Optimization of Resource Scheduling Based On Genetic Algorithm in Cloud Computing Environment

Huaqiao Ye

Wuhan Institute of Shipbuilding Technology,
Wuhan 430050, Hubei, China

Abstract
This paper discusses the optimization of resource scheduling based on genetic algorithm in cloud computing environment. The research adopts methods based on chromosome coding and genetic operators in terms of workshop scheduling to improve resource scheduling in cloud computing environment. By testing the execution efficiency of the algorithm with simulation examples, the result shows that the algorithm is better than traditional genetic algorithms in performance and service quality. The optimized resource scheduling based on genetic algorithm plays an active role in optimization, which can be more adaptive for resource scheduling for large-scale tasks in cloud computing environment.
Key words: GENETIC ALGORITHM, CLOUD COMPUTING, OPTIMIZATION, RESOURCE SCHEDULING

1. Introduction

Genetic algorithm (GA for short)–a random optimization algorithm supported by evolutionary thought was first proposed by Professor Holland. In the western research tradition, project scheduling was proposed in the middle of 20th century and measures such as Gantt chart, key activity diagram and network are taken to carry it out. In the meantime, the fifties of 20the century saw scholars' proposition of genetic algorithm in terms of single-mode resource constraint project scheduling, which inherits coding scheme based on permutation and priority and implements serial scheduling, for using renewable resource in domestic study in China. In this research, strategies are proposed to optimize resource scheduling for genetic algorithm in cloud environment, to effectively improve resource allocation rate and to fulfill the development needs of this era.

Cloud computing is now a computing mode based on the internet. In cloud computing environment, multiple computers can be assigned with different tasks as to constitute a resource pool and the application system in cloud computing environment can obtain different software service functions [1]. Due to the rapid development of information technology, data has gradually become huge and complicated on the internet, so resource scheduling based on traditional genetic algorithm is unable to meet the demand of office and research. Specifically, current resource scheduling by genetic algorithm is constrained in terms of the cloud, which not only has a low convergence rate but is extremely prone to premature convergence as well.

In the practical application of genetic algorithm, due to its serious deficiency it is emergent to innovate an effective one to carry out resource scheduling in a more suitable manner. This paper discusses the question as how to optimize resource scheduling based on genetic algorithm in cloud computing environment, exemplified by an enterprise, so as to give full play to the optimizing role of resource scheduling and provides effective optimization measures in terms of this issue [2].

2. Material and methods

This research seeks the optimal strategy and optimizes scheduling plan for the optimization of workshop scheduling of an enterprise in order to ensure that workshop resource scheduling reaches optimization level. The discussion divides the methods of resource scheduling in terms of different genetic algorithms into traditional group and optimization group. The former adopts traditional genetic algorithm for resource scheduling while the latter employs the algorithm based on chromosome coding and genetic operators. The two algorithms are compared in terms of their effect as the discussion goes on.

For the traditional group, traditional genetic algorithm is adopted to achieve cloud resource scheduling in workshop scheduling, wherein the scheduling and searching process does not directly affect a variable but the coding individual based on parameter set. In the meantime, in the process of the workshop resource scheduling and searching, it can iterate from a set of solutions to another, implementing parallelization for resource scheduling [3]. While for the optimization group, it is through advanced genetic algorithm of fitness functions based on chromosome coding to bring about effective task scheduling of workshop resource. The count of tasks can be taken as the string length of the chromosome genes in the cloud while the total amount of resources can be taken as the string length of the chromosome genes in practice to ensure that total resource mapping tasks and task IDs can be taken as the value of genes in the chromosome for effectively accelerating convergence velocity. Resource-task indirect real number encoding can be applied to carry out predictive encode for the chromosome, for example: firstly, setting m task with task ID = {1, 2, 3, … , m}; and secondly, setting n resource nodes with resource ID = {1, 2, 3, … , n}. Now the string length of the chromosome genes equals the count of tasks m while the gene value of the chromosome can range from 1 to n; thus both form complete chromosome encoding. In the meantime, genetic algorithm evaluates all the chromosomes to define how they are adapted to problem solving with optimized resource scheduling algorithm. Therefore, it is necessary to evaluate solutions to chromosome candidates in terms of developed function.

3. Procedures of optimizing genetic algorithm based scheduling in the cloud

3.1 Basic ideas

A genetic algorithm is both a search heuristic that mimics the process of natural selection and an adaptive probabilistic algorithm for global optimization [4]. In computing, it usually defines a 5-turple including individual (chromosome) coding method, fitness function, initial solution group, genetic operation and parameters of genetic algorithm. It enables crossover, selection and mutation among population to generate species better than their parents [5].

3.2 Coding

For optimized genetic algorithm suitable karyotypes can be selected to ensure legitimacy and feasibility of chromosome coding. This research adopts a procedure based coding method, using direct decimal expression, as shows in Table 1 below:

Table 1. Code mapping

Procedure	21	11	31	41	42	32	22	23	33	12	13	43
Code	2	1	3	4	4	3	2	2	3	1	1	4

In Table 1, “2” in its first occurrence represents “21” and “23” in its third appearance. It is not difficult to figure out that feasible scheduling can always be achieved with any-to-any permutation and coding in this way also entails the optimal scheduling scheme. The solve loop of genetic algorithm is shown in Figure 1.

3.3 Generating initial groups

This paper adopts decimal coding to generate initial groups at random by generating two random numbers on the basis of a given chromosome, which take this for exchange position and promote the creation of new chromosome by the way of position exchange.

Meanwhile, the individual adaptation degree to living environment determines its survival chance. The value of fitness function is not negative in resource scheduling based on genetic algorithm, shown as the equation below:

$$F(s) = f(T(s)) = \sum_{i=1}^{m} \sum_{j=1}^{k_i} w_{ij} - T(s) \quad (1)$$

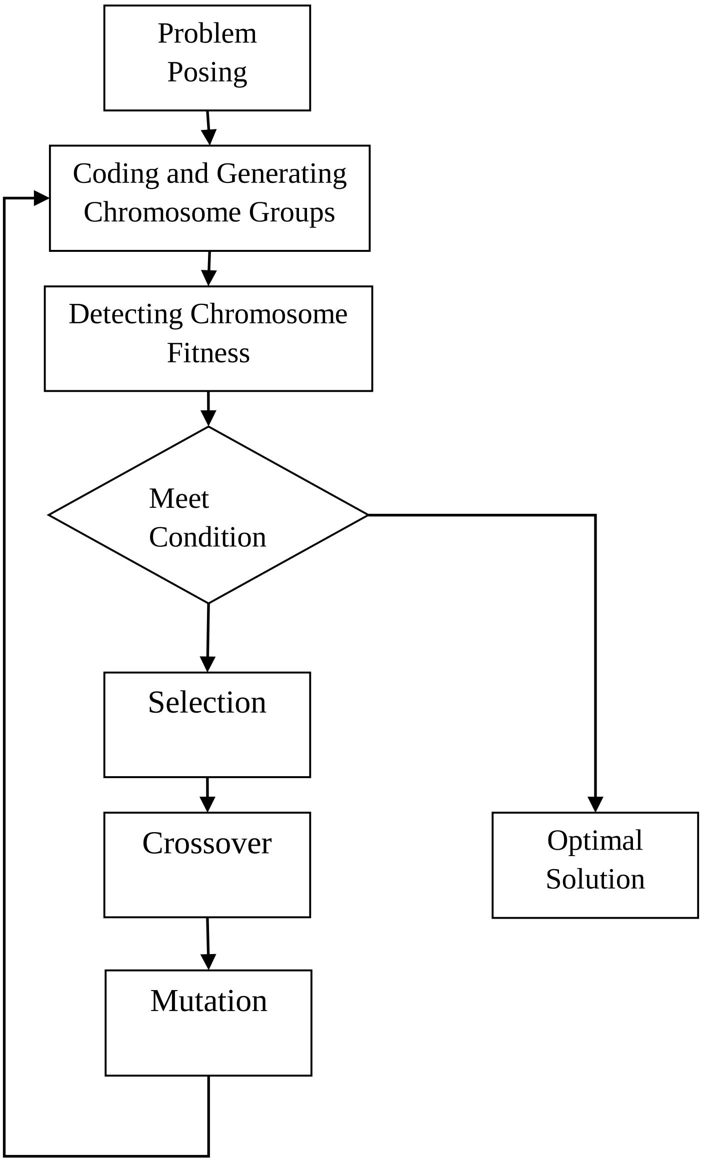

Figure 1. Solve loop of genetic algorithm

3.4 Selection of operator

The selection of operator should be based on the fitness value for the selection of several excellent individuals such that they can be handed down by heredity to the next generations. The group scale is set as 2*N for adaptation comparison each time and the individuals with the highest adaptation are chosen to pass on to the later generations [6].

4. Results and Simulation Research

4.1Set-up of simulation test environment

The simulation tool for cloud computing, namely, CloudSim can be employed to simulate local domain for examining the operating condition of the algorithm in the special grid. In terms of the simulation test of resource scheduling based on genetic algorithm in cloud computing environment, and it is necessary to use a PC with a 1.7GHz CPU and 256M memory and graphic interfaces such as VC++6.0 and Matlab. Thus can be achieved the simulation analysis of resource scheduling based on genetic algorithm and so can be examined the performance of the algorithm and accuracy of the results [7]. The environmental parameters for the application system are listed in Table 2 below:

Table 2. System environment parameters

Performance Parameters	Nodes Parts
Intel Corel2, Q9450, 1.7GHz	CPU parameters
GA-EP45-UD3LR, Chipset based on Intel P45+ICH10	Motherboard
Two modules of D9GKX 1GB, 256M	Memory
500GB	Hard disk

In the test, the algorithmic operation is carried out by MATLAB programming for the optimization design of the scheduling [8]. Below are a part of the pseudocodes:

The parameters involved in genetic algorithm include: m—total number of work pieces, n—total number of procedures, P—total number of machines, M—evolutional generations of genetic algorithm, N—population size, Pc—crossover rate, Pm—mutation rate, Q—matrix of machine serial numbers, W—matrix of operation time, Xp—optimal scheduling scheme, and farm—population. Parameter initialization codes are listed as below:

```
pn=length(find(W~=0));% recording the number of procedures for all work pieces
[m,n]=size(W);%m as total number of work pieces, n as total number of procedures
Xp=zeros(M,pn);% as optimal decision variable
LC1=zeros(1,M);% as convergence curve 1
LC2=zeros(1,N);% as convergence curve 2
pron=zeros(1,m);% recording the number of procedures for each work piece
for i=1:m
    pron(i)=length(find(W(i,:)~=0));
end
OPT  S=zeros(M,pn);
```

Excellent individuals can be selected from the current groups to serve as the parents for genetic algorithm [9]. Then in terms of vk of each chromosome its fitness degree can be calculated with

$$eval(vk);k=1,2,3,...,m \quad (2)$$

and selection probability is calculated in terms of each chromosome's vk:

$$P=eval\ (vk)/\sum eval\ (vk) \quad (3)$$

Thus initial population codes are generated as below:

```
farm=zeros(N,pn);% using matrix structure to store population
fori=1:N
for j=1:pn
gene=randint(1,1,[1 m]);
while length(find(farm(i,:)==gene))>=pron(gene)
gene=randint(1,1,[1 m]);
end
farm(i,j)=gene;
end
end
counter=0;% setting iteration counter
while counter ~=M
```

The individuals in the species are matched in pairs here and through the exchange of chromosome of different individuals new individuals are generated. Codes for crossover are listed as below:

```
Newfarm=zeros(N,pn);% new population generated by crossover
    Ser=randperm(N);
 A=zeros(N,pn);
 for i=1:N
       B(i,:)=farm(Ser(i),:);% parents
       A(i,:)= B(i,:);
          cp(i,:)=unidrnd(pn,1,4);% randomly selected intersections
           % two-point crossover of single parent and single offspring
           a(i)=A(i,cp(i,1));
           b(i)=A(i,cp(i,2));
           A(i,cp(i,1))=b(i);
           A(i,cp(i,2))=a(i);
           c(i)=A(i,cp(i,3));
   d(i)=A(i,cp(i,4));
           A(i,cp(i,3))=d(i);
           A(i,cp(i,4))=c(i);
        newfarm(i,:)=A(i,:);% storing of offspring after crossover newfarm
           for j=1:6
oo(i,j)=length(find(newfarm(i,:)==j));
           end
     end
```

4.2 Evaluation results

Enter the parameters below to evaluate the optimization algorithm for resource scheduling in the simulating process:

```
        M=100; % M genetic iterations
        N=50; % N population size (getting even numbers)
        Pc=0.8; % crossover rate
        Pm=0.1; % mutation rate
        P=6; % P number of machine tools
        %Q m×n matrix, to store processing machinery corresponding to m work pieces
and n procedures
        Q=[ 3  1  2  4  3  5 ;
               2  3  5  2  1  4 ;
               3  2  4  1  2  5 ;
               2  1  3  4  5  4 ;
               1  2  5  6  1  4 ;
               2  4  2  1  5  3 ];
        % W m×n matrix, to store processing time corresponding to m work pieces and
n procedures
        W= [  1   3   6   7   3   6;
                8   5 10  10  10   4;
                5   4   8   9   1   7;
                5   5   5   3   8   9;
                9   3   5   4   3   1;
                3   3   9  10   4   1 ];
```

Genetic algorithm simulates the survival of the fittest rule of biological evolution, able to search for and get close to the optimal solution in a wider range with theaveragefitness of thepopulation of each generation inconstantchange, shown in Figure 2 below:

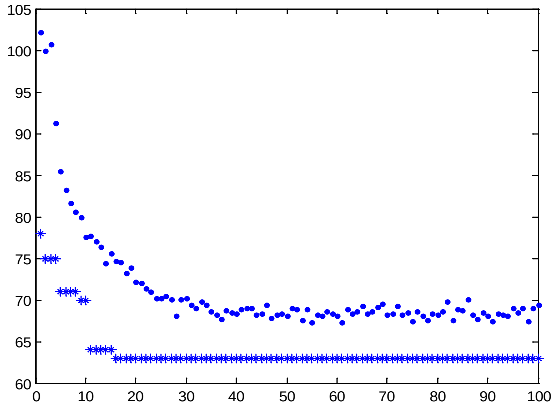

Figure 2. Optimal fitness value of convergence curve

Resulting output after running: optimal scheduling scheme Xp (optimal chromosome); optimal fitness value of convergence curve and curve of average fitness value of population of each generation; Gantt chart of working procedure time based on work pieces; Gantt chart of working procedure time based on machinery; and runtime. Optimal scheduling scheme:

Xp=[4 3 3 5 1 6 1 2 5 3 4 6 2 4 6 3 5 4 5 1 5 3 2 1 6 4 6 1 2 4 2 5 1 2 6 3]

In cloud computing environment, it not only reduces the runtime of resource scheduling program but also achieves the optimal scheduling scheme as well as playing an important role in practical application to adopt chromosome coding method and genetic operators resourcescheduling in resource scheduling based on genetic algorithm [10].

5. Conclusions

In general, this paper has proposed a resource scheduling optimization strategy based on chromosome coding and genetic operators. Firstly, it can be better adapted to resource schedulingcharacteristic of large-scale and sharing in cloud computing environment. Secondly, it can quick find the right virtual machine to assign tasks, thus effectively ensuring user jobs to be finished on schedule. Thirdly, in dealing with resource scheduling as a key issue of cloud computing, traditional genetic algorithms are unable to meet the demand of current scheduling due to their slow convergence and premature as well as constraints on task scheduling that prevent it from being applicable to intensive tasks. Therefore, the optimized genetic algorithm for resource scheduling can play an active role in practical application.

Acknowledgements

This work was supported by GZYLX1213134 (Project Topic of China's Higher Vocational and Technical Education Research Association) and 2012B352 (Planned Project of Education Science of Hubei Province).

References

1. Xu X. Y., Pan Y., Ling C. (2012) Power-aware Resource Scheduling under Cloud Computing Environment. *Journal of Computer Applications*, 32(7), p.p.1913-1915, 1946.
2. Cheng C. L., Pan Y., Zhang D. Y. (2013) Energy Saving Resource Scheduling Algorithm in Cloud Environment. *Systems Engineering and Electronics*, 35(11), p.p.2416-2423.
3. Xu W. Z., Peng Z. P., Zuo J. L. (2015) Research on Cloud Computing Resource Scheduling Strategy based on Genetic Algorithm. *Computer Measurement & Control*, 23(5), p.p.1653-1656.
4. ZhangS. P.,WuH.Y. (2012) Cloud Resource Schedule based on Cellular Automata Genetic Algorithm. *Computer Engineering*, 38(11), p.p.11-13.
5. Liu Y., Zhao Z. W., Li X. L., Kong L. R., Yu S. H., Yun Y. F. (2012) Resource Scheduling Strategy based Optimized Genetic Algorithm in Cloud Computing Environment. *Journal of Beijing Normal University (Natural Science)*, 48(4), p.p.378-384.
6. Chen L., Chen D. L., Gui Y. J., Wu Z. (2014) Cloud Computing Federation Resource Scheduling for Maximizing Users' Benefits. *Journal of Wuhan University of Technology (Information & Management Engineering)*, (3), p.p. 369-373.
7. Li D. C., Wei D. Y., Jin W. P., Liu P. (2012) Research on Resource Scheduling Method of Simulated Platform in Cloud Environment. *Journal of Chinese Computer Systems*, 33(12), p.p.2625-2630.
8. Liu W. N., Jin H. B., Lou B. (2013) Cloud Computing Resource Scheduling based on Improved Quantum Genetic Algorithm. *Journal of Computer Applications*, 33(8), p.p.2151-2153.
9. Zhang J. X., Gu Z. M., Zheng C. (2010) Survey of Research Progress on Cloud Computing. *Application Research of Computers*, 27(2), p.p.429-433.
10. Tian G. Z., Xiao C. B., Zhao J. J. (2014) Evolutionary Algorithm towards Resource Allocation of Concurrent Scheduling Multiple DAGs in Clouds. *Application Research of Computers*, 31(9), p.p.2798-2802.