International Journal of Grid and Distributed Computing
Vol. 9, No. 6 (2016), pp.51-62
http://dx.doi.org/10.14257/ijgdc.2016.9.6.06

Service Cost of Resource Scheduling in Cloud Computing based on an Improved Algorithm Combining Support Vector Machine with Genetic Algorithm

Hongyan Chu

Nanjing College of Information Technology, Nanjing, Jiangsu in China, 210023
chu_hy@njcit.cn

Abstract

It is important to achieve high effective scheduling in cloud computing environment and further guarantee quality of cloud service. In order to do it, fitness function is designed to synthetically reflect completion time and cost of tasks. Based on crossover operation of genetic algorithm and components selection of partial regression, some significant parameters in classification algorithm of support vector machine (SVM) are redefined to strengthen mutual learning capability and sorting ability between each items of subgroup. Meanwhile, it can improve convergence performance of algorithm. Moreover, from the view of results in simulation experiment, the proposed method represents better convergence performance and resource scheduling capability in the different number of resources than artificial neural network (ANN) and genetic algorithm (GA). In the final, according to the proposed method, high effective scheduling will be acquired and service cost may be reduced in cloud computing environment.

Keywords: *genetic algorithm (GA), classification, cloud computing, cost management*

1. Introduction

As we know, there is a wide prospect in the application of cloud computing. However, the quick development of this application depends on key technology of cloud computing. Hereinto, the most key technology is resource management, including the unified management of heterogeneous resources, reasonable scheduling and allocation of resource and so on. Recently, many researchers focused on resource management of cloud computing, which mostly includes several orientations as follows: (1) the research in which the aim of resource sorting and scheduling is to reduce power consumption of data center in cloud computing; (2) the research in which the aim of resource management and scheduling is to improve resource utilization of systems; (3) the research which focus on modeling of cloud resource management based on economics; and (4) other relative researches. The research of resource scheduling and allocation whose aim is to reduce power consumption of data center in cloud computing is a mainstream in existing studies. [1,2] focused on the problem of power consumption of data center in cloud computing and proposed a framework of green cloud computing. Here, optimal location and selection algorithm energy-aware virtual machine are developed, which can reduce power consumption of data center through allocation ways of energy aware.

The nature of resource scheduling in cloud computing is a complex combination and optimization problem. Group intelligence algorithm was introduced to solve this optimization problem of complex function and achieved good performances [3]. Much research has been done. For example, [4] proposed a Double-fitness Genetic Algorithm for the programming framework of cloud computing using completion time of total task and average task as objective fitness function. [5] developed an Chaos Ant Colony Algorithm for resource scheduling in cloud computing, which can improve convergence rate of optimal solution and avoid into partial optimization. [6] proposed an improved artificial firefly algorithm to reduce the average completion time of the requested tasks, solve network load and further achieve a rational resource allocation. Lei and Xiong [7] proposed an effective GA to minimize the expected makespan and the expected total tardiness, and con-firmed that it outperformed the traditional dispatching rules. Jin [8] studied two metaheuristic algorithms with the objective tominimize the makespan based on shop partitioning and simulated annealing for multistage hybrid flow shop scheduling problems, and the proposed approaches had been implemented in a real-life printed circuit board assembly line. Furthermore, theMulti-objective particle swarm optimization (MOPSO) is another class of MOEAs that has been addressed by Coello [9] and Mostaghim [10]. There are other multi-objective optimization algorithms, which include multi-objective simulated annealing (MOSA) by Nam and Park [11], multi-objective ant colony optimization (MOACO) by Garcia-Martinez [12], multi-objective memetic algorithm (MOMA) by Chi-Keong and Yew-Soon [13], *etc*.

Because resource scheduling is a NP-hard problem, under the condition of rapid increasing of solution space, the method which has high complexity of computation time could lead to some limitations of it performance. Therefore, intelligence algorithm has increasingly applied in solving resource scheduling problems, such as Genetic Algorithm (GA) [12-14]. In addition, cloud computing decrease threshold of resource access and collect more source to develop a variety of candidate resource pool. These characteristics greatly increase the space of solution of resource scheduling problem in cloud computing, which may generate a challenge to computation efficiency resource scheduling problem in cloud computing. Therefore, Quantum Genetic Algorithm (QGA) has attracted more attention owing to its wide searching capability of solution space and optimization with probability.

In this paper, we design a fitness function which can synthetically reflect finished time and cost of assignments. Then, based on interlace operation of genetic algorithm and components selection of partial regression, some significant parameters in classification algorithm of support vector machine are defined again to strengthen mutual learning capability and sorting ability between each items of subgroup. Meanwhile, it can improve convergence performance of algorithm. From the results of simulation experiment, compared to artificial neural network (ANN) and genetic algorithm (GA), the proposed method represents better convergence performance and resource scheduling capability in the different number of resources. In addition, high effective scheduling will be acquired and service cost may be reduced in cloud computing environment.

The rest of this paper is organized as follows. In Section 2, some basic concepts of cloud resource scheduling are introduced. Section 3 introduces the model of classification algorithm of SVM. Based on genetic algorithm (GA), Section 4 constructs the proposed model in this paper. A simulation experiment is developed to verify the validity of the proposed model and results are further analyzed in Section 5. Section 6 concludes this paper.

2. Cloud Resource Scheduling

2.1 Scheduling Model

The model of resource scheduling in cloud computing environment can be described by an element set including *n*consumers and *m* resources as follows:

$M = (U, V, F, \theta)$,

where *U* is a consumer set including *n*consumers; *V* is a virtual machine resource set including m virtual machine; *F* is an object function of cloud resource scheduling and optimization; θ is scheduling and optimization algorithm where intelligence algorithm always be used. Only the situation with one consumer who carries out assignments is considered in this paper and several consumers can repeat to carry out assignments[15-17].

Then, the characteristics of scheduling model are demonstrated in the following:

(1) In virtual machine resource set V = $\{v_1, v_2, ..., v_m\}$, each virtual machine is divided according to the number of core, the size of internal storage, and the space of disc. Suppose v_i= $\{\lambda_i, \mu_i, \varphi_i\}$ $i = 1, 2, ..., m$, where i denotes the number of virtual machine and λ_i, μ_i, φ_i denote the number of core, the size of internal storage, and the space of disc respectively.

(2) Meta task set T = $\{t_1, t_2, ..., t_n\}$ denotes n dependent metatask of a consumer.

(3) Task time matrix $S_{n\times m} = (s_{ij})$ $(i = 1, 2, ...n; j = 1, 2, ...m)$ is proposed where s_{ij} denotes the implementing time of task i on resource j. The values of task time matrix are based on performance of assigned virtual machine.

(4) Resource scheduling matrix $E_n = (e_{ij})$ $(i = 1, 2, ...n)$ is developed to denote implementing task i of resource e_i. If E_n =[3 2 2 4 1] denotes implementing task 1 of number 3 virtual machine, implementing task 2 of number 2 virtual machine,…, implementing task 5 of number 1 virtual machine.

(5) Resource using matrix $X_{n\times m} = (x_{je_i})$ $(j = 1, 2, ...n)$ is demonstrated satisfying $e_i \in E_n, x_{je_i} = 1$ and others equal to 0. The generation of $X_{n\times m}$ is based on resource scheduling matrix. Here, x_{je_i} denotes virtual machine e_i is used by task j.

Therefore, as the features of resource scheduling model mentioned above, three hypothesizes are provided in the following:

Hypothesis 1. The performance of virtual machine can satisfy the requirement of each task.

Hypothesis 2. A task can be assigned a virtual machine [8-9].

Hypothesis 3. All tasks can be assigned absolutely [8-9].

Completion time of task in virtual machine vj can be obtained from task time matrix and resource using matrix:

$$T_j = \max_{1\le i\le n}\{s_{ij}, x_{je_i}\}\ (i = 1, 2, ...n; j = 1, 2, ...m). \tag{1}$$

Therefore, the total completion time of implementing task in m virtual machines can be obtained

$$\text{Makespan} = \sum_{j=1}^{m}\max_{1\le i\le n}\{s_{ij}, x_{je_i}\}\ (i = 1, 2, ...n; j = 1, 2, ...m). \tag{2}$$

2.2 Cost of Cloud Service

In cloud computing environment, service provider is concerned with not only completion time of tasks offered by consumer but also the cost of cloud service. Service model Pay-as-you-go of cloud computing requires that service provider quantifies the charge of service used by consumers. In this paper, virtual machine of cloud resource is considered as scheduling unit and task requirement of consumer is assigned and scheduled according to virtual machine instance. Completion time of consumer's task in virtual machine will be different with respect to different configuration of virtual machine. Thus, a good scheduling algorithm is required to reduce the cost of cloud service of service provider and further increase the benefit of service provider. Based on pricing model of virtual machined in Google Compute Engine, the cost p_j in the per unit time of virtual machine of cloud service provider is defined as

$$P_j = p_{i1}\times\alpha_i + p_{i2}\times\beta_i + p_{i3}\times\gamma_i, \tag{3}$$

where p_{i1}, p_{i2}, p_{i3} are the cost of unit core, memory and disc space, respectively [9]. Then, unit time of each virtual machine is obtained by

$$P = \{p_1, p_2, ..., p_m\} \quad (4)$$

With respect to a task of consumer, the cost v_j of virtual machine offered by cloud service provider can be defined as follows

$$P_j = (p_{i1} \times \alpha_i + p_{i2} \times \beta_i + p_{i3} \times \gamma_i) \times \max_{1 \le i \le n} \{s_{ij}, x_{je_i}\} = p_j \times \max_{1 \le i \le n} \{s_{ij}, x_{je_i}\} \quad (5)$$

Here, Eq. (5) is obtained by multiplying Eq. (3) with Eq. (2). Thus, with respect to a task of consumer, the total cost of cloud service provider is

$$P = \sum_{j=1}^{m} p_j \times \max_{1 \le i \le n} \{s_{ij}, x_{je_i}\} \quad (6)$$

Obviously, Eq. (6) is deduced from Eqs. (2) and (5). Therefore, the objective function can be created as min *P*.

3. The Basic Concepts of SVM

3.1 Kernel Function

Definition 1. (Feature Space) Given an initial dataset $S = \{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\}$, where $x_i \subseteq R^n$ and $y_i \subseteq R^m$. A feature mapping is

$$\phi : x \in R^n \to H \subseteq R^n \quad (7)$$

where *H* is feature space, denoting space *Hilbert* .

Definition 2. (Kernel function) Let binary function $K(x, x')$ belonging to $R^n \times R^n$ be a kernel function of $R^n \times R^n$. If there is a mapping from R^n to any feature space H , then it can be obtained that

$$K(x, x') = \langle \phi(x_i), \phi(x_j) \rangle \quad (8)$$

where $\langle \cdot, \cdot \rangle$ denotes inner product of *H*.

Definition 3. (Positive semi-definite matrix) Let binary function $K(x_i, x_j) : R^n \times R^n \to R$ be positive definite. If it is symmetric such as $K(x_i, x_j) = K(x_j, x_i)$, and satisfies

$$\sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) \ge 0, \quad (9)$$

where $m \in l$ (positive integer), $x_1, x_2, ..., x_m \in R^n$ and $\alpha_1, \alpha_2 ..., \alpha_m \in R$.

Definition 4. (*Gram* matrix) Given $K(x, x') : R^n \times R^n \to R$ and $x_1, x_2, ..., x_l \in R^n$. Then, the element in the *i* line of the *j* column constructs a $l \times l$ matrix of $K_{ij} = K(x_i, x_j)$. Therefore, K is a *Gram* matrix of $K(x, x')$ related to $x_1, x_2, ..., x_l$.

3.2 Modelling of SVR

The ideal of SVR is demonstrated as follows. Firstly, suppose $x \in R^n$ and $y \in R$, where R^n denotes output space. Then, through nonlinear transformation, input space of *x* can be mapped into a high dimensional feature space. Finally, in this space, linear function can be used to fit the sample data and guarantee to obtain good generalization ability.

Step 1. In feature space, linear estimation function is defined as

$$y = f(x, \omega) = \omega^T \phi(x) + b \quad (10)$$

Step 2. For nonlinear regression problem, non-negative relaxation variables ξ and ξ^* are introduced to solve the optimization problem which can be showed as

$$\min \frac{1}{2}\|\omega\|^2 + C\cdot\sum_{i=1}^{N}(\xi_i+\xi_i^*) \tag{11}$$

$$\text{S.t.}\begin{cases} y_i-\omega\cdot\phi(x_i)-b\le\varepsilon+\xi_i \\ -y_i+\omega\cdot\phi(x_i)+b\le\varepsilon+\xi_i^* \quad i=1,2,\cdots,N \\ \xi_i>0,\xi_i^*>0,C>0 \end{cases} \tag{12}$$

where penalty factor $C>0$ is a preset constant. The more the value of C is, the more the penalty degree is.

Step 3. In order to deal with this binary optimization problem, lagrange equation where optimal solution is under constraint condition is constructed as follows:

$$\begin{aligned} L(\omega,b,\xi_i,\xi_i^*) &= \frac{1}{2}\|\omega\|^2 + C\cdot\sum_{i=1}^{N}(\xi_i+\xi_i^*) - \sum_{i=1}^{N}a_i\cdot(\varepsilon+\xi_i-y_i+\omega x_i+b) \\ &-\sum_{i=1}^{N}a_i^*\cdot(\varepsilon+\xi_i^*+y_i-\omega x_i-b)-\sum_{i=1}^{N}(\eta_i\xi_i+\eta_i^*\xi_i^*) \end{aligned} \tag{13}$$

$$\text{S.t.}\begin{cases} \dfrac{\partial L}{\partial\omega}=\omega-\sum_{i=1}^{N}a_ix_i+\sum_{i=1}^{N}a_i^*x_i=0 \\ \dfrac{\partial L}{\partial b}=\sum_{i=1}^{N}a_i^*-\sum_{i=1}^{N}a_i=0 \\ \dfrac{\partial L}{\partial\xi_i}=C-a_i-\eta_i=0 \\ \dfrac{\partial L}{\partial\xi_i^*}=C-a_i^*-\eta_i^*=0 \end{cases} \tag{14}$$

Step 4. Based on Eq.(15) andthrough selecting component $\overline{a_k}$ in open interval $(0,C)$, bias term b can be obtained as

$$b=y_k-\sum_{i=1}^{N}(a_i^*-a_i)K(x_i,x_k)-\varepsilon \tag{15}$$

Step 5. Finally, the nonlinear regression function in high dimension space is developed as

$$y=f(x)=\omega\phi(x)+b=\sum_{i=1}^{N}(a_i^*-a_i)K(x_i,x)+b \tag{16}$$

In Eq(16), $K(x_i,x)=(\phi(x_i)\cdot\phi(x))$ is the kernel function, $\phi(x)$ and $\phi(x_i)$ are the image of mapping of the points x and x_i in sample space. The principle of SVR is illustrated in Figure 1.

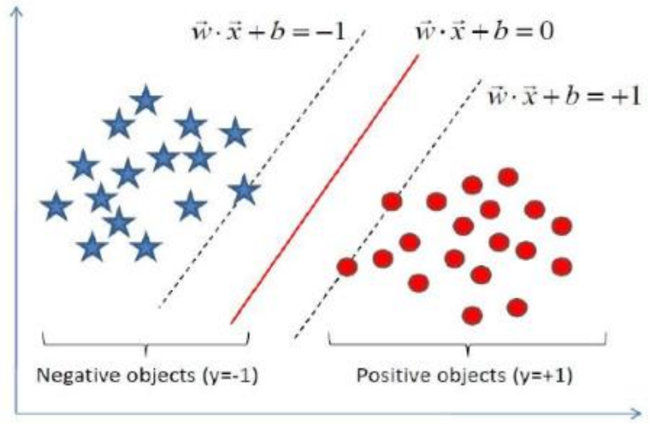
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=0$
$\vec{w}\cdot\vec{x}+b=+1$
Negative objects (y=-1)
Positive objects (y=+1)

Figure 1.Classification of Support Vector Regression

4. The Model of GA- PLS-SVR

The basic ideal of PLS-SVR includes the following content. First of all, the first principal component t_1 and u_1 are respectively extracted from independent variable E_0 and dependent variable F_0. Then, the regression from dependent variable F_0 to t_1 is built. If regression degree has been reached satisfaction, then the algorithm is discontinued. On the contrary, the second principal component will be extracted using residual information after explanation of E_0. This process will be continued until a satisfied precision is obtained where the precision is estimated by testing cross validity. Furthermore, let principal component be input variable by using SVR algorithm. Then, input variable can be mapped into a high dimension space through kernel function. Therefore, after linear processing, a final PLS-SVR regression equation can be obtained.

Step 1. Normalize the data. After normalization of *X*, a data matrix is acquired as $E_0 = [E_{01}, E_{02}, \cdots, E_{0p}]_{n*p}$. As same as *X*, a data matrix $F_0 = [F_{01}, F_{02}, \cdots, F_{0q}]$ is obtained after normalization of *Y*. Suppose t_1 is the first principal component of E_0, w_1 is the first axis of E_0, which is a unit vector as $\|w_1\| = 1$.

Step 2. In step 1, the feature vector w_1 and principal component t_1 in accordance with the biggest feature value of matrix $E_0^T F_0 F_0^T E_0$ can be calculated.

$$\begin{cases} w_h = \dfrac{E_{h-1}^T F_0}{\left\|E_{h-1}^T F_0\right\|} \\ t_h = E_{h-1} w_h \\ p_h = \dfrac{E_{h-1}^T t_h}{\|t_h\|^2} \\ E_h = E_{h-1} - t_h p_h^T \end{cases} \tag{17}$$

Step 3. In *h* steps denoted as $h = (2, 3, \cdots, m)$, E_{h-1} and F_0 are known. Then, it can be obtained that

$$\begin{cases} w_1 = \dfrac{E_0^T F_0}{\left\|E_0^T F_0\right\|} \\ t_1 = E_0 w_1 \\ p_1 = \dfrac{E_0^T t_1}{\|t_1\|^2} \\ E_1 = E_0 - t_1 p_1^T \end{cases} \tag{18}$$

Step 4. Cross validation $Q_h^2 \geq (1 - 0.95^2) = 0.0975$ should be tested to obtain a satisfied input variable, where include *h* principal components $t_1, t_2, \cdots, t_h$.

Step 5. The *h* principal components from step 4 can be applied in the following optimization problem to solve the parameters using quadratic programming method. The objective function can be defined as

$$\min_{\omega,b,\xi} J(\omega, b, \xi) = \frac{1}{2}\omega^T \omega + \frac{1}{2} C \sum_{i=1}^{N} \xi_i^2 \tag{19}$$

$$s.t. \quad u_i = \phi(t_i)\,\omega + b + \xi_i, \quad i = 1, \ldots, N \tag{20}$$

where $\omega \in R^h$ is weight vector, $\phi(\cdot)$ is nonlinear mapping function, $\xi_i \in R^{N\times 1}$ is error vector, $b \in R$ is bias term and $C > 0$ is penalty coefficient.

Step 6. *Lagrange* is introduced to multiply with α_i to construct *Lagrange* function, where $\alpha_i \in R^{N\times 1}$.

$$L(\omega,b,\xi_i,\alpha_i)=\frac{1}{2}\|\omega\|^2+\frac{1}{2}C\sum_{i=1}^{N}\xi_i^2-\sum_{i=1}^{N}\alpha_i\left[\phi(t_i)\omega+b+\xi_i-u_i\right] \tag{21}$$

$$\begin{cases}\dfrac{\partial L}{\partial\omega}=\omega-\sum_{i=1}^{N}\alpha_i\phi(t_i)=0\\ \dfrac{\partial L}{\partial b}=\sum_{i=1}^{N}\alpha_i=0\\ \dfrac{\partial L}{\partial\xi_i}=\alpha_i-C\xi_i=0\\ \dfrac{\partial L}{\partial\alpha_i}=\phi(t_i)+b+\xi_i-u_i=0\end{cases}. \tag{22}$$

Meanwhile, the mentioned equations can be transformed into the form of matrix.

$$\begin{bmatrix}0 & E^T\\ E & \phi\phi^T+C^{-1}I\end{bmatrix}\begin{bmatrix}b\\ \alpha\end{bmatrix}=\begin{bmatrix}0\\ u\end{bmatrix}. \tag{23}$$

Here, E is a $N\times1$ matrix where the all elements equal to 1, I is a $N\times N$ unit matrix, and

$$\alpha=\left[\alpha_1,\alpha_2,...,\alpha_N\right]^T,\ \ u=\left[u_1,u_2,...,u_N\right]^T,\ \ \phi=\left[\phi(t_1)^T,\phi(t_2)^T...,\phi(t_N)^T\right]^T \tag{24}$$

Step7. a model of PLS-SVM regression function is defined as follows.

$$f(t)=\sum_{i=1}^{N}\alpha_iK(t_i,t_j)+b \tag{25}$$

The common kernel function includes linear kernel, polynomial kernel, radial basis kernel and so on. Because radial basis kernel has better generalization ability, it is applied in this paper as follows.

$$K(t_i,t_j)=\phi(t_i)\phi(t_j)=\exp(-\|t_i-t_j\|/2\delta^2) \tag{26}$$

where penalty coefficient C, kernel width δ and the number of principal component t are three important parameters of $LS-SVR$.

Step8. From the computation process of SVR, the different values of ε in non-sensitive loss function, penalty coefficient C and σ^2 in radial basis function will lead to the different support vector regression model. Therefore, in this paper, we select approximate optimization of parameter sets (C,σ^2) based on GA by controlling the value of error ε to construct task scheduling algorithm of GA-PLS-SVM.

As mentioned above, there are two aspects should be completed. On the one hand, a bit string space should be built as $S^L=\{a_1,a_2,\cdots a_K\},a_k=(a_{k1},a_{k2},\cdots,a_{kL})$. Then, $a_{kl}\in\{0,1\}$ makes a bit string of individual from a string of space transform into decoding function $\Omega:\{0,1\}^L\to[u,v]$ in problem parameter space as follows.

$$x_k=\Omega(a_{k1},a_{k2},\cdots,a_{kL})=u+\frac{v-u}{2^L-1}(\sum_{j=1}^{L}a_{kj}2^{L-j}) \tag{27}$$

where binary can be used to code p,q. In this algorithm, five binary code are applied to code p,q, where the length of code is $L=10$. Suppose the former five denote p, the last five denote q, which construct a space S of candidate solution that includes 2^L elements.

On the other hand, operators should be selected. First of all, individual in population can be arranged as decreasing order from big one to small one. Second, a method is selected using roulette. That is, suppose the fitness degree of each individual in population is $f_i(i=1\cdots N)$, then the total fitness degree of population is $\sum_{i=1}^{N}f_i$. Thus, the survival possibility of each individual is $f_i / \sum_{i=1}^{N} f_i$.

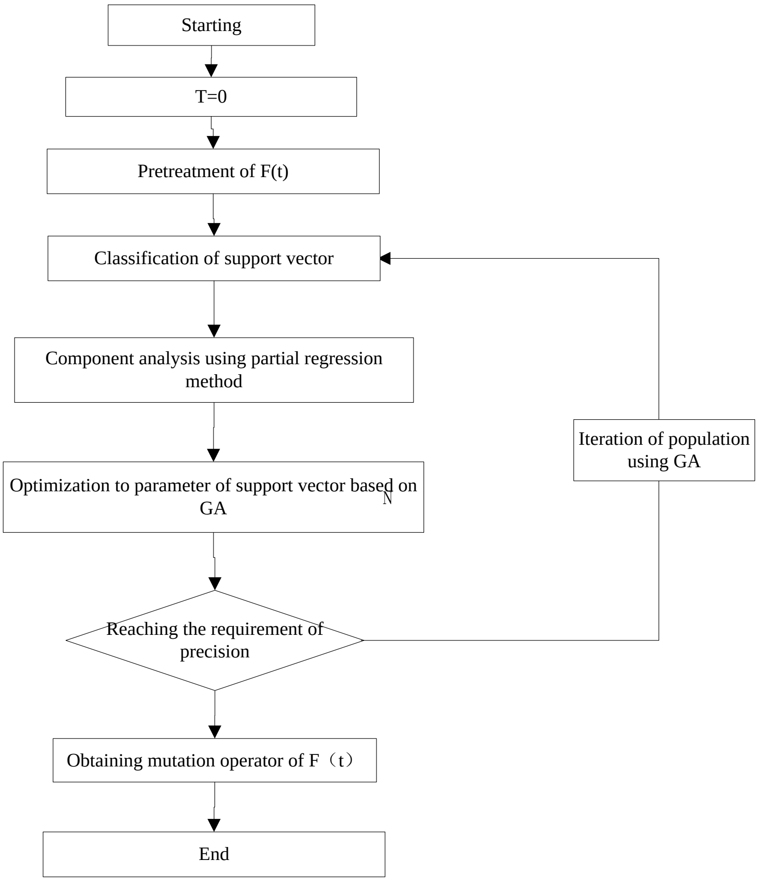
Starting
T=0
Pretreatment of F(t)
Classification of support vector
Component analysis using partial regression method
Optimization to parameter of support vector based on GA
N
Iteration of population using GA
Reaching the requirement of precision
Obtaining mutation operator of F（t）
End

Figure 2. The Process of Improved Algorithm

Two operators are introduced in this section. First of all, a new self-adaptive genetic operator is introduced to build a relationship between cross operator and fitness function $f(x)$ to make cross probability P_c can flexibly change with the fluctuation of fitness. In this situation, P_c satisfies time-variation characteristic to promote flexible of the algorithm.

$$P_c = \begin{cases} 1 - \dfrac{\sum_{i=1}^{N}\left(f_i - \overline{f}\right)^2}{N \max\left(f_i - \overline{f}\right)^2}, & T \le t \\ k_1, & T > t \end{cases} \tag{28}$$

Second, based on basic bit mutation, self-adaptive mutation operator is used to carry out mutation operation for individual coding string according to some genes assigned by mutation probability P_m .The relationship between P_m and fitness function $f(x)$ is constructed as follows.

$$P_m = \begin{cases} k_2\left(1 - \dfrac{\sum_{i=1}^{N}\left(f_i - \overline{f}\right)^2}{N \max\left(f_i - \overline{f}\right)^2}\right), T \le t \\ k_3, \qquad T > t \end{cases} \tag{29}$$

In order to ensure the minimum cost, *P* in Eq. (6) is changed to *f*(*t*) in Eq. (25).

5. Simulation Experiment

In this paper, CloudSim is applied as the imitative experiment platform, which inherits programming model of Gridsim to support construction and simulation of infrastructure in cloud computing. Improved support vector genetic algorithm is realized in CloudSim.

Experiment 1. The population is 5 and the number of iterations is 100. Then, the algorithm in this paper is compared with genetic algorithm(GA) and artificial neural network(ANN).

From the Figure 3, it is obvious that when the number of iterations is less than 20, the minimum cost in the proposed algorithm is slightly bigger than that of GA and ANN. However, along with the increase of the number of iterations, the minimum cost in the proposed algorithm is apparently smaller than that of GA and ANN.

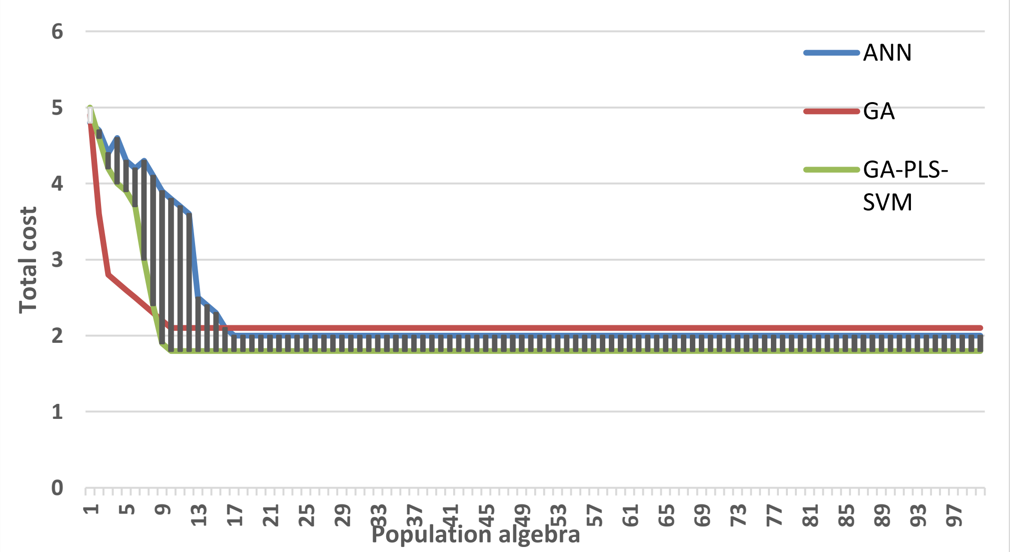

Figure 3. Comparison between each Algorithm when the Number of Iterations is 100

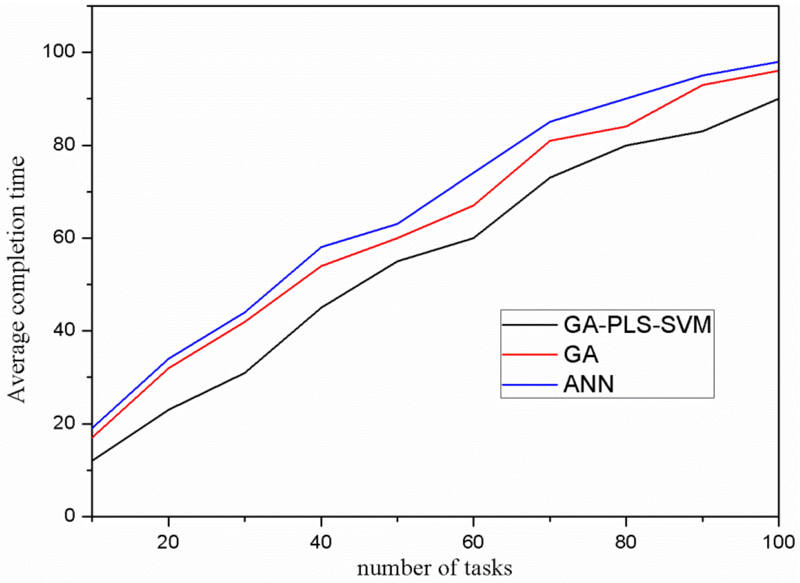

Figure 4. Comparison of Average Completion Time between Three Algorithms under the Different Number of Tasks

Experiment 2. The comparison of resource scheduling performance between three algorithms which have the different number of tasks is demonstrated in this experiment when the number of resources is constant. The number of resources is 30 and the average result is determined by repeating 10 experiments showed in Figure 4.

From Figure 4, with the gradual increasing of the number of tasks, average completion time of task becomes gradually increasing. The completion time of the proposed algorithm is smaller than that of GA and ANN and can keep better performance of resource scheduling.

6. Conclusion

In order to achieve effective resource scheduling in cloud computing and guarantee the quality of cloud service, we design a fitness function which can reflect both completion time and cost of tasks. Based on cross operation of genetic algorithm and components selection of partial regression, some significant parameters in classification algorithm of support vector machine are defined again to strengthen mutual learning capability and classification ability between each items of subgroup and improve convergence performance of algorithm. Furthermore, from the results in simulation experiment, the proposed method represents better convergence performance and resource scheduling capability in the different number of resources than artificial neural network(ANN)and genetic algorithm (GA). In the final, according to the proposed algorithm, high effective scheduling will be obtained and service cost may be reduced in cloud computing environment. In the further, the more applications of the proposed algorithm will be demonstrated to deal with more complex problem in cloud computing context.

References

[1] A. Beloglazav, J. Abawajy, R. Buyya, “Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing[J]”, Future Generation Computer Systems, vol. 28, no. 5, **(2012)**, pp.755—768.

[2] R. Buyya, A. belogiazov, J. Abawajy, “Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision Architectural Elements”, and Open Challenges[C]‖Proceedings of the 2010 Imitational Conference 0n Parallel and Distributed Processing Techniques and Applications(PDPTA2010). Las Vegas,USA, July **(2010)**.

[3] V. Sesum-Cavic, E. Kuhn, “Applying Swarm Intelligence Algorithm for Dynamic Load Balancing to a Cloud Based Call Center [C] “,/ / Proceedings of the 4th IEEE International Conference on Self Adaptive and Self organizing.Systems. [S.l.]: IEEE Press, **(2010)**, pp. 255-256.

[4] J.F. Li, J. Peng, “a task scheduling algorithm based on improved genetic algorithm in cloud computing environment”, Journal of Computer Applications, vol. 31, no. 1, **(2011)**, pp. 184-186.

[5] F. Wang, M.A. Li, W.J. Duan, “Cloud computing task scheduling based on dynamically adaptive ant colony algorithm”, Journal of Computer Applications, vol. 33, no. 11, **(2013)**, pp. 3160-3162.

[6] L. Li, Y. Yao, T. Li, “Study on improved artificial firefly algorithm in cloud computing resources”, Journal of Computer Applications, vol. 30, no. 8, **(2013)**, pp. 2298-2333.

[7] D.M. Lei, H.J. Xiong, “An efficient evolutionary algorithm for multi-objective stochastic job shop scheduling”, in: Proceedings of International Conference on Machine Learning and Cybernetics, Hong Kong, China, **(2007)**, pp. 867–872.

[8] Z. Jin, Z. Yang, T. Ito, “Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem”, International Journal of Production Economics 100, **(2006),** pp. 322–334.

[9] C.A. Coello, G.T. Pulido, M.S. Lechuga, “Handling multiple objectives with particle swarm optimization”, IEEE Transactions on Evolutionary Computation8, no. 3, **(2004),** pp. 256–279.

[10] S. Mostaghim, J. Teich, Covering Pareto-optimal fronts by sub-swarms in multi-objective particle swarm optimization, Evolutionary Computing 2 **(2004)**, pp.1404–1411.

[11] D. Nam, C.H. Park, Multi-objective simulated annealing: a comparative study to evolutionary algorithms, International Journal of Fuzzy Systems 2, no. 2, **(2000)**, pp. 87–97.

[12] G-Martinez, O. Cordon, F. Herrera, “A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP”, European Journal of Operational Research 180, no. 1, **(2007)**, pp. 116–148.

[13] G. Chi-Keong, O. Yew-Soon, T.K. Chen, Multi-Objective Memetic Algorithms, pringer-Verlag, Berlin, ermany, 2009.

[14] Y.M. Xu, K.L Li, J.T. Hu, K.Q. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, Information Science, vol. 270, **(2014)** 255-287.

[15] J.P. Qi, X.F. Cha, “Multi-task and multi-resource optimization scheduling of virus genetic algorithm”, Journal of Computer Applications, vol. 31, no. 7, **(2011)**, pp.1773-1775.

[16] X.Y. Xu, Y. Pan, C. Ling, “Power-aware resource scheduling under cloud computing environment”, Journal of Computer Applications, vol. 32, no. 7, **(2012)**, pp. 1913-1915.

[17] D.W. Sun, G.R Chang, F.Y. Li, “Optimizing Multi-Dimensional QoS Cloud Resource Scheduling by Immune Cloud with Preference”, Acta Electronic Sinica**,** vol. 39, no. 8, **(2011)**, 1824-1831.