Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5188335, 9 pages
https://doi.org/10.1155/2022/5188335

*Research Article*

# Research on Biological Population Evolutionary Algorithm and Individual Adaptive Method Based on Quantum Computing

**Xuandiyang Lu**

*Dong Hua University, Shanghai 201620, China*

Correspondence should be addressed to Xuandiyang Lu; 181100221@mail.dhu.edu.cn

Received 19 January 2022; Revised 17 February 2022; Accepted 22 February 2022; Published 22 March 2022

Academic Editor: Kalidoss Rajakani

On the basis of classical computer, quantum computer has been developed. In dealing with some large-scale parallel problems, quantum computer is simpler and faster than traditional computer. Nowadays, physical qubit computers have many limitations. Classical computers have many ways to simulate quantum computing, the most effective of which are quantum superiority and quantum algorithm. Ensuring computational efficiency, accuracy, and precision is of great significance to the study of large-scale quantum computing. Compared with other algorithms, genetic algorithm has more advantages, so it can be more widely used. For example, strong adaptability and global optimization ability are the advantages of genetic algorithm. Through the research in Chapter 4, we can conclude that the variance of A2C is obviously smaller than that of PPO. Furthermore, it can be concluded that A2C has better robustness.

## 1. Introduction

The purpose of this paper is to summarize all the important aspects and results affecting the field of quantum computation and quantum information. First, we are familiar with the characteristics and principles of quantum mechanics and provide more effective and secure information processing. Then, their applications in general information theory, cryptography, algorithms, computational complexity, and error correction are discussed [1]. If one day the bits of a computer are reduced to the size of a single atom, the quantum mechanical effect may profoundly change the nature of computing itself [2]. This special issue on mathematical structure in computer science contains several contributions related to modern quantum information and quantum computing [3]. Computer is usually regarded as a general-purpose computing device. In other words, it is believed that it can simulate any physical computing device at a computational time cost of up to one polynomial factor [4]. Quantum information is stored in a state containing multiple quasiparticles, which have topological degeneracy. The unitary gate operation required by quantum computation is realized by weaving quasiparticles and then measuring the states of multiple quasiparticles. The fault tolerance of topological quantum computer comes from the nonlocal coding of quasiparticle state, which makes quasiparticle immune to errors caused by local disturbance [5]. Decoherence in quantum computer is expressed by semigroup method. It is proved that these subspaces are stable to perturbation, and it is possible to carry out universal quantum computation in these subspaces [6]. It has been proved that the combination of genetic algorithm and $K$-nearest neighbor classifier can extract protein-water binding information from X-ray crystal protein structure data well [7]. Combining the advantages of traditional real genetic algorithm and structured genetic algorithm, this paper proposes a real structured genetic algorithm to deal with biological logic circuit design problems [8]. The interference cancellation system is designed by using LM algorithm of artificial neural network with adaptive learning ability, and the number of hidden layer neurons, learning rate, and momentum factor of neural network is optimized by using swarm intelligence algorithm [9]. We use the main biological learning methods instead of machine learning to adjust synaptic weights. We use evolutionary algorithm to optimize the learning meta-parameters. These algorithms are implemented in parallel and use the island model approach

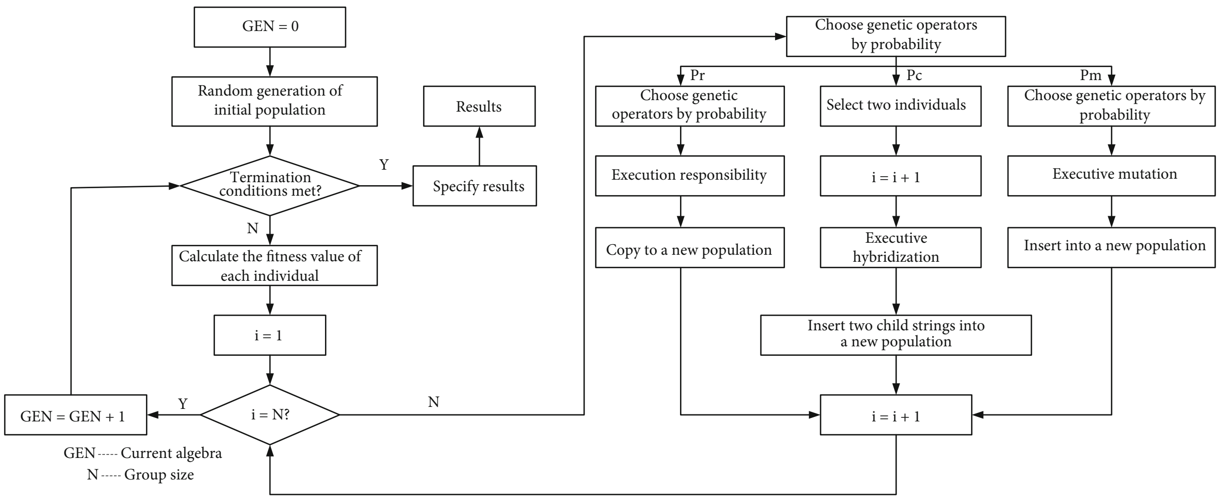
GEN = 0
Random generation of initial population
Termination conditions met?
Y
N
Specify results
Results
Calculate the fitness value of each individual
i = 1
i = N?
GEN = GEN + 1
GEN----- Current algebra
N----- Group size
Choose genetic operators by probability
Pr
Pc
Pm
Choose genetic operators by probability
Select two individuals
Choose genetic operators by probability
Execution responsibility
i = i + 1
Executive mutation
Copy to a new population
Executive hybridization
Insert into a new population
Insert two child strings into a new population
i = i + 1

Figure 1: Basic workflow chart.

to achieve sufficient speed [10]. In this paper, the biological evolutionary algorithm is combined with other artificial intelligence algorithms, and the adaptive and self-learning ability of improved support vector machine optimized by genetic computing is used to predict the early failure of gears [11]. The algorithm proposed in this paper is based on the concept of variable step-size control. We prove that this weakness of variable step size control can be avoided by making relatively small changes to the algorithm, and the derandomization scheme of variable step size control is helpful to reliably adapt to a single step size [12]. We attempted to analyse the interaction networks leading to individual differences in perceived odor quality by assessing the effects of self-adaptation and cross-adaptation on odor intensity and quality reporting in 18 human subjects [13]. In this paper, we will study a generalized evolutionary minority game model, in which agents are divided into several groups, and the performance of each agent is averaged in each group. We find that there are three different effects in the generalized model: (1) group average effect, (2) left-right asymmetry effect, and (3) self-interaction effect [14]. The statistical results show that the positive coping style of night shift nurses significantly positively predicts the positive self-suggestion of sleep and negatively predicts the importance of sleep. Negative coping style can negatively predict the positive self-suggestion of sleep, while coping style can well predict the cognitive aspect of individual sleep adaptation [15].

## 2. Genetic Algorithm

*2.1. Standard Genetic Algorithm.* The standard genetic algorithm has two important characteristics for simulating the whole learning process of the population, which are the population search strategy and the information exchange among individuals in the population. First, create a specific function to test the adaptability of individuals. The algorithm steps are as follows:

(a) Randomly select $N$ starting points to form a group. The total number $N$ is the size of the whole population. In order to express it conveniently, it is necessary to use a specific coding form to represent each individual in the population. An encoded content may represent a specific feature of a chromosome, and the content it represents varies according to the type of problem to be solved

(b) Choose suitable individuals according to certain selection requirements. We need to select the $M$ individuals with the best adaptability within the standard to form a new population to breed the next generation

(c) Among the selected $M$ individuals, pairwise grouping is carried out, and then each group is hybridized and recombined. Hybridization is an important way for two chromosomes to exchange information

(d) If the situation requires, we can select some suitable individuals from $M$ individuals by the given mutation probability, and then carry out mutation operation on the suitable individuals according to certain requirements. Mutation operation plays a very important role in the ability of genetic algorithm to find the optimal solution

(e) It is judged whether the bundle condition of the junction is met, if not, it returns to process $B$, continues the process, and if the condition is met, it ends the process. Its basic workflow is shown in Figure 1

*2.2. Adaptive Genetic Algorithm.* Its most important idea is to adjust the size of $Pc, Pm$ at any time according to the actual situation of the population in the evolution process. That is, when a population converges, reduce $Pc$ and increase $Pm$, that is, increase the probability of mutation and reduce the probability of hybridization, thus ensuring the diversity of the population and avoiding "premature maturity;" when the population deviates, $Pc$ is increased and $Pm$ is decreased, that is, the probability of hybridization is increased and the probability of mutation is reduced, so that the population tends to converge and the concentration rate of the algorithm is increased.

The index that defines the degree of "premature maturity" of population is $\Delta$. $F_{t\ \max}$ Is the fitness of the optimal individual, and the difference $\Delta$ between $F_{t\ \max}$ and $F_{t\ \max}$ is defined as follows

$$\Delta = F_{t\ \max} - F_{t\ \max}. \tag{1}$$

The "premature maturity" degree of the population can be obtained by the size of formula (1). According to the index $\Delta$, a new strategy is proposed, so that $\Delta$ can influence the values of hybridization probability $Pc$ and mutation probability $Pm$. Its mathematical description is shown in the following formulas

$$P_c = \frac{1}{1 + \exp(-k_1 \cdot \Delta)}, \tag{2}$$

$$P_m = 1.0 - \frac{1}{1 + \exp(-k_2 \cdot \Delta)}, \tag{3}$$

where $k_1, k_2 > 0$. Since $\Delta$ is always greater than or equal to 0, the value range of $P_c$ is between [0.5, 1] and the value range of $P_m$ is between [0, 0.5]. It can be seen from the above formula that $\Delta$ can affect the value of $Pc, Pm$ in the evolution process: when $\Delta$ increases, $Pc$ increases, $Pm$ decreases, and the ability to develop excellent individuals increases.

## 3. Quantum Machine Algorithm

*3.1. Quantum Nearest Neighbor Algorithm.* QNN algorithm firstly loads the training sample set $v_j (j = 1, 2, \cdots, M)$ and the sample to be classified $v_0 := u$ onto the quantum state by probability amplitude coding, as shown in the following formulas.

$$|v\rangle = d^{-1/2} \sum_{i:v_i \neq 0} |i\rangle \left( \sqrt{1 - \frac{r_{ji}^2}{r_{\max^2}}} e^{-i\phi_i} |0\rangle + \frac{v_{ji}}{r_{\max}} |1\rangle \right) |1\rangle, \tag{4}$$

$$|u\rangle = d^{-1/2} \sum_{i:v_{0i} \neq 0} |i\rangle |1\rangle \left( \sqrt{1 - \frac{r_{0i}^2}{r_{\max^2}}} e^{-i}\phi_i |0\rangle + \frac{v_{0i}}{r_{\max}} |1\rangle \right), \tag{5}$$

where $v_{ji} = r_{ji} e^{i\varphi ji}$; $r_{\max}$ is the upper bound of eigenvalue; $r_{ji}$ is a number greater than 0. You can get the square of the inner product of $|v\rangle$ and $|u\rangle$ as shown in the following formula

$$|\langle u \mid v \rangle|^2 = (2P(0) - 1) d^2 r_{\max}^4. \tag{6}$$

Although Euclidean distance is widely used to calculate the Euclidean distance of quantum states, the experimental results show that the method using Euclidean distance is more complicated than the inner product method.

*3.2. Quantum Support Vector Machine Algorithm.* The combination of quantum computing and support vector machine is to use the second acceleration effect of Grover search algorithm to deal with the optimization calculation of SVM model at the earliest. QSVM algorithm first encodes $j$ eigenvectors of each data sample point $x_i = (x_{i1}, x_{i2}, \cdots, x_{ij})$, $j = 1, 2, \cdots m$ into quantum states by probability amplitude coding, as shown in the following formula.

$$|x_i\rangle = |x_i|^{-1} \cdot \sum_{j=1}^{m} x_{ij} |j\rangle, \tag{7}$$

where $|x_i|^{-1}$ is the normalized vector and $m$ is the feature dimension. The quantum state preparation of the training set is shown in the following formula.

$$|\chi\rangle = \left(\sqrt{N_\chi}\right)^{-1} \cdot \sum_{i=1}^{n} |x_i| |i\rangle |x_i\rangle, \tag{8}$$

where $\sqrt{N_\chi} = \sum_{i=1}^{n} |x_i|^2$ and $x_i$ are the $i$ training samples. The inner product operation $K_{ij} = x_i \cdot x_j$ of the training data set can be obtained by solving the offset trace of the density matrix $|\chi\rangle\langle\chi|$, as shown in the following formula.

$$tr_2(|\chi\rangle\langle\chi|) = \frac{1}{N_\chi} \sum_{i,j=1}^{M} |x_i| |x_j| \langle x_i \mid x_j \rangle |i\rangle |j\rangle = \frac{K}{trK}, \tag{9}$$

where $x_i \cdot x_j = |x_i| |x_j| \langle x_i \mid x_j \rangle$. The above formula can be transformed into solving a linear equation, as shown in the following formula.

$$\begin{pmatrix} 0 & 1^T \\ 1 & K + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ y \end{pmatrix}, \tag{10}$$

where $K_{ij} = x_i \cdot x_j$ is the kernel matrix, $(b, \alpha)^T$ can be solved according to HHL algorithm to obtain the following formula.

$$|b, \alpha\rangle = \frac{1}{\sqrt{N_{b,\alpha}}} \left( b|0\rangle + \sum_{k=1}^{M} \alpha_i |i\rangle \right), \tag{11}$$

where $N_{b,\alpha} = b^2 + \sum_{k=1}^{M} \alpha_k^2$. Finally, the quantum state $|\tilde{u}\rangle$ and the quantity $|\tilde{x}\rangle$ to be classified are constructed, and the class of $|\tilde{x}\rangle$ can be obtained by calculating the similarity between the two quantum states by using the control exchange gate, as shown in the following formulas

$$|\tilde{u}\rangle = \frac{1}{\sqrt{N_u}} \left( b|0\rangle|0\rangle + \sum_{k=1}^{M} \alpha_k |x_k| |k\rangle |x_k\rangle \right), \tag{12}$$

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N_{\tilde{x}}}} \left( (|0\rangle|0\rangle) + \sum_{k=1}^{M} |x| |k\rangle |x\rangle \right), \tag{13}$$

where $N_{\tilde{u}} = b^2 + \sum_{k=1}^{M} a_k^2 |x_k|^2$, $N_{\tilde{x}} = M|x|^2 + 1$.

Table 1: Gym excuses return data list.

| Return value | Description |
|---|---|
| Observation | Returns the pixel picture of the video game |
| Reward | The scale of the reward for the previous action varies according to the environment |
| Done | Used to determine whether a reset environment is required for the surface. When the value is true, the surface game ends |
| Info | Contains diagnostic information for debugging |

Table 2: Video game description.

| Games | Describe |
|---|---|
| BeamRider | Initialize 3 lives, need to avoid bombs, and destroy enemy planes. There are 15 enemy planes in each level, and the reward for destroying enemy planes is $40 + 4 * i$ (level $I$). After level 4, there are special rewards with high scores. |
| Breakout | Initialization has 5 lives. Move the flat plate to make the ball rebound and break the color block to score. The ball falls into the bottom line and loses a life. The more the color block is hit, the faster the rebound speed is. |
| MsPacman | Initialize three lives, avoid the enemy, hit the enemy, and lose one. Swallow the pink square on the blue road to get a reward. The enemy will become weak periodically. At this time, you can swallow the enemy to get a higher reward. |
| Qbert | Initialize 3 lives, jump on the pyramid to light up each square to get a reward, all light up to enter the next level to give a large reward, jump outside the pyramid, or hit the enemy will lose a life. |
| Seaquest | Initialize 3 lives, destroy sharks and enemy submarines to get rewards, avoid sharks and enemy submarines and launch torpedoes, hit and lose one life, submarine runs out of oxygen and loses one life, and returns to the sea without saving people to lose one life |
| SpaceInvaders | Initialize 3 lives, destroy the enemy to get rewards, and lose one life when hit by the enemy. The enemy will invade at a faster and faster speed. If the enemy reaches the bottom line, no matter how many lives are left, the game will end. |

*3.3. Quantum Linear Regression.* The purpose of quantum linear regression algorithm is to use quantum algorithm to solve the optimal fitting parameter $w$, as shown in the following formula

$$w = X^{+}y = (X^{+}X)^{-1}X^{+}y, \tag{14}$$

where $y = (y_1, y_2, \cdots, y_n)^T$; $X = (x_1, x_2, \cdots, x_n)^T$ is the data matrix. In order to calculate $w$ more conveniently, formula (14) is decomposed into two formulas, and the two formulas are solved, respectively, namely, $y' = X^{+y}$ and $w = (X^{+}X)^{-1}y'$. For $y' = W^{+}y$, the matrix $y$ is first loaded on the quantum state by probability amplitude coding, as shown in the following formula

$$|y\rangle = \sum_{P=M+1}^{M+N} \frac{y_p}{|y|}|p\rangle. \tag{15}$$

Since matrix $X$ is not a Hermite matrix, we will transform it into a Hermite matrix, as shown in the following formula

$$I(X) = \begin{pmatrix} 0 & X \\ X^{+} & 0 \end{pmatrix}. \tag{16}$$

At this time, $y' = X^{+}y$ can be transformed into $|y'\rangle = I(X)|y\rangle$ to solve. For $w = (X^{+}X)^{-1}y'$, the matrix $(X^{+}X)^{-1}$ needs to be transformed into the Hermite matrix $I((X^{+}X)^{-1})$, as shown in the following equation

$$I\left((X^{+}X)^{-1}\right) = \begin{pmatrix} X^{+}X & 0 \\ 0 & XX^{+} \end{pmatrix}^{-1} = \begin{pmatrix} (X^{+}X)^{-1} & 0 \\ 0 & (XX^{+})^{-1} \end{pmatrix}. \tag{17}$$

At this time, for $w = (X^{+}X)^{-1}y'$, it can be transformed into solving $|w\rangle = I((X^{+}X)^{-1})|y'\rangle$. $|y'\rangle = I(X)|y\rangle$ and $|w\rangle = I((X^{+}X)^{-1})|y'\rangle$ solve the transformed formula by quantum algorithm and then get the quantum state $|w\rangle$ of fitting parameters.

*3.4. Quantum K-Means Algorithm.* In 2013, the quantum version of the nearest center algorithm was proposed, which can be regarded as a subprocess of the quantum $K$-means algorithm and can accelerate the classical algorithm exponentially, as shown in the following equations.

Table 3: A2C game scores under 10 M, 20 M, 30 M, and 40 M calculation budget.

| Game (A2C) | 10 M | 20 M | 30 M | 40 M |
|---|---|---|---|---|
| BeamRider | 1894 | 4972 | 6039 | 7162 |
| Breakout | 189 | 416 | 471 | 517 |
| MsPacman | 1564 | 2156 | 2152 | 2283 |
| Qbert | 3468 | 9763 | 14514 | 15576 |
| Seaquest | 1249 | 1754 | 1752 | 1740 |
| SpaceInvaders | 669 | 742 | 899 | 1746 |

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|u\rangle + \frac{1}{M}\sum_{j=1}^{M}|j\rangle|v_j\rangle\right), \tag{18}$$

$$|\varphi\rangle = \frac{1}{\sqrt{Z}}\left(|u\rangle|0\rangle - \frac{1}{\sqrt{M}}\sum_{j}|v_j\rangle|j\rangle\right), \tag{19}$$

where $Z = |u|^2 + 1/M\sum_j |v_j|^2$. Quantum $K$-means algorithm includes minimum quantum algorithm and phase estimation algorithm. Its application can greatly reduce the overall computational complexity.

*3.5. Quantum Principal Component Analysis.* On the feature space, $\rho$ can be expressed as the following formula.

$$\rho = \sum_j \lambda_j |u_j\rangle\langle u_j|, \tag{20}$$

where $\lambda_j$ is the eigenvalue; $u_j$ is its corresponding eigenvector. The most critical part of QPCA is the construction of controlled $e^{-i\rho t}$. The method of constructing $e^{-i\rho t}$ can be expressed by the following equation:

$$\begin{aligned} tr_p e^{-iS\Delta t}\rho\otimes\delta e^{iS\Delta t} &= \left(\cos^2\Delta t\right)\delta + \left(\sin^2\Delta t\right)\rho - i\sin\Delta t[\rho,\delta] \\ &= \delta - i\Delta t[\rho,\delta] + O\left(\Delta t^2\right), \end{aligned} \tag{21}$$

where $tr_p$ is the bias trace for the first variable; matrix $S$ is a commutative operator, $\delta = |u\rangle\langle u|$ is a feature vector. Thus, $e^{-i\rho t}$ can be realized by partial tracing operation with the aid of $N$ replica density operator $\rho$ and a sparse commutation matrix $S$.

Controlled $e^{-i\rho t}$ can be realized by applying unitary operator $\sum_n |n\Delta t\rangle\langle n\Delta t| \otimes \prod_{j=1}^{n} e^{-iS_j\Delta t}$ to $|n\Delta t\rangle\langle n\Delta t| \otimes \delta\otimes\rho\otimes\cdots\otimes\rho$, and then performing partial trace operation on each subsystem, as shown in the following equation:

$$tr_p\left(\begin{array}{c}\left(\sum_n |n\Delta t\rangle\langle n\Delta t| \otimes \prod_{j=1}^{n} e^{-iS_j\Delta t}\right) \\ |n\Delta t\rangle\langle n\Delta t| \otimes\delta\otimes\rho\otimes\cdots\otimes\rho\end{array}\right). \tag{22}$$

## 4. Experimental Research on Deep Reinforcement Learning

Deep reinforcement learning has the characteristics of various methods, extreme sensitivity to super parameters and great randomness. Even the same algorithm will bring about performance differences due to different implementation methods or different test environment platforms. The best way to integrate these problems is to hope that the domain can unify the model and test environment, so that the domain can replicate and improve and try new ideas. Therefore, many research institutions have begun to develop their own algorithm models and test environments.

Table 4: PPO game scores under 10 M, 20 M, 30 M, and 40 M calculation budget.

| Game (PPO) | 10 M | 20 M | 30 M | 40 M |
|---|---|---|---|---|
| BeamRider | 2711 | 4302 | 5408 | 6495 |
| Breakout | 227 | 290 | 376 | 413 |
| MsPacman | 2022 | 2357 | 3051 | 3762 |
| Qbert | 10737 | 15061 | 18211 | 21695 |
| Seaquest | 1072 | 1642 | 1850 | 1881 |
| SpaceInvaders | 782 | 1536 | 2021 | 2137 |

Table 5: DDPG game scores under 10 M, 20 M, 30 M, and 40 M calculation budget.

| Game (DDPG) | 10 M | 20 M | 30 M | 40 M |
|---|---|---|---|---|
| BeamRider | 2304 | 4264 | 5201 | 6207 |
| Breakout | 179 | 259 | 329 | 436 |
| MsPacman | 1978 | 2205 | 2988 | 3652 |
| Qbert | 11454 | 14987 | 17892 | 20194 |
| Seaquest | 1104 | 1569 | 1760 | 1798 |
| SpaceInvaders | 773 | 1435 | 2018 | 2065 |

Table 6: Sampling data under different thread numbers.

| Thread quantity | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| A2C | 139 | 222 | 404 | 679 | 976 |
| PPO | 153 | 198 | 353 | 580 | 840 |
| DDPG | 147 | 216 | 388 | 634 | 893 |

According to the algorithm model, DeepMind, OpenAI, and Baidu have all opened up some self-implemented deep reinforcement learning algorithms and applications, among which OpenAI is the most abundant, including the high-quality implementation of nine algorithms: TRPO, PPO, HER, GAIL, DQN, DDPG, AVKTR, ACER, and A2C, which is also the preferred benchmark code for researchers in the field at present. For the test environment, there are mainly OpenAIGym, MuJoCo, rllab, DeepMindLab, TORCS, PySC2, and so on. Among them, MuJoCo is mainly a robot control problem, Gym, rllab, and DeepMindLab are similar, mainly focusing on video games, TORCS is a racing car control problem, and PySC2 is the environment of StarCraft II. At present, Atari video game environment is widely used in the field of machine game, and Gym interface is usually used.

Deep learning of machine games is divided into many parts, the most important of which is to interact with some simulators. If the algorithm supports multiprocess mode, the training period for obtaining state information will be shortened. Therefore, A2C, DDPG, and PPO, which support multiprocess mode, are currently the most used algorithms in the field of machine game. In this section, the A2C, DDPG, and PPO algorithm for some related experimental research and analysis, to provide reference for scholars in the field.

*4.1. Test Environment.* Related experiments will be conducted in Atari 2600 video game environment. There are

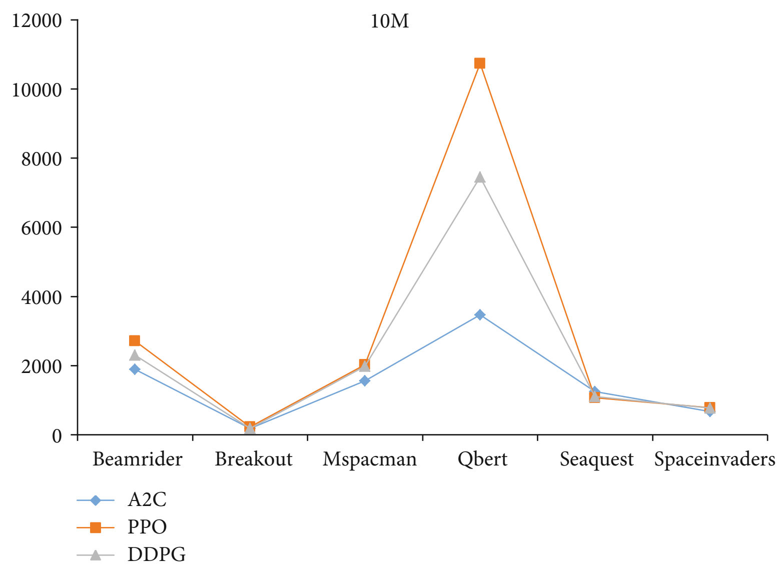

Figure 2: Performance comparison in 10 M environment.

many test environments for deep reinforcement learning, but Atar video game environment is generally used most. Gym environment is usually needed for interaction in reinforcement learning. The specific step is to enter the action produced by the agent into the function step, and then you will get four return values: done, reward, observation, and info, as shown in Table 1.

It is used by agent in a very simple way. In order to ensure the reproducibility of the experimental results, all the video game experiments involved in this paper are completed on Gym platform. In this paper, six popular classic video games: BeamRider, Breakout, MsPacman, Qbert, Seaques, and SpaceInvaders are selected as specific test environments, and the task description of each game is shown in Table 2.

*4.2. Experimental Setup and Configuration.* In order to ensure the reproducibility of the results, this paper uses A2C and PPO provided by baselines of OpenAI as benchmark experimental codes, Gym version is 0.10.5, baselines version is 0.1.5. All the experiments were completed on a PC with the following configuration: CPU: Inter (R) Core (TM) i9-7900X @ 3.3 GHz, GPU: NVIDIAGTX1080Ti, and memory: 32 G. All the experiments were repeated five times under its related settings.

*4.3. Performance Analysis of Reinforcement Learning for Machine Game.* As OpenAI updated the V4 version of Gym and announced that the ∗NoFrameSkip-v4 environment is considered the standard Atari environment, it was reminded that the V4 version will not provide the same experimental results as the previous V3 version. The above problems lead to the fact that when researchers need to do comparative experiments, they often need to reexperiment in the new version environment, which is very time-consuming. At the same time, our research found that there is no brand-new baseline result for industry reference. In addition, different researchers often need the performance results of benchmark algorithms under different computational budgets, but there is no complete and detailed experimental results under different computational budgets at present. In this paper, six mainstream classic video games are used as the test environment, and the latest Gym version is used to do standard experimental tests on A2C, PPO, and DDPG under 10 M, 20 M, 30 M, and 40 M calculation budgets, respectively, in which the default number of processes is 16. The final performance data of its model are shown in Tables 3–5.

Sampling data tables of A2C, PPO, and GGPG algorithms with different thread numbers are shown in Table 6.

In addition, we also give the performance comparison diagrams of A2C, PPO, and DDPG under different environments and different computing budgets, as shown in Figures 2–5.

By observing graph 2, graph 3, graph 4, and graph 5, we can conclude that PPO is the largest in any environment, DDPG is the middle, and A2C is the smallest, which is particularly easy to observe in Qbert.

We use the sampled data of A2C and PPO algorithms with different thread numbers to draw an efficiency diagram as shown in Figure 6.

By observing the data table and the schematic diagram of the performance curve obtained from the experiment, some new discoveries are also obtained in this paper. For example, in some test tasks, the performance of A2C will be lower than PPO and DDPG in the early stage, but with the increase of computing budget, the performance will exceed PPO and DDPG. However, in PPO's paper, only the experiment was carried out at the calculation cost of 10 M, and it was concluded that its effect was better than A2C, which is not rigorous and correct at present. Usually, due to the long training time of video games and the limitation of computing resources, researchers often only conduct experiments under a small computing budget, so it is difficult to find this

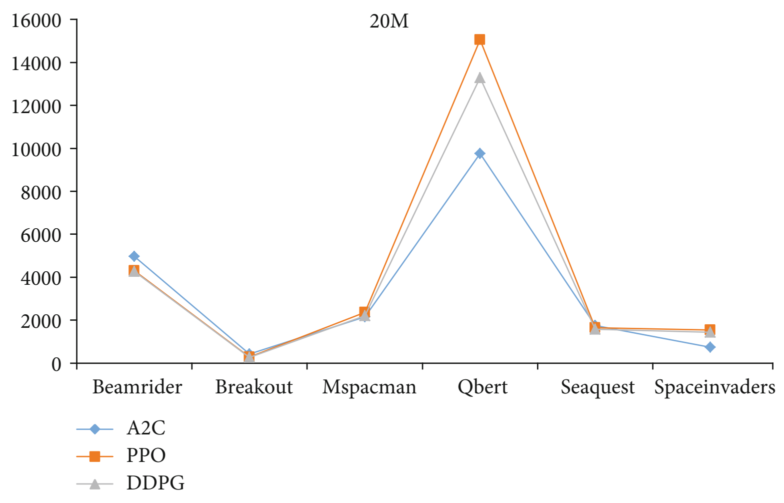

Figure 3: Performance comparison chart in 20 M environment.

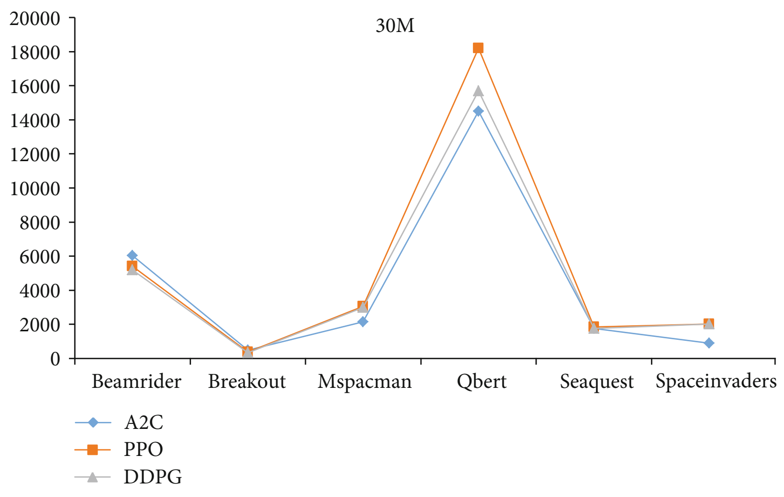

Figure 4: Performance comparison in 30 M environment.

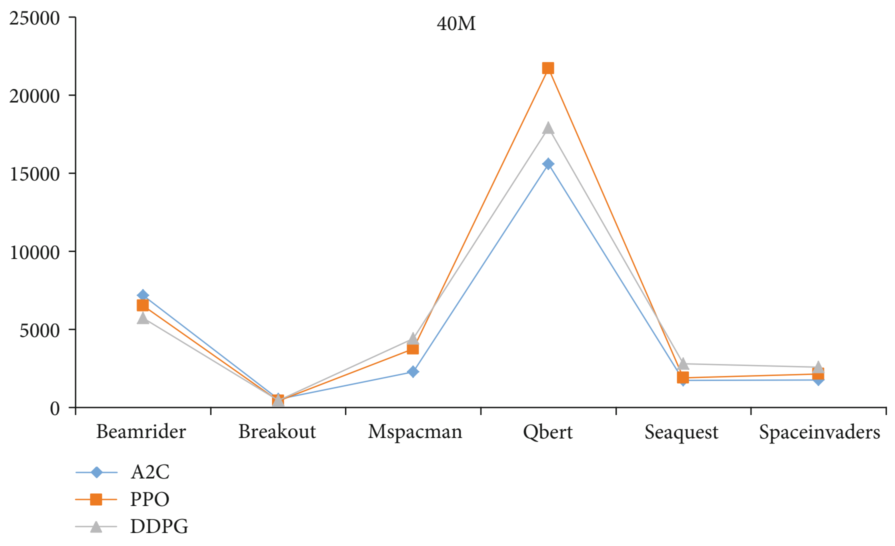

Figure 5: Performance comparison in 40 M environment.

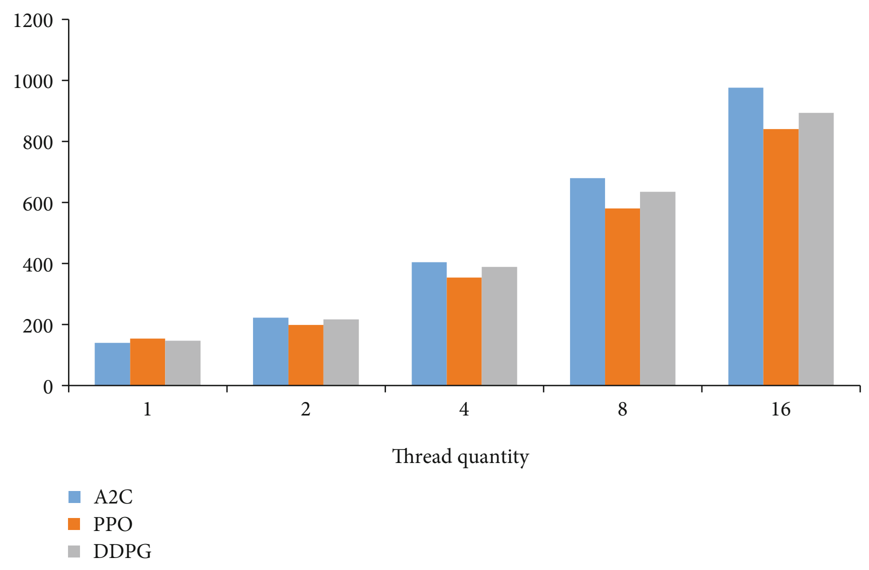

Figure 6: Schematic diagram of sampling efficiency of A2C, PPO, and DDPG algorithms under different thread numbers.

phenomenon, which also provides a new focus for the comparison of algorithms in the future. In addition, it can be found that the variance of A2C is significantly smaller than that of PPO algorithm in almost all test environments, especially in Seaquest and Spaceliwaders games. Therefore, it can be considered that A2C has better robustness.

In this chapter, we first briefly introduce the background of reinforcement learning and deep learning, which leads to the deep reinforcement learning method. On this basis, the deep reinforcement learning methods for machine game are systematically investigated, including reinforcement learning methods based on value function and reinforcement learning methods based on strategy gradient. In addition, hierarchical reinforcement learning and multiagent reinforcement learning are introduced. In the experimental research part, we briefly introduce the current mainstream code base and standard test platform for deep reinforcement learning. Finally, the mainstream reinforcement learning algorithms are studied experimentally, the performance of A2C, DDPG, and PPO under different computational costs and the specific training curves is given, and the experimental results are analyzed.

## 5. Conclusion

In this paper, the current situation of biological population evolutionary algorithm, quantum machine learning algorithm, and deep reinforcement learning is analyzed and studied. This paper analyzes that in the upcoming era of "quantum computing," we are facing many unprecedented opportunities and challenges: how to cross-integrate "biological population evolutionary algorithm + individual adaptation + quantum computing" to quantitatively and accurately describe the uncertainties of big data; in the coming era of "quantum computing," how to make use of the advantages of parallel and efficient processing of quantum computing to efficiently deal with the complexity problems of biological population evolutionary algorithm and individual adaptive method. These opportunities and challenges will become important problems to be solved urgently in the era of "quantum computing" and will also become research hotspots in the field of big data in the era of intelligence.

## Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

## References

[1] M. A. Nielsen and I. L. Chuang, "Quantum computation and quantum information," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 21, no. 1, pp. 1–59, 2010.

[2] D. P. Divincenzo, "Quantum computation," *Science*, vol. 270, no. 5234, pp. 255–261, 1995.

[3] M. A. Nielsen and I. L. Chuang, "Quantum computation and quantum information," *Mathematical Structures in Computer Science*, vol. 17, no. 6, pp. 1115–1115, 2002.

[4] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Los Alamitos, CA, 1994.

[5] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, "Non-abelian anyons and topological quantum computation," *Reviews of Modern Physics*, vol. 80, no. 3, pp. 1083–1159, 2008.

[6] D. A. Lidar, I. L. Chuang, and K. B. Whaley, "Decoherence free subspaces for quantum computation," *Physical Review Letters*, vol. 81, no. 12, pp. 2594–2597, 1998.

[7] M. L. Raymer, T. E. Doom, L. A. Kuhn, and W. F. Punch, "Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm," *IEEE Transactions on Systems, Man, and Cybernetics , Part B (Cybernetics)*, vol. 33, no. 5, pp. 802–813, 2003.

[8] C. H. Chuang, C. L. Lin, Y. C. Chang, T. Jennawasin, and P. K. Chen, "Design of synthetic biological logic circuits based on evolutionary algorithm," *IET Systems Biology*, vol. 7, no. 4, pp. 89–105, 2013.

[9] J. Mahil and T. Raja, "An intelligent biological inspired evolutionary algorithm for the suppression of incubator interference in premature infants ECG," *Soft Computing*, vol. 18, no. 3, pp. 571–578, 2014.

[10] S. Dura-Bernal, S. A. Neymotin, C. C. Kerr et al., "Evolutionary algorithm optimization of biological learning parameters in a biomimetic neuroprosthesis," *IBM Journal of Research & Development*, vol. 61, no. 2/3, pp. 6:1–6:14, 2017.

[11] X. H. Chen, L. Xiao, Z. X. Liu, and C. Liu, "Prophase failure prediction of the mechanical transmission systems based on the biological evolutionary algorithm," *Key Engineering Materials*, vol. 572, pp. 447–450, 2013.

[12] A. Ostermeier, A. Gawelczyk, and N. Hansen, "A derandomized approach to self-adaptation of evolution strategies," *Evolutionary Computation*, vol. 2, no. 4, pp. 369–380, 1994.

[13] R. J. O'Connell, D. A. Stevens, and L. M. Zogby, "Individual differences in the perceived intensity and quality of specific odors following self- and cross-adaptation," *Chemical Senses*, vol. 19, no. 3, pp. 197–208, 1994.

[14] Y. Xie, B.-H. Wang, W. Yang, and W. Wang, "The mechanism for the self-adaptation behavior in the evolutionary minority game model," *Chinese Science Bulletin*, vol. 49, no. 5, pp. 432–437, 2004.

[15] T. Bormann, S. Wolfer, and L. Konieczny, "Individual differences between two letter-by-letter readers: impaired letter processing versus impaired verbal working memory," *Procedia-Social and Behavioral Sciences*, vol. 94, no. 1, pp. 173-174, 2013.