

A Study on Multi-core Task Scheduling Algorithm based on Artificial Intelligence

Hu Zhiyu and Li Li

Jingdezhen University, Jingdezhen city, Jiangxi province, China, 333000

Abstract

With the rapid development of science technology, the multi-core processor system has been become one of the hottest issues in the high performance computation field at present. At the same time, there are some problems in the application and the process of development. In order to find the more efficient task scheduling algorithm, this paper will research the multi-core processors task scheduling algorithm. With exploring the existing task scheduling algorithm principle, the heterogeneous multi-core processor system task scheduling mathematical model is be built, and based on the genetic algorithm, the paper proposes the heterogeneous multi-core processor system scheduling based on population genetic algorithm. Then, through the feasibility, parameter analysis, verification algorithm, the improved genetic algorithm effectively improves the system performance, and reduces the running time. The model, in a certain extent, increases the application and development of artificial intelligence, and provides a theoretical basis for related research.

Keywords: multi-core processor system, task scheduling algorithm, heterogeneous multi-core system task scheduling algorithm model, genetic algorithm, verification

1. Introduction

With the development of science technology, people have entered the high-speed information age; processor system has become the foundation to support this time. At the same time of tasks running in parallel, how to ensure the instantaneity of multi-core tasks system and improve the system operation efficiency? How to shorten the length of task scheduling and keep the system load balance? How can the multi-core processor system give full play to its performance by the high efficient scheduling tasks? These problems have become a research focus of new era and the important problems that people must face. With the constant progress of the era, the single core processor has transformed into the multi-core processor. However, task scheduling algorithm of isomorphic multi-core processor system has been already mature, there are many disadvantages in the heterogeneous multi-core processor system scheduling and there is lots of improved room. Although in recent years, researchers at home and abroad has explored actively, they want to improve the efficiency of the process of processor scheduling through the process rapid distribution and fitness adjustment. But recently the task scheduling algorithm of multi-core processor is still not ideal, and affects the task scheduling performance of multiple core processors. According to the status quo of real-time task scheduling algorithm in the multi-core processor system, the paper analyzes the advantages and disadvantages of existing algorithm. And the paper follows the principle of foster strengths and circumvents weakness, and solves the task scheduling problems of heterogeneous multi-core processor system. Then, the paper makes the related adjustment and optimizes the scheduling algorithm of artificial intelligence.

2. The Related Research about Artificial Intelligence Multi-core Task Scheduling

2.1. Artificial Intelligence

Artificial intelligence, the English abbreviation is AI. It is a new technology science of the theory, method, technology and application system, using for researching, simulating and expanding the human intelligence. Artificial intelligence is a discipline that can use the computer for simulating human thinking process and intelligent behavior (such as learning, reasoning, thinking, planning and so on). It mainly includes the theory of computer intelligence and computer that is similar to the human brain intelligence. It is a branch of computer science. It attempts to understand the essence of intelligence, and can produce a new kind of intelligent machines that it can make the human intelligent response. The research about this field includes robots, speech recognition, image recognition, natural language processing and expert system. Since the birth of artificial intelligence, the theory and technology are maturing, the application fields are expanding. We can image that the artificial intelligence technological products will be the “container” of human intelligence in the future.

2.2. The Status Quo of Multi-core Task Scheduling

In fact, multi-core task scheduling faces the opportunity and challenge at the same time. Task scheduling algorithm is a typical problem in optimization combination problems, and also is the core issue of computer research. At present, there is a better algorithm for the heterogeneous multi-core processor task scheduling. But heterogeneous multi-core processor task scheduling also faces many problems, such as the division of application, the task assigned to the processor, and keep the system load balanced. Therefore, there is not an exact algorithm is recognized as the most optimal until today, and there is lots of space for the researchers. The multi-core task scheduling mainly classifies as follows:

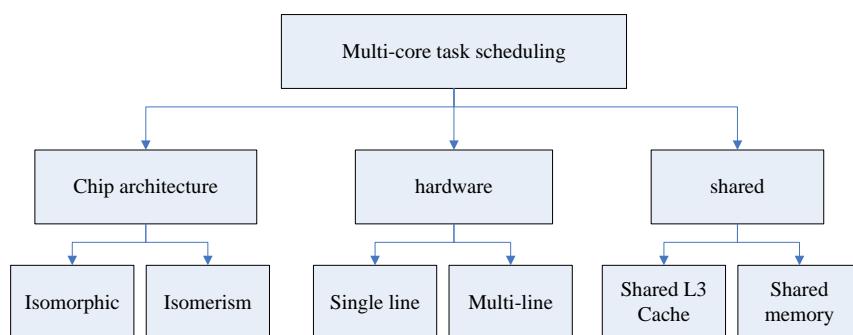


Figure 1. Multi-core Tasks Distribution

2.3. The Basic Techniques of Multi-core Task Scheduling

Now some artificial intelligence methods have already formed the system. These artificial intelligence methods are genetic algorithm, simulated annealing algorithm, ant colony algorithm, tabu search algorithm and so on. There are advantages and shortcomings of each algorithm, so experts are worth doing further research. Fully considering the measurement scale of final task execution time, the complexity of the algorithm time, the scalability, and processing nodes utilization, the main algorithm is as

follows:

(1) Uncertain scheduling technology

The other name of this technology is random search scheduling technology. In fact, this method is got by random contrast function. The corresponding process is the process that the scope of existence space is shrinking. So the running efficiency is low. It includes simulated annealing algorithm. In the process of comparison, this algorithm constantly produces the optimal solution, until the final most optimal solutions to produce. In fact, optimization algorithm, in this paper, is one of the random searches scheduling technology, named genetic algorithm. Actually, due to paired comparison method, its efficiency is higher than other algorithms. Genetic algorithm is established on the simulation of biological genetic and evolutionary process and is a kind of highly parallel global probability search algorithm. Genetic algorithm reflects “survival of the fittest” competition mechanism. Because genetic algorithm is directly search optimization method and is generated by the theory of evolution and the genetics mechanism, so in this algorithm, many concepts of evolution and genetics are used. Genetic algorithm is a bionic algorithm with macro sense, and its mechanism is the simulation of the formation and evolution of all life and wisdom.

(2) Heuristic scheduling algorithm

The main focus of this algorithm is time. With the constraint of time, the solution with the optimal scheduling can be achieved. The algorithm mainly includes the following sub-options:

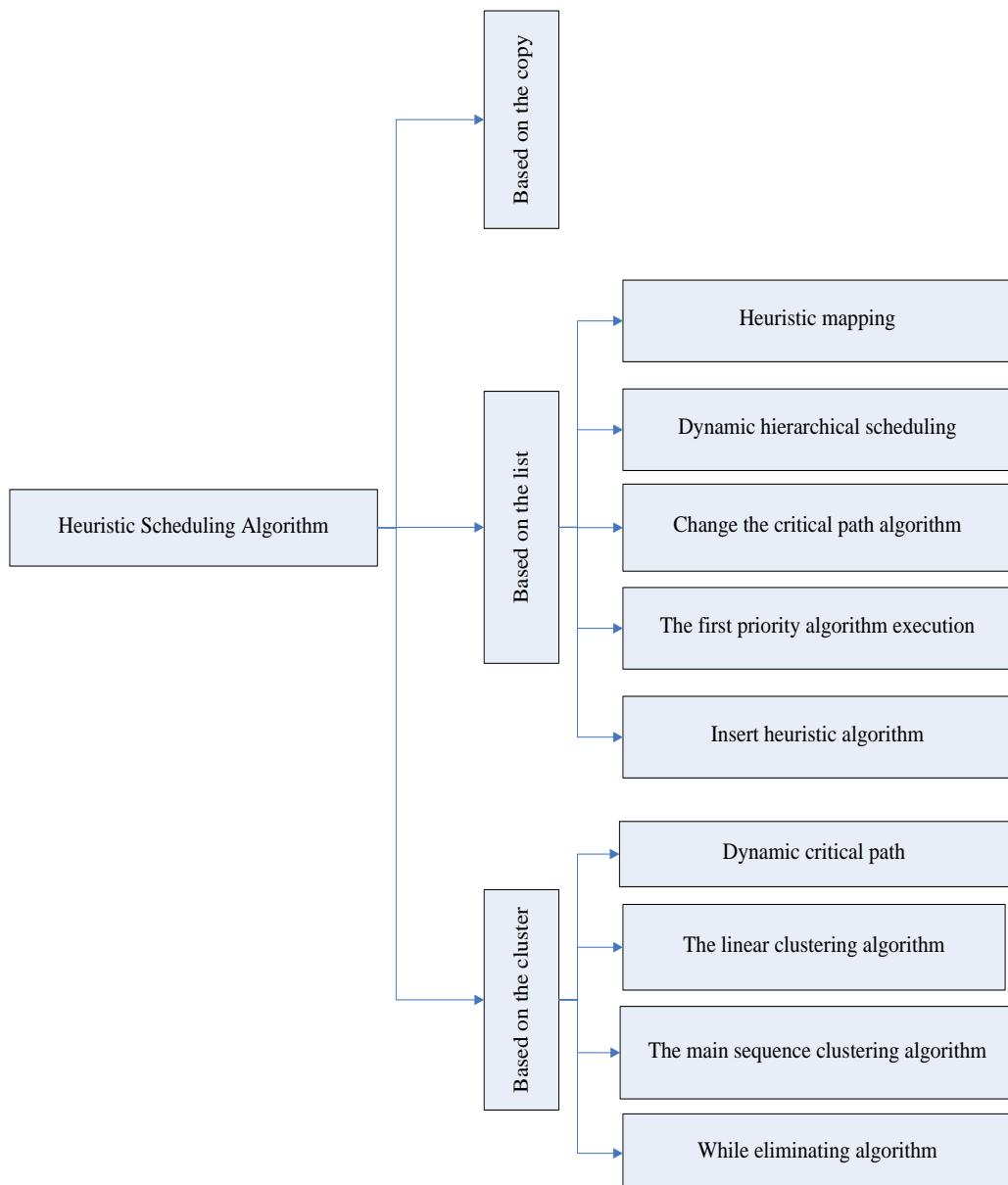


Figure 2. Distribution of Heuristic Scheduling Technology

3. Make sure the Multi-core Task Scheduling based on Artificial Intelligence

3.1. Genetic Algorithm and the Steps to Solve the Problem

The specific process of genetic algorithm is shown in picture 3.

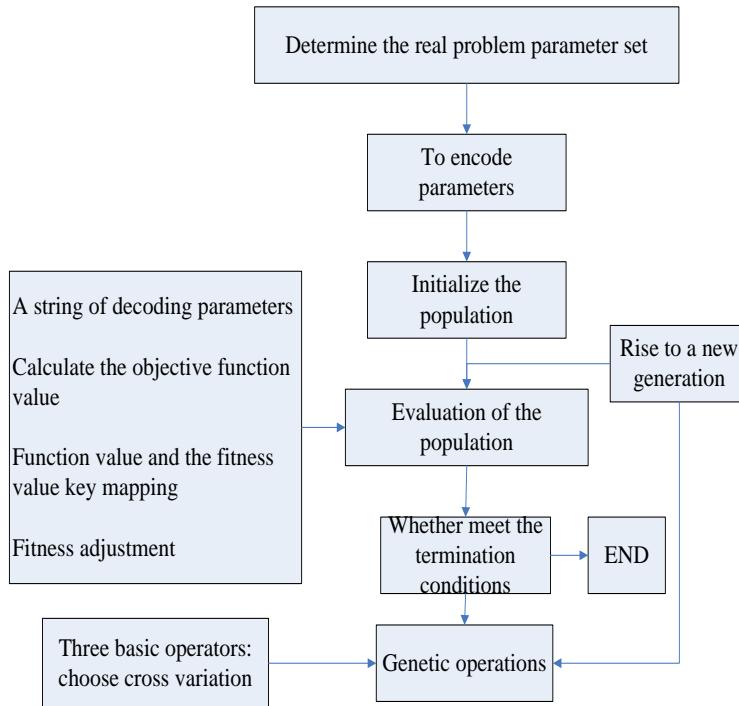


Figure 3. The Specific Process of Genetic Algorithm

3.2. Genetic Algorithm Model Theorem

The basic theory of genetic algorithm is a mathematical tool provided by schema theorem. This mathematical tool can explain the operational mechanism of genetic algorithm. It contains some behavior criterion and guidelines of developing code strategy, also contains some behavior criterion and guidelines of genetic operation strategy. The patterns in the schema theorem are the similar modules. It is described as a subset. This subset and individual coding string on some location of gene string have the similar structure character. So the schema can also be interpreted as the same configuration. For example, in the binary code, the schema consists of character string based on three value character set $V = \{0, 1, *\}$. * can be replaced by either 0 or 1. The schema is a set of strings. The described string must be consistent with the gene of location of "*" in the schema. So we can find that "*" in the schema is more, the number of strings in the described string sets are more. For example, a schema is $1**0$, so the represent string set is $\{1000, 1010, 1100, 1110\}$. There are four main parameters in a schema E : the original length e , the schema order $O(E)$, the defining length $\delta(E)$, and the computational formula:

$$D(E) = 2^{e-O(s)}$$

(1)

If the schema is $E = 1**0$, the original length of the schema is 4, the schema order is $O(E) = 2$, the defining length is $\delta(E) = 4 - 1 = 3$, the model dimension is $D(E) = 2^{4-2} = 4$. If a given time t (the generation of t), there are m individuals in the population $P(t)$ belonging to schema E , $m = m(E, t)$, so it means that in the era of genetic to t , there are m individuals belonging to schema E . On the choice of individual populations, every string will be replicated (selected) through its fitness rate value. In

the process of copying, the strings, containing E , according to

$$P_r = F(E) / \sum_j^n F_j, \quad F(E) \text{ is the fitness rate of containing } E \text{ strings.}$$

$\sum_j^n F_j$ is the sum of fitness rate of all individuals in the populations. So the individuals of copying populations are $nF(E) / \sum_j^n F_j$. If $\overline{F(E)}$ is the average fitness rate of strings of containing E , so $m(E, t+1)$ is:

$$m(E, t+1) = \frac{n m(E, t) \overline{F(E)}}{\sum_j^n F_j} = m(E, t) \frac{n \overline{F(E)}}{\sum_j^n F_j} \quad (2)$$

Because the average fitness rate of populations is

$$\overline{F} = \frac{\sum_j^n F_j}{n} \quad (3)$$

So the formula 2 can be rewritten:

$$m(E, t+1) = m(E, t) \frac{n \overline{F(E)}}{n \overline{F}} = m(E, t) \frac{\overline{F(E)}}{\overline{F}} \quad (4)$$

If the average fitness rate of E is $\overline{F(E)} = (1+C)\overline{F}$, so

$$m(E, t+1) = m(E, t) \frac{(1+c)\overline{F}}{\overline{F}} = (1+c)m(E, t) \quad (5)$$

If $t=0$, it means that the gene starts from the first generation. c is permanent, so from the formula 5, we can get

$$m(E, t) = (1+c)^t m(E, 0) \quad (6)$$

From the formula 6, we can get that if $c > 0$, $m(E, t)$ presents the exponential increase. Crossover operation and mutation operation in genetic algorithm will destroy the schema with a certain probability. The probability of crossover operation is P_c , the damage probability of crossover operation for schema E is

$P_c \cdot \frac{\delta(S)}{e-1}$. If the probability of mutation operation is P_m , the damage probability of mutation operation is $P_m \cdot O(S)$. Above all, through formula 4, the schema theorem can be summed up as:

$$m(E, t+1) \geq m(E, t) \frac{\bar{F}(E)}{F} \left(1 - P_c \cdot \frac{\delta(E)}{e-1} - P_m \cdot O(E) \right) \quad (7)$$

If any pattern is destroyed, a gene of the wildcard “*” in the schema changes. So the survival rate of schema is $P_s = (1 - P_m)O(E) \approx 1 - P_m \cdot O(E)$ (8)

Obviously, with the function of variety factors, the survival rate of schema can be summed as: $O(E)$ is smaller, the schema is easier to survive. $O(E)$ is bigger, the schema is easier to be destroyed.

The summarize of genetic algorithm schema theorem is: in the genetic algorithm, with the function of choice, crossover and variety factors, the schema, which has the definition of length with the low order and short and the average fitness value is higher than populations average fitness value, follows the exponential increase. So the definition length $\delta(E)$ is low, and the average fitness value $\bar{F}(E)$ is higher than the population average fitness value \bar{F} , the number of string sets in the schema E will follow the exponential increase in the genetic process.

3.3. The Advantages and Disadvantages of Genetic Algorithm

By the above research shows that in the current era, genetic algorithm is not good to solve the problems of measurement, and it is easy to fall into “premature”. So from the genetic algorithm there is a problem that is difficult to solve: how to select an algorithm and this algorithm can not only retain the better individuals, but also can keep the diversity of population. Now the paper will summarize the advantages and disadvantages of genetic algorithm.

Table 1. The Advantages and Disadvantages of Genetic Algorithm

advantages	disadvantages
No concern with the problem areas fast random search capabilities	Genetic programming is more complex, first need to encode the problem, then find the optimal solution to the problem also needs to be decoded
starting from groups. It has the potential parallelism. Comparison of multiple individuals at the same time	Further implementation of the algorithm has many parameters, such as crossover rate, mutation rate, selection of these parameters also affect the quality of the solution, and the selection of parameters largely dependent on experience
Search using heuristic evaluation function, the process is simple	Not timely feedback of information using the Internet to get a more accurate solution requires more time
Probabilistic mechanism iterates, randomness is scalable, easy to combine with other algorithms	Underutilized algorithm parallelism mechanisms underlying the ability
Probabilistic mechanism iterates, randomness is scalable, easy to combine with other algorithms	There is a certain dependence, it can be combined with improved heuristic algorithm

3.4. The Improvement of Genetic Algorithm

In order to better use the genetic algorithm, the paper proposes the improvement scheme for its problems. The scheme is as follows:

(1) Give certain guidance for genetic algorithm initialization operation. Because genetic

algorithm randomly generates the solution sets, so when initializing the population operation, we always choose a code of conduct from the problem space that meet the randomness to avoid the uniqueness of genes. Such not only can guarantee the species diversity, but also can make it to the superior species.

(2) Set the mixed ending condition of genetic algorithm. The mixed ending condition consists of the number of iterations and the optimal solution to determine conditions. The optimal solution to determine conditions are extracted from the corresponding problems. For this algorithm, the population iterative phase needs to be considered. If the early phases of algorithm meet the ending condition, the current solution is in the scope of locally optimal solution, so it's best to use the population fitness differences as a ending condition latterly, it is advantageous to the algorithm to search the optimal solution, and try to save calculation time.

Use the dynamical crossover rate P_c and mutation rate P_m . In the early, crossover rate P_c can remain relatively low. In order to improve the optimization speed, people can improve the crossover rate to optimize species, and the excellent genes can retain and propagate to the next generation as possible. In the later, population has a certain optimization improvement, species can also be relatively concentrated, so people use a higher mutation rate P_m and leave a excellent parent as many as possible, reduce the crossover rate P_c slightly. The adaptive ability of crossover rate and mutation rate and the self-organized ability of crossover rate and mutation rate can ensure the execution efficiency of algorithm and “premature” phenomenon. Next, the improved genetic algorithm model is set up as follows:

$$QPR = (YU, PO, QWA) \quad (9)$$

The task model of genetic algorithm is YU , the system model of genetic algorithm is PO , the task scheduling algorithm with genetic algorithm is QWA .

The task scheduling algorithm model YU can be expressed as:

$$YU = \{Y, L\} \quad (10)$$

Y represents the set of task scheduling, if the entire research process has k business, so $Y = \{Y_1, Y_2, \dots, Y_k\}$. L represents the constraint relation among the factors. The main relationship is as follows:

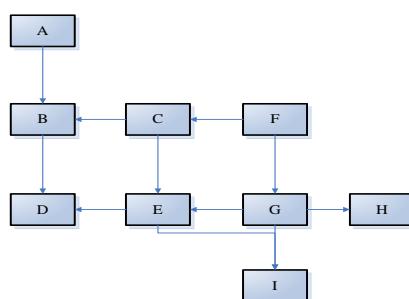


Figure 4. The Constraint Relation Graph

Task scheduling system model PM can be represented as: $PM = \{P, V\}$

(11)

P is processing core set, and is a form of collection, $P = \{P_1, P_2, \dots, P_N\}$.

B represents the connection structure among the processor cores. Here, the paper introduces the cost formula:

$$\text{cosstt} = E / AH \quad (12)$$

By combining the formula 10 and formula 11, the improved artificial intelligence multi-core task scheduling model is as follows:

$$QPR = (Y, L, P, V, QWA) \quad (13)$$

The improved model has the following characters:

- (1) Confirm to consider the needs of the whole process, and ensure the calculation of computation to estimate the task.
- (2) The whole process is a continuous process, and can not be interrupted and stopped.
- (3) In the process of computation, the computation only operates on a computation kernel, on the multiple computation kernel operation is not allowed.
- (4) In the process of computation, there are the precedence relations. It means ensure the process of prior running and subsequent running and the jumping steps are not allowed. If there is not the sequencing in the process of computation, people can adopt the following constrains to determine priority conditions:

$$\text{costc} = R/T \quad (14)$$

- (5) The computation nuclear in the process of calculation, the different running speeds are allowed.
- (6) In fact, computation capacity of computation nuclear has the direct relation with the capacity of calculation and communication. The relation is as follows:

$$\bar{df} = df + \frac{1}{n-1} \sum F_{mn} \quad (15)$$

- (7) The model has the stability in the process of calculation. Stability refers to the stability of the structure itself, and in the process of continuous calculation, the influence from others is smaller.

In general, in a certain extent, the improved model fully consider the whole process of task scheduling and computer running characters, then establish a model that has the more stable structure, the running time is shorter, and the structure order is more smooth.

4. The Practical Application of Multi-core Task Scheduling Algorithm

Based on the parameter settings in the dynamic genetic algorithm of better population, the DNA task graph is generated randomly by 10 control experiments. When generating randomly DNA task graph, the parameters are: nodenum, the scope of the number of task execution data [minnode, maxnode], task DAG graph maxdegreeout and [minedge, maxedge]

Laboratory random DAG task parameters are as follows:

Table 2. Laboratory Random DAG Task Parameters

DAG	nodenum	[minnode,maxnode]	[minedge,maxedge]	maxdegreeout
1	20	[200, 400]	[10, 50]	2
2	30	[200, 400]	[20, 60]	3
3	50	[100, 300]	[10, 50]	3
4	60	[100, 300]	[20, 60]	4
5	70	[200, 400]	[30, 80]	3
6	80	[200, 500]	[10, 50]	4
7	90	[100, 500]	[20, 40]	3
8	100	[200, 500]	[10, 50]	4
9	110	[100, 500]	[10, 30]	2
10	120	[200, 500]	[20, 60]	3

(1) Choose the tournament initial g in the operation

If g is too small, the tournament population of the whole operation process remains the same. If g is too big, in the late algorithm, g keeps the large value, so in a certain extent it will kill the population diversity. So g can't be too small or too big, it can ensure the population diversity. As shown in the picture below:

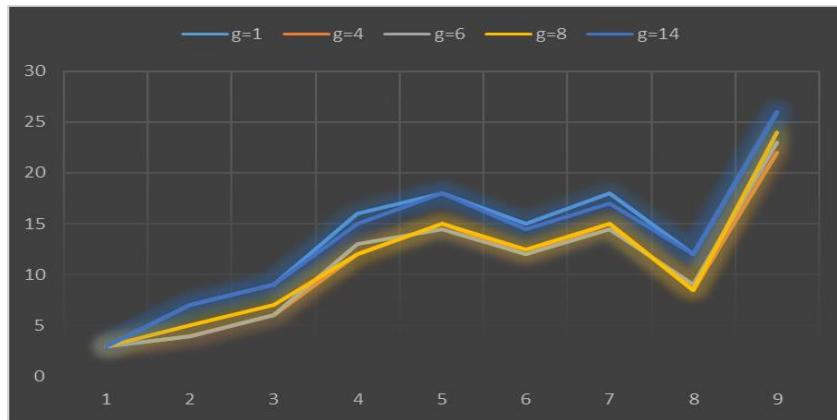


Figure 5. g Change Trend Line

(2) The basic value kc in the crossover rate P_c . $kc=0.6$, the solution is the worst. $kc=0.9$, the solution is relative bad. $kc=0.7$ and $kc=0.8$, the solution is relative good. The optimal value scope is $[0.7, 0.8]$.

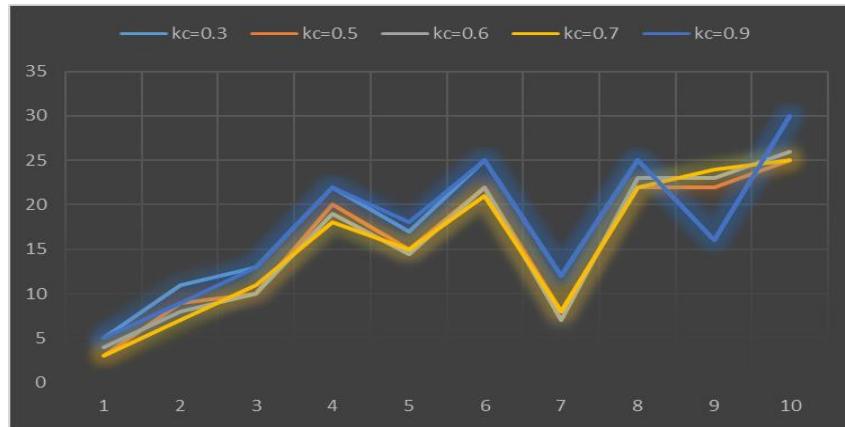


Figure 6. kc Change Trend Line

(3) The basic value km in the mutation rate P_m . Km=0.1, the probability of population mutation is small, so the solution is relative bad. Km=0.4, the probability of population mutation is big, so the solution is relative bad. km=0.2, the number of population is less, km is small, so the mutation rate is less, the capacity of population mutation is not strong, the search space is relatively insufficient and the solution is better. Km=0.3, it can cover the shortage of km=0.2, so the final solution is relative better. When the population is bigger, km=0.3 is too big, so the algorithm convergence is insufficient. When km=0.2, the mutation rate will be reduced, the convergence will be improved, so the solution is relatively better.

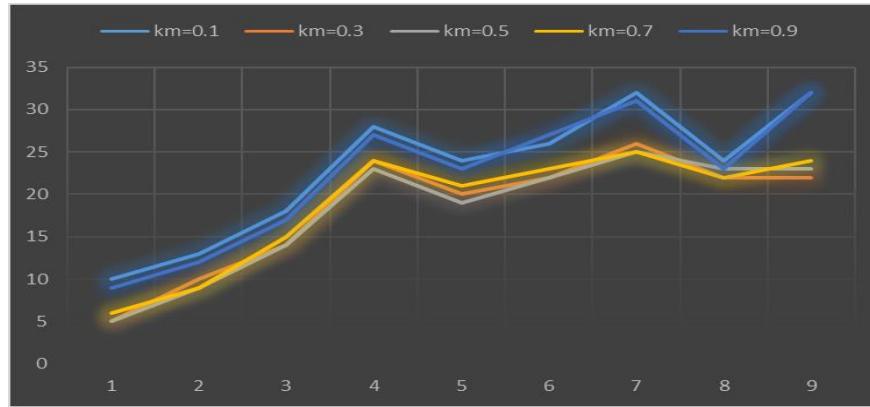


Figure 7. km Change Trend Line

(4) The population similarity cut-off value in the end conditions: k=0.3 and k=0.4, the solution is the worst, but the number of iteration is less. k=0.1 and k=0.2, the solution is better. The number of iterations is less and the calculation running time is less, compensating for the shortcomings of low precision.

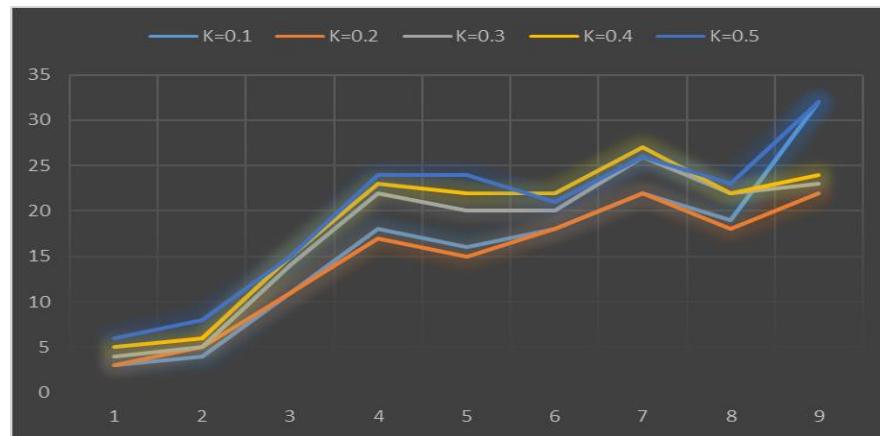


Figure 8. K Change Trend Line

Through the paper, we can find that heterogeneous multi-core processor is according to high and low collocation, it can not only ensure the execution with high efficient, but also save the system cost, so it becomes the inevitable trend in the future. For the research of heterogeneous multi-core processor, it directly affects the system performance, so the system task scheduling problems are also important. System task scheduling problems have been proved to be NP safety problems. So the execution time of general heuristic algorithm is complex, but the precision of solutions is low. Therefore, some artificial similar algorithm draw the more attention in recent years, and the application is used in the NP safety problems. Genetic algorithm had a cold era in the early; the research about it is more and more in recent years. In this paper, the improved genetic algorithm is used in the heterogeneous multi-core processor system task scheduling. The paper summarizes the research contents and innovation about heterogeneous multi-core processor task scheduling. The summary is as follows:

- (1) Through analyzing the development of heterogeneous multi-core processor system and the task scheduling algorithm, the paper proposes the basic model to be more intuitive, and show it concretely.
- (2) The paper introduces the basic genetic algorithm, and proposes its advantages and disadvantages.
- (3) The improved genetic algorithm is used in the heterogeneous multi-core processor system task scheduling, and the paper proposes dynamic genetic algorithm of excellent populations. The diversity of population is ensured through the heuristic algorithm, and then the excellent populations are produced.
- (4) The fitness function needed by dynamic genetic algorithm based on the excellent populations is dynamic, and provides references for tournament selections.

Finally, the paper introduces the advantages and disadvantages of the algorithm, and proposes the corresponding improvement direction. Because of the limitation of individual research level and research time, there are still many deficiencies and inadequate, the further study is needed.

5. Conclusion

Today the artificial intelligence is continuously developing; the practical problems can be solved by the computer characters, the optimal form of multi-core task scheduling algorithm, and the improvement of calculation capacity and computation efficiency. Through the establishment of the genetic algorithm model, the paper get the conclusion: with the function of choice, crossover, and mutation, the model has the definition length

of low order and short, the average fitness value is higher than the population average fitness value, and the model is according to exponential increase. But there is the uniqueness and time consuming in genetic algorithm model, so the paper proposes the optimal model and establishes the improved genetic algorithm model based on the integrity and efficiency. Finally, through the verification of algorithm practical application, the paper get that if the initial group and crossover rate are more concentrated, the mutation rate degree is less; if the population is more similar, the efficiency of algorithm is higher. Therefore, the collocation of heterogeneous multi-core processors is vital for the future application of artificial intelligence. Reasonable device configuration not only can ensure high efficient execution, but also can save the system expenses at the same time. It is vital for the development of multi-core task scheduling and the extension of related fields.

References

- [1] K. Deb, S. Aarawal and A. Pratap, "A fast elitist no dominated sorting genetic algorithm for multi-objective optimization: NAGA-II", Proceeding of the Parallel Problem Solving from Nature VI Conf, Paria, (2000).
- [2] N. Hirabayashi, H. Nagasawa and N. Nishiyama, "A decomposition scheduling method for operating flexible manufacturing systems", International Journal of Production Research, vol. 2, no. 1, (1994), pp. 161-178.
- [3] M. John. Usher, "Negotiation-based routing in job shops via collaborative agents", Journal of Intelligent Manufacturing, vol. 14, no. 5, (2003), pp. 485-499.
- [4] T. Kis, "On the complexity of non-preemptive shop scheduling with two jobs", Computing, vol. 69, no. 1, (2002), pp. 37-49.
- [5] C. Meloni, D. Pacciarelli and M. Pranzo, "A rollout metaheuristic for job shop scheduling problems", Annals of Operations Research, vol. 131, no. 1, (2004), pp. 215-235.
- [6] M. Gunes, U. Sorges and I. Bouazzi, "ARA the ant colony based routing algorithm for MANETs: Proceedings International Conference on Parallel Processing Workshops", (2002); Vancouver, BC, Canada.
- [7] C. David, Bushcc and A. Joseph, "A bi-directional scheduling approach for job shops", International Journal of Production Research, vol. 37, no. 16, (1999), pp. 3823-3837.
- [8] J. Carretero, F. Xhafa and A. Abraham, "Genetic Algorithm Based Schedulers for Grid Computing Systems", International Journal of Innovative Computing, Information and Control, vol. 3, no. 6, (2007).
- [9] P. Dutot, "Complexity of master-slave tasking on heterogeneous trees", European Journal on Operational Research, vol. 164, no. 3, (2005), pp. 690—695.
- [10] N. Maleolm and W. Zhao, "HarderAll-time Communication in MultiPle-access Netwokrs", Real Time system, vol. 9, (1995), pp. 75-107.
- [11] F. Chang, J. Dean and S. Getaway, "Bitable: A distributed storage system for structured data", ACM Transactions on Computer Systems (TOCS), vol. 26, no. 2, (2008), pp. 4-6.

Authors



Hu Zhiyu. He is born in March, 1983 in Yixian, Anhui Province. His research direction is Graphics, artificial intelligence and computer network. He works at Jingdezhen University in Jingdezhen City, Jiangxi Province now.



Li Li. She is born in December, 1983 in Jingdezhen, Jiangxi Province. Her research direction is computer network technology and software engineering. She works at Jingdezhen University in Jingdezhen City, Jiangxi Province now.