

An Optimized Scheduling Algorithm on a Cloud Workflow Using a Discrete Particle Swarm

Cao Jianfang*, Chen Junjie**, Zhao Qingshan*

* Department of Computer Science & Technology, Xinzhou Teachers' University, Xinzhou City,
CO 034000 China

** College of Computer & Software, Taiyuan University of Technology, Taiyuan City, CO 030024
China

Emails: kcxjdj122@126.com

chenjj@tyut.edu.cn

qszhao1226@126.com

Abstract: In order to solve the problems of security threats on workflow scheduling in cloud computing environments, the security of tasks and virtual machine resources are quantified using a cloud model, and the users' satisfaction degree with the security of tasks assigned to the virtual resources is measured through the similarity of the security cloud. On this basis, combined with security, completion time and cost constraints, an optimized cloud workflow scheduling algorithm is proposed using a discrete particle swarm. The particle in the particle swarm indicates a different cloud workflow scheduling scheme. The particle changes its velocity and position using the evolution equation of the standard particle swarm algorithm, which ensures that it is a feasible solution through the feasible solution adjustment strategies. The simulation experiment results show that the algorithm has better comprehensive performance with respect to the security utility, completion time, cost and load balance compared to other similar algorithms.

Keywords: Scheduling algorithm, cloud workflow, discrete particle swarm, evolution equation.

1. Introduction

The cloud workflow, as a new type of an application mode in cloud computing environment, must ensure partial auto execution or execution of the whole business process in the distributed virtual resources according to predefined rules that

develop and deploy the business processes in a cloud computing platform, by providing computing resources and massive storage resources with high performance. The emergence of cloud computing brings convenience to the construction, execution, management and monitoring of the cloud workflow, but its business model of on-demand payment and the unprecedented openness and autonomy bring new challenges for the cloud workflow at the same time: (1) how to complete the implementation of the cloud workflow task using little payment [1]. (2) how to satisfy the security requirements of the cloud workflow task in order to avoid the risk of sensitive data being leaked or tampered in the process of transmission or execution [2].

According to the above problem, this paper puts forward a scheduling algorithm of the cloud workflow using discrete particle swarm optimization, which achieves better performance of the cloud workflow scheduling system in the following aspects: security, completion time, cost and load balancing, etc. This paper is connected with the following: (1) quantize the security of tasks and virtual machine resources in the cloud workflow scheduling using a cloud model, and quantize users' satisfaction degree to tasks assigned to the virtual machine resources, using similarity calculation of the cloud security; (2) establish a scheduling model on the cloud workflow of multi-dimensional QoS (Quality of Service) perception considering security, completion time and cost in the cloud workflow scheduling; (3) propose an optimized scheduling algorithm of the cloud workflow based on a discrete particle swarm to solve the above model.

2. Related works

The problem of hidden security troubles in cloud workflow scheduling has attracted the attention of the traditional grid computing environment. In [3] failure probability is used to quantify the influence on independent tasks when being in the batch processing caused by security and reliability of resources. In [4], aiming at the heterogeneity problem of fault-tolerant mechanism, a job scheduling scheme based on a genetic algorithm is proposed, and a new chromosome coding method is designed to integrate the different fault-tolerant mechanisms. In [5] a workflow scheduling algorithm based on particle swarm optimization is proposed and the problem of workflow scheduling having security constraint conditions is formalized. In [6] a scheduling model of time-safe driving is put forward, and it is solved using the discrete particle swarm optimization algorithm by quantifying the relationship between the task security requirements and the security properties of the resource nodes using security benefits membership functions. The simulation experiments have shown that all these algorithms can achieve better performance in relation to time and safety, but these algorithms do not consider the costs. Therefore, these algorithms cannot be directly used for a cloud workflow scheduling system.

With the advancement of the cloud computing application research, many scholars have carried on thorough research on the cloud workflow scheduling. In [7] a hierarchical scheduling market oriented strategy on a cloud workflow system

is put forward, including scheduling of the service layer and the task layer. The service layer scheduling allocates the appropriate service for the cloud workflow instances, while task scheduling can shorten the completion time and reduce the operation cost. In [8] the scheduling problems of large-scale figure data processing tasks under cloud computing environment are discussed, and the objective function is set up to minimize the scheduling length and the resource rent expense, and a scheduling algorithm based on particle swarm optimization is put forward. The researches consider the completion time and the costs, but ignore the hidden threats towards security in the cloud computing platform. In [9] the user utility in a multi-dimension QoS is quantified, and the objective function of the multi-dimensional QoS optimization is established, and the cloud resource scheduling algorithm based on immune clone optimization is put forward according to user's application preferences. It cannot manifest the characteristics, such as fuzziness and randomness in security, and has the disadvantage that the quantitative strength is too coarse at the same time, using a piecewise function to describe security.

In view of the disadvantages of these algorithms, this paper is trying to use a cloud model to describe its security in the cloud workflow scheduling system. At the same time, a cloud workflow scheduling model and an algorithm of multidimensional QoS perception is proposed in order to improve QoS of the cloud service, considering the completion time and the cost.

3. Formalization description and problem analysis of the cloud workflow

3.1. Cloud computing environment

Definition 1 (cloud computing environment). Assuming that the cloud computing environment is a set of Z data centres $DC = \{dc_1, dc_2, \dots, dc_z\}$, each data centre being composed by a number of computational resources and storage resources.

$BW = \begin{pmatrix} b_{11} & \dots & b_{1Z} \\ \vdots & \ddots & \vdots \\ b_{Z1} & \dots & b_{ZZ} \end{pmatrix}$ denotes the network bandwidth among different data centres.

$\forall i, j = 1, 2, \dots, Z$, b_{ij} is the value of the network bandwidth between dc_i and dc_j , and $b_{ij} = +\infty$ when $i=j$.

3.2. Cloud workflow description

The cloud workflow makes a description of $G = \{V, E, W\}$ by using the Directed acyclic Graph DAG (Directed Acrylic Graph), as shown in Fig. 1. $V = \{v_1, v_2, \dots, v_N\}$ denotes that the cloud workflow is composed by N dependent tasks, $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$ denotes the dependencies between tasks, $(v_i, v_j) \in E$ expresses that v_i is the precursor of v_j ; v_j is the subsequence of v_i . The direct precursor tasks set of task v_i is expressed by $Pre(v_i)$, and the direct successor set is

$\text{Succ}(v_i)$. Only when all $\text{Pre}(v_i)$ have been completed, v_i reaches the ready condition. The entrance task v_{entry} expresses the task of no precursor nodes, the export task v_{exit} expresses the task without any subsequent nodes. For any edge $e = (v_i, v_j) \in E$, $W(e) = w_{i,j}$ is the weight of the edge e , expressing the size of the amount of transporting data from v_i to v_j , using the matrix storage $F_{N \times N}$ to store G :

$$F[i][j] = \begin{cases} w_{ij} & \text{if } v_i \text{ link to } v_j, \\ -1 & \text{else.} \end{cases}$$

The corresponding storage matrix of Fig. 1 is F .

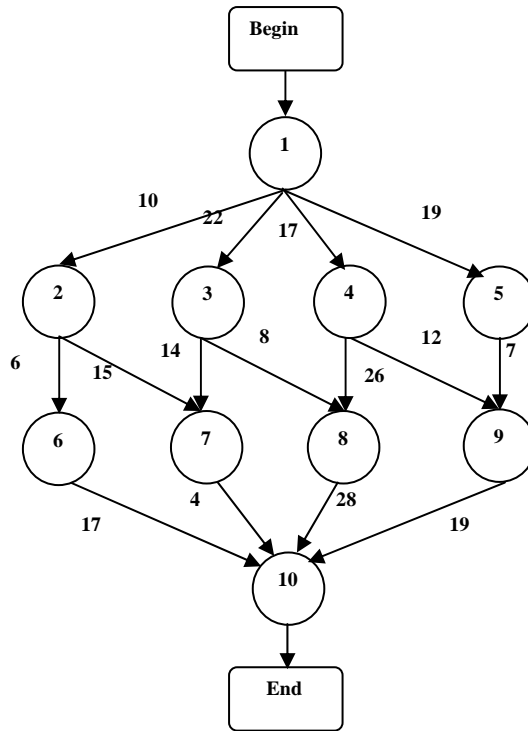


Fig. 1. An example of a cloud workflow

3.3. The quantification of multi-dimension QoS

Assuming that the cloud computing environment can provide M virtual machines, the computing capacity is $P = \{p_1, p_2, \dots, p_M\}$; the cloud workflow consists of N tasks, the calculating length is $L = \{l_1, l_2, \dots, l_N\}$, the essence of the cloud workflow scheduling is to establish a mapping relationship between the N tasks $V = \{v_1, v_2, \dots, v_N\}$ and M virtual machines $M = \{m_1, m_2, \dots, m_M\}$; N tasks and M virtual machines' numbers are respectively: $V_L = \{1, 2, \dots, N\}$ and $M_L = \{1, 2, \dots, M\}$. This paper considers three kinds of QoS attributes: security, completion time and costs.

$$F = \begin{bmatrix} -1 & 10 & 22 & 17 & 19 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 6 & 15 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 14 & 8 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 26 & 12 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 7 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 17 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 4 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 28 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 19 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}.$$

3.3.1. Security

Because the security has the characteristics of randomness and fuzziness, this paper adopts a quantitative conversion model – a cloud model [10] to describe security, which is named a security cloud. The security cloud similarity is the characterization of users' satisfaction by the virtual machine assigned by tasks.

Definition 2 (cloud and cloud droplets) [10]. Assuming that U is a quantitative domain being expressed by a precise numerical, C is the qualitative concept of U . If the quantitative value $x \in U$ is a random realization of the qualitative concept C , and the certainty degree of x to C , that is $\mu(x) \in [0, 1]$ is a random number having a stable tendency $\mu: U \rightarrow [0, 1]$, $\forall x \in U$, $x \rightarrow \mu(x)$, the distribution of x on domain U is called a cloud, each x is called a cloud droplet.

The cloud model uses a forward cloud generator and a backward cloud generator to complete the mutual transformation between the qualitative concept and the quantitative numerical [10]. The forward cloud generator transforms the overall features of the qualitative concept into quantitative values and finishes the transformation of the conceptual space to a numerical space. The backward cloud generator converts a set of quantitative data into a qualitative concept represented by using digital features $\{Ex, En, He\}$, and achieves the transformation from the qualitative concept to a quantitative numerical. The expectations of the cloud droplets in the spatial distribution of the domain are named as Ex ; En denotes the measurement of uncertainty of the qualitative concept; He denotes the uncertainty of En .

Definition 3 (one-dimensional normal cloud) [10]. Assuming that U is a quantitative domain expressed by a precise numerical, C is the qualitative concept of U . The quantitative value $x \in U$ is a random realization of the qualitative concept C . If x meets the following condition: $x \sim N(Ex, En'^2)$, $En' \sim N(En, He^2)$ and

the certainty degree of x to C meets the condition $\mu = e^{\frac{-(x-Ex)^2}{2(En)^2}}$, the distribution of x on domain U is called an one-dimensional normal cloud.

Definition 4 (one-dimensional normal security cloud). Assuming that $\Omega = [a, b]$ is a quantitative domain being expressed by a precise numerical, S is the

qualitative concept of Ω , the quantitative value $x \in \Omega$ is a random realization of the qualitative concept S . If x meets the condition $x \sim N(Ex, En'^2)$, $En' \sim N(En, He^2)$

and the certainty degree of x to S meets the condition $\mu = e^{\frac{-(x-Ex)^2}{2(En)^2}}$, the distribution of x on domain Ω is called an one-dimensional normal security cloud.

Assuming that the security quantitative domain is $[0, 10]$, the corresponding safety level, safety concept, safety range and standard safety cloud are shown in Table 1. The characteristic values of a standard security cloud can be concluded according to Algorithm 1. A standard security cloud generated by a forward cloud generator is shown in Fig. 2.

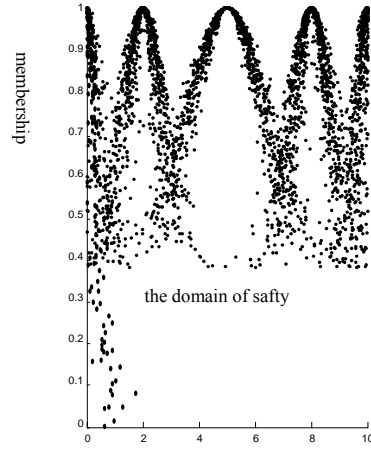


Fig. 2. Illustration of a standard security cloud

Table 1. The security level and the standard security cloud

Safety level	Safety concept	Safety range	Standard security cloud
1	unsafe	[0, 1]	TBC1(0.0, 0.06, 0.5)
2	weak security	[1, 3]	TBC2(0.3, 0.03, 0.5)
3	safe	[3, 7]	TBC3(0.5, 0.03, 0.5)
4	very safe	[7, 9]	TBC4(0.7, 0.03, 0.5)
5	strong safe	[9, 10]	TBC5(0.8, 0.01, 0.5)

Algorithm 1. Standard security cloud generator

Input: A number of security subspaces.

Output: A standard security cloud $SSC_i(Ex_i, En_i, He_i)$.

Step 1. Calculate the expected value Ex_i according to the upper and lower limit value of each interval,

$$Ex_i = \begin{cases} R_i^{\min}, & i = 1, \\ \frac{R_i^{\min} + R_i^{\max}}{2}, & 1 < i < n, \\ R_i^{\max}, & i = n. \end{cases}$$

Step 2. Calculate the entropy En_i according to the upper and lower limits of each interval,

$$En_i = \frac{R_i^{\max} - R_i^{\min}}{3}.$$

Step 3. Calculate the super entropy $He_i = \eta$ (η is a constant) .

The security of the cloud workflow scheduling is quantified by the security cloud. First of all, according to the security requirements of the task, different security levels will be chosen, and the security requirements of the task will be quantified through the corresponding standard security cloud, such as $TSC_i(Ex_i, En_i, He_i)$. Next, according to the history behaviour of the virtual machine (users' experience about the use of cloud services), the security of the virtual machine can be quantified by using a backward cloud generator, such as $VSC_j(Ex_j, En_j, He_j)$. Finally, the appropriate virtual machine can be chosen by using an algorithm of security cloud similarity. The calculation process of the security cloud similarity is listed in Algorithm 2.

Algorithm 2. Calculation algorithm of the security cloud similarity

Input: Safety cloud of the task $TSC_i(Ex_i, En_i, He_i)$, $1 \leq i \leq N$, safety cloud of the virtual machine $VSC_j(Ex_j, En_j, He_j)$, $1 \leq j \leq M$, and the number of cloud droplets K .

Output: Similarity $SimD(TSC_i, VSC_j)$.

Step 1. for $i=1$ to N do

Step 2. for $j=1$ to M do

Step 3. for $k=1$ to K do

Step 4. generate a normal random number $En'_k : En'_k \sim (En_i, He_i^2)$

Step 5. generate a normal random number $x_k : x_k \sim (Ex_i, (En'_k)^2)$

Step 6. generate a normal random number $En'_k : En'_k \sim (En_j, He_j^2)$

Step 7. calculate $y_k = e^{\frac{-(x_k - Ex_j)^2}{2(En'_k)^2}}$

Step 8. end for

Step 9. end for

Step 10. average similarity $SimD(TSC_i, VSC_j) = \frac{1}{K} \sum_{k=1}^K y_k$

Step 11. end for

The accuracy of $SimD(TSC_i, VSC_j)$ is related to the size of cloud droplets, the more the number of cloud droplets is, the more accurately $SimD(TSC_i, VSC_j)$

reflects the similarity between two security clouds. Obviously, greater $\text{SimD}(\text{TSC}_i, \text{VSC}_j)$ suggests that cloud droplets $\text{TSC}_i(\text{Ex}_i, \text{En}_i, \text{He}_i)$ fall in the virtual machine security cloud $\text{VSC}_j(\text{Ex}_j, \text{En}_j, \text{He}_j)$, it gets more closely to the virtual security cloud $\text{VSC}_j(\text{Ex}_j, \text{En}_j, \text{He}_j)$. For tasks v_i it has a priority selection of the virtual machine m_j corresponding to the maximum security cloud similarity and establishes a mapping relationship with them.

3.3.2. Completion time

The time of completing the task v_i includes the time of getting the data (using a parallel manner to access the required data) and the time of executing a mission, which is expressed as

$$(1) \quad T(v_i) = \text{Max}(T(v_k) + C_{v_k, v_i} \mid v_k \in \text{Pre}(v_i)) + T_e(v_i, m_j),$$

C_{v_k, v_i} is the time of the transmitted data from task v_k to task v_i , calculated by formula (2). $T_e(v_i, m_j)$ is the time cost when performing task v_i in the virtual machine m_j , calculated by formula (3);

$$(2) \quad C_{v_k, v_i} = w_{v_k, v_i} / \text{BW}[m_x, m_j],$$

w_{v_k, v_i} is the amount of data transferred from task v_k to the task v_i , $\text{BW}[m_x, m_j]$ is the network bandwidth of the data centre where the virtual machines m_x and m_j respectively carry out task v_k and task v_i . To be sure when the virtual machines m_x and m_j are at the same data centre, $C_{v_k, v_i} = 0$,

$$(3) \quad T_e(v_i, m_j) = l_{v_i} / p_{m_j}.$$

The completion time of the workflow Makespan is the completion time of the export task,

$$(4) \quad \text{Makespan} = T(v_{\text{exit}}).$$

3.3.3. Costs

At present, cloud computing service providers usually take three kinds of a charging method: On-Demand, Reservation and Spot. This paper uses the On-Demand method to charge, buying the computing power by the hour. The cost of executing task v_i includes the transmission costs and the processing costs, calculated by

$$(5) \quad \text{Cost}(v_i) = \sum_{v_k \in \text{Pre}(v_i)} \text{Cost}_t(w_{v_k, v_i}) + \text{Cost}_e(v_i),$$

where the transmission costs can be calculated by formula (6), the processing costs by formula (7),

$$(6) \quad \text{Cost}(w_{v_k, v_i}) = \sum_{v_k \in \text{Pre}(v_i)} w_{v_k, v_i} \text{Price}(m_j, m_k).$$

$\text{Price}(m_j, m_k)$ is the cost of transporting data between the data centres where the two virtual machines are located,

$$(7) \quad \text{Cost}(w_{v_k, v_i}) = \sum_{v_k \in \text{Pre}(v_i)} w_{v_k, v_i} \text{Price}(m_j, m_k).$$

$\text{Price}(m_j)$ is the unit price of using the virtual machine.

The required cost of the cloud workflow is equal to the sum of the costs of all tasks, calculated by

$$(8) \quad \text{Cost}_{\text{Total}} = \sum_{v_i \in V} \text{Cost}(v_i).$$

3.4. The utility function of multi-dimensional QoS perception

In order to illustrate the dimension of the completion time, using cost and security is inconsistent, and on one hand, it needs to be normalized. On the other hand, the completion time, and the price cost belong to cost type attributes, and its value may be as small as possible, while security belongs to quality-benefit type attributes, and its value may be as big as possible. So according to the theory of a multi-attribute utility function, three QoS attributes can be converted into a real value of comprehensive measure by

$$(9) \quad \max \left\{ w_1 \frac{T_{\max} - T'}{T_{\max} - T_{\min}} + w_2 \frac{C_{\max} - C'}{C_{\max} - C_{\min}} + w_3 \frac{S' - S_{\min}}{S_{\max} - S_{\min}} \right\},$$

where T' , C' and S' denote respectively the completion time, the costs and security similarity of the current scheduling scheme; T_{\max} (T_{\min}), C_{\max} (C_{\min}) and S_{\max} (S_{\min}) express respectively the largest (smallest) completion time, the costs and security similarity of the current scheduling solution concentration, $w_k \in R^+ \left(\sum_{k=1}^3 w_k = 1 \right)$ denotes user's preferences.

4. Cloud workflow scheduling algorithm of multi-dimension QoS perception

Taking the ideas of [11], the cloud workflow scheduling problem belonging to discrete optimization can be solved by using the standard PSO (Particle Swarm Optimization, PSO) algorithm namely, to obtain the approximate integral of the particle position, the rest of the algorithm inherits the standard PSO algorithm.

4.1. Particle coding and update mode

Considering the discrete space characteristics of the cloud workflow scheduling problem, the way of encoding the assigned task to the virtual machine is $X_i^t = (x_{i1}^t, \dots, x_{ij}^t, \dots, x_{iN}^t)$, $x_{ij}^t \in M_L$, x_{ij}^t expresses the virtual machine number assigned by the j -th place of the i -th particles at the t -th iteration; N denotes the dimension of the particles, namely the number of tasks of the cloud workflow; $V_i^t = (v_{i1}^t, \dots, v_{ij}^t, \dots, v_{iN}^t)$, $v_{ij}^t \in M_L$, denotes the speed of i -th particles at the t -th

iteration, v_{ij}^t expresses the moving distance of the j -th place of the i -th particle at the t -th iteration. The following is the particle' update formula:

$$(10) \quad V_i^{t+1} = \omega V_i^t + c_1 r_1 (pbest_i^t - X^t) + c_2 r_2 (gbest - X^t),$$

$$(11) \quad X_i^{t+1} = X_i^t + V_i^{t+1}.$$

Among them, t is the number of iterations, ω is the inertia weight, c_1 is a cognitive coefficient, c_2 is a society coefficient, $pbest_i = (pbest_{i1}, \dots, pbest_{ij}, \dots, pbest_{iN})$ is the individual optimal solution, the global optimal solution is expressed as $gbest = (gbest_1, \dots, gbest_j, \dots, gbest_N)$, r_1 and r_2 are the random numbers of $[0, 1]$.

4.2. The adjustment strategy of the feasible solution

Assuming that the initial position of particle i is $X_i^0 = (x_{i1}^0, \dots, x_{ij}^0, \dots, x_{iN}^0)$, and is a feasible solution, namely $\forall j, x_{ij}^1 \in M_L$, after an iteration, the updated position of the particle i is $X_i^t = (x_{i1}^t, \dots, x_{ij}^t, \dots, x_{iN}^t)$, and $\exists j, x_{ij}^t \notin M_L$. At this point, the adjustment strategy of the feasible solution needs to be designed to ensure that the location code of the particle is still a feasible solution after t -th iteration, namely, $\forall j, x_{ij}^t \in M_L$. Therefore, firstly, the particle' number Num must be counted, which does not meet the number of the feasible solution, and its location must be recorded; Then, for $x_{ij}^t \notin M_L$, $(x_{ij}^t)' \in M_L$, is randomly generated from M_L by equal probability and it replaces the corresponding positions x_{ij}^t , executed Num times continuously; finally, the new particles $(X_i^t)' = ((x_{i1}^t)', \dots, (x_{ij}^t)', \dots, (x_{iN}^t)')$ can be obtained, and it becomes a feasible solution.

4.3. Cloud workflow scheduling algorithm based on discrete particle swarm optimization

Algorithm 3. CWDPSO algorithm (Cloud Workflow based on Discrete Particle Swarm Optimization)

Input: Storage matrix of the cloud workflow F , calculated length L_i of the task $v_i \in V$, the requirements grade of security S_i , the computing power p_j and unit time cost $Price(m_j)$ of the virtual machine $m_j \in M$, security cloud of virtual machines $VSC_j(Ex_j, En_j, He_j)$, network bandwidth of the data centres BW, required cost for data transmission $Price(m_j, m_k)$, the population size of the particle swarm P , the biggest evolution algebra T .

Output: Scheduling scheme.

Step 1. Initialize the position and speed of particles in the particle swarm, produce the initial group $pop(t)$, $t = 0$, which size is P

Step 2. for $t=1$ to T do

Step 3. for $p=1$ to P do

Step 4. adjust the particle's position using adjustment strategy of the feasible solution

Step 5. calculate each particle's completion time, cost and security using formulas (4), (8) and algorithm 2

Step 6. end for

Step 7. construct the contemporary decision matrix $D_t^p = [d_{ij}]_{P \times 3}$, x_{ij} denotes the value of the i -th particle on the j -th principle

Step 8. assuming that $D_p^t = (d_{p1}^t, d_{p2}^t, \dots, d_{pP}^t)^T$, determine the maximum value $\text{Max}_j = \max\{d_{ij} \mid i \in [1, P], j \in [1, 3]\}$ and the minimum value $\text{Min}_j = \min\{d_{ij} \mid i \in [1, P], j \in [1, 3]\}$ of each column in the contemporary decision matrix D_p^t

Step 9. according to formula (9), standardize the contemporary decision matrix D_p^t with weight, being expressed as $\bar{D} = [\bar{d}_{ij}]_{P \times 3}$, calculate the contemporary individual optimal solution pbest_p^t and the global optimal solution gbest^t , and then store pbest_p^t in a global decision matrix D^g

Step 10. according to pbest_p^t and gbest^t , judge whether to update the individual optimal solution pbest_i and the global optimal solutions gbest

Step 11. update the particle's speed and position using formulas (10) and (11)

Step 12. end for

Step 13. normalize the global decision matrix D^g using formula (9), and then choose the best individual as the scheduling scheme

The algorithm is divided into three stages: initialization, execution and end. At the initialization phase (line 1), the initial population whose size is P , is randomly generated according to the cloud workflow coding scheme; at the execution phase (line 2-11), local adjustment of the particle position can be made to ensure that it is a feasible solution, and three kinds of QoS attribute values of the particles can be calculated on line 3~6; a decision matrix is constructed according to the QoS attribute values, combined with the users' preferences, the objective function of multi QoS perception is aggregated into a single objective function, making possible that the particles are selected by numerical comparison among each other, in order to find out the individual optimal values and the global optimal values on line 7~9; new evolution groups are produced by using the particle evolution equation to update the particle's speed and position on line 10; at the end phase (line 12), the optimal solution of the cloud workflow scheduling scheme is output. Since the cloud workflow contains N tasks and M virtual machines, the required time of the particles is $O(NM)$ in each iteration process; and because the population size is set to be P , the number of iterations is set T , the time complexity of the CWDPSPSO algorithm is as follows: $T \times P \times O(NM)$.

5. The simulation experiments

In order to validate the CWDPSO algorithm, a comprehensive comparative analysis with LGPPSO algorithm (Large Graph Processing-based on Particle Swarm Optimization, LGPPSO) proposed in [8], will be done under different simulation parameters and performance indices.

5.1. Experimental environment and parameter setting

The simulation experimental platform is based on Cloudsim [12] and the development tool is Myeclipse 8.5. The data used in the experiments are randomly generated. The cloud workflow contains an interval of the tasks number [5, 500], interval of the length of the task [100, 200], interval of safe level demand [1, 5], interval of the amount of transferred data between tasks [1, 30], interval of the number of virtual machines [5, 100], and interval of ability to calculate [10, 20], interval of the calculating price of a unit time [1, 3], interval of the expectations of the virtual machine security cloud [0, 10], interval of entropy [0, 2], interval of hyper entropy [0, 1]; interval of the network bandwidth of the data centres [1, 9], the costs per unit of data transmission is 0.01. Set the parameters of DPSO according to reference [13]: the inertia weight ω is 0.6, the cognitive coefficient c_1 is 2.3, the social factor c_2 is 1.7, the population size of P is 50.

5.2. Experimental results and performance analysis

This section will accomplish the comparative analysis between CWDPSO algorithm and LGPPSO algorithm with respect to four performance indicators: the security utility value, completion time, cost and load balance deviation. The security utility value is used to measure the users' satisfaction by the security of the virtual

machine assigned by tasks, calculated as $SecU = \sum_{i=1}^N SimD(TSC_i, VSC_j)$, the higher

its value is, the higher the users' satisfaction by the security of the virtual machine assigned by tasks is. The completion time refers to the total time spent from the beginning of executing the workflow's first task to completion of the last task, calculated by formula (4). The cost of using includes the transport cost and the computation cost, calculated by formula (5). The shorter the completion time is, the smaller the cost of using is, the higher the QoS of the cloud is. Load balancing deviation is used to react to fair utilization efficiency achieved by its own abilities of virtual machine resources, its value can be obtained by

$$(12) \quad \sigma = \sqrt{\frac{\sum_{j=1}^m (LB_j - \overline{LB_j})^2}{m-1}}$$

where $\overline{LB_j}$ is the average of the load balancing factor LB_j , the load balancing factor LB_j is calculated by

$$(13) \quad LB_j = \frac{\text{Cost}(i, j)}{\sum_{i=1}^N \sum_{j=1}^M \text{Cost}(i, j)} / \frac{l_i}{\sum_{i=1}^N l_i}.$$

$\text{Cost}(i, j)$ is the fee of performing tasks v_i on virtual machine m_j , l_i is the calculating length of task v_i .

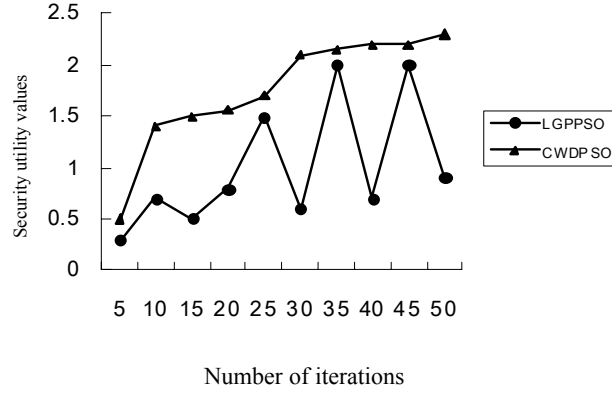


Fig. 3. Comparison of the security utility value

As it can be seen from Fig. 3, with the increase of the number of iterations, the security utility value of CWDPSO algorithm has steadily grown, while the security utility value of LGPPSO algorithm is volatile. This is because this paper considers the completion time, the cost and safety at the same time when constructing the objective function, which makes the algorithm have a priority selection of the virtual machine meeting the task security requirements in the process of iteration. At the same time LGPPSO algorithm only considers the completion time and the cost, it has some blindness and randomness in the aspect of whether the virtual machine meets the task security requirements or not, which leads to low safety utility value and great volatility.

As can be seen from Figs 4 and 5, with the increase of the number of iterations, CWDPSO algorithm's completion time and the cost are lower than that of LGPPSO algorithm. This is because the objective function designed by LGPPSO algorithm is 0 when the completion time and the cost of a particle gets the maximum value at the same time, and it is easy to fall into a local optimum, and hence it is hard to find the global optimal solution. Aiming at this problem, the best individual of every generation is preserved by the global decision matrix, then the global decision matrix carries on normalized processing and selects the largest individual of the utility function as a cloud workflow scheduling scheme among them, thus the problems of the existing prematurity is efficiently avoided in the LGPPSO algorithm proposed in the paper.

As can be seen from Fig. 6, with the increase of the number of tasks, the load balance deviation of CWDPSO algorithm and LGPPSO algorithm are both increasing, but the increasing speed of CWDPSO algorithm is less than LGPPSO algorithm. This is because CWDPSO algorithm chooses virtual machines most

similar to task security requirements by the calculation method of security similarity. In this way, the fast calculation speed and the heavy load of virtual machine resources whose computational cost is low can be efficiently avoided, ensuring task security requirements at the same time.

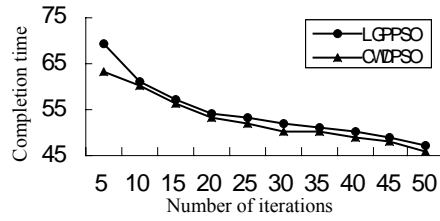


Fig. 4. Comparison of the completion time

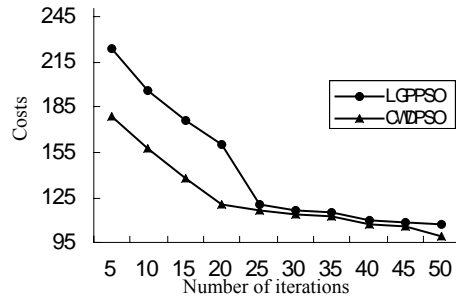


Fig. 5. Comparison of the cost

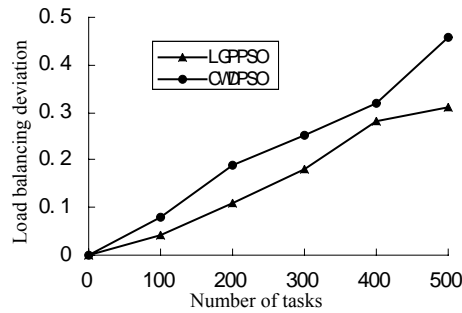


Fig. 6. Comparison of the load balancing time

6. Conclusions

Based on comprehensive analysis of the cloud workflow scheduling problem, this paper uses the cloud model to quantify the security of the task and virtual machine resources and the security cloud similarity to show the user's security satisfaction. Besides, a cloud workflow scheduling model merged with security is built. On this

basis, the objective function with minimized completion time and minimized cost and maximized security satisfaction is established, and the cloud workflow scheduling algorithm of multi-dimension QoS perception based on a discrete particle swarm is put forward. The experimental results have shown that, compared with other similar algorithms, the algorithm has better performance with respect to the completion time, the cost of using and security satisfaction, thus the feasibility and validity of this algorithm can be validated. The way to build a dynamic multi-target cloud workflow scheduling model to depict the virtual machine resources price's situation along with the dynamic change of the supply and demand is the main research work of the next step.

Acknowledgement: This work was supported by the National Natural Science Foundation of China under Grant No 61202163 and by the Natural Science Foundation of Shanxi Province under Grant No 2013011017-2 and by the Technology Innovation Project of Shanxi Province under Grant No 2013150 and by the Key disciplines supported by Xinzhou Teachers University under Grant No XK201308. The authors are grateful for the constructive and valuable comments made by the expert reviewers.

References

1. Shang, S. F., J. L. Jiang, W. M. Zheng. CWFlow: A Cloud-Based Workflow Framework with Adaptive Resource Utilization. – J. Tsinghua Univ (Sci & Tech), Vol. **53**, 2013, No 3, 415-420.
2. Chakrabarti, A., A. Damodaran, S. Sengupta. Grid Computing Security: A Taxonomy. –IEEE Security & Privacy, Vol. **6**, 2008, No 1, 44-51.
3. Kołodziej, J., F. Xhafa. Integration of Task Abortion and Security Requirements in GA-Based Meta-Heuristics for Independent Batch Grid Scheduling. – Computers and Mathematics with Applications, Vol. **63**, 2012, No 2, 350-364.
4. Wu, C., R. Sun. An Integrated Security-Aware Job Scheduling Strategy for Large-Scale Computational Grids. – Future Generation Computer Systems, Vol. **26**, 2010, No 2, 198-206.
5. Liu, H., A. Abraham, V. Snášel et al. Swarm Scheduling Approaches for Work-Flow Applications with Security Constraints in Distributed Data-Intensive Computing Environments. –Information Sciences, Vol. **192**, 2012, No 6, 228-243.
6. Zhu, H., Y. P. Wang. Integration of Security Grid Dependent Tasks Scheduling Double-Objective Optimization Model and Algorithm. – Journal of Software, Vol. **22**, 2011, No 11, 2729-2748.
7. Wu, Z., X. Liu, Z. Ni et al. A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems. – J. Supercomput., Vol. **63**, 2013, No 1, 256-293.
8. Li, J., Q. J. Huang, Y. Y. Liu et al. A Task Scheduling Algorithm for Large Graph Processing Based on Particle Swarm Optimization in Cloud Computing. – Journal of Xi'an Jiaotong University, Vol. **46**, 2012, No 12, 116-122.
9. Sun, D. W., G. R. Chang, F. Y. Li et al. Optimizing Multi-Dimensional QoS Cloud Resource Scheduling by Immune Clonal with Preference. – Acta Electronica Sinica, Vol. **39**, 2011, No 8, 1824-1831.
10. Li, D., C. Liu. A New Cognitive Model: Cloud Model. – Int. J. of Intelligent Systems, Vol. **24**, 2009, No 3, 357-375.
11. Salaman, A., I. Ahmad, S. Al-Madani. Particle Swarm Optimization for Task Assignment Problem. – Microprocessors and Microsystems, Vol. **26**, 2002, No 8, 363-371.
12. Calheiros, R. N., R. Ranjan, A. Beloglazov, et al. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. – Software: Practice and Experience, Vol. **41**, 2011, No 1, 23-50.
13. Parsopoulos, K. E., M. N. Vrahatis. Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization. – Natural Computing, Vol. **1**, 2002, No 2-3, 235-306.