Buletin Teknik Elektro dan Informatika
(Bulletin of Electrical Engineering and Informatics)
Vol.1, No.4, December 2012, pp. 279~284
ISSN: 2089-3191

A Fuzzy Differential Evolution Scheduling Algorithm Based on Grid

Fu Qiao [1,2], Guoyin Zhang[1]
1. Harbin Engineering University, Harbin, China Technology, Harbin, China
2. Heilongjiang Institute of Science and Technology, Harbin, China
NO.1 Tangchang Street Songbei District Harbin China of Heilongjiang Institute of Science and Technology, (+86)15804665036
e-mail:463667708@qq.com

Abstract

Aim at the problem of tasks scheduling problem for grid computing based on fuzzy constraints, propose tasks scheduling model of grid computing about tasks scheduling problem. Using fuzzy AHP(analytic hierarchy process) processes fuzzy constraints weight value. Utilize the merit of differential evolutionary algorithm, that is, the algorithm don't need objective function is continuous and differentiable. Propose fuzzy differential evolutionary algorithm, the algorithm can process tasks scheduling for grid computing based on fuzzy constraint. In the experiments, compare the proposed algorithm with the existing algorithms. the proposed algorithm is more favorable than the existing algorithm in term of reliability, security and drooped task numbers.

Keywords*: Grid computing, scheduling algorithm, differential evolution*

1. Introduction

In grid computing, resource management is NP problem. With the increase of the scale of the problem and constraints, the complexity of these problems exponentially increase, especially the size of the task scheduling problem increase faster with fuzzy constraints. In some literatures, different optimal technologies are proposed, for instance, genetic algorithm [1], artificial fish swarm algorithm [2], simulated annealing algorithm [3]. The idea of the literature [3] is to allow the searched inferior solutions to enter solution space, this inevitably reduce the efficiency of the algorithm. In those algorithms, the genetic algorithm is considered to be one of effective solutions to high-dimensional combination optimization problem [4]. In the literature [5], the combination of Genetic algorithms and simulated annealing algorithm solves grid tasks scheduling problems, this results in decreasing the complexity of genetic algorithm, but scheduling time of tasks can not be reduced. To solve multi-constraints problem, the general approach is to aggregate multiple objective functions into a single objective function, so that these algorithms have low time complexity, and so easy to implement. For example, the literatures [6] propose some multi- constraints heuristic task scheduling algorithms. These algorithms are used single-objective as the final optimization strategies, while modeling the system, the treatment of multi- constraints using weighted method leads to the non-distinct constraints ineffectively addressed. The literature [7] proposed a GA based on multi-objective evolutionary algorithm, the algorithm can solve the problem of multi-objective scheduling grid, but the algorithm searches near an optimal solution after several genetic, crossover and mutation. The literature [8] avoids the fault of the literature [7], but the literature [8] needs that constraints function is continuous and differentiable, this results in limited use of the literature [8].In this paper, we first solve weight value of fuzzy constraint using fuzzy AHP (analytic hierarchy process), propose a fuzzy differential evolution scheduling algorithm to finish tasks scheduling in grid computing.

The rest of this paper covers the following sections, Section 2 is concerned with task scheduling model, Section 3 details descriptions of the algorithm, Section 4 shows experimental results and Section 5 gives conclusions.

Received July 22, 2012; Revised September 26, 2012; Accepted October 20, 2012

2. Task scheduling model

$R=\{r_1,r_2,\cdots,r_i,\cdots,r_m\}$ is *m* compute nodes with a heterogeneous grid platform, $1\le i\le m$. $T=\{t_1,t_2,\cdots,t_j,\cdots,t_n\}$ is a set of *n* independent tasks, $1\le j\le n$. $X=\{x_1,x_2,\cdots,x_k,\cdots,x_n\}$ is a mapping scheme set, $1\le k\le n$, the tupelo $x_k=(t_j,r_i)$ denotes running scheme of task t_j scheduled on r_i. $Q=\{Q^1\times Q^2\times\cdots\times Q^h\times\cdots Q^s\}=\{(q^1,q^2,\cdots,q^h,\cdots,q^s)\mid q^h\in Q^h\}$ is s-dimension constraint space, the set Q^h is a value in the first *h* constraint, set Q_j^h is a value in Q^h. A constrain value of task t_j in s-dimension is $q_j=(q_j^1,q_j^2,\cdots,q_j^i,\cdots,q_j^s)\in Q_j$, q_j^i is a value of task t_j in the first *j* constraint. The least constraint needs is $q^{\min}=(q_1^{\min},q_2^{\min},\cdots,q_j^{\min},\cdots,q_n^{\min})$, the $q_j^{\min}$ is the least constraint needs of task t_j. A clear objective function model of multi-constraints scheduling is

$$Max(f^h)=\sum_{j=1}^{p}q_j^h x_k \quad \text{s.t. . } x_k\in X,\ \sum_{j=1}^{p}Q_j(x_k)\ge q_j^{\min} \tag{1}$$

In the equation, the *h*=1,2,…,*s* and *j*=1,2,…,*p* express, respectively, the number of constraint objective and the number of constraint space. q_j^h is a constant value, different tasks have different constraint value, q_j^h obtained by prediction method. If $X=\{x_1,x_2,\cdots,x_p\}$ satisfies the Eq. (1), that is a feasible scheduling scheme. If, in the Eq. (1), the coefficients are known, scheduling scheme is easily obtained. In fact, some objects is non-distinct, so we can substitute Eq. (2) for Eq. (1).

$$\max(f(x)) \text{ s.t.. } x_k\in X,\ \tilde{q}_{k,j}x_jw_j \underset{\sim}{<} \tilde{p}_k \tag{2}$$

In the equation, $j=1,2,\cdots,n$, $k=1,2,\cdots,s$, the symbol $\underset{\sim}{<}$ is fuzzy expression of the symbol $\le$, w_j is weight value. To determine the weight value w_j, we use fuzzy analytic hierarchy process.

If $\mathbf{A}=(a_{ij})_{n\times n}$ is judgment matrix, $i,j=1,2,\cdots,n$, $\forall i,j$, $a_{ii}=0.5$, $0<a_{ij}<1$, $a_{ij}+a_{ji}=1$, we consider $\mathbf{A}$ as fuzzy complementary judgment matrix.

Definition 1. fuzzy complementary judgment matrix $\mathbf{A}=(a_{ij})_{n\times n}$, if $\forall k, 1\le k\le n$, $a_{ij}=a_{ik}-a_{jk}+0.5$, we regard $\mathbf{A}$ as fuzzy unanimous judgment matrix.

If fuzzy complementary judgment matrix $\mathbf{A}_k=(a_{ij}^{(k)})_{n\times n}$, $k=1,2,\cdots,s$, and $\bar{a}_{ij}=\sum_{k=1}^{s}\lambda_k a_{ij}^{(k)}$, $\lambda_k>0$, $\sum_{k=1}^{s}\lambda_k=1$, The $\bar{\mathbf{A}}=(\bar{a}_{ij})_{n\times n}$ is named after composed matrix $\mathbf{A}_k$ $(k=1,2,\cdots,s)$, that is, $\bar{\mathbf{A}}=\lambda_1\mathbf{A}_1\oplus\lambda_2\mathbf{A}_2\oplus\cdots\oplus\lambda_s\mathbf{A}_s$.

Theorem 1. composed matrix $\bar{\mathbf{A}}=\lambda_1\mathbf{A}_1\oplus\lambda_2\mathbf{A}_2\oplus\cdots\oplus\lambda_s\mathbf{A}_s$ of fuzzy unanimous judgment matrix is fuzzy unanimous judgment matrix.

Proof:

$$\because\ \overline{a}_{ij} = \sum_{k=1}^{s} \lambda_k a_{ij}^{(k)} = \lambda_1 a_{ij}^{(1)} + \lambda_2 a_{ij}^{(2)} + \cdots \lambda_s a_{ij}^{(s)}$$
$$= \lambda_1 (a_{ik}^{(1)} - a_{jk}^{(1)} + 0.5) + \lambda_2 (a_{ik}^{(2)} - a_{jk}^{(2)} + 0.5) + \cdots + \lambda_s (a_{ik}^{(s)} - a_{jk}^{(s)} + 0.5)$$
$$= (\lambda_1 a_{ik}^{(1)} + \lambda_2 a_{ik}^{(2)} + \cdots + \lambda_s a_{ik}^{(s)}) - (\lambda_1 a_{jk}^{(1)} + \lambda_2 a_{jk}^{(2)} + \cdots + \lambda_s a_{jk}^{(s)})$$
$$+0.5(\lambda_1 + \lambda_2 + \cdots + \lambda_s)$$
$$\therefore \overline{a}_{ij} = \overline{a}_{ik} - \overline{a}_{jk} + 0.5$$

$\therefore$ According to Definition 1, the $\overline{\mathbf{A}} = \lambda_1 \mathbf{A}_1 \oplus \lambda_2 \mathbf{A}_2 \oplus \cdots \oplus \lambda_s \mathbf{A}_s$ is fuzzy unanimous judgment matrix.

Theorem 2. $\forall i, j, k, 1 \le i, j, k \le n$, $\mathbf{A} = (a_{ij})_{n\times n}$ is fuzzy judgment matrix, $\boldsymbol{W} = (w_1, w_2, \cdots, w_n)^T$ is weight vector of $\mathbf{A}$, (1) if $a_{ij} = w_i - w_j + 0.5$, $\mathbf{A} = (a_{ij})_{n\times n}$ is fuzzy unanimous judgment matrix. (2) if $a_{ij} \ne w_i - w_j + 0.5$, the $\boldsymbol{W}$ satisfies $w_i = (\sum_{j=1}^{n} a_{ij} + 1 - \frac{n}{2}) \Big/ n$.

Proof:

(1) $\because\ a_{ij} = w_i - w_j + 0.5$, $a_{ik} = w_i - w_k + 0.5$ and $a_{jk} = w_j - w_k + 0.5$

$$\therefore\ a_{ik} - a_{jk} + 0.5 = (w_i - w_k + 0.5) - (w_j - w_k + 0.5) + 0.5$$
$$= w_i - w_j + 0.5 = a_{ij}$$

$\therefore$ According to Definition 1, the $\mathbf{A} = (a_{ij})_{n\times n}$ is fuzzy unanimous judgment matrix

(2) $\because\ a_{ij} \ne w_i - w_j + 0.5$

$\therefore$ Suppose $F(\boldsymbol{W}) = \sum_{i=1}^{n}\sum_{j=1}^{n} (a_{ij} - (w_i - w_j + 0.5))^2$, $\sum_{i=1}^{n} w_i = 1$

$\therefore$ A Lagrangian function is $L(\boldsymbol{W}, \lambda) = F(\boldsymbol{W}) + \lambda(\sum_{i=1}^{n} w_i - 1)$

Suppose $\frac{\partial L}{\partial w_i} = 0$

$$\therefore\ \sum_{j=1}^{n} 2[a_{ij} - (w_i - w_j + 0.5)](-1) + \lambda = 0$$

$$-2[\sum_{j=1}^{n} a_{ij} - nw_i + 1 - 0.5n] + \lambda = 0$$

$$\sum_{i=1}^{n} (-2[\sum_{j=1}^{n} a_{ij} - nw_i + 1 - 0.5n)] + \lambda = 0$$

$$-2[\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} - n + n - 0.5n^2)] + \lambda n = 0$$

$$-2[\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}-0.5n^2)]+\lambda n=0$$

$$\therefore\ \sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}=0.5n^2$$

$$\therefore\ \lambda=0$$

$$\sum_{j=1}^{n}a_{ij}-nw_i+1-0.5n=0$$

$$\therefore\ w_i=(\sum_{j=1}^{n}a_{ij}+1-\frac{n}{2})\Big/n$$

The $\boldsymbol{W}$ can be obtained by $w_i=(\sum_{j=1}^{n}a_{ij}+1-\frac{n}{2})\Big/n$ from fuzzy complementary judgment matrix. If $a_{ij}\le 1-n/2$, the $\boldsymbol{W}$ value is negative or zero, the $\boldsymbol{W}$ will be judged again.

3 Task scheduling algorithm

The following is implementation steps for FDEA (Fuzzy Differential Evolution Algorithm), the procedure of the MONCDA is the following:

Step1: Produce initial population, compute fitness value of chromosome.

(1.1) An initial population of size *n* is randomly generated from $[0,1]^{i+1}$ according to the uniform distribution in the closed interval $[0,1]$, produce randomly *n* population, chromosomes with the mapping scheme $\{x_k(t_i,r_j)\}$ is $\tilde{Q}_{y,G}=\frac{\mu_{y_0,G}}{pt_0}+\frac{\mu_{y_1,G}}{pt_1}+\cdots+\frac{\mu_{y_i,G}}{pt_i}+\cdots+\frac{\mu_{y_r,G}}{pt_r}$, $y_i\in\{x_k(t_i,r_j)\}$, μ_{y_i} is a real number in the closed interval $[0,1]$, pt_i is partition in given interval, G is evolution algebra, $k=0,1,\cdots,n$, $i=0,1,\cdots,r$.

(1.2) Estimate value of each constraint coefficient, evaluate every chromosome in the population by fitness value.

(1.3) If task set T is not empty, a computing node $r_i\in R$ and a task $t_i\in T$ are selected randomly, the task t_i is distributed to r_i of least constraint and deleted from tail of ready queue and the task set T.

Step2: Mutation operation

For each $\tilde{Q}_{y,G}$, after mutation operation, $v_{y,G+1}=\tilde{Q}_{r1,G}+F\cdot(\tilde{Q}_{r2,G}-\tilde{Q}_{r3,G})$, the number $r1\ne r2\ne r3$ is randomly selected, $F\in[0,1]$ is factor, the factor can control offset variation.

Step3: Crossover operation

(3.1) For the *G*th population $\tilde{Q}_{y,G}$ and mutation population $v_{y,G+1}$, execute crossover operation, $u_{y,G+1}=\begin{cases}v_{y,G+1} & ran(0,1)\le CR\\ \tilde{Q}_{y,G} & otherwise\end{cases}$, crossover probability is $CR=0.5*(1+rand(0,1))$.

(3.2) Execute remedy operation. If the numbers of a task scheduled is more than one, the least constrain schedule is retained, other schedules is deleted. If the numbers of a task scheduled is zero, the task is distributed to a computing node of the least constrain. The task is inserted into the tail of ready queue, is deleted from the task set T.

Step4: Selection operation

(4.1) Compute and add every chromosome fitness value of the ***Step3***, S_y is the fitness value, $y=1,2,\cdots,n$, construct the intervals of the fitness value, $[0,S_1]$, $[S_1,S_2]$, $\cdots$, $[S_{j-1},S_j]$, $j=2,\cdots,n-1$.

(4.2) Order the intervals $[0,S_1]$, $[S_1,S_2]$, $\cdots$, $[S_{j-1},S_j]$, Obtain the order of each chromosome, a random number ran_num is generated from the interval $[0,T]$ and regard as distance. If $ran\_num \in [S_{j-1},S_j]$, the *j*th chromosome $\tilde{Q}_{j,G}$ is selected.

(4.3) Reproduce the new population of the step (4.2).

From the step (1.2) of the ***Step1*** to the step (4.3) of the ***Step4***, execute G times iterations. The algorithm is terminated after G generations are produced. The last population is $\tilde{Q}^*_{1,G},\tilde{Q}^*_{2,G},\cdots\tilde{Q}^*_{n,G}$, the maximum fitness value is the best chromosome in the population. The best chromosome represents the optimal solution for the problem. Let the best chromosome be $\tilde{Q}^*_{y,G}=\frac{\mu^*_{y_0,G}}{pt_0}+\frac{\mu^*_{y_1,G}}{pt_1}+\cdots+\frac{\mu^*_{y_i,G}}{pt_i}+\cdots+\frac{\mu^*_{y_r,G}}{pt_r}$, $1\le y\le n$. The estimated value of each fuzzy coefficient in Eq. (1).

4. Experimental results and analysis

The characteristic of a PC is the following: CPU is Pentium 2.8G, RAM is 2G and operating system is Windows Xp. Simulation software is GridSim. The numbers of computing nodes are 20, tasks are randomly generated, the numbers of tasks is between 100 and 5000. The initial population is 100, the crossover probability is 0.2, the maximum times of iteration is 100, the factor $F\in[0,1]$, the evolution algebra is 100. Using the evaluation indicators are, security and dropped tasks numbers. The comparisons of security and the dropped tasks numbers for the FDEA algorithm and the GA algorithm is respectively shown in Fig. 1 and Fig. 2.

From Fig.1, with the increase of the numbers of tasks, the security value of the two algorithms increase, but the security value of the FDEA is more than the security value of the GA. From Fig. 2, with the increase in the tasks numbers of two algorithms, increase the dropped number of tasks, but the FDEA for dropped tasks numbers is less than the GA.

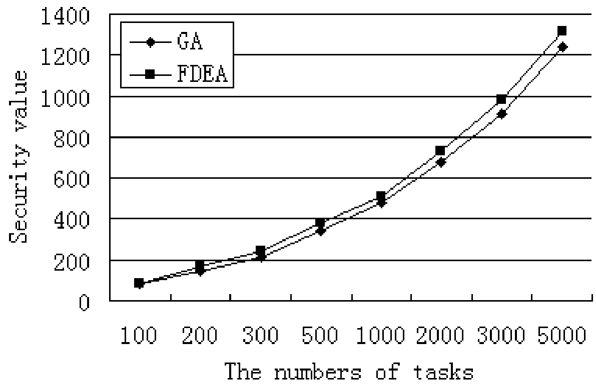

Figure 1. Comparisons of security

5. Conclusions

In this paper, we solve fuzzy constraints tasks scheduling problem of grid computing. Establish the model about fuzzy constraints tasks scheduling problem and give weight value of fuzzy constraints using fuzzy AHP. Propose a fuzzy differential evolution scheduling algorithm to deal with tasks scheduling in grid computing. The experiments show that the FDEA algorithm is better than the GA algorithm. The GA-MNCDA algorithm is helpful for grid computing tasks scheduling.

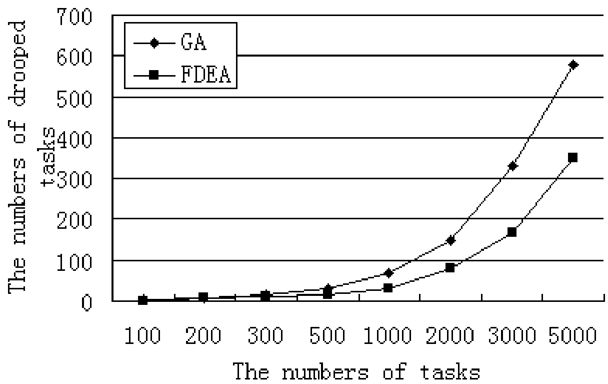

Figure 2. Comparisons of the Dropped Tasks Numbers

Acknowledgements

This work was financially supported by foundation of Science and Technology Research Project of Heilongjiang Provincial Department of Education in China (12521473).

References

[1] J. Carretero, F. Xhafa, A. Abraham. Genetic algorithm based schedulers for grid computing systems. Journal of innovative computing, information and control, 2007, 3(6):1-19.
[2] S. Farzi. Efficient job scheduling in grid computing with modified artificial fish swarm algorithm. Journal of computer theory and engineering, 2009, 1(1):1793-8201.
[3] P. Tormos, A. Lova, F. Barber, et al. A genetic algorithm for railway scheduling problem. Studies in computational intelligence, 2008, 128:255-276.
[4] R. Vijayalakshmi, G. Padamavathi. A performance study of GA and LSH in multiprocessor job scheduling. Journal of computer science, 2010,7(1):37-42.
[5] A. Baruah. A GA approach to static task scheduling in grid based systems. Journal on computer science and engineering,2012,4(1):54-61.
[6] Ding Ding, Luo Si-wei, Gao Zhan. An Object-Adjustable Heuristic Scheduling Strategy in Grid Environments. Journal of Application Research of Computers, in chinese, 2007, 44(9):1572-1578.
[7] Zhang Weizhe, Hu Mingzeng, Zhang Hongli et al. A multiobjective evolutionary algorithm for grid job scheduling of multi-Qos constraints.Journal of computer research and development, in chinese, 2006,43(11):1855-1862.
[8] Qiao Fu, Zhang Guoyin, Liu Zhongyan. Grid tasks scheduling strategy based on segree of multi-objective conflict.Journal of application and research of computers, in chinese, 2009, 04.