Optimizing the evaluation parameters of Amazon chess with parallel genetic algorithm

1st Shengran Qiu
School of Navigation
Wuhan University of Technology
Wuhan, China
email: qiushengran@whut.edu.cn

2nd Yiyun Ma
School of Vehicle Engineering
Dalian University of Technology
Dalian, China
email:2199588586@qq.com

3rd Yutao Xu
School of Computer and Artificial Intelligence
Wuhan University of Technology
Wuhan, China
email: 2932901873@qq.com

4th Yucheng Ji
School of Computer and Artificial Intelligence
Wuhan University of Technology
Wuhan, China
email: 2243470906@qq.com

5th Hongbo Deng
School of Computer and Artificial Intelligence
Wuhan University of Technology
Wuhan, China
email: 1139432393@qq.com

2023 35th Chinese Control and Decision Conference (CCDC) | 979-8-3503-3472-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCDC58219.2023.10326586

***Abstract*—In the process of computer game, in order to improve the accuracy of the situation evaluation of Amazon AI program, an optimization scheme based on parallel genetic algorithm is proposed. This scheme uses a multi-node distributed parallel algorithm to optimize the weight of the parameters of Amazon's chess game evaluation function. The whole optimization system is mainly composed of server and client. Each client uses multithread parallel computing and transmits the result data to the server. Through experiments, the optimization results of the parameters of the situation evaluation function can be obtained in 5 hours. It is proved that the situation assessment is more accurate and the capability of the AI program is significantly improved after the optimization of this scheme.**

Keywords—computer games, Amazons, evaluation, optimization, genetic algorithm

I. INTRODUCTIONS

Computer games is one of the most important and challenging research fields of Artificial Intelligence [1]. It's very difficult to design and implement a high-level computer game system. With the development of related technologies, such as genetic algorithm, deep learning, neural network, computer games is becoming more difficult and enjoyable as well, whose level has been improved greatly[2].

AlphaGo, an AI Go program developed by Google DeepMind team, defeated the top professional player by using neural networks and deep learning technology, thus establishing the important position of neural networks and deep learning in the field of AI[3,4]. Although neural networks and deep learning technology have achieved great success in artificial intelligence, they require excellent computing ability of computer hardware (CPU/GPU). Take AlphaGo as an example, it has carried out 20 million games of self-game training to obtain experience and patterns similar to those of human chess players, which requires a lot of computing ability. In practical applications, if the computer hardware doesn't have enough computing ability or sufficient training , the effect of using neural networks and deep learning technology will not be ideal. In the case of the present conventional computing power, we consider the use of bionics swarm intelligence algorithm (genetic algorithm), such as Amazons chess, to achieve lightweight computing optimization program. Take AlphaGo as an example, it has trained itself in 20 million games to gain experience and patterns similar to those of human chess players.

Amazons was invented in 1988. It is a typical two-player board game played on a rectangular board, with standard size 10×10, and it is a complete information game with simple rules, but it is very complex[5,6]. Each player controls 4 queens. Two players, Black and White, move alternately. When a player has no legal move, he loses the game.

A move in Amazons is comprised of two compulsory phases: queen move and arrow shot. Firstly, a player moves one queen with the constraint that it may not cross or enter a square occupied by an amazon. Secondly, the queen has to shoot an arrow, which travels in the exact same way as a queen, from the current queen just moved to [7]. This special rule leads to many moves in every step of Amazons. There are about 1500 kinds of moves, with great uncertainty in the evaluation of most moves[8-9]. Because of these reasons, Amazons attracted a lot of scholars and researchers.

In this paper, an optimization scheme was proposed, and an Amazons game system was implemented. The paper is composed of 5 sections, including Introduction, Technologies, Scheme, Experiments and Conclusion. Section 1 introduces computer games, the game of Amazons and the organization of this paper. Section 2 describes the relation technologies used for Amazons game, including search engine, evaluation function and parameters optimization about the game. Section 3 gives details of solution design, including architecture, optimization algorithm as well as an island model. Section 4 presents the experiments, it contains results of optimization. The conclusion follows in the final Section 5.

II. TECHNOLOGIES

A. Search Engine

Search engine is one of the most important modules in computer game. It usually adopts variational algorithms based on mini-max algorithm, such as PVS (principal variation search) algorithm, MTD (Memory-enhanced Test Driver with node n and value f) algorithm, etc [1]. The idea of mini-max algorithm is widely used in computer games. In the process of game, taking the assessed value as the standard, a best search method is obtained by repeated maximal search and minimal search, which perfectly solves the search problem of computer game, but the game tree expansion of a complex chess game is very large. It is impossible to do exhaustive search in many occasions. The accuracy of the evaluation function greatly affects the effectiveness of the search algorithm [10].

In reality, when playing chess, the biggest difference between the master and the ordinary player is that the master has a deeper prediction of the chess game. In the same way, this idea is adopted in computer game. Through the game tree expansion, nodes of each layer are repeatedly searched for maximum and minimum search, and finally an optimal node is found. In order to improve the search efficiency, the game tree is pruned. It greatly improves the efficiency of the search engine.

B. *Evaluation Function*

Evaluation function is the other key module additional to search engine in computer game. For Amazons' evaluation function, it is generally accepted that the evaluation scheme is based on queen-move algorithm and king-move algorithm raised by Professor Jens Lieberum[11]. In reference [12], the calculation and theoretical description of their evaluation factors w, t1, t2, c1 and c2 are introduced in detail. According to the reference, the expression of evaluation function can be expressed as follow in Formula (1).

$$\text{value} = f_1(w)\times t1 + f_2(w)\times t2 + f_3(w)\times c1 + f_4(w)\times c2 \quad (1)$$

There into, t1 and t2 are the factors based on the queen-move algorithm and the king-move algorithm, which represent the ownership of the calculation space. c1 and c2 are the factors that calculate the strength of the space ownership. w is an assessment factor that describes the progress of games according to the obstacles set by both sides, and can judge the emergence of the Filling situation. The expressions of f_1 (w), f_2 (w), f_3 (w) and f_4 (w) in the Formula (1) are difficult to ascertain. A layered scheme based on chess step number is proposed in reference [13]., in which the benefits of the w parameter are not used. Therefore, this study proposes an evaluation scheme which is based on layered w parameters. The w interval is divided into n segments, w_1, w_2, w_3..., w_n, each of which corresponds to a set of t1, t2, c1, c2. For each group of t1, t2, c1 and c2, there are corresponding weights wt1, wt2, wc1 and wc2. If the interval is smaller, the set of evaluation parameters will be more reasonable. So the evaluation function are described in Formula (2).

value =

$$\begin{cases} wt1_1 \times t1 + wt2_1 \times t2 + wc1_1 \times c1 + wc2_1 \times c2, w \in (0, w_1) \\ wt1_2 \times t1 + wt2_2 \times t2 + wc1_2 \times c1 + wc2_2 \times c2, w \in [w_1, w_2) \\ \ldots\ldots \\ wt1_n \times t1 + wt2_n \times t2 + wc1_n \times c1 + wc2_n \times c2, w \in [w_{n-1}, w_n) \end{cases} \quad (2)$$

C. *Parameters Optimization*

Machine learning requires a large amount of prior data and long learning time. Deep learning or reinforcement learning requires high hardware configuration. For the computer games on notebook with poor computing ability, the traditional lightweight game algorithm still makes sense. It is necessary to develop a lightweight game system.

Traditional genetic algorithm searches slowly and is easy to converge too early. The distributed parallel system is composed of the main processor and several sub processors. Its genetic algorithm system divides the initial populations into several small populations and distribute them to several sub processors. Each sub processor independently calculates fitness, selection and crossover and mutation. Part of the individuals are regularly transmitted to the main processor. After receiving the individuals, the main processor disposes the information and sends it to other sub processors. The operation repeats until the genetic process is terminated.

In this study, each group parameters in the evaluation function is taken as an evaluation value. It corresponds to the individual in the genetic algorithm. Each parameter in the group corresponds to a segment of chromosome in the genetic algorithm.

III. SOLUTION DESIGN

A. *Algorithm Model*

In this study, a parallel genetic algorithm model was proposed (shown in Fig. 1). The client played the game by itself. Lightweight machine learning and optimization of evaluation parameters was realized. The server was responsible for optimizing task allocation, scheduling and control.

The main task of the server was to buffer the genetic information from the clients, and transmitted the data to other clients according to the requirements. The clients communicated with other clients through the server. In this way, information exchanged among different sub populations was completed. The information exchange among sub populations was based on island model.

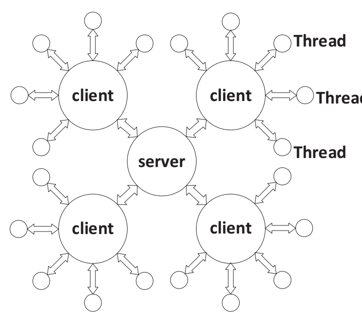

Fig. 1. Diagram of parallel genetic algorithm model

Genetic information is transmitted from one sub population to all other sub populations. The diversity of each sub population is enriched. The island model avoids the shortcomings of traditional genetic algorithm which are easy to be precocious.

In the client program, the most important part of the parameter optimization algorithm is multi-thread parallel genetic algorithm. It generally includes: gene coding, fitness function determination, individual selection, gene crossover and mutation.

B. *Gene Encoding*

The weight of each parameter corresponds to the expressive form of genetic algorithm. The mapping from the phenotype to the genotype is called coding. The commonly codings are float-encoding and binary-encoding. In this solution design scheme, binary-coding is used. The weight of each parameter is represented by a 0/1 binary string, and the string length depends on the accuracy of the solution. This scheme requires to be accurate to two decimal places, and the weight range is between [0,1]. It is known that $2^6 < 1\times10^2 <= 2^7$, so the encoded binary string needs at least 7 bits. Since there are four parameters in total, a two-dimensional array gene[4][7] can be defined to store relevant data.

C. Fitness Function Determination

The genetic algorithm does not use the external information in the evolutionary search but the fitness function as the only basis, using the fitness of each individual in the population to search. Fitness function is determined by objective function. There is no a set of fixed evaluation functions at present. Genetic algorithm is used to solve the problem of computer games evaluation. To solve the problem of computer game evaluation with genetic algorithm, tournament algorithm is mainly used. That is to say, different individuals (different weights of parameters) form a group to play chess with each other. In order to avoid the difference between the two sides, each two individuals exchange order to play the game. The winner gets 1 point and the loser loses 1 point. After finishing the game, each group scored their fitness.

This method of calculating fitness has a bug, that is, the uncertainty of chess strength. Assuming that role-A is superior to role-B, role-B is better than role-C, but the actual result may be that role-C is superior to role-A. It has been verified many times in experiments.

D. Individual selection

The first step of individual selection is to determine the fitness of each individual. Then, by selecting algorithm, crossover and mutation of selected individuals are picked out. This study proposes tournament selection algorithm for individual selection. First, the initial population is divided into n groups, and then each group is played by each player, each score is calculated, and the scores are taken as their fitness respectively.

Assume that the initial population has m individuals, the number of races after grouping is N1, and its calculation is shown in Formula (3).

$$N1 = \left[\left(\frac{m}{n}\right)^2 - \frac{m}{n} \right] \times n \tag{3}$$

Assume that the number of games in non-group individuals is N2, and its calculation is shown in Formula (4).

$$N2 = (m)^2 - m \tag{4}$$

Compared with N1 and N2, the number of games played after grouping is significantly less than that without grouping, and the more the groups are, the fewer the games are played. However, the diversification loss increases with the growing scale and intensity of competition. After grouping, the group is followed up in turn, and the better individuals in the group are recorded. The initial population size of this scheme is 120 individuals. The population is divided into 30 groups. Each group has 4 individuals, and then each individual is played with others. Finally, the adaptability of each individual is obtained.

Because each group of individuals is sometimes independent, the scheme uses multi-thread technology. The game between the 30 groups is assigned to 3 threads. It not only improves the utilization ratio of CPU, but also improves the speed of data processing. After playing chess, the fitness of each individual is obtained. The better individuals in each group are marked. Using these better individuals as parent, crossover and mutation operations are applied. And then, the next generation is produced. In the end, the new generation is blended with the former superior individuals to form a new generation of population.

E. Gene Crossover and Mutation

Gene crossover, that is, gene recombination, is the replacement and reorganization of some parts of the two male parents and the operation of new individuals. The main crossover operations include single-point crossover, multi-point crossover and uniform crossover.

Considering that multiple parameters need to be optimized, uniform crossover is adopted. Four (0,1) masks are generated, corresponding to 4 parameters that needs to be optimized. If it is 1, crossover and mutation operations are carried out according to the probability of crossover and mutation. This ensured stable changes between parameters. The crossover and mutation probability directly affects the convergence of genetic algorithm. The greater the crossover probability is, the faster the new individual will generate. But the damage to the gene also increases. If too small, it might lead to slow convergence of solutions. Similarly, the greater the probability of mutation is, the more likely it is to generate new individuals. But it's too big to lose the characteristics of the genetic algorithm.

An adaptive genetic algorithm is adopted in this reference. According to the distribution of fitness in population, the probability of crossover and mutation are set. When the fitness is lower than the average fitness, the cross mutation rate goes higher. Conversely, when the fitness level is higher than the average fitness, a smaller cross mutation rate is adopted. Cross probability Pc and mutation probability Pm can be obtained from the following Formula 5 and Formula 6.

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}}, f' \geq f_{avg} \\ P_{c1}, f' \leq f_{avg} \end{cases} \tag{5}$$

$$P_m = \begin{cases} P_{m1} - \dfrac{(P_{m1} - P_{m2})(f_{\max} - f)}{f_{\max} - f_{avg}}, f \geq f_{avg} \\ P_{m1}, f \leq f_{avg} \end{cases} \tag{6}$$

Among them, f' crosses two individuals in a larger degree of fitness. f is a variation of individual fitness. $f_{\max}$ is the largest fitness in the population. f_{avg} is the average fitness in the population. P_{m1}=0.1, is a constant value. P_{m2}=0.001, is the mutation probability of the largest fitness population. P_{c1}=0.9, it's a constant value. P_{c2}=0.6 is the crossover probability of the largest fitness population.

IV. EXPERIMENTS

Amazon Gaming Software includes several clients and one server. The optimization experiment is simulated in the local computer environment (shown in Figures 2). The hardware environment of the local machine is: CPU Core i5-2400, logical processor 4 Core, and RAM 16G.

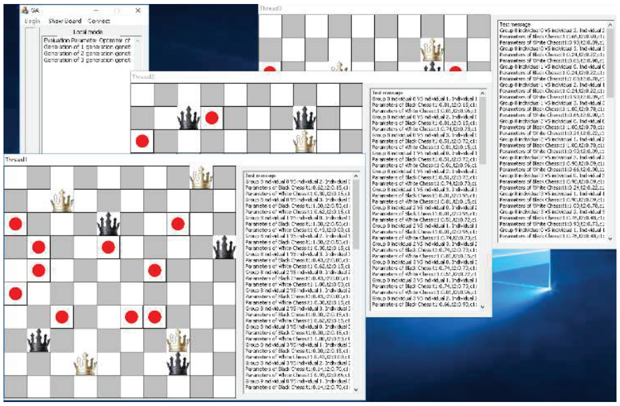

Fig. 2. Process of single machine and multi-threaded parallel optimization

Based on previous experiences, the w parameters are generally stable between 0 and 70. So in the experiments, w is divided into five parts: (70,45), (45,30), (30,20), (20,10), and (10,0). The genetic termination condition was the 100 generation of heredity. Through tens of thousands of experiments in Amazon games (a screenshot of a gene operation was shown in Fig. 3), a large amount of data is obtained after running for 5 hours.

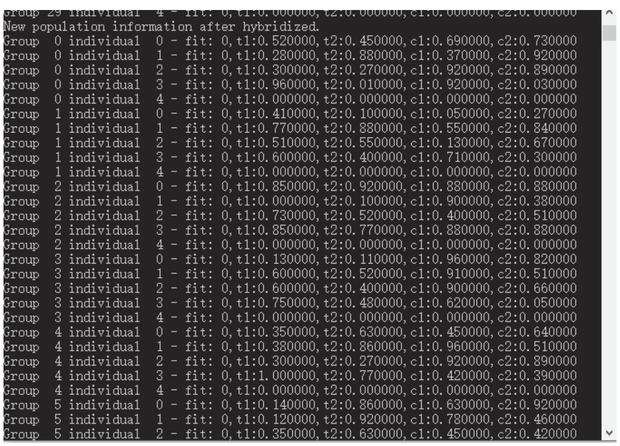

Fig. 3. Part of a gene operation

The experiments data is collected and collated, shown in TABLE Ⅰ and TABLE Ⅱ.

TABLE I. PARAMETERS(C1,C2,T1,T2,W) OF EVALUATION FUNCTION

moves	c1	c2	t1	t2	w
1	7.00	10.83	18.50	26.10	68.00
5	-1.25	-0.67	1.20	-4.90	62.50
10	1.25	1.33	4.40	7.90	52.75
15	3.75	4.17	7.20	5.00	40.25
20	2.88	6.17	6.40	10.20	27.38
25	-0.66	17.00	4.10	4.10	2.66
30	-1.00	18.00	3.00	3.00	0.00
35	-0.63	13.00	3.00	3.00	0.00
40	0.63	8.00	3.00	3.00	0.00
45	3.00	4.00	4.00	4.00	0.00

TABLE II. PARAMETERS(M1,M2,VALUE) OF EVALUATION FUNCTION

moves	m1	m2	value
1	-25.36	40.00	12.06
5	-11.31	-1.00	-1.85
10	-4.48	-8.00	3.37
15	-1.17	-17.00	2.90
20	-0.82	-5.00	4.54
25	-0.14	10.00	3.86
30	0.00	0.00	3.00
35	0.00	0.00	3.00
40	0.00	0.00	3.00
45	0.00	0.00	4.00

The parameters' trend is shown in Fig. 4 and Fig. 5. Through the data, we found that w and the number of moves basically satisfy the linear relationship. According to the experimental data, the w parameters are stable between 0 and 70. According to the parameters, the evaluation of the Amazons game can be calculated. In this way, using these parameters to describe the progress of game is feasible.

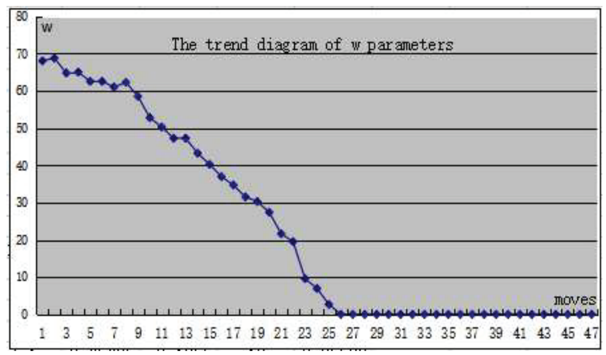

Fig. 4. The trend diagram of w parameters

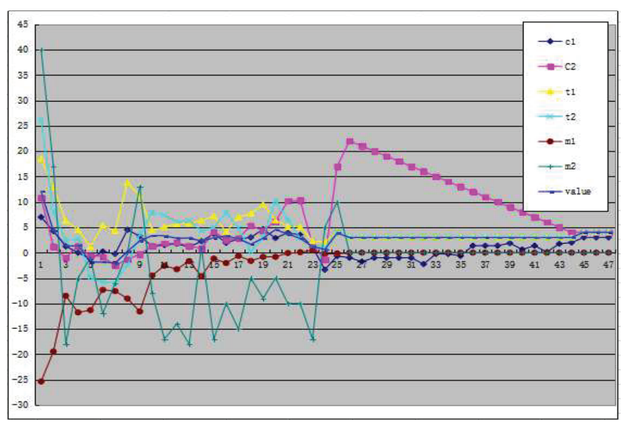

Fig. 5. The trend diagram of other parameters

V. CONCLUSION

In this study, the parallel genetic algorithm is used to optimize the parameters in the evaluation function of Amazons. Theoretically, it has changed the original adverse situation based on the experience of chess to determine the weights of parameters. When the game program has a complete set of correct evaluation functions, the chess ability will be obviously improved. If some parameters needs to be added, it can also be optimized by this method instead of manual adjustment.

In this way, it is possible to maximize the effect of the evaluation function. Provide more accurate evaluation value for search engine, so as to further enhance the power of chess. Experiments show that the parallel genetic algorithm is effective in optimizing Amazons evaluation parameters.

ACKNOWLEDGMENT

REFERENCES

[1] Wang Y , Qiu H , Yanyan W U , et al. Research and development of computer games[J]. CAAI Transactions on Intelligent Systems, 2016, 11(6), pp.788-798.

[2] Qiu H, Zhang P, Wang Y, et al. Analysis of Search Algorithm in Computer Game of Amazons[C], Control and Decision Conference (CCDC), 2014 26th Chinese. IEEE, 2014, pp.3947-3950.

[3] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search.[J]. Nature, 2016, 529(7587):484-489.

[4] Silver D , Schrittwieser J , Simonyan K , et al. Mastering the game of Go without human knowledge[J]. Nature, 2017, 550(7676):354-359.

[5] Kloetzer J , Iida H , Bouzy B . The Monte-Carlo Approach in Amazons[C]// Computer Games Workshop, 2007, pp. 185–192.

[6] Kloetzer J , Iida H , Bouzy B . Playing Amazons Endgames[J]. ICGA journal, 2009, 32(3):140-148.

[7] Song J , Muller M . An Enhanced Solver for the Game of Amazons[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2015, 7(1):16-27.

[8] Kloetzer J , Iida H , Bouzy B . Playing Amazons Endgames[J]. ICGA journal, 2009, 32(3):140-148.

[9] Song J , Muller M . An Enhanced Solver for the Game of Amazons[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2015, 7(1):16-27.

[10] Snatzke R G . New results of exhaustive search in the game Amazons[M]. Elsevier Science Publishers Ltd. 2004.

[11] Qiu H, Wang H, Wang Y, Auto parameter adjustment of amazon chess evaluation function realized by Q-Learning[J]. Journal of Chongqing University of Technology (Natural Science), 2022, 12

[12] Lieberum J . An evaluation function for the game of amazons[J]. Theoretical Computer Science, 2005, 349(2):230-244.

[13] Guo Q , Shuqin L I . Research on evaluation function computer game of Amazon[J]. Computer Engineering & Applications, 2012, 48(34), pp.50-54.