Send Orders for Reprints to reprints@benthamscience.net

The Open Automation and Control Systems Journal, 2013, *5*, 87-95

LoBa-Min-Min-SPA: Grid Resources Scheduling Algorithm Based on Load Balance Using SPA

Zhongping Zhang[1,2], Lijuan Wen[1,*] and Zhiping Wang[3]

[1]*College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, 066004, China*

[2]*The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao, Hebei 066004, China*

[3]*Foreign Language College of Dalian Jiaotong University, Dalian,Liaoning, 116052, China*

Abstract: In the grid environment, there are a large number of grid resources scheduling algorithms. According to the existing Min-Min scheduling algorithm in uneven load, and low resource utilization rate, we put forward LoBa-Min-Min algorithm, which is based on load balance. This algorithm first used Min-Min algorithm preliminary scheduling, then according to the standard of reducing Makespan, the tasks on heavy-loaded resources would be assigned to resources that need less time to load balance, raise resource utilization rate, and achieve lesser completion time. We used benchmark of instance proposed by Braun et al. to prove feasibility and effectiveness of the algorithm. At Last, We introduced the SPA and gave the experimental results of Min-Min-SPA and LoBa-Min-Min-SPA.

Keywords: Grid Computing, Task Scheduling, Min-Min Scheduling Algorithm, Load Balance, Set Pair Analysis (SPA).

1. INTRODUCTION

Grid computing [1] is often considered the most important technology after the Internet, and grid computing is developing rapidly with the Internet. Grid computing is a new calculation model for large-scale complex scientific experiments, allowing using a large number of geographically distributed idle computing resources. These heterogeneous resources contain supercomputers which belong to different dynamic virtual organizations, storage systems, workstations, data resources, networks, software, special equipment etc. In summary, grid computing organizes computers dispersed in different geographical locations with the Internet and becomes a virtual "super computer". Grid computing has two advantages, one is the strong data processing ability, and the other is the ability of making full use of idle computing resources with the network.

As the grid resources with the distributed, dynamic and multiple managerial characteristics [2], resource management and task scheduling in the grid become essential. In grid computing, task scheduling objective is to maximize resource utilization and minimize Makespan. The effectiveness of the scheduling algorithm is measured by the throughput of the system. The focus of this study is to minimize Makespan and improve the system throughput.

The scheduling of tasks on heterogeneous grid resources is known to be a NP-complete problem, and heuristics are used for task scheduling. Task scheduling is to allocate tasks to resources, mapping task set to resource set. Min-Min scheduling algorithm [3] has achieved a very good performance in the existing resources scheduling algorithms, however, because Min-Min scheduling algorithm always preferentially schedules short tasks on resources with high computational power, resulting in load imbalance and low resource utilization [4].

To avoid the drawbacks of Min-Min scheduling algorithm, we put forward grid resources scheduling algorithm based on load balance--LoBa-Min-Min (Load Balance Min-Min) scheduling algorithm. This algorithm first uses Min-Min algorithm preliminary scheduling, then according to the standard of reducing Makespan, the tasks on heavy-loaded resources will be assigned to resources that need less time to load balance, raise resource utilization rate, and achieve lesser completion time. The paper gave the experimental results of Min-Min scheduling algorithm and LoBa-Min-Min scheduling algorithm.

Because of the uncertainty nature of a grid, traditional task scheduling algorithms do not work well in an open, heterogeneous and dynamic grid environment of real world. SPA (Set Pair Analysis) [5], a new soft computation method is used to process the synthetic uncertainty in the task scheduling of a computing grid. After introducing SPA, The Paper gave the experimental results of Min-Min-SPA (Min-Min scheduling algorithm on SPA) and LoBa-Min-Min-SPA (LoBa-Min-Min scheduling algorithm on SPA).

2. RELATED WORKS

Currently, there are a lot of heuristic grid resources scheduling algorithms, they map the meta-task set to the het-

*Address correspondence to this author at the College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, 066004, China; Tel: 086+335+8047606; Fax: 086+335+8074806;
E-mail: zpzhang@ysu.edu.cn

1874-4443/13

erogeneous resource set. Here are several common scheduling algorithms.

2.1. Opportunistic Load Balancing Algorithm (OLB)

OLB schedules tasks to the next earliest available idle resources. If there are multiple available idle resources, OLB will select one of them randomly. OLB does not consider the execution time of the tasks on those resources [6, 7], resulting in poor system throughput.

2.2. Minimum Execution Time Algorithm (MET)

MET belongs to the online mode scheduling, which considers only one task once. MET assigns each task to the resource with the minimum expected execution time without considering the availability and current load of the resource.

2.3. Minimum Completion Time Algorithm (MCT)

The scheduling policy of MCT is similar to MET, the difference lies in the completion time rather than the execution time. MCT calculates the completion time for a task on all resources by adding the availability time of resources and the expected execution time of the task on the resources.

2.4. Min-Min Algorithm

Min-Min scheduling algorithm which belongs to the batch mode scheduling is different from the MET and the MCT, Min-Min scheduling algorithm starts with a set of all unmapped tasks. Firstly, the algorithm calculates the completion time of each task on each resource by adding the expected execution time and the resource available time. Secondly, the algorithm calculates the earliest completion time of each task. Then the task with the overall minimum completion time is selected and mapped to the resource which obtains the overall minimum completion time. The process is repeated until all the unmapped tasks are assigned.

2.5. Max-Min Algorithm

Similar to Min-Min scheduling algorithm, Max-Min scheduling algorithm also considers all unmapped tasks. Min-Min scheduling algorithm always preferentially schedules short tasks, resulting in the waiting time of long tasks increases and the long tasks can't be implemented in a timely manner. In the case of the number of the short tasks more than the number of the long tasks, we can preferentially schedule a long task, reducing the waiting time of long tasks. Max-Min scheduling algorithm preferentially schedules long tasks on resources with high computational power. In the case of the number of the short tasks less than the number of the long tasks, the preferential scheduling of long tasks may increase the Makespan.

2.6. QPS $_{\text{Max-Min<>Min-Min}}$ Algorithm

QPS $_{\text{Max-Min<>Min-Min}}$ scheduling algorithm is a QoS based predictive Max-Min, Min-Min switcher algorithm for job scheduling in a grid [8]. Most of the resources in the grid aren't dedicated resources. While performing the tasks in the grid, the resources also perform their own internal tasks. In order to measure the completion time of grid tasks better, the algorithm introduces prediction mechanism in grid resource scheduling. Since the existing Min-Min, Max-Min scheduling algorithm are suitable for the cases where we have more long tasks and more short tasks respectively, the authors put forward a QoS based predictive Max-Min, Min-Min Switcher Algorithm for job scheduling in a grid. QPS $_{\text{Max-Min<>Min-Min}}$ scheduling algorithm calculates the standard deviation SD of the execution time of the task set. If the difference between any two of task execution time is greater than the standard deviation and appears in the front half, Min-Min scheduling algorithm is used; otherwise Max-Min algorithm is used.

2.7. Min-Mean Algorithm

Min-mean scheduling algorithm, which is an improved Min-Min scheduling algorithm based on the average execution time of resources—mean CT [9,10]. The algorithm is divided into two phases. In the first phase, Min-Min scheduling algorithm pre-schedules the tasks; In the second phase, the resources whose completion time is greater than meanCT are defined as heavy-loaded resources. Tasks on the heavy-loaded resources whose execution time are more than the mean-CT need reschedule. This algorithm balances the load and improves resource utilization.

3. LOBA-MIN-MIN ALGORITHM

3.1. Problem Description

Grid resource scheduling is task scheduling in heterogeneous grid environment. In this paper the experimental study is based on a benchmark simulation model proposed by Braun et al. Static mapping of meta-tasks is considered in this model. A resource can only perform one task at one time, tasks on resources non-preemptive sequentially execute in accordance with the order of distribution. Meta-tasks static mapping mechanism requires necessary prerequisite elements, the known number of tasks and resources, and the known ETC matrix. Based on the above conditions, the premise of the grid resource scheduling problem can be described as follows:

(1) The execution of the applications consist of meta-tasks, which can't be divided and have no dependency among each other. Meta-tasks [11] have no deadlines and priorities.

(2) The number of available resources to participate in the allocation of tasks is known.

(3) The workload of each task is known.

(4) The computing capacity of each resource is known.

(5) The ready time of the resource after completing the previously assigned task.

(6) ETC (m*n) matrix is known, representing the execution time of m tasks on the n resources.

Based on the above conditions, the grid resource scheduling problem can be described as follows:

(1) Task set $\{T_1, T_2, T_3, \ldots, T_m\}$ submit to the grid resource scheduler.

(2) Available resource set $\{R_1, R_2, R_3, \ldots, R_n\}$ is known When the task set has submitted to the grid resource scheduler.

(3) Makespan=max(CT_j), $CT_j = \sum_i CT_{ij}$,

(4) $CT_{ij}=r_j+ET_{ij}$, Makespan represents the completion time of the task set, CT_j represents the completion time of each resource, ET_{ij} represents the execution time of task Ti on resource R_j. r_j represents the ready time of the resource after completing the previously assigned task.

(5) Makespan is one of the main indicators of evaluating scheduling algorithms; the main objective of optimizing the scheduling algorithm is to reduce Makespan.

3.2. LoBa-Min-Min Scheduling Algorithm Description

LoBa-Min-Min scheduling algorithm is an improved scheduling algorithm based on Min-Min scheduling algorithm. In the first phase, Min-Min scheduling algorithm pre-schedules the tasks; LoBa-Min-Min scheduling algorithm defines the resources obtained the Makespan as heavy-loaded resources, Makespan=max(CT_j). The goal of LoBa-Min-Min scheduling algorithm is to balance load and reduce the Makespan. LoBa-Min-Min scheduling algorithm re-schedules tasks on the heavy-loaded resources to the resources having a smaller completion time. In the second phase, LoBa-Min-Min scheduling algorithm sorts the resources in the increasing order of CT_j(j=1,2,...,n) and sets the resource as R_{maxCT} which has obtained the maximum completion time maxCT. LoBa-Min-Min scheduling algorithm reschedules tasks assigned to the resource R_{maxCT} to other resources that can reduce the maxCT, then updating CT_j (j = 1, 2, ..., n) and resorting the resources in increasing order of CT_j (j = 1,2, ..., n).Repeating these steps to obtain new maxCT and the corresponding R_{maxCT} until the tasks on the R_{maxCT} can no longer be assigned to other resources , then reaching smaller Makespan.

3.3. LoBa-Min-Min Scheduling Algorithm

According to the thinking of the above description, we will give LoBa-Min-Min scheduling algorithm based on load balance, the algorithm will describe in the following algorithm 1.

Algorithm 1: LoBa-Min-Min sheduling algorithm

INPUT: All tasks T_i (i = 1,2, ..., m), All resource R_j (j = 1,2, ..., n), the ETC matrix

OUTPUT: Makespan, Scheduling table

LoBa-Min-Min (T_i, R_j, ETC)

BEGIN

//Min-Min scheduling algorithm

(1) For all tasks T_i

(2) For all resources

(3) $C_{ij}=E_{ij}+r_j$

(4) End for

(5) End for

(6) Do until all tasks are mapped

(7) For each task find the earliest completion time and resource that obtains it

(8) Find the task T_k with the minimum earliest completion time assign task T_k to the resource R_l that gives the earliest completion time

(9) Delete task T_k from list

(10) Update ready time for resource R_l

(11) Update C_{il} for all i

(12) End do

//Rescheduling the tasks of the heavy-loaded resources

(13) Sort the resources in the increasing order of CT_j

(14) Do While the R_{maxCT} with $maxCT$ has tasks rescheduling

(15) Sort the tasks on R_{maxCT} in the increasing order

(16) For all tasks T_{kmaxCT} on R_{maxCT}

(17) For the other n-1 resources

(18) IF($CT_j+ET_{kmaxCT}<CT_{maxCT}$&& CT_j+ET_{kmaxCT} is the minimum)

(19) Reschedule T_{kmaxCT} on R_j and update CT

(20) End for

(21) IF a task is assigned, exit the loop

(22) End for

(23) Sort the resources in the increasing order of the CT_j

(24) End do

(25) Output the Makespan and tasks scheduling table

END

In summary, the implementation process of LoBa-Min-Min scheduling algorithm is as follows. Step 1)-12), Min-Min scheduling algorithm pre-schedules tasks. Min-Min scheduling algorithm cause load imbalance and low resource utilization, and LoBa-Min-Min scheduling algorithm use step 13) -24) to reschedule the tasks. First, LoBa-Min-Min scheduling algorithm sorts resources in increasing order according to the completion time, then taking task set T_{kmaxCT} on the resource R_{maxCT} with the maximum execution time maxCT out; Second, according to the algorithm step 18), If T_{kmaxCT} is assigned to the other n-1 resources can reduce CT_{maxCT}, LoBa-Min-Min scheduling algorithm will assign T_{kmaxCT} out; If there are multiple resources enable to reduce CT_{maxCT}, the algorithm will select resource with the minimum completion time. Third, the algorithm sort the resources in increasing order again with the completion time, then we carry out the cycle step 14) -24) until tasks on resource R_{maxCT} with the maximum completion time maxCT can't be assigned out. LoBa-Min-Min scheduling algorithm performs the above steps to get a smaller Makespan. It is proved that performance of LoBa-Min-Min scheduling algorithm is significantly superior to Min-Min scheduling algorithm.

4. AN ILLUSTRATIVE EXAMPLE

Supposing task set {T_1,T_2,T_3,T_4,T_5} and resource set {R_1,R_2,R_3} have submitted to the grid resource scheduler, and ETC matrix has been given in Table **1**. Through a simple

Table 1. ETC Matrix

	R_1	R_2	R_3
T_1	1	2	3
T_2	2	4	5
T_3	3	7	8
T_4	4	11	12
T_5	5	16	17

example to verify performance of LoBa-Min-Min scheduling algorithm is superior to Min-Min scheduling algorithm.

The first stage: The results of Min-Min scheduling algorithm pre-scheduling are shown in Fig. (**1**).

As seen in Fig. (**1**), the Makespan of Min-Min scheduling algorithm is 15 sec.

The second stage: At this time, the completion time of the resource R_1, R_2, and R_3 is 15sec, 0sec, and 0sec respectively. The resource set $\{R_1,R_2,R_3\}$ sorts in increasing order resource set $\{R_2,R_3,R_1\}$ with the completion time. Then LoBa-Min-Min scheduling algorithm takes task set $\{T_1,T_2,T_3,T_4,T_5\}$ on resource R_1 out, according to the step 18) in LoBa-Min-Min scheduling algorithm, the task T_1 can be assigned to resource R_2 to obtain the result shown in Fig. (**2**).

At this time, the completion time of the resource R_1, R_2, and R_3 is 14sec, 2sec, and 0sec respectively. The resource set $\{R_2,R_3,R_1\}$ sorts in increasing order resource set $\{R_3,R_2,R_1\}$ with the completion time. Then LoBa-Min-Min scheduling algorithm takes task set $\{T_2,T_3,T_4,T_5\}$ on resource R_1 out, according to the step 18) in LoBa-Min-Min scheduling algorithm, the task T_2 can be assigned to resource R_3 to obtain the result shown in Fig. (**3**).

At this time, the completion time of the resource R_1, R_2, and R_3 is 12sec, 2sec, and 5sec respectively. The resource set $\{R_3,R_2,R_1\}$ sorts in increasing order resource set $\{R_2,R_3,R_1\}$ with the completion time. Then LoBa-Min-Min scheduling algorithm takes task set $\{T_3,T_4,T_5\}$ on resource R_1 out, according to the step 18) in

LoBa-Min-Min scheduling algorithm, the task T_3 can be assigned to resource R_2 to obtain the result shown in Fig. (**4**).

At this time, the completion time of the resource R_1, R_2, and R_3 is 9sec, 9sec, and 5sec respectively. The resource set $\{R_2,R_3,R_1\}$ sorts in increasing order resource set $\{R_3,R_2,R_1\}$ with the completion time. Then LoBa-Min-Min scheduling algorithm takes task set $\{T_4, T_5\}$ on resource R_1 out. LoBa-Min-Min scheduling algorithm ends when tasks on resource R_1 can no longer be allocated out. At last, the final result is shown in Fig. (**4**), Makespan = 9sec <15 sec.

5. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

We used the benchmark proposed in [12] to present the computational result, Benchmark instances are classified into 12 different types according to the consistency of ETC matrix, job heterogeneity and resource heterogeneity. We simulated LoBa-Min-Min scheduling algorithm and Min-Min scheduling algorithm in these 12 cases.

5.1. Experimental Standard

Instances benchmark in 12 cases are labeled as u-x-yy-zz.

(1) u -uniform distribution used in generating ETC matrix

(2) x -the type of consistency (c-consistent, i-inconsistent,

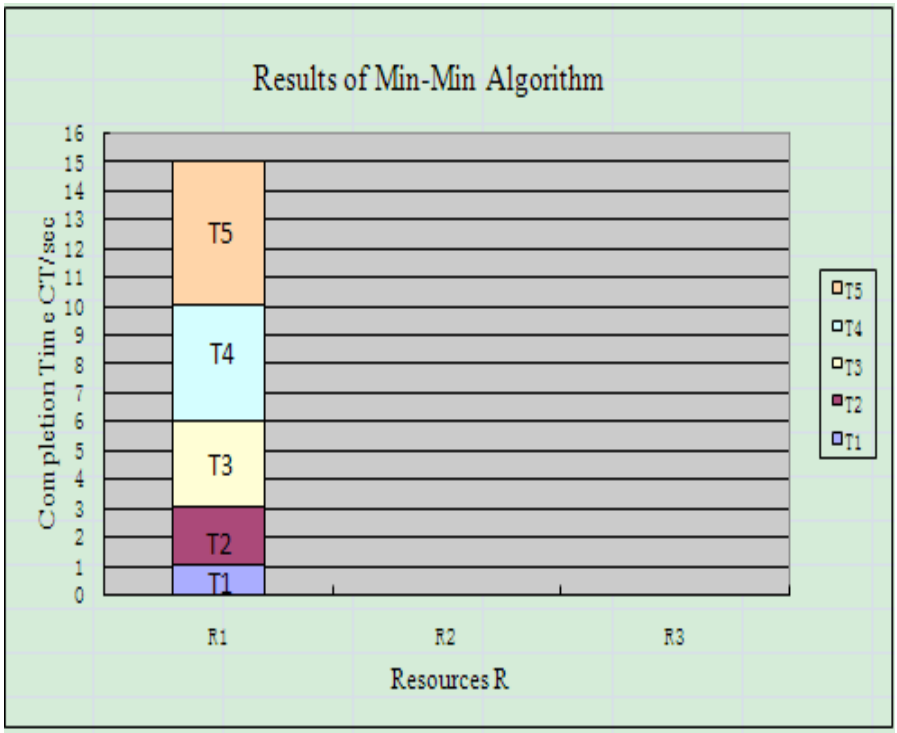

Fig. (1). Results of Min-Min Algorithm.

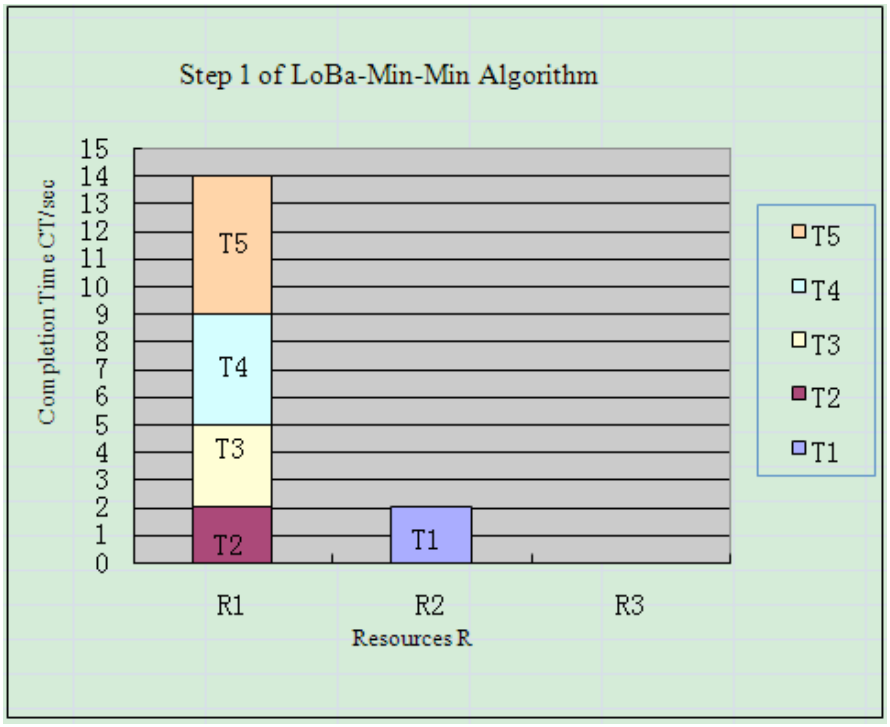

Fig. (2). Step 1 of LoBa-Min-Min Algorithm.

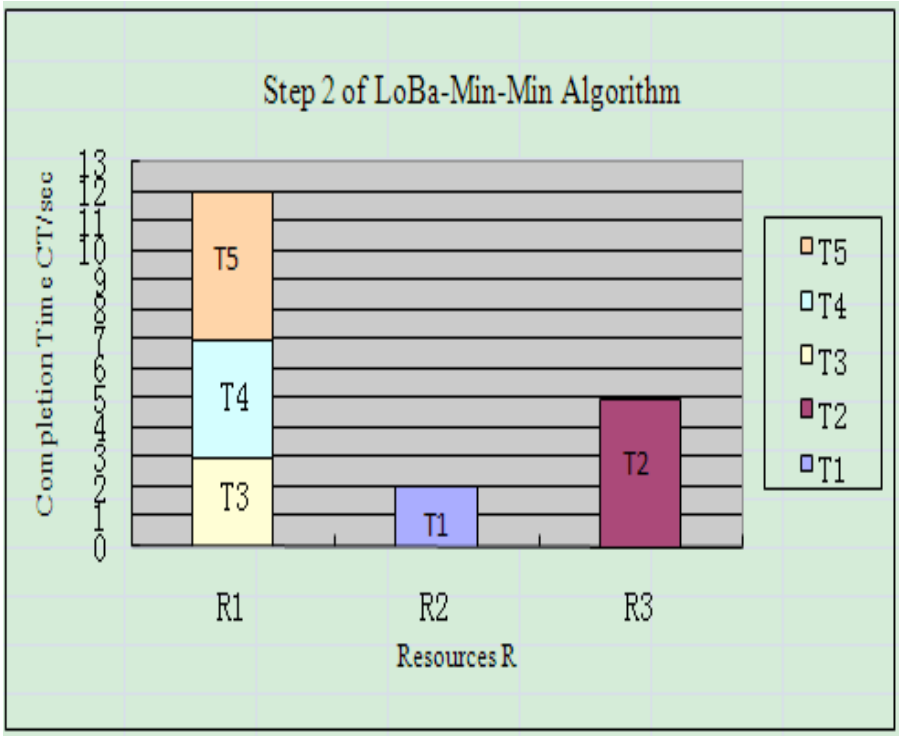

Fig. (3). Step 2 of LoBa-Min-Min Algorithm.

s-semi-consistent or partially-consistent)

(3) An ETC matrix is consistent if a resource R_j executes any task T_i faster than resource R_k, then resource R_j executes all tasks faster than resource R_k.

(4) An ETC matrix is inconsistent if a resource R_j is faster than resource R_k for some tasks and slower for other tasks.

(5) An ETC matrix is semi-consistent or partially consistent if it includes a consistent sub-matrix.

(6) yy -task heterogeneity(hi-high, lo-low)

(7) task heterogeneity:Variation in the execution time of the entire tasks for a given resource.

(8) zz - resource heterogeneity (hi-high,lo-low)

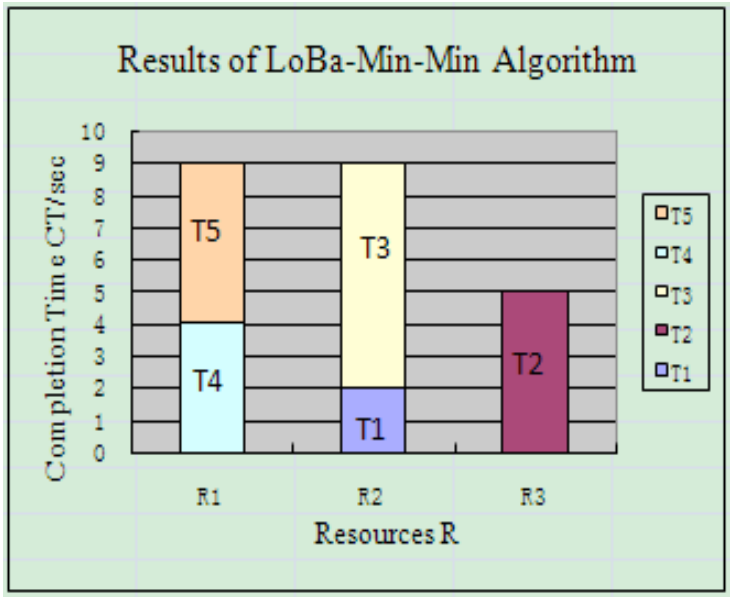

Fig. (4). Results of LoBa-Min-Min Algorithm.

Table 2. Results of LoBa-Min-Min and Min-Min

Instances	LoBa-Min-Min(ms)	Min-Min(ms)
u_c_hihi	3.446704×10^{7}	1.26670067×10^{9}
u_c_hilo	5.7116310×10^{6}	1.56062448×10^{8}
u_c_lohi	5.288029×10^{3}	2.1409888×10^{5}
u_c_lolo	6.230497×10^{2}	1.8261639×10^{4}
u_i_hihi	1.4216358×10^{7}	1.15505184×10^{8}
u_i_hilo	2.7787848×10^{6}	2.6237578×10^{7}
u_i_lohi	2.282899×10^{3}	6.948441×10^{4}
u_i_lolo	3.2669702×10^{2}	3.3645144×10^{3}
u_s_hihi	1.6272901×10^{7}	2.245968×10^{8}
u_s_hilo	3.090145×10^{6}	4.3790376×10^{7}
u_s_lohi	1.412832×10^{3}	9.745546×10^{4}
u_s_lolo	3.5311795×10^{2}	5.2183545×10^{3}

According to Table **2**, we can give Fig. (**5**)

(9) resource heterogeneity:Variation in the execution time of a particular task among the entire resources.

5.2. Experimental Environment

The experimental environment is desktop computer, System is Microsoft Windows XP Professional Version 2002 Service Pack 3, Inter (R) Core (TM) 2 Duo CPU E4500@2.20GHz 2.19GHz, 2.00GB memory; Simulator is GridSim-toolkit 5.0 version.

5.3. Performance Analysis

The experiment was conducted in the above experimental environment and instance benchmark proposed by Braun et al, the ETC matrix generated based on 512 tasks and 16 resources in instances benchmark using uniform distribution, the results of LoBa-Min-Min and Min-Min scheduling algorithm are shown in Table **2**.

From the above experimental results can be seen, in the case of ETC matrix consistency, tasks and resources high heterogeneous, the Makespan of LoBa-Min-Min scheduling algorithm proposed in this paper is significant less than Min-Min scheduling algorithm. In the case of ETC matrix semi-consistency and inconsistency, tasks and resources high heterogeneous, the Makespan of LoBa-Min-Min scheduling algorithm proposed in this paper is also less than Min-Min scheduling algorithm, In brief, In the case of tasks and resources high heterogeneous, the performance of the proposed LoBa-Min-Min scheduling algorithm is significant

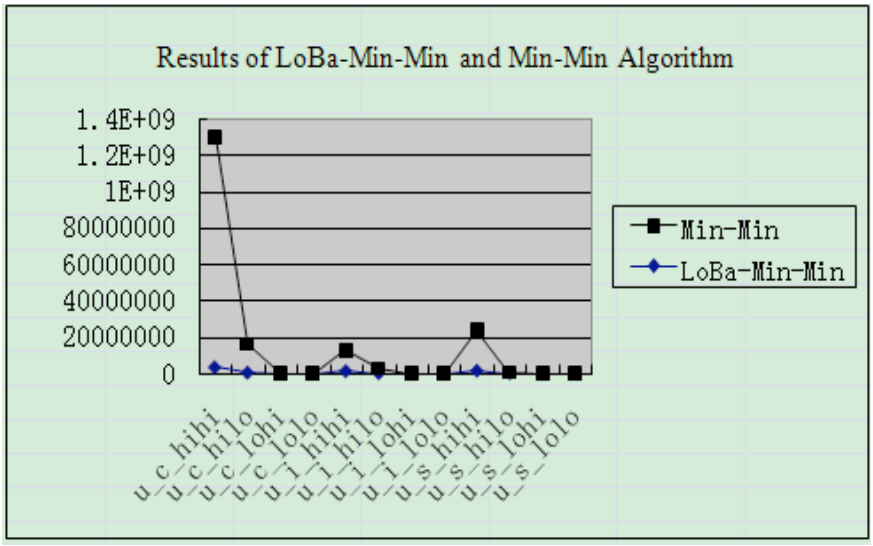

Fig. (5). Results of LoBa-Min-Min and Min-Min Algorithm.

Table 3. Results of LoBa-Min-Min-SPA and Min-Min-SPA

Instances	LoBa-Min-Min-SPA(ms)	Min-Min-SPA(ms)
u_c_hihi	1.52390477×10^{9}	1.56373389×10^{9}
u_c_hilo	1.77107088×10^{8}	1.83389088×10^{8}
u_c_lohi	194.35764	194.35764
u_c_lolo	186003.36	192255.52
u_i_hihi	1.15482768×10^{8}	1.23916472×10^{8}
u_i_hilo	3.3976628×10^{7}	3.7381328×10^{7}
u_i_lohi	121.98419	121.98419
u_i_lolo	31810.09	34777.97
u_s_hihi	2.28789072×10^{8}	2.44194832×10^{8}
u_s_hilo	5.6848404×10^{7}	6.03354×10^{7}
u_s_lohi	171.50237	171.50237
u_s_lolo	52679.555	56099.004

According to Table **3**, we can give Fig. (**6**).

superior to the Min-Min scheduling algorithm, improving the system throughput.

6. SET PAIR ANALYSIS METHOD

The binary connection number a+bi of Set Pair Analysis (SPA) proposed by Professor Zhao Ke-qin firstly in 1989 [13]. SPA is a new soft computing method which can express and process the synthetic uncertainty caused by fuzzy, random, indeterminate known uncertainty etc. Now, It gradually becomes a new uncertainty theory by which we can research certainty and uncertainty as a whole now.

A Set Pair is a system made up of two sets (A,B) which there are some similar attributes or tight relations, such as (Control, Decision), (Computers, Human Brains), (Products, Sell), etc can be seen examples of Set Pair under specified conditions. The main thought of SPA is as follows: To two sets A, B under specified conditions, first analyzing their identical, discrepancy and contrary properties or attributes, and then describing them quantificationally, expressing their relations by a formula called connection degree finally.

6.1. Connnection Degree

We gave two sets A and B, and both of them have N attributes. Their connection degree can be defined and denoted by $u(A,B)$.For brevity, we will usually denote $u(A,B)$ simply as u,Thus

$$u=\frac{S}{N}+\frac{F}{N}i+\frac{P}{N}j \tag{1}$$

where $S+F+P=N$ (2)

and S is the number of their identical attributes, P is the number of their contrary attributes, the $F=N-S-P$ is the num-

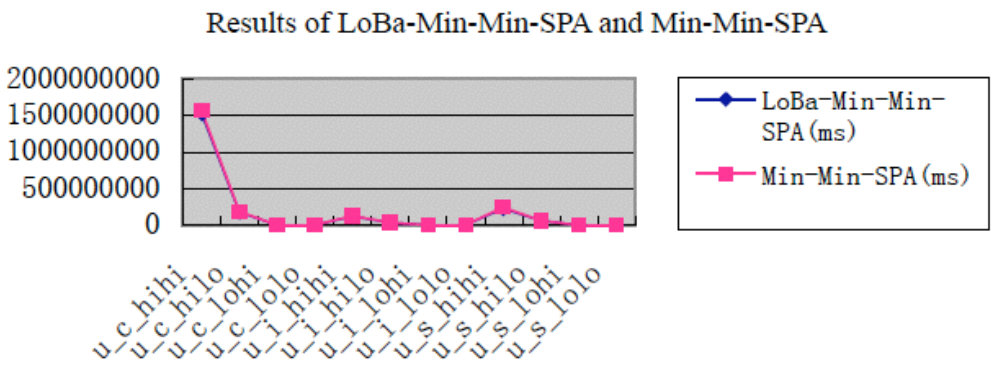

Fig. (6). Results of LoBa-Min-Min-SPA and Min-Min-SPA Algorithm.

ber of their discrepancy attributes. $\frac{S}{N}$, $\frac{F}{N}$, $\frac{P}{N}$ are called identical degree, discrepancy(uncertain) degree, and contrary degree respectively. The $i \in [-1,1]$, $j = -1$ are called discrepancy coefficient, contrary coefficient respectively, and the value of i needs further analysis to determine it according to a practical applications. But the i, j are usually as signs of discrepancy degree and contrary degree when we doesn't computing the value of connection degree μ in some situations such as only concerning about operations or macroscopical analysis.

If let $a=\frac{S}{N}$, $b=\frac{F}{N}$, $c=\frac{P}{N}$, then the formula (1) and (2) can be rewritten as fellows.

$$\mu=a+bi+cj \quad (3)$$

where $a+b+c=1$ (4)

6.2. Binary Connection Number

We say that a $\mu=a+bi+cj$ is a ternary connection number if a,b,c are arbitrary nonnegative real number, and i,j are discrepancy coefficient, contrary coefficient respectively. $\mu=a+bi$ is a binary number when $c=0$.

Obviously, the connection number (binary or ternary) is an extension and generalization of connection degree by deleted the constraint condition a+b+c=1, We only use binary connection number to represent the uncertain Execution Time to Compute of grid tasks in this paper, that is Min-Min-SPA and LoBa-Min-Min-SPA scheduling algorithms.

The experiment was conducted in the above experimental environment and instance benchmark proposed by Braun et al, the ETC matrix generated based on 512 tasks and 16 resources in instances benchmark. We generate a and

$b \in [0,a]$ randomly in the uniform distribution. the results of LoBa-Min-Min-SPA and Min-Min-SPA scheduling algorithm is shown in Table **3**.

7. CONCLUSIONS AND FUTURE WORK

Due to the disadvantages in load imbalance and low resource utilization of Min-Min scheduling algorithm, this paper has improved Min-Min algorithm. The main scheduling policy is rescheduling, tasks on heavy-loaded resources will be reallocated to the resources with the smaller completion to balance load, improve resource utilization, and to produce a smaller Makespan. The performance of the proposed LoBa-Min-Min scheduling algorithm is significant superior to Min-Min scheduling algorithm. The main tasks for the future are to optimize the rescheduling policy and apply LoBa-Min-Min scheduling algorithm to the actual grid computing system.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work was financially supported by National Natural Science Foundation of China (61272124), Hebei Provincial Natural Science Foundation of China (F2012203087) and National Natural Science Foundation of China (61073060).

REFERENCES

[1] A. Foster, C. Kasselman and S. Tuecke, "*The Anatomy of the Grid*", 2001.

[2] Gird Computing [M] in Chinese, translated by Zhan xiao-su, Zhang shao-hua., Tsinghua University Press, Beijing, China, pp.21-36, 2005.

[3] H. Xiaoshan, X-H. Sun and G. Laszewski, "QoS Guided Min-Min Heuristic for Grid Task Scheduling", *Journal of Computer Science and Technology*, vol. 18, pp. 442-451, 2003.

[4] T. Kokiavani and G. Amalarethinam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing", *International Journal of Computer Application*, vol. 20, pp. 0975-8887, 2011.

[5] H. Decai, Y. Yuan, Z. Li-jun and Z. ke-qin, "Research on tasks scheduling algorithm for dynamic and uncertain computing grid based on a+bi connection number of SPA", *Journal of Software*, vol.4, pp.10, 2009.

[6] R. Armstrong, D. Hensgen and T. Kidd, "The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions", In 7th IEEE Heterogeneous Computing Workshop(HCW'98), pp.79-87, 1998.

[7] R.F. Freund and H.J. Siegel, "Heterogeneous processing", *IEEE Computer*, vol. 26, pp. 13-17, 1993

[8] M. Singh and P.K. Suri, "$QPS_{Max\text{-}Min<>Min\text{-}Min}$: a QoS based predictive max-min, min-min switcher algorithms for job scheduling in a Grid", *Information Technology Journal*, vol. 7, pp. 1176-1181, 2008.

[9] G. K. Kamalam and V. Murali Bhaskaran, "A new heuristic approach: min-mean algorithm for scheduling meta-tasks on heterogeneous computing systems", *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, pp. 24-31, 2010.

[10] G. K. Kamalam and V. Murali Bhaskaran, "New Enhanced Heuristic Min-mean Scheduling Algorithm for Scheduling Meta-Tasks on

Heterogeneous Grid Environment, *European Journal of Scientific Research*, vol. 70, pp. 423-430, 2012.

[11] T.D.Braun, H.J. Siegel and N. Beck, "*A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems*", *Journal of Parallel and Distributed Computing*, vol. 61, pp. 810-837, 2001.

[12] T.D. Braun, H.J. Siegal, N. Beck, L.L. Boloni, M. Maheshwaran, A. Reuther, J. Robertson, M. Theys, B. Yao, D. Hensgen, and R. Freund, "A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing system", In 8th IEEE Heterogeneous Computing Workshop(HCW'99), pp. 15-29, 1999.

[13] Z. Ke-qin and X. Ai-li, "Set pair theory-a new theory method of non-define and its application", *Systems Engineering*, pp. 18-24, 1996.

Received: August 13, 2013 Revised: September 11, 2013 Accepted: September 11, 2013