

# Multi-Dimensional Constrained Cloud Computing Task Scheduling Mechanism Based on Genetic Algorithm

<http://dx.doi.org/10.3991/ijoe.v9iS6.2793>

Youchan Zhu, Peng Liu

North China Electric Power University, Baoding, China;

**Abstract**—A new Cloud Computing task scheduling mechanism with multi-dimensional constrained based on Genetic Algorithm (GA) was proposed to reduce time to complete tasks and to improve users' satisfaction by reducing expenses. The algorithm improves fitness function with comprehensive efficiency constituted of shortest completion time and minimum cost. Finally, simulation results showed that this algorithm is highly efficient and meet not only performance needs but also economic needs of users.

**Index Terms**—Task Scheduling, Genetic Algorithm, Multidimensional Constraints, Cloud Computing

## I. INTRODUCTION

Currently IT resources were used as services in Cloud Computing to provide to user through high-speed Internet. Users could dynamic access to acquire resources what their need. As a new computing model, tasks in Cloud Computing would be distributed across hardware resources pool constituted by IaaS (Infrastructure as a Service), and would dynamic gain computing, storage, networking and software resources on demand. Because of hyper-scale, on-demand service, virtualization, high reliability, high availability, low price and others, Cloud Computing is attracting more and more people.

As an important part of Cloud Computing environment, the core function of task scheduling[1], in essence, is assigning tasks to resource nodes that were selected for meeting user's demand and some maximize evaluation standards based on resource nodes running status. However, because of heterogeneity and dynamics of Cloud Computing environment, traditional task scheduling algorithm simply to improve system throughput or shorten task completion time does not applied in Cloud Computing efficiently. At the same time, the dynamic demand of users often results in uneven loading of Cloud Computing environment and other issues. So the traditional task scheduling algorithm is not effective to meet demand of performance and user's satisfaction. Therefore, people proposed a lot of different algorithms to resolve combinational optimization problem effectively, such as Immune Evolutionary Algorithm[2], GA [3-4], Euclidean Distance Liked-Load Balance

Scheduling Algorithm[5], Hybrid Particle Swarm Algorithm[6]and Time-shared, Space-shared allocation algorithm, etc. Those algorithms had made good progress obviously.

GA with using shortening task completion time as the only evaluation criteria would not meet requirements of entire Cloud Computing load balancing and user's satisfaction when it was applied to Cloud Computing task scheduling process. Especially it was very difficult to maintain the demand balance between performance and economy. This paper putted forward a multi-dimensional constraints task scheduling algorithm based on GA to improve the task accomplished comprehensive benefit(TACB), which is consisted of performance benefit and economy benefit. Taking both task completion time and the economic needs of user into account would improve efficiency of task scheduling in Cloud Computing environment.

## II. CLOUD COMPUTING TASK SCHEDULING MODEL

### A. Cloud Computing Task- Resource Model

In this paper, defined one set  $T = \{T_1, T_2, T_3, \dots, T_n\}$  as the set of user tasks, which is consisted of  $n$  independent tasks, and each task has a quintuple form description:

$$T_i = \langle \text{ID}, \text{LENGTH}, \text{INSIZE}, \text{OUTSIZE}, \text{PERF} \rangle$$

Where ID represents the task identifier, LENGTH is task length, INSIZE is the input file size of Map-Reduce operation, OUTSIZE is the output file size of Map-Reduce operation, and PERF represents the TACB value.

$R = \{R_1, R_2, R_3, \dots, R_m\}$  is a set of resource nodes consisted of  $m$  heterogeneous resources, and each resource node could be expressed with a septet form:

$$R_i = \langle \text{ID}, \text{pesNUMBER}, \text{MIPS}, \text{RAM}, \text{SIZE}, \text{BW}, \text{COST} \rangle$$

In this septet, ID is the resource node identifier, pesNUMBER, MIPS, RAM, SIZE, BW represent respectively CPU numbers, CPU processing performance,

memory sizes, file system sizes and bandwidth in each resource node, and COST is total expenses of each node.

### B. Multi-dimensional Constraints Description

Here, defined the task expectation time of completion matrix with ETC.

$$ETC = \begin{bmatrix} et_{11}, & \dots, & et_{1m} \\ \dots, & \dots, & \dots \\ et_{n1}, & \dots, & et_{nm} \end{bmatrix}$$

In this matrix,  $et_{ij}$  is the expectation completion time that task  $i$  is executed on resource node  $j$ .

Matrix AL represents resource-task allocation matrix [7], which record allocation situation of resource nodes and tasks.

$$AL = \begin{bmatrix} al_{11}, & \dots, & al_{1m} \\ \dots, & \dots, & \dots \\ al_{n1}, & \dots, & al_{nm} \end{bmatrix}$$

In this matrix,  $al_{ij} = 1$  means that task  $i$  was assigned to resource node  $j$ , and  $al_{ij} = 0$  means not assigning.

In order to meet performance and economy needs, task scheduling used TACB as the evaluation criteria. Performance benefit referred to the task completion time, and economy benefit used task execution expenses as reference. Using evaluation criteria with TACB would avoid high costs because of excessive pursuit for high performance, or low costs resulting in low performance. On basis of maintaining performance requirements of users, GA with multi-dimensional constrained would reduce as few expenses as possible.

Defined Performance Benefits Evaluation Function(PBEF):

$$TimePer(i, j) = \frac{k}{\sum_{j=1}^M \sum_{i=1}^N et_{ij} \times al_{ij}} \quad (1)$$

$et_{ij}$  is an element of ETC matrix, and  $al_{ij}$  is an element of AL matrix.  $et_{ij}$  has same position in ETC with  $al_{ij}$  in AL.  $k$  is a coefficient of reference. The PBEF indicates that different way to assign tasks would have different completion time. The shorter task completion time is, the higher performance benefits are. On the contrary, the assigning way has low performance benefits if it has long executing time.

The economic benefits are primarily based on the costs of task scheduling. The evaluation function is:

$$CostPer(i, j) = \frac{k}{PE_{cpu/num} + PE_{mem/MB} + PE_{stor/MB} + PE_{bw/Mbps}} \quad (2)$$

$PE_{cpu/num} + PE_{mem/MB} + PE_{stor/MB} + PE_{bw/Mbps}$  is the total expenses of all virtual resources used by task scheduling.  $k$  is the coefficient of reference.  $P$  is the number of tasks,  $E_{cpu/num}$ ,  $E_{mem/MB}$ ,  $E_{stor/MB}$ ,  $E_{bw/Mbps}$

represent the unit price of CPU, memory, storage, and bandwidth severally.

In summary, TACB of Cloud Computing was consisted of performance benefits and economic benefits to evaluate the efficiency of task scheduling:

$$Perf(i, j) = \alpha \times TimePer(i, j) + \beta \times CostPer(i, j) \quad (3)$$

$\alpha + \beta = 1$ ,  $\alpha, \beta \in [0,1]$ , where  $\alpha$  and  $\beta$  represent respectively user's preference of task completion time or costs.  $\alpha$  is larger when users are more interesting in the completion time. On the contrary,  $\beta$  being larger indicates that lower costs are more popular.

### III. THE TASK SCHEDULING MECHANISM WITH GA

GA was first proposed in 1975 by Professor John Holland, and its two most notable features are parallelism and global solution space search. It is an effective parallel global search algorithm based on the theory of natural evolution[8] and genetic variation to solve global optimal solution in nature. According to specific issues of Cloud Computing environment, GA would construct a fitness standard to evaluate genetic population, which are consisted of multiple-solutions (each solution mappings of a chromosome), and execute selection, crossover, variation operation through multi-generation reproduction to find best individual as the optimal solution. Its implementation process is shown in Fig.1.

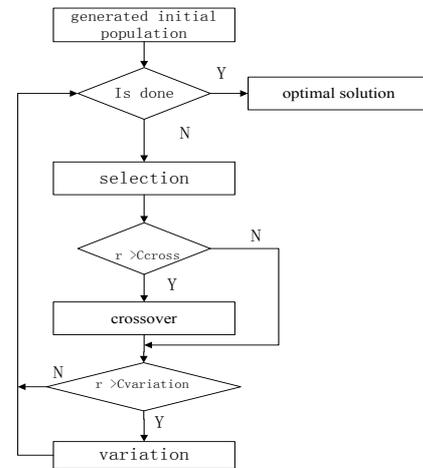


Figure 1. GA operation process

#### A. The GA Encoding

The initialization of populations is random in GA, and its specific initialization method relies on encoding. GA should confirm encoding of chromosome to ensure that each chromosome could map a solution of the optimization problem. Encoding of chromosome is variable, and is based on the specific usage scenarios of problem. People can use task execution status directly as codes, and also the indirect encoding according to the typical characteristics of Cloud Computing environment. In paper, GA encoding used indirect pattern, and each code's length is equal to the number of tasks. The value of a gene in each chromosome is in  $\{0,1,2, \dots, M-1\}$ , and

M is the number of resource nodes. For example, there are five tasks numbered 0, 1, 2, 3, 4, and four resource nodes 0, 1, 2, 3 in resource pool.  $N = 5$ ,  $M = 4$ . Three chromosomes:  $\{1,2,0,2,3\}$ ,  $\{2,1,0,2,3\}$ ,  $\{3,0,3,2,1\}$ . Decoding would confirm how many tasks each resource node could execute and those task ID. Just like  $\{1,2,0,2,3\}$ , decoding result is:  $R0 = \{2\}$ ,  $R1 = \{0\}$ ,  $R2 = \{1,3\}$ ,  $R3 = \{4\}$ . Those show that task 0 was assigned to the resource node 1, task 1 and task 3 were assigned to the node 2, task 2 was assigned to the node 0, and task 4 was assigned to node 3. When got all decoding results, we could determine the value of TACB based on fitness function.

## B. GA Operation Processes

### 1) Fitness Function Selection

The fitness function of GA is very important and would relate to the convergence speed of GA and the effectiveness of solution. According to the comprehensive consideration of user's performance and economic needs, GA used TACB as the fitness evaluation standards in Cloud Computing environment:

$$f(x) = \alpha \times \text{TimePer}(i, j) + \beta \times \text{CostPer}(i, j) \quad (4)$$

$\alpha + \beta = 1$ ,  $\alpha, \beta \in [0,1]$ .  $\alpha$  and  $\beta$  represent user preference of task completion time and expense.  $\text{TimePer}(i, j)$ ,  $\text{CostPer}(i, j)$  were got respectively from function (1) and function (2). This paper focuses on the effectiveness of algorithm compared with other existing algorithms, so it chose the no significant demand preference to simulate. Assumed that:  $\alpha = \beta = 0.5$ . Whether or not different demand preference affecting the efficiency of algorithm is not considered here.

### 2) Selection Operation

Selection operation with Roulette-Selection is used to determine how to select individuals from parent populations to inherit in next generations for determining reorganization or cross-individual. According to the proportion of each chromosome's fitness value in current population, GA selected the chromosome with largest value to execute crossover operation and mutation operation. The probability of the individual selection is:

$$P(i) = \frac{f(i)}{\sum_{j=1}^{SCALE} f(j)} \quad (5)$$

$f(i)$  is the TACB value of chromosome  $i$ .  $\sum_{j=1}^{SCALE} f(j)$  is the sum of all chromosomes TACB values.  $SCALE$  is the population size. The individual would have more possible to generate with larger TACB value.

### 3) Crossover Operation and Variation Operation

Crossover operation would select two chromosomes to reform new chromosomes with interchanging their part of genes according to the crossover probability, and it's the main method to generate new individuals. This paper chose single point crossover, which is simple and

effective, as the crossover method. Variation operation would reform new individuals by changing some genes in chromosomes with variation probability, and it's the helper method to generate new individuals. It determines the local search ability of GA, and maintains the diversity of population. In GA, Crossover operation and variation operation cooperate with each other to support for completion global and local search on solution search space. We used constant  $C_{cross}$ ,  $C_{variation}$  to represent the crossover probability and mutation probability. Variable  $r$  was acquired from Random function to confirm whether execute crossover operation or variation operation or not.  $r \in [0,1]$ . If  $r > C_{cross}$ , execute crossover operation, or not; If  $r > C_{variation}$ , execute variation operation, or not.

### (4) Catastrophe Optimal Operation

To further improve global search ability and avoid emerging local optimal solution in the process of evolution, GA introduced catastrophe theory, which is natural evolutionary process. When external environment changes tremendous, the vast majority of creatures would disappear, and individual species would survive and have adequate room for evolution. In this paper, GA used catastrophe operation as a way to avoid producing local optimal solution. It eliminated all current outstanding individuals at one point and permitted some individuals away from the current extremes value point to have room for full evolution to avoid falling into local optimal solution. The disaster counter  $COUNT$  was used to control the time of catastrophe operation in this algorithm. Here assumed:  $COUNT = 300$ . If there was not produce optimal solution in 300 generations, it would execute catastrophe operation, and stop to execute catastrophe operation without the optimal solution changing after several times.

## IV. SIMULATION

This simulation was implemented on the CloudSim [9,10] platform to simulate and implementation task scheduling algorithm. This simulation software of Cloud Computing was developed by the Melbourne University of Australia.

5 virtual datacenters were created with 200 physical hosts in the Cloud Computing simulation environment. Allowing for specific application of Cloud Computing environment, those physical hosts were divided into three levels of price according to different performance. Using C1, C2, C3 as unit prices of each virtual machine, which is total prices of CPU, memory, storage and bandwidth. Here multi-dimensional constraints GA was compared with time-sharing algorithm, space-sharing algorithm and traditional GA that is only in order to shorten task completion time. In this simulation, Fig.2 of contrast of task completion time and Fig.3 of contrast of TACB were obtained.

Fig.3 shows that those algorithms would have little difference in task completion time and costs when there were small number of tasks. So their TACB values are similar. With the number of tasks increasing, completion time and costs of time-sharing algorithm and

space-sharing algorithm were increasing rapidly, which result in decline of TACB. Traditional GA only in order

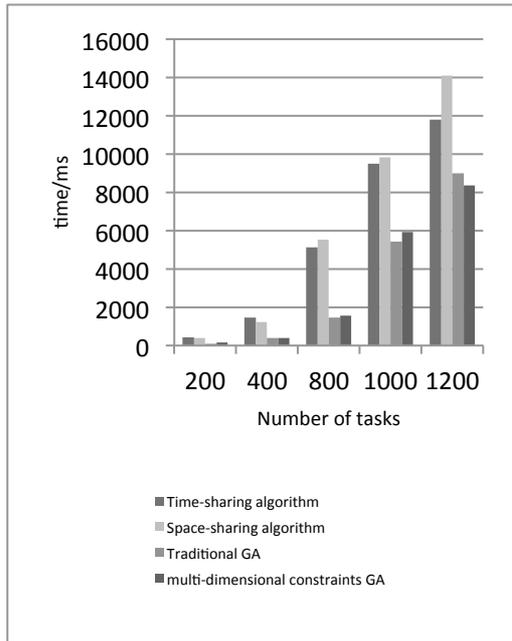


Figure 2. Contrast of task completion time

In Fig.2, the result indicated that both traditional GA and GA with multi-dimensional constraint conditions could effectively shorten task completion time.

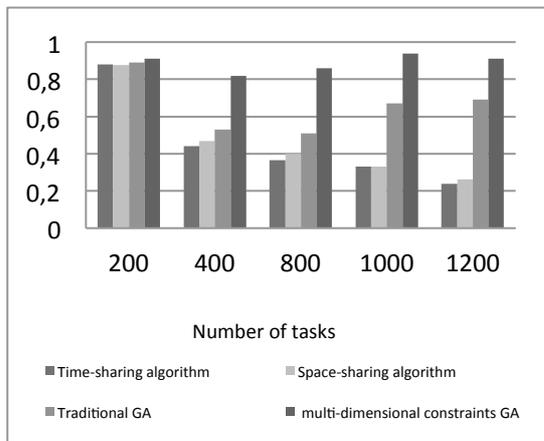


Figure 3. Contrast of TACB

to shorten completion time also reduced its TACB because of not effective control of expenses increasing. Multi-dimensional constrained GA with an effective control in terms of completion time and expenses could keep TACB at a high level abidingly.

## V. SUMMARIZES

Although traditional GA in order to shorten task completion time only shortened completion time of task

effectively when it is applied to task scheduling in Cloud Computing, it could not improve user's satisfaction of Cloud Computing services. This paper provided a new task scheduling mechanism based on GA with multi-dimensional constraints, and it used TACB, which is consisted of performance benefit and economic benefit, as the evaluation standard of GA. At Last, simulation experiments showed that this algorithm can complete Cloud Computing task scheduling efficiently.

## REFERENCES

- [1] Iosup A, Ostermann S, Yigitbasi M N, "Performance Analysis of Cloud Computing Services for Many-tasks Scientific Computing," *On Parallel and Distributed System. Washington*, vol. 22, pp. 931-945, June 2011.
- [2] SHEN Lijun, LIU Li, LU Rui, CHEN Yuting, TIAN Pingping, "Task Scheduling in Cloud Computing Based on Improved Immune Evolutionary Algorithm," *Computer Engineering. China*, vol. 38, pp. 208-210, July 2012.
- [3] Fan Jie, Peng Jian, Li Hongyou, "Demand elasticity algorithm for cloud computing based on ant colony optimization algorithm," *Journal of Computer Applications, China*, vol.31, pp. 1-4, June 2011.
- [4] Li Jianfeng, Peng Jian, "Task scheduling algorithm based on improved genetic algorithm in cloud computing environment," *Journal of Computer Applications, China*, vol. 31, pp. 184-186, Jan 2011. <http://dx.doi.org/10.3724/SP.J.1087.2011.00184>
- [5] Zhu Jianchen, Xu Jie, Lu ke, "Task Scheduling Algorithm Based on Euclidean Distance Liked Load Balance in Cloud Computing Environment," *Computer Simulation, China*, vol. 29, pp. 159-162, June 2012.
- [6] Sheng Jun Xue, Wu Wu, "Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm," *TELKOMNIKA Indonesian Journal of Electrical Engineering, Indonesia*, vol.10, pp. 1560-1566, Nov 2012.
- [7] SUN Jianmei, ZENG Biqing, "Optimal Model Redundant Task Scheduling for Cloud Computing," *Computer System & Applications, China*, vol. 21, pp. 56-59, April 2012.
- [8] Xuesong Yan, Qinghua Wu, Can Zhang, Wei Li, Wei Chen, Wenjing Luo, "An Improved Genetic Algorithm and Its Application," *TELKOMNIKA Indonesian Journal of Electrical Engineering, Indonesia*, vol. 10, pp. 1081-1096, Jan 2012.
- [9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and Experience, U S*, vol. 41, pp. 23-50, Aug 2011.
- [10] R Buyya, R Ranjan, R N Calheiros, "Modeling and simulation of scalable Cloud Computing environments and the CloudSim toolkit: Challenges and opportunities," *High Performance Computing & Simulation, 2009 HPCS' 09. International Conference*, pp. 1-11, June 2009.

## AUTHORS

**Youchan Zhu and Peng Liu** are with the Information & Network Management Center, North China Electric Power University, Baoding, China.

This article is an extended and modified version of a paper presented at the 2012 International Conference on Artificial Intelligence and Its Application in Industry Production (AIAIP 2012), held in Wuhan, China in December 2012. Manuscript received 28 May 2013. Published as resubmitted by the authors 26 June 2013.