1. Likewise, 60% alleles from parent 1 will be combined with 40% alleles from parent 2 to produce offspring 2. Both offspring will be processed again at the Mutation stage.

The results of the marriage at the Cross-Over stage will be changed back to the alleles value at the Mutation stage. This stage aims to correct the value of alleles that are not in accordance with the rules of the game, such as double alleles which means the same card on a chromosome and giving a value of 0 to alleles whose card positions are not available or are already occupied by other cards. If there is no improvement in the allele value, then Mutation will change one of the allele values randomly. The chromosome results from the Mutation stage will be made into a second-generation chromosome population and will be reprocessed from the Evaluation to Termination stages. If the chromosome still does not meet the requirements of the Termination stage, then repeat Selection, Cross-Over, and Mutation stage again and the results will be carried over to the third generation. This process will continue until the tenth generation of chromosome populations. Termination will choose the elite chromosome in the tenth generation as a choice of cards to be summoned by the NPC.

The computational stages of the Genetic Algorithm can provide the possibility for the NPC to summon cards based on the fitness value of a chromosome. Cards can be summoned on the empty enemy arena when circumstances are favorable for NPCs. For example, if an attack from the opponent's card
in the 3rd position cannot be blocked by the NPC, the opponent will only reduce the LP (Life Point) of the NPC by 1 point. Of course, issuing cards with more than 1 point of ATK to reduce the opponent's LP will be more advantageous compared to cards summoned only to block attacks by 1 point from the opponent. Besides, using an algorithm can make NPC save SP to summon better cards.

4. Conclusion
Based on the results of the study it can be concluded that the Genetic Algorithm can optimize the playing ability of the Game Agent in the Card Battle game. Game Agent is modified by replacing the previous algorithm (MiniMax) with the Genetic Algorithm that has been adapted to the rules of the Card Battle. Modified Game Agent Genetic Algorithm can process every attribute in the Card Battle game, so Game Agent has capabilities that the previous algorithm did not have. The new ability possessed by the Game Agent Genetic Algorithm modification is to summon cards in an empty arena and save SP to summon stronger cards. Genetic Algorithm is adjusted to the playing rules of the Card Battle game to reduce the computational time required. Adjustment of Genetic Algorithm is done by changing the computation flow, as well as changing variables and designing the formula used based on each attribute contained in the Card Battle game. The computational time of the adjusted Genetic Algorithm is 0.1ms to 44ms, but there is no fixed pattern for computing the computation time.

5. Acknowledgment
This research would not have been possible without the help of my lecturers. Mr. Silvester Dian Handy Permana and Ms. Maya Cendana. Their enthusiasm, knowledge, and exacting attention to detail have been an inspiration to me. Hence, keeping my work on track from what I planned since the start of this research. I am also grateful for Malfius Arva Prasetyo, my colleague at Universitas Trilogi who helped me through my most difficult times during the writing of this paper, not only by encouraging me but also by giving me some references for my research.

References
[1] Adisusilo A K, Hariadi M, Zaini A and Nugroho S M S 2016 Optimizing of Boxing Agent Behavior Using Elitism Based Genetic Algorithm Kursor 7 55–62
[2] Kalayci C B, Polat O and Gupta S M 2016 A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem Ann. Oper. Res. 242 321–54
[3] Marchetti E and Valente A 2015 Learning via game design: From digital to card games and back again Electron. J. e-Learning 13 167–80
[4] Norton D, Ripamonti L A, Ornaghi M, Gadia D and Maggiorini D 2017 Monsters of Darwin: A strategic game based on artificial intelligence and genetic algorithms CEUR Workshop Proc. 1956
[5] Permana S D H 2019 Implementation of Min Max Algorithm as Intelligent Agent on Card Battle Game IJIISTECH (International Journal of Information System & Technology) 2(2) 53-58.
[6] Permana S D H, Bintoro K B Y, Arifitama B, Syahputra A and Cendana M 2019 Improvisation of Minimax Algorithm with Multi Criteria Decision Maker (MCDM) in the Intelligent Agent of Card Battle Game International Conference on Electrical Engineering and Computer Science (ICECOS) 254-258 IEEE.
[7] Roy S K and Mula P 2016 Solving matrix game with rough payoffs using genetic algorithm Oper. Res. 16 117–30
[8] Russell S and Norvig P 2013 Artificial Intelligence A Modern Approach
[9] Widodo D K 2017 Generator Teka Teki Silang Menggunakan Algoritma Genetika dan Multithreading untuk Menghitung Fitnessnya PROXIES 1 75–82
[10] Yundong C, Zhiqi S, Siyuan L, Han Y, Xiaogang H, Jun J, Martin J M, Cyril L and Chunyan M 2014 An agent-based game for the predictive diagnosis of parkinson's disease Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems AAMAS’14 1663-1664.