Journal of Artificial Intelligence Practice (2022)
Clausius Scientific Press, Canada
DOI: 10.23977/jaip.2022.050405
ISSN 2371-8412 Vol. 5 Num. 4

High-Performance Computing Option Pricing Algorithm on Hybrid Heterogeneous Many-Core Platforms

Ling Yang[a*], Yijia Zhang[b], Lei Wang[c], Yu Guan[d]

School of Informatics, Harbin Guangsha College, Harbin, Heilongjiang, 150025, China
[a]yangling869@hit.edu.cn, [b]18846793339@139.com, [c]409600586@qq.com, [d]841707819@qq.com
[*]Corresponding author

Keywords: Internet of Things, Smart Environment, Hybrid Heterogeneous Many-Core Platform, High-Performance Computing, Option Pricing Algorithm

Abstract: In recent years, financial derivative securities have developed rapidly, and the pricing of contingent rights has also attracted widespread attention from domestic and foreign scholars. Option pricing theory has become an important theory that won the Nobel Prize in Economics after asset portfolio theory and capital asset pricing model. This paper aims to study the pricing algorithm of high-performance computing options on hybrid heterogeneous many-core platforms. In this paper, particle swarm optimization algorithm (PSO), quantum behavioral particle swarm algorithm (QPSO), differential evolution algorithm (DE), and evolution strategy (ES) are used to solve the parameter estimation of nonlinear option pricing models, using the nonlinear approximation of the algorithm ability to establish an algorithm model for solving parameter estimates, and use the weighted sum of the squared errors of the experimental value and the predicted value of the algorithm as the objective optimization function. Find the most suitable optimization algorithm to solve this problem through experiments. In addition, the Internet of Things technology is also used to design the Internet of Things data collection system, and the RFID technology, sensor technology, wireless network technology, artificial intelligence technology, and cloud computing technology in the Internet of Things are analyzed. The results used in this paper show that the parameters of option pricing are estimated by four algorithms. Among them, the QPSO algorithm has the best convergence performance and the algorithm fitting results are 1.7 times more accurate than the worst algorithm, followed by the DE algorithm, PSO, and ES The effect of algorithm parameter fitting is poor.

1. Introduction

With the development of the big data era, users have higher and higher requirements for processor performance, and the ability of single-core processors to process information can no longer meet the needs of users. The traditional way to increase the main frequency of a single-core processor is to integrate more transistors into the processor. However, according to Moore's theorem, the number of transistors integrated into the semiconductor will double every 18 months. Therefore, the Increasingly transistors. At the same time, the problem of heat dissipation and power consumption of transistors has become increasingly prominent, and the manufacturing process is complicated. It is increasingly difficult to increase the processor's main frequency simply by integrating transistors. The development of single-core processors has encountered a bottleneck. Various obstacles faced by single-core processors have triggered changes in architecture technology, and multi-core processors have become a breakthrough. Multi-core processors, also known as on-chip multiprocessors, have replaced single-core processors because of their high frequency, low power consumption, and simple structure, and have become the mainstream of processor development in one fell swoop. After years of development, multi-core processors have been widely used, their market is getting bigger and bigger, and the competition among manufacturers is getting fiercer. With the development of computer architecture, the internal structure of multicore processors is becoming increasingly complex. In order to fully improve the execution efficiency of multicore processors, it is necessary to study and design efficient task scheduling algorithms. The quality of task scheduling algorithms directly affects the performance of multicore processors. Inefficient task scheduling algorithms not only fail to take advantage of the parallel execution of multicore processors, but also reduce the utilization of each core in the multicore processor.

Cuenca J studied the matrix matrix multiplication kernel optimized for different computing system components through guided experiments. He also analyzed the adaptability of basic linear algebra routines to hybrid multicore+multiGPU and multicore+multiMIC auto-tuning technologies. These computer components share different types of multiprocessors and coprocessors, usually several graphics processing units (GPU) or many integrated cores (MICS). However, due to the complex internal structure of the components, the results obtained are not very accurate [1]. The Internet of Things (IoT) is a dynamic global information network composed of objects connected to the Internet, which have become an integral part of the Internet in the future. Perera C surveyed more than one hundred IoT smart solutions on the market and carefully checked them to identify the technologies, functions, and applications used. Based on the application field, he classified these solutions into five categories: 1) smart wearable devices; 2) smart wearable devices;3) Smart city; 4) Smart environment; 5) Smart enterprise. This survey is intended to serve as a guide and conceptual framework for future research on the Internet of Things to inspire and inspire further development. However, there are many unstable factors in the survey, which will lead to certain errors in the results [2]. Samimi O introduced the pricing of American options under the Heston–Hull–White stochastic volatility and stochastic interest rate models. To this end, he first used Euler discretization scheme to discretize random processes. Then, the least squares Monte Carlo algorithm is used to price American options. He also compared the numerical results of the model with the Heston-CIR model. Finally, the numerical results show the efficiency of the proposed algorithm for pricing American options under the Heston-Hull-White model. However, due to the errors in the proposed algorithm, the results obtained are inconsistent with the expected [3].

The innovation of this paper is (1) introduces the current development status and challenges of swarm intelligence algorithms. It needs a better computer platform to promote the further development of EDA technology. Researchers have high hopes for EDA technology. (2) Solve the inverse problem, use an intelligent optimization algorithm to estimate the unknown parameters in the B-S option pricing model, and draw a large number of experimental conclusions.

2. High-Performance Computing Option Pricing Method under the Internet of Things Environment

2.1 Internet of Things Technology

(1) RFID technology

RFID is abbreviated as radio frequency identification technology, which mainly uses the unique identification number corresponding to the tag to identify the indicator light. RFID is a simple radio system with only two basic elements. The system is used to control, detect, and monitor objects. The POS machines we use with swipe bus cards and go to the supermarket to shop are all applications of radio frequency technology [4].

(2) Sensor technology

For the collection of basic data, according to specific rules, it is converted into electrical signals or other forms of output information to meet the needs of data transmission, processing, storage, projection, recording, and control requirements [5].

(3) Wireless network technology

Artificial intelligence is the study of how to use computers to simulate certain processes of human thinking and intelligent behavior (such as learning, thinking, thinking, designing, etc.). In the Internet of Things, artificial intelligence technology is mainly responsible for analyzing the content of "voice" objects and starting automatic processing by computers.

(4) Artificial intelligence technology

Used for network communication. In the Internet of Things, communication between objects and people is bound to be inseparable from high-speed wireless networks, which can transmit large amounts of data.

(5) Cloud computing technology

The cloud in cloud computing technology actually represents the Internet. Computer capabilities through the network, instead of using the software originally installed on the computer or replacing the original data storage energy on your own hard disk, using the network to perform various operations, these operations will lead to the realization structure exceeding the maximum operating range, resulting in a large excess of one, and even large fluctuations, which is not allowed in production [6].

2.2 Option Pricing Algorithm

(1) Monte Carlo simulation method

Monte Carlo simulation is a method of evaluating European derivative assets. The basic idea is to assume that the underlying asset price distribution function is known, and then divide the actual selection period into several small-time intervals. Using a computer, we can randomly sample the distribution samples and simulate each change in stock price to calculate the final value of the option. Time and possible stock price paths [7]. This result can be seen as a random sample of all possible final values, and another random sample can be obtained through another path of this variable. More sample paths can generate more random samples. Repeat this operation a thousand times to obtain the total selection value in T, and run a simple arithmetic average on thousands of random samples, and you can get the expected performance of options in T [8]. According to the principle of non-arbitrary pricing, the expected X_T return of the option to be repaid early at the risk-free interest rate in the future can get the price of the option at the current time [9].

$$P = e^{-rT} D(X_T) \tag{1}$$

Among them, P represents the option price, and r represents the risk-free interest rate (X_T is the expected rate of return of the option at time T). The advantage of Monte Carlo simulation is that it can be used to compare the expected return and volatility of the underlying asset. In complex cases, the simulation calculation time increases linearly with the increase of the number of variables, and its calculation efficiency is higher. However, the limitation of this method is that it can only be used for the evaluation of European options, not for American contracts, because its selection is executed in advance, and the accuracy of the results depends on the number of simulation operations [10].

(1) Finite difference method

The finite difference method should include in particular two methods: internal and external. The basic idea is to use numerical methods to solve the differential equations satisfied by derivatives and to estimate the choices obtained in the Black-Scholes model of derivative assets. After the differential equation is transformed into a series of differential equations, it is solved by an iterative method. Generally speaking, the basic idea of the finite difference method is basically similar to that of the binary tree method. They can be used to resolve the prices selected in Europe and the prices selected in the United States [11].

The Black-Scholes option pricing method is an analytical method. Its main advantage is that this method can obtain detailed expressions of compensation parameters and leverage results, thereby providing clearer quantitative conclusions and analytical solutions for derivative asset trading strategies. It is more convenient to use when calculating directly with the number of small options. However, this method also has disadvantages, that is, it can only provide a detailed solution for European choices, and this method is also difficult in variable states and other more complicated situations. The martingale method can also give the exact pricing type of derivative assets, but sometimes it is difficult to find a single equivalent martingale measure [12].

2.3 Intelligent Algorithm High-Performance Computing

When an ant feeds to the feed, it will leave the pheromone on the way it walks. The ants behind will walk along the path with the pheromone and will also run to the food along the shortest path [13]. The more the pheromones, the stronger the ability to attract ants. Based on this phenomenon, scientists have designed an "ant colony algorithm" to find the best path. The basic idea of the algorithm is:

(1) A group of ants randomly start from the starting point, encounter food, grab the food, and return to the same route. Ants leave traces of pheromone on the way back [14].

(2) Pheromone will gradually evaporate over time (usually described by a negative exposure function). For ants starting from the nest, the probability of choosing a path is proportional to the concentration of pheromone on each path [15]. The same principle can also be used to describe the food status of ant colonies from multiple food sources.

To reach all designated cities in the fastest way and finally return to the starting point, the problem of city n and briefly introduces the ant algorithm based on the principle of ant colony foraging to solve the shortest path ant colony [16].

Suppose there is an ant in m, and there are $a_x(t)$ in the x city at time t, and the amount of pheromone left on the road between the two cities in x, y at time t is $b_{xy}(t)$. Assuming that each ant does not repeat the cities it has walked when completing a loop, the probability of the kth ant going from city x to y is:

$$P_{xy}^k = \frac{b_{xy}(t)}{\sum b_{xy}(t)} \tag{2}$$

The amount of pheromone $b_{xy}(t)$ can be defined as:

$$D_{xy} = \sum_{k=1}^{m} \frac{e}{L_k} a_{xy}^k(t) \tag{3}$$

Where L_k is the loop length obtained by the k ant [17].

$$b_{xy}(t+1) = c \cdot b_{xy}(t) + D_{xy} \tag{4}$$

$$a_{xy}^{k}(t) = \begin{cases} 1, t \geq k \\ 0, t < k \end{cases} \tag{5}$$

This way, each ant gets a loop after n migration, and its length is L_k. Use the new formula to calculate the pheromone concentration of each path, and proceed to the next search.

3. High-Performance Computing Option Pricing Experiment on Hybrid Heterogeneous Many-Core Platforms

3.1 IoT Data Collection Experiment

The Internet of Things collection system will inevitably involve the perception of device information and the distinction between different devices. At the same time, if the device is operated by personnel, it also involves the verification of personnel identity, so the data collection module of this design is divided into sensor data collection. The storage synchronization of identity data and the collection of video images[18].

The data collection of the sensor adopts the ModbusRTU protocol, and the ModbusRTU protocol uses the communication mode of query to response. This protocol is generally used for industrial data transmission. It has the advantages of simple protocol and high flexibility. If the sensor module is increased, only the number of bytes of the protocol needs to be changed, and the real-time data can be added to the protocol. Therefore, this sensor data acquisition architecture has high scalability and is highly compatible with the IoT data acquisition module discussed in this paper[19].

The synchronization of identity data refers to the collection of identity data through external devices or the acquisition of identity data through data transmission and synchronization between the local database and the remote management platform to unify the data on both sides, thereby achieving the purpose of storing identity data[20].

Video image capture uses the V4L2 interface dedicated to video capture under Linux. According to the relevant knowledge of the operating system, it is very time-consuming for the application to copy data between the kernel mode and the user mode, while the video capture is an operation on the external device and involves system calls. V4L2 uses memory mapping technology to map the memory space of the external camera to the user address space, so that programmers can use the image content collected by the camera like a memory, reducing data copy time and speeding up image transmission. In addition, V4L2 also provides related operations on video capture equipment, including the format and size of the captured video. The video captured by the camera is saved according to a certain color coding method. In this design, it is captured and saved in the YUV format[21].

3.2 Estimation of Option Pricing Parameters

The nonlinear estimation of model parameters has always been one of the most popular problems in the control field [22]. Traditional parameter estimation methods include square method and maximum probability method. However, the least square method and the method of estimating the maximum probability are both based on the assumption of a normal search area with continuous derivatives. It is a local technique that optimizes the search along an oblique direction and may fall on a local edge under certain conditions. With the expansion of the scope, the time and space complexity of the nonlinear model parameter estimation problem makes it very difficult to solve. Traditional parameter estimation methods are difficult to meet the needs of the problem [23]. In recent years, the emergence of intelligent optimization algorithms has provided new ideas and methods for nonlinear estimation of model parameters and has been widely used in the development of scientific research, economic, and technological engineering. Particle algorithm, quantum particle behavior algorithm, differential evolution algorithm, and development strategy are the characteristics of many intelligent cluster optimization algorithms. They have shown great potential in solving the problem of nonlinear model parameter estimation and have become research hotspots in recent years [24].

The option price in the Black-Scholes option pricing model depends on the following five parameters: the basic purchase price of the asset, the warning price, and the period, the risk-free interest rate, the volatility of the underlying price, and the standard deviation of asset returns [25]. Among these parameters, the first three are easily determined values, but the basic risk-free interest rate and price volatility must be estimated through certain calculations [26]. Therefore, the parameters of the option pricing model need to be inverted. This section mainly uses intelligent optimization algorithms to solve the parameters r and σ. The purpose of parameter estimation is to determine r and σ, so that the actual observation value and the algorithm estimate have a higher degree of fit. In this paper, particle swarm optimization (PSO), quantum behavioral particle swarm optimization (QPSO), differential evolution algorithm (DE), and evolution strategy (ES) are applied to the parameter estimation problem of option pricing [27].

3.3 Establishment of High-Performance Computing Platform

According to the logic equivalence principle of software and hardware, the operations that can be realized by any software method can be realized by hardware. Whether a function is realized by hardware or software depends on system performance, price, power consumption, volume, and operating frequency. And many other considerations. Since the general hardware platform has the flexibility to support most tasks, it is impossible to provide special optimization for a certain task. If you want to obtain higher computing performance, the usual method is to upgrade the hardware and increase the operating frequency of the system. But this method will be limited by factors such as volume, price, power consumption, etc. [28]. The application-centric dedicated hardware can provide specialized acceleration based on the characteristics of the algorithm, and has obvious advantages over general-purpose hardware in terms of volume, price, and power consumption when achieving the same or higher performance. Because of the flexible system reconfigurability, FPGA is very suitable for the method of software and hardware co-design and can provide the best support scheme according to the different needs of the algorithm. With the advancement of FPGA technology, its design integration and efficiency, signal integrity, power consumption, operating frequency, reliability, and other indicators are constantly optimized, and it has the conditions for use in actual systems. Taking automotive electronics as an example, various types of racing electronic control systems first began to adopt FPGA design technology, and the automotive engine control unit (ECU) is the most widely used [29]. It is the use of FPGA features to make the racing car improve the flexibility, controllability, and reliability of electronic components.

What this paper realizes is that 20 particles are used as a population, the particles are three-dimensional, and iterate 2000 times. With reference to the general pattern of particle-based algorithms implemented on hardware platforms, the entire design adopts a modular design structure, as shown in Figure 1. On the whole, the population is divided into three independent parallel hardware structures of one-dimensional QPSO algorithm, which also adopts modular design [30].

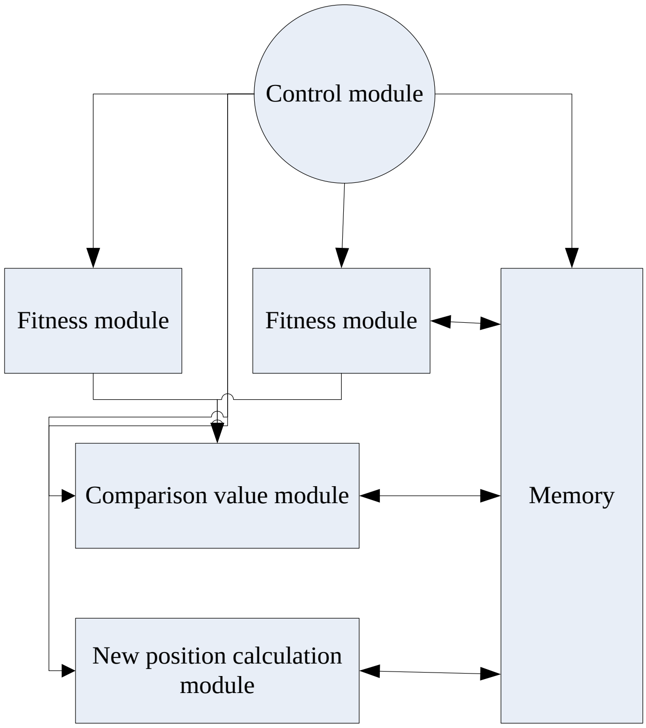

Figure 1: The structure of the high-performance computing platform

It can be seen from Figure 1 that the system consists of 5 parts: control module, memory, fitness value module, comparison module, and new position calculation module. The entire system is controlled by the control module sending control signals, the memory is responsible for the storage of the system during operation, and the final result, and other modules are responsible for various complex operations. Through the analysis of the software-implemented QPSO algorithm, it can be seen that operations such as comparison and selection, updating particles, and calculation of fitness function account for 80% to 90% of the entire algorithm time. Assuming that the population size is a and the number of iterations is b, then these operations will be performed at times. For a complex problem, both a and b will be large. It can be seen that improving the efficiency of algorithm operation is the key to improving the efficiency of QPSO algorithm. Clearly, due to its own limitations of the software QPSO algorithm (programs can only be executed sequentially), there is not much room for improvement in this area; on the contrary, when the QPSO algorithm is implemented by hardware, the parallelism of the algorithm can be fully utilized. For example, in the QPSO algorithm, at least two fitness calculation operations are required to provide the individuals required for the comparison selection operation. When implemented by hardware, these two selections can be performed at the same time, so that each selection time required is reduced by half; similarly, the parallelization of the new position calculation greatly improves the efficiency of the algorithm execution.

4. Analysis of Pricing Algorithm for High-Performance Computing Rights Period of Hybrid Heterogeneous Many-Core Platforms

4.1 Performance Analysis of Mixed Heterogeneous Many-Core

The first set of experiments mainly tested the performance of ILSGA, GA, and CPOP algorithms under different task nodes. The number of selected processor cores is 4, CCR=5, and the normalized task scheduling length NSL of ILSGA and GA and CPOP algorithm when task node N is 20, 40, 60, 80, 100, and ILSGA and the number of iterations for GA to obtain the optimal solution, and the test result is shown in Figure 2.

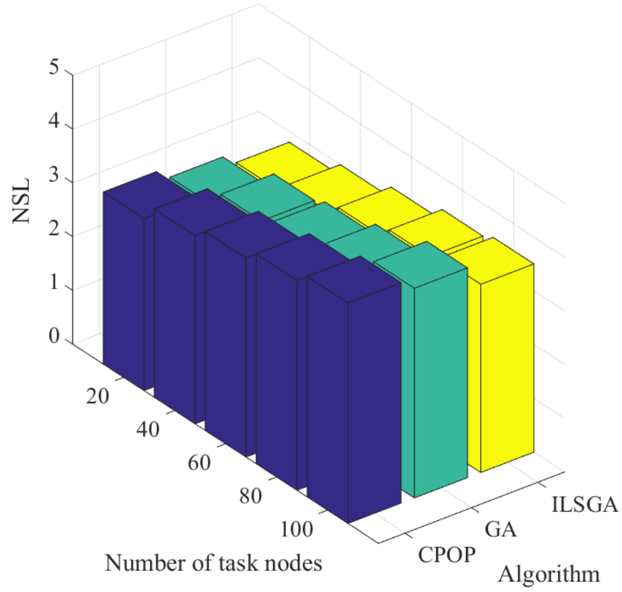

Figure 2: NSL of three algorithms under different task nodes

According to the data analysis in Figure 2, under the same number of task nodes, the quality of the optimal solution obtained by ILSGA is the highest, and as the number of tasks increases, the advantages of ILSGA become increasingly obvious. And under the same number of task nodes, the number of iterations for ILSGA to obtain the optimal solution of task scheduling length is less than that of GA, and the convergence speed is faster. The reason is that ILSGA adds the task scheduling sequence obtained by CPOP to the initialization population, which improves the quality of the initialization population. Therefore, the algorithm can find the optimal solution for task scheduling more quickly, so that the heterogeneous multicore processor has better scheduling efficiency. In order to compare the evolution of ILSGA and GA more intuitively, when the number of task nodes is 20, the task scheduling length SL of the two algorithms changes with the increase of the evolution time. The statistical results are shown in Figure 3.

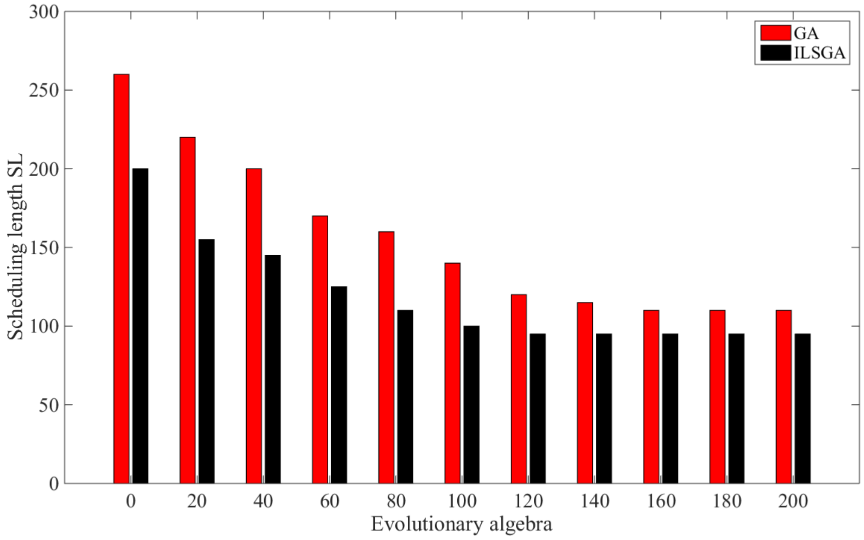

Figure 3: The iterative process of ILSGA and GA

4.2 Analysis of Option Pricing Algorithm

This paper applies PSO algorithm, QPSO algorithm, DE algorithm, and ES algorithm to solve the parameters of the Black-Scholes option pricing model. The data analysis and drawing work is done under the Matlab environment.

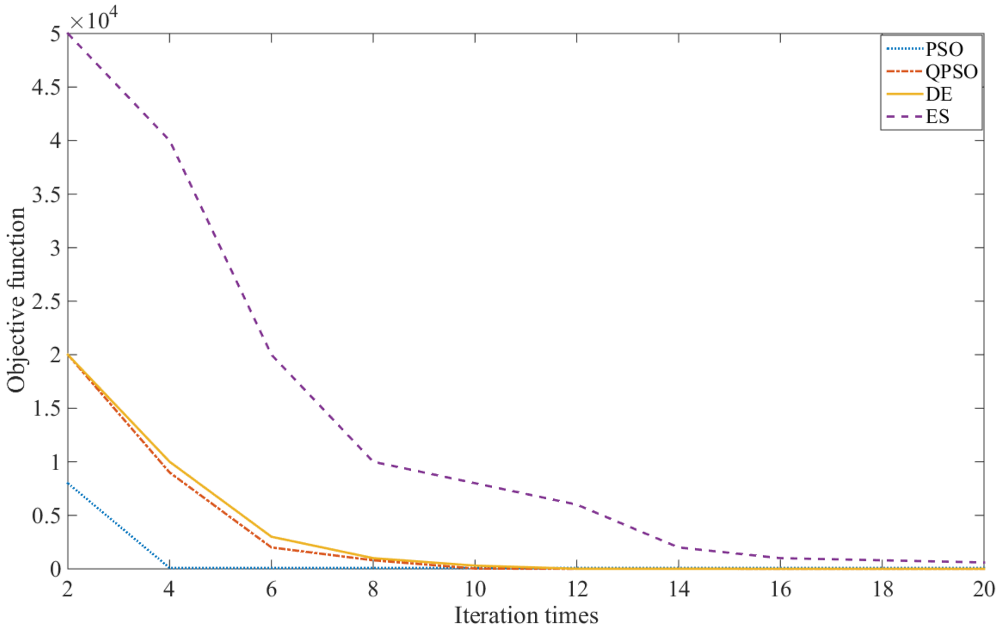

Figure 4: Convergence curves of objective functions of different algorithms

Figure 4 shows the objective function convergence curve of PSO, QPSO, DE, and ES with 20 iterations. The number of particles is 20, and the objective function value is the average value of the program running 20 times. From the data in the figure, it can be seen that the value of the objective function of the QPSO algorithm after 11 iterations is close to 0, while the value of the objective function of the DE algorithm tends to be stable after about 20 iterations. The PSO algorithm is easy to mature prematurely, and the result of ES algorithm fitting is better. difference. The reasons for the poor convergence of the objective function of the PSO and ES algorithms may have the following aspects: (1) The parameters are not sensitive to the model, and the changes of the parameters cannot reflect the changes of the model. This requires more accurate data to simulate a more accurate model. (2) The performance of the algorithm prevents the parameters from getting rid of the local optimal solution, and the algorithm needs to be improved to improve the algorithm's global search ability. (3) The experiment has errors. The experimental results show that the QPSO algorithm has the best convergence performance, followed by the DE algorithm. The fitting values of the PSO algorithm and the ES algorithm are poor, and the PSO is easy to fall into the local optimum. It is concluded that the QPSO algorithm performs better in solving the problem. Well, PSO and ES algorithms are not suitable for solving this problem.

Table 1 shows the experimental values given by r and the fitting values of various algorithms. Among them, the number of particles is 20, the number of algorithm iterations is 10, and the fitting values of various algorithms are the best fitting values after running the program 10 times. From the table, it can be seen that the fitting values of the QPSO algorithm and the DE algorithm are closest to the exact value, while the fitting values of the PSO algorithm and the ES algorithm are worse.

Table 1: Experimental values of r and fitting values of various algorithms

Serial number	E	s	T	o	experimental value of r	Algorithm fitting value of r			
						PSo	QPSo	DE	ES
1	280	316	0.07	0.16	0.08	0	0.08	0.078	0.09
2	290	316	0.07	0.16	0.08	0	0.08	0.078	0.07
3	295	316	0.07	0.16	0.08	0	0.08	0.079	0.06
4	300	316	0.07	0.16	0.08	0	0.08	0.078	0.08
5	305	316	0.07	0.16	0.08	0	0.08	0.079	0.08
6	310	316	0.068	0.17	0.078	0	0.077	0.079	0.03
7	315	316	0.068	0.17	0.078	0.1	0.077	0.082	0.02
8	280	316	0.068	0.17	0.078	0	0.078	0.076	0.07
9	290	316	0.068	0.17	0.078	0	0.078	0.077	0.09
10	295	316	0.068	0.17	0.078	0.09	0.078	0.077	0.07

Figure 5 shows the experimental and predicted values of r, where the continuous curve represents the experimental value, and the marked points represent the predicted data. The predicted value is the optimal value of the program running 10 times. From the data in the figure, it can be seen that the values predicted by the QPSO and DE algorithms basically fit the experimental values, while the values predicted by the PSO and ES algorithms have a large gap with the experimental values.

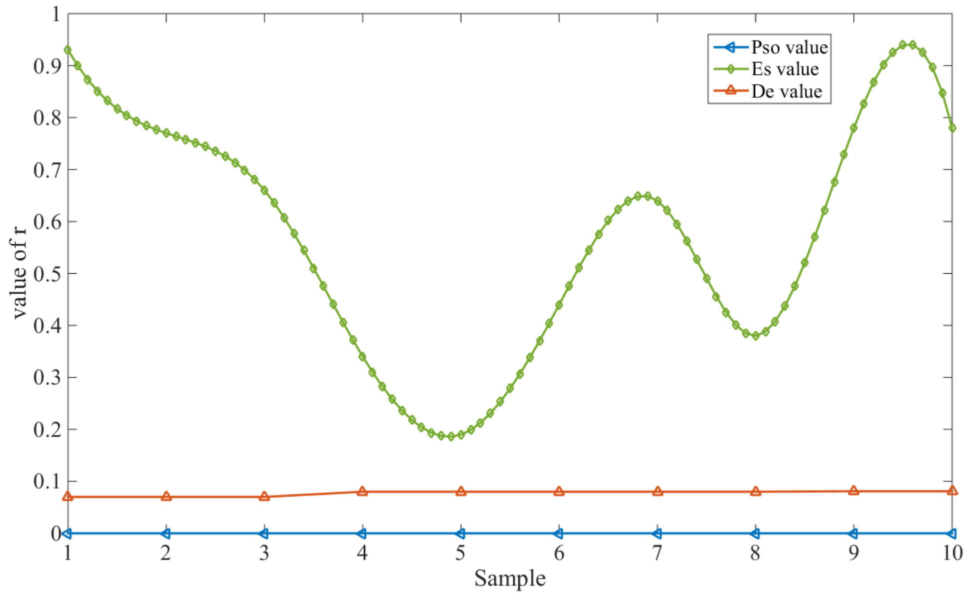

Figure 5: Experimental and predicted values of r

Table 2 is the statistical analysis table of the algorithm. It lists the maximum value, minimum value, average value, and variance value of the objective function value of the four algorithms of PSO, QPSO, DE, and ES after the program is run 20 times. Among them, the number of iterations of the algorithm is 20. It can be seen from the data in the table that when iterating 20 times, the average value of the objective function of the QPSO algorithm is the smallest, which is 0.001, followed by the average value of the objective function of the DE algorithm, which is 3.9. ES has the largest mean value. The variance of the QPSO algorithm is the smallest, and the variances of other algorithms are larger, indicating that the QPSO algorithm has the smallest volatility and the most stable. The mean and score difference of the PSO algorithm are 54 and 864, respectively, followed by DE, PSO, and ES. In the QPSO algorithm, the mean and variance are both 0.001, the mean variance of DE is small, and the mean variance of PSO and ES are both large.

Table 2: Statistical analysis of algorithm

Statistics	PSO	QPSO	DE	ES
max	75	0.2	74	936
min	3.5	0.00003	0.0007	0.6
mean	54	0.01	3.9	330
var	864	0.02	145	91600

It can be seen from the data in the above table that the QPSO algorithm has the smallest mean and variance, the smallest objective function value, and the best stability. At the same time, as the number of iterations increases, the mean and variance of various algorithms gradually decrease. It shows that the number of iterations has an impact on the convergence of the objective function. As the number of iterations increases, the value of the objective function becomes smaller and smaller. Based on the above experiments, it is concluded that the objective function value of the QPSO algorithm is the best, the parameter fitting effect is the best, and the algorithm is the most stable.

4.3 Analysis of Intelligent Algorithm High-Performance Computing Platform

This article uses the hardware description language VHDL to design the computing platform and its peripheral control. After compiling, synthesis, layout and other steps, a.bit object file is finally generated, and the.bit object file is downloaded to the target board through the serial download line. Among the chips, this experimental chip becomes a high-performance computing platform for the QPSO algorithm.

In FPGA design, simulation is an important step in material design using VHDL language, which runs through the entire design process. In the system design process, three kinds of simulations are usually performed: behavior-level simulation, RTL-level simulation, and portal-level simulation. The simulation goals at all levels are different, and the description requirements for the VHDL language are also different. Before logic synthesis, the modules need to be simulated to verify the correctness of the platform functions. After each module is connected, a VHDL platform needs to be established. The whole design is described in VHDL and implemented on Xilinx's spartan3fpga chip. In order to verify the correctness of the system and check the performance of the system, this paper uses two reference test functions f1 (x) and f2 (x) for system testing.

Among them, the program implemented by the software is written in MATLAB language and runs on a Pentium computer with a main frequency of 2.4G. Because the setting program needs to run for a period of time before the results appear, the previous section of the program is in the high impedance state. The test results are shown in Table 3.

Table 3: System performance test results

function	Implementation	Result	Cputime	Number of clock cycles
f1(x)	Software Implementation	0.00069	1.29s	3000
	Hardware implementation	0	2.69s	800
f2(x)	Software Implementation	0.0004	2.34s	5900
	Hardware implementation	0	3.15s.	940

In the results, the calculation results of the software and hardware implementation algorithms are not much different. The hardware implementation is fixed-point, while the software implementation is floating-point. From the perspective of CPU running time, software is faster than hardware, but this does not mean that the algorithm implemented by software runs faster than hardware. This is because the main frequency of the Pentium microcomputer cpu is 2.4G, and the main frequency of the FPGA is only 300M, so whoever has the fastest speed should be compared with their clock cycles, and whoever has less is faster. The number of clock cycles can be roughly obtained by multiplying the main frequency by the CPU time. From the results in the table, it can be seen that the number of clock cycles required by the software is about 4 times the number of clock cycles required by the hardware-implemented computing platform, which means that if you in the case of the same CPU frequency, the hardware implementation platform runs much faster than the software.

5. Conclusions

With the continuous development and improvement of the options market and the continuous improvement of the quality of option practitioners, the use of intelligent optimization algorithms to solve some problems in option pricing will become increasingly widely used. Intelligent optimization algorithms have broad applications in the field of option pricing prospects. However, the option market is a complex non-linear system. The environment of the system is always changing. Usually, people estimate the interest rate and volatility of the unknown parameters in the option pricing model based on past conditions, and then make decisions and invest. Therefore, there are always differences in the investment process. Therefore, the use of intelligent algorithms to estimate the parameters of option pricing has a certain reference value for actual investment.

Although the research in this article has good reference value, however, the research focus of this article is the full pricing algorithm. It is based on the tasks that have been divided. There is no in-depth study of the rules of task division, and it is based on certain assumptions that make the algorithm there is a certain distance between the user environment and the actual application, which is only suitable for basic theoretical research.

The computing core designed in this paper contains rich types of computing resources and large-capacity register files, and these two parts constitute the main body of the computing core. However, in actual use, it does not mean that more is better or bigger is better. It is the best strategy to reasonably select and configure computing resources and memory according to the actual situation. Therefore, from a local perspective, calculating the type of computing unit in the core, and the configurability of number and memory capacity is worth studying.

Acknowledgements

The main research results of the research project of education and teaching reform of Harbin Guangsha College "Research and Practice on the Hierarchical Training Mode of Applied Undergraduate Information Technology Talents Based on U-Learning". No: JY2022B08

References

[1] Cuenca J, L. P. García, D. Giménez, et al. (2017) Guided installation of basic linear algebra routines in a clusterrnwith manycore components. Concurrency and Computation, 29(15):1-14.

[2] Perera C, Liu C H, Jayawardena S. (2017) The Emerging Internet of Things Marketplace from an Industrial Perspective: A Survey. IEEE Transactions on Emerging Topics in Computing, 3(4):585-598.

[3] Samimi O, Mardani Z, Sharafpour S, et al. (2017) LSM Algorithm for Pricing American Option Under Heston–Hull–White's Stochastic Volatility Model. Computational Economics, 50(2):1-15.

[4] Alrawais A, Alhothaily A, Hu C, et al. (2017) Fog Computing for the Internet of Things: Security and Privacy Issues. IEEE Internet Computing, 21(2):34-42.

[5] Kan G, He X, Ding L, et al. (2020) Heterogeneous parallel computing accelerated generalized likelihood uncertainty estimation (GLUE) method for fast hydrological model uncertainty analysis purpose. Engineering with Computers, 36(1):75-96.
[6] Xiao G, Chen Y, Liu C, et al. (2020) ahSpMV: An Autotuning Hybrid Computing Scheme for SpMV on the Sunway Architecture. IEEE Internet of Things Journal, 7(3):1736-1744.
[7] Xiang Y, Pasricha S. (2018) Mixed-criticality scheduling on heterogeneous multicore systems powered by energy harvesting. Integration, 61(MAR.):114-124.
[8] Vaisband B, Friedman E G. (2018) Heterogeneous 3-D ICs as a platform for hybrid energy harvesting in IoT systems ☆. Future Generation Computer Systems, 87(OCT.):152-158.
[9] Akpakwu G, Silva B, Hancke G P, et al. (2018) A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. IEEE Access, 5(12):3619-3647.
[10] Yang Y, Wu L, Yin G, et al. (2017) A Survey on Security and Privacy Issues in Internet-of-Things. Internet of Things Journal, IEEE, 4(5):1250-1258.
[11] Mosenia A, Jha N K. (2017) A Comprehensive Study of Security of Internet-of-Things. IEEE Transactions on Emerging Topics in Computing, 5(4):586-602.
[12] Razzaque M A, Milojevic-Jevric M, Palade A, et al. (2017) Middleware for Internet of Things: A Survey. IEEE Internet of Things Journal, 3(1):70-95.
[13] Rahmani A M, Gia T N, Negash B, et al. (2017) Exploiting Smart E-Health Gateways at the Edge of Healthcare Internet-of-Things: A Fog Computing Approach. Future Generation Computer Systems, 78(PT.2):641-658.
[14] Zhang Y, Wen J. (2017) The IoT electric business model: Using blockchain technology for the internet of things. Peer-to-Peer Networking and Applications, 10(4):983-994.
[15] Longchao Z, Jianjun X, Limei Y. (2017) Research on congestion elimination method of circuit overload and transmission congestion in the internet of things. Multimedia Tools and Applications, 76(17):18047-18066.
[16] Mostafa H, Kerstin T, Regina S. (2017) Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices. Healthcare Informatics Research, 23(1):4-15.
[17] Mishra D, Gunasekaran A, Childe S J, et al. (2017) Vision, applications and future challenges of Internet of Things: A bibliometric study of the recent literature. Industrial Management & Data Systems, 116(7):1331-1355.
[18] Lin J, Yu W, Zhang N, et al. (2017) A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. IEEE Internet of Things Journal, 4(5):1125-1142.
[19] Singh J, Pasquier T, Bacon J, et al. (2017) Twenty Security Considerations for Cloud-Supported Internet of Things. IEEE Internet of Things Journal, 3(3):269-284.
[20] Rubén Braojos, Bortolotti D, Bartolini A, et al. (2017) A Synchronization-Based Hybrid-Memory Multi-Core Architecture for Energy-Efficient Biomedical Signal Processing. IEEE Transactions on Computers, 66(4):575-585.
[21] Kan, G., He, X., Ding, L., Li, J., Hong, Y., & Liang, K. (2018) Heterogeneous parallel computing accelerated generalized likelihood uncertainty estimation (GLUE) method for fast hydrological model uncertainty analysis purpose. Engineering with Computers.
[22] Alonso-Monsalve S, Garcia-Carballeira F, Calderon A. (2018) A heterogeneous mobile cloud computing model for hybrid clouds. Future Generation Computer Systems, 87(OCT.):651-666.
[23] Zhang X, Gu W, Zhang H, et al. (2019) Hybrid Communication Path Orchestration for 5G Heterogeneous Ultra-Dense Networks. IEEE Network, 33(4):112-118.
[24] Chen C, Du Y, Zuo K, et al. (2019) toward fault-tolerant hybrid programming over large-scale heterogeneous clusters via checkpointing/restart optimization. Journal of supercomputing, 75(8):4226-4247.
[25] Xiao S, Ma S H, Li G, et al. (2017) European Option Pricing With a Fast Fourier Transform Algorithm for Big Data Analysis. IEEE Transactions on Industrial Informatics, 12(3):1219-1231.
[26] Xiaoli G, Xintian Z. (2017) Option pricing and hedging for tempered stable jumps driven stochastic volatility models based on improved PSO algorithm. Systems Engineering-Theory & Practice, 37(11):2765-2776.
[27] Sakuma T. (2017) Application of Fast N-Body Algorithm to Option Pricing under CGMY Model. Journal of Mathematical Finance, 07(2):308-318.
[28] Zhu D M, Lu J, Ching W K, et al. (2019) Option Pricing under a Stochastic Interest Rate and Volatility Model with Hidden Markovian Regime-Switching. Computational Economics, 53(2):555-586.
[29] Ma J, Zhou Z. (2017) Convergence Analysis of Iterative Laplace Transform Methods for the Coupled PDEs from Regime-Switching Option Pricing. Journal of entific Computing, 75(3):1656-1674.
[30] Song H, Wang X, Zhang K, et al. (2017) Primal-Dual Active Set Method for American Lookback Put Option Pricing. East Asian Journal on Applied Mathematics, 7(03):603-614.