# Research on Tasks Scheduling Algorithms for Dynamic and Uncertain Computing Grid Based on *a+bi* Connection Number of SPA

HUANG Decai
College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou, China
Email: hdc@zjut.edu.cn

YUAN Yuan
College of Computing Sciences, New Jersey Institute of Technology, Newark, New Jersey, USA
Email: yy916@hotmail.com

ZHANG Li-jun
College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou, China
Email: zhlj206@163.com

ZHAO Ke-qin
Institute of Zhuji connection mathematics. Zhuji,China
Email: zjzhaok@sohu.com

*Abstract*—**Task scheduling algorithms are key techniques in task management system of computing grid. 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. In this paper, Set Pair Analysis (SPA), a new soft computation method is used to process the synthetic uncertainty in the task scheduling of a computing grid. After introducing SPA and its application, the paper goes on to introduce the definition of connection number to express the uncertain Expected Time to Compute of tasks, analysis operation properties and linear order relation suitable for computing grid scheduling. Three online uncertain dynamic scheduling algorithms, OUD_OLB, OUD_MET, OUD_MCT, and three batch uncertain dynamic scheduling algorithms BUD_Min-min, BUD_Min-max, BUD_Surferage, are presented for the uncertain dynamic computing grid. Theoretical analysis and experimental results illustrate that these algorithms are capable of representing the dynamics and uncertainty in a computing grid environment. These algorithms are the generalization of traditional grid scheduling algorithms, and they possess high value in theory and application in a grid environment. Certainly it will be a new method to design tasks scheduling algorithm in uncertain computing grid environment.**

*Index Terms*—**computing grid; Dynamic;uncertainty; task scheduling; algorithm**

## I. Introduction

Grid technique will be the key feature of new application on Internet, and it is already the front research area of computer and communication. Although there are several kinds of computing grid such as data grid, storage grid, service grid and manufacturing grid etc, tasks scheduling system with the function of tasks management, task scheduling and resource management is always the key system of any computing grid[1,2]. A suitable scheduling model and optimal scheduling algorithm is desirable for grid application. The efficiency of tasks scheduling algorithm directly influences the efficiency of network communication, reliability of computing grid, makespan of the grid system[3]. There are many research on the realm of grid task scheduling[3,4,5,6,7], but these algorithms seldom concern the dynamics and uncertainty in the grid environment. The traditional task scheduling algorithms cannot be applied effectively in an open, heterogeneous, uncertain and dynamic grid environment of real world.

Dynamics and uncertainty of grid system makes it different from the traditional heterogeneous computing systems, and how to represent and treat with these dynamics and uncertainty is considered as one of the "Top Ten" questions[8] unanswered. Just as pointed out by paper [9], the dynamics and uncertainty of computing grid render the ETC (Expected Time to Compute) of a task to be uncertain. Paper [9] proposed an artificial immune algorithm for task scheduling in grid environment in which the ETC of tasks is represented by fuzzy uncertainty, and paper [10] uses stochastic method to get a prediction of task execution times. However, apart from the fuzzy uncertainty factor in grid systems, there are some other uncertain factors such as random, indeterminate-known factors and unexpected incidents and so on. Sometimes, the uncertainty of task execution time is usually the combined effect caused by fuzzy, random, indeterminate-known uncertainty and unexpected incidents and so on. And thus any single uncertain method, such as fuzzy or random cannot represent and process well the synthetic uncertainty caused by more than one uncertain factor. Now the


doi:10.4304/jsw.4.10.1102-1109

binary connection number $a+bi$ of Set Pair Analysis (SPA) proposed by ZHAO Ke-qin[11] gives a new method to represent and process the synthetic uncertainty in dynamic and uncertain computing grid. SPA is applied successfully in many fields such as network planning, weather forecast, incomplete information system and product design[12-17].

In this paper, we use binary connection number of SPA, a new soft computing method[16] to represent the uncertain Execution Time to Compute of tasks and to process the synthetic uncertainty in task scheduling of a computing grid. After introducing SPA and its application briefly, three online uncertain dynamic scheduling algorithms, OUD_OLB, OUD_MET, OUD _MCT, three batch uncertain dynamic scheduling algorithms BUD_Min-min, BUD_Min-max, BUD _Surferage, are presented for a computing grid. Theoretical analysis and experimental results illustrate that these algorithms can represent the dynamics and uncertainty of “expected time to compute a task” in the computing grid environment. Certainly it will be a new modeling and algorithm designing method for computing grid scheduling.

## II. INTRODUCTION OF SPA AND CONNECTION NUMBER

Set Pair Analysis (SPA) is a new soft computing method which can express and process the synthetic uncertainty caused by fuzzy, random, indeterminate-known uncertainty etc. it is presented by Professor Zhao Ke-qin firstly in 1989[11]. After twenty years theory and application research about SPA, 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. Then we can research a series of issues about the system, such as decision, control, evaluation, simulation, evolution and mutation, etc.

### A. Connection Degree

***Definition 1***[11]. Let A, B be two sets, and both of them have N attributes. We define their connection degree and denote it by $\mu(A,B)$. For brevity, we will usually denote $\mu(A,B)$ simply as $\mu$. Thus

$$\mu = S/N+F/Ni+P/Nj \qquad (1)$$

$$\text{where } S+F+P=N \qquad (2)$$

and S is the number of their identical attributes, P is the number of their contrary attributes, the F=N-S-P is the number of their discrepancy attributes. S/N,F/N,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 $\mu$ in some situations such as only concerning about operations or macroscopical analysis.

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

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

$$\text{where } a+b+c=1 \qquad (4)$$

The thought of connection degree is directly coming from the decision analysis research. For example, let 10 experts (set A) decide a project (set B) whether to be put in practice by ballots. If the voting results are 5 persons voting for it, 2 experts voting against it, and other 3 abstain from voting. Thus the voting results can be expressed by connection degree as fellow.

$$\mu=5/10+3/10i+2/10j=0.5+0.3i+2j$$

Thus, we usually can do further research and analysis according practice application by the connection degree $\mu$.

### B. Binary Connection Number

***Definition 2***[12].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 connection 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,and the main theory meaning of the connection number is extended the conception of number. We only use binary connection number to represent the uncertain Execution Time to Compute of grid tasks in this paper, thus we give the relationship of constant, variables, uncertain variable on the macroscopy and microscopy (Table 1). From table 1 we can see that, on the one hand, binary connection number linked an instant with an interval number, on the other hand, it linked macroscopical certainty quantity with microcosmic uncertainty quantity of a system.

TABLE 1.
THE RELATIONSHIP OF CONSTANT, VARIABLES,UNCERTAIN VARIABLE ON THE MACROSCOPY AND MICROSCOPY

| quantity | expressing | macroscopy | microscopy |
|---|---|---|---|
| instant | A | certain | certain |
| variable | X | uncertain | certain |
| uncertain | a+bi | certainty | uncertain |
| super uncertain | unknown | uncertain | uncertain |

### C. Binary Connection Number and probability

According to the classical definition of probability, the probability (denoted as p) of an event to occur was defined as number of cases favorable for the event, over the number of total outcomes possible in an equiprobable sample space. For example, if the event is "occurrence of an even number when a die is rolled", the probability is given by p=3/6=0.5, since 3 faces out of the 6 have even numbers and each face has the same probability of appearing, so a probability p is a real number and $p\in[0,1]$.

Firstly it is possible to transfer a probability p to a binary connection number u=p+(1-p)*i*. Secondly, it is necessary to transfer a probability p to μ=p+(1-p)*I* in some cases. Because probability is a mathematical describing to stochastic uncertainty just on the macroscopy level, and expressed by a certainty probability p of stochastic uncertainty, but on the microscopy level, the stochastic uncertainty is still representing its essential uncertain. So when we need to consider the influence and effect of stochastic uncertainty on macroscopy and microscopy level at one time, it is very necessary and useful to transfer a probability p to μ=p+(1-p)*i*.

*D. Binary Connection Number and interval number*

Let I=[C,D] is a positive interval number, C≥0 and D>C, we can transfer this interval U to a binary connection number μ=C+(D-C)*i* where *i*∈[−1,1], thus the μ is another interval number instead of I, its variation interval is [C-(D-C),C+(D-C)], but the length of μ is doubleness of U's length. If we doesn't require this extention of length, we can restrict the *i*∈[0,1]. I=[C,D] and μ=C+(D-C)*i* are different despite the restriction of i∈[0,1] (see table 2) and μ=C+(D-C)*i* includes more uncertain information than I=[C,D].

TABLE 2.
THE DIFFERENCES BETWEEN THE INTERVAL NUMBER [C,D] AND BINARY CONNECTION NUMBER μ

| interval number I=[C,D] | binary connection number μ=C+(D-C)i |
|---|---|
| The two end points is certain middle of [C,D]is uncertain | The two end points is uncertain middle of μ is certain |
| D>C | can be C>(D-C) or C< (D-C) |
| Expected value is (C+D)/2 | Expected value is C |
| There is not other information besides C and D | There is uncertain coefficient *i* besides C and D |

## III. Operations and Linear Order

### *A. Arithmetic operations*

From above we known that connection number is an important concept of SPA. Now we give the operation definition of binary connection number $u=a+bi$ used in uncertain computing grid.

If $u=a+bi$ is a binary connection number, then we can it to represent the uncertain execution time of a task on computing grid. Where $a$ is normal finish time of a task. $b$ is the variational amplitude of the uncertain time, it usually represents the possible delay time or saving time to process the task influenced by random, fuzzy, indeterminate-known factors. $i$∈[–1,1] is the uncertainty coefficient, and it is usually used as a sign of uncertainty and its value represents the variational direction and size of $b$. It is usually determined by the practical application problem. Specially, denote $u=0$ when $a=0$ and $b=0$. We denote the set of all binary connection number by $R_{cn}$.

For example, if it takes 100 second to finish task α on a computing grid under normal condition. However, if some new tasks are added to the machine concurrently or some tasks on the machine are terminated, it may delay 31 seconds or may save 31 seconds to finish task α on the machine. And thus the uncertainty execution time of task α on the machine may be represented as 100+31*i* by binary connection number, where *i* is an uncertainty coefficient and *i*∈[–1,1]. In the same way, we may simply denote it as 100+31*i* if the task influenced by uncertainty will delay 28s or save 31s at most.

Now we give some basic operations of binary connection numbers.

Obviously $i\times i$ *is* uncertain, because $i$ is uncertain coefficient, thus we can give the following definition.

***Definition3***[12]:If $i$ is the uncertain coefficient of binary connection number, then $i=i^2=\cdots=i^n$, (n=1,2,3,……).

***Definition 4***[12]:If $u_1=a_1+b_1i$, $u_2=a_2+b_2i\in R_{cn}$,we define their addition as a connection number $u=a+bi$, denote it by $u_1+u_2$, where $a=a_1+a_2$, $b= b_1+b_2$.

***Definition*** **5**[12]:If $u_1=a_1+b_1i$, $u_2=a_2+b_2i\in R_{cn}$,we define their addition as a binary connection number $u=a+bi$, denote it by $u_1+u_2$, where $a=a_1+a_2$, $b= b_1+b_2$.

From definition 5 we know the addition of binary connection numbers has the commutative and associative properties.

Next paragraph is the definition of subtraction of binary connection numbers.

***Definition*** **6**: If $u_1=a_1+b_1i$, $u_2=a_2+b_2i\in R_{cn}$, we define their subtraction as a binary connection number $u=a+bi$, denote it by $u_1-u_2$, where $a=a_1-a_2$, $b= b_1-b_2$.

We can easily get the following inference from definition 4, definition 5 and definition 6.

***Inference*** *1*: If $u_1=a_1+b_1i$, $u_2=a_2+b_2i \in R_{cn}$, Then $u_1=u_2$ if and only if $a_1=a_2$, $b_1=b_2$ .

***Definition*** **7**: Let U={u1,u2,⋯,un} where $u_k=a_k+b_ki$ (k=1,2,⋯,n), U's mean is a binary connection number and typically denoted with a horizontal bar over the variable $\overline{U}=\overline{A}+\overline{B}i$ ,and is computed in accordance with the following methods.

$$\overline{U}=\overline{A}+\overline{B}i=\frac{1}{n}\sum_{k=1}^{n}u_k=\frac{1}{n}\sum_{k=1}^{n}(a_k+b_ki)$$
$$=\frac{1}{n}\sum_{k=1}^{n}a_k+\frac{1}{n}\sum_{k=1}^{n}b_ki$$

***Definition*** *8*: If $u_1=a_1+b_1i$, $u_2=a_2+b_2i\in R_{cn}$, we define their product as a binary connection number $u= a+bi$, denote it by $u_1\times u_2$, where $a=a_1a_2+b_1b_2$, $b= a_1b_2+a_2b_1$.

***Definition*** *9*: If $u_1=a_1+b_1i$, $u_2=a_2+b_2i \in R_{cn}$, and if there is a binary connection number $u=a+bi$ and $u_1=u\times u_2$, we call $u$ is the quotient of $u_1$ divided by $u_2$, and denote it by $u=u_1\div u_2$. In this case, we say $u_1$ can be divided by $u_2$, otherwise $u_1$ can not be divided by $u_2$.

Furthermore, we can easily prove the following theorem.

***Theorem 1***: If $u_1=a_1+b_1i$, $u_2=a_2+b_2i\in R_{cn}$, then $u_1$ can be divided by $u_2$ if and only if the following matrix M is not singular.

$$M=\begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix}$$

And thus, the quotient $u=a+bi$ of $u_1$ divided by $u_2$ can be gotten, where $a$ and $b$ are the solutions of the following linear equation.

$$\begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$$

Of course, we can also get some properties of the product of two binary connection numbers, such as commutative and associative laws etc. we do not give the proof because of the limit on length of paper.

***Definition 10:*** The square root of u=a+b$i$ is a binary connection number and denoted by $\sqrt{u} = \sqrt{a + bi} = a'+b'i$, Where $a' = \sqrt{a}$ and $b' = -\sqrt{a} \pm \sqrt{a+b}$ ,because (a'+b'$i$) (a'+b'$i$)= a+b$i$, namely $a'^2+2(a'b'+b'^2)i$= a+b according to the product definition of two binary connection number.

*B. Grid scheduling and linear order*

Meta-task scheduling problem is a basic and key problem, because many new scheduling model and new scheduling algorithms are derived from it. This scheduling problem can be described as following: if there are $n$ independent tasks T={$t_1$, $t_2$,⋯,$t_n$} and $m$ computing resources M={$M_1$, $M_2$,⋯,$M_m$}, the execution time of task $t_i$ on machine $M_j$ is E(i,j). A scheduling is to assign each task to someone computing resource and the object is to minimize makespan of the computing grid. This problem seems very simple because it has only single object without any restriction, however it is not a trivial problem but a NP-Hard problem. Thus many heuristic algorithms are proposed, such as Max-min, Min-min and Sufferage etc. According to above description of traditional meta-task scheduling problem, we use binary connection number E(p,q)=$a_{pq}$+$b_{pq}i$ to represent the uncertainty execution time of task $t_p$ on computing resource $M_q$. Thus we get a new scheduling algorithm based on binary connection number.

Because the expected execution time of task $t_p$ on machine $M_q$ and the makespan of system are uncertainty value represented by binary connection number according to the definition and operations properties of binary connection number, we must know how to compare one binary connection number $u_1$ with another $u_2$, and know which one is lager, namely we ought to build linear order on the set $R_{cn}$ before we build the scheduling model and give task scheduling algorithms of uncertain computing grid.

Because the value of a binary connection number $u=a+bi$ is uncertain, we can definition different order on set $R_{cn}$ according different practical application problem. Paper [11] gives a state order on set $R_{cn}$ according to the cases of $a/b$ is larger, less or equal to 1. But this is only a partial order not a linear order. Paper [12] gives a different partial order on set $R_{cn}$ according to the requirement of uncertain network planning, and gives some new concepts such as primary critical paths, 2nd critical paths and 3rd critical paths etc, and applied successfully in network planning. Both orders defined above are partial order instead of linear order. It means some elements in set $R_{cn}$ may not be comparable. But we definitely need a linear order on $R_{cn}$ for gird task scheduling, namely every pair of elements in set $R_{cn}$ ought to be comparable.

In the following paragraphs we define two type of linear order on set $R_{cn}$ according to the requirement of grid task scheduling.

***Definition 11***: Let $u_1=a_1+b_1i$, $u_2=a_2+b_2i$, $i\in R_{cn}$. If $a_1<a_2$, or $a_1=a_2$ and $b_1<b_2$, we define $u_1$ is less than $u_2$, and denote it by $u_1<_p u_2$.

Obviously, the relation $<_p$ is a linear order on set $R_{cn}$. It compares $a_1$ with $a_2$ firstly, If $a_1=a_2$, then the order determined by $b_1$ and $b_2$. This definition is reasonable because the $a_1$ and $a_2$ are determination value. This order is called a worst case order, or pessimistic order, because when $a_1=a_2$, $u_1<_p u_2$ if and only if $b_1<b_2$.

From definition 11 we can get the following theorem easily.

***Theorem 2***: if $u_1$, $u_2\in R_{cn}$, then one and only one of the following three conclusions must be true.

(1) $u_1= u_2$; (2) $u_1<_p u_2$; (3) $u_1>_p u_2$

Now we give an optimistic order definition on set $R_{cn}$.

***Definition 12:*** Let $u_1=a_1+b_1i$, $u_2=a_2+b_2i$ , $i\in R_{cn}$. If $a_1<a_2$, or $a_1=a_2$ and $b_1>b_2$, we define $u_1$ is less than $u_2$, and denote it by $u_1<_o u_2$ .

Similar to theorem 2, we can prove the following theorem 3.

***Theorem 3***: If $u_1$, $u_2\in R_{cn}$, then one and only one of the following three conclusions must be true.

(1) $u_1= u_2$; (2) $u_1<_o u_2$; (3) $u_1>_o u_2$

In the following section, a $u\in R_{cn}$ is “less”, or “larger” etc, which is under one of the mean of linear order $<_p$ or $<_o$. Which linear order to be used in a practical grid system is determined by the current state prediction of a computing grid.

## IV. Dynamic Scheduling Algorithm

*A. Parameters definition*

(1) $M_q$: One the one side it denotes the $q^{th}$ computing resource, on the other hand, it denotes the tasks set assigned to $M_q$, and all tasks in the $M_q$ is on the order of its assigned time.

(2) $S_q$: The time of resource $M_q$ to start a new task when $M_q$ finished all tasks assigned to it.

(3) E(p,q):Expected execution time if task $t_p$ assigned to resource $M_q$.

(4) C(p,q):Completion time of a new task $t_p$ assigned to resource $M_q$, C(p,q)=$S_q$ +E(p,q).

(5) CT: a set of ordered pair ($t_p$, $M_q$), it means task $t_p$ is going to be assigned to resource $M_q$ and C(p,q) =min{C(p,k) | k=1,2,⋯,m} in a loop of scheduling algorithm.

⑹ Assigned$_q$: a flag of $M_q$, it denotes if $M_q$ is assigned a new task in a loop of scheduling algorithm. Its value is either 'True' or 'False'.

*B. Online Dynamic algorithms*

⑴ Algorithm OUD_OLB

Input:Tasks set T, Computing resources set M, ETC Matrix based on connection number and $S_q \geq 0$

Output: Scheduling results $M_1, M_2, \cdots, M_m$.

Step0. for every $M_q \in M$
  $M_q = \phi$
  endfor
Step1. for every $t_p \in T$
  $S_{min} = \min\{S_1, S_2, \cdots, S_m\}$
  $M^{min} = \{M_k \mid S_k = S_{min}, k=1,2,\cdots,m\}$
  Find a computing sources $M_q$ from $M^{min}$ at random
  $M_q = M_q \cup \{t_p\}$, $T = T - \{t_p\}$
  $S_q = S_q + E(p,q)$
Step3. endfor

⑵ Algorithm OUD_MET

The input and output is the same as that of Algorithm OUD_OLB.

Step0. for every $M_q \in M$
  $M_q = \phi$
  endfor
Step1. for every $t_p \in T$
  $E_{min} = \min\{E(p,1), E(p,2), \cdots, E(p,m)\}$
  $M^{min} = \{M_k \mid E(p,k) = E_{min}, k=1,2,\cdots,m\}$
  Find a computing sources $M_q$ from $M^{min}$ at random
  $M_q = M_q \cup \{t_p\}$, $T = T - \{t_p\}$
Step3. endfor

⑶ Algorithm OUD _MCT

The input and output is the same as that of Algorithm OUD_OLB.

Step0. for every $t_p \in T$
  for every $M_q \in M$
  $C(p,q)=0$, $M_q = \phi$, $S_q = 0$,
  endfor
  endfor
Step1. for every $t_p \in T$
Step2. for every $M_q \in M$
  $C(p,q) = S_q + E(p,q)$
  endfor
  $C_{min} = \min\{C(p,1), C(p,2), \cdots, C(p,m)\}$
  $M^{min} = \{M_k \mid C(p,k) = C_{min}, k=1,2,\cdots,m\}$
  Find a computing sources $M_q$ from $M^{min}$ at random
  $M_q = M_q \cup \{t_p\}$, $T = T - \{t_p\}$
  $S_q = S_q + E(p,q)$
Step3. endfor

*C. Batch Dynamic algorithms*

⑴ algorithms BUD_Min-min

The input and output is the same as that of Algorithm OUD_OLB.

Step0. for every $M_q \in M$
  $M_q = \phi$
  endfor
Repeat the following steps until ($T = \phi$)
Step1. CT＝Φ
Step2. for every $t_p \in T$
Step3. for every $M_q \in M$
  $C(p,q) = S_q + E(p,q)$;
Step4. endfor
  Find the minimum earliest completion time of $t_p$, and the corresponding resource $M_q$, Let $CT = CT \cup \{<t_p, M_q>\}$
Step5. endfor
Step6. $C_{min} = \min\{C(p,q) \mid <t_p, M_q> \in CT\}$
  $M^{min} = \{M_k \mid C(p,k) = C_{min} \text{ and } <t_p, M_k> \in CT\}$
Step7. Choose any one $<t_r, M_k>$ from $M^{mim}$, Let $M_k = M_k \cup \{t_r\}$, $T = T - \{t_r\}$
  $S_k = S_k + E(r,k)$

⑵ Algorithm BUD_Max-min

We can get Algorithm BUD_Max-min if we just use the following Step6. and Step7. to replace the Step6. and Step7. of Algorithm BUD_Min-min. Because the Input, Output, and Step0 to Step5 of Algorithm BUD_Max-min are the same as Algorithm BUD_Min-min.

Step6. $C_{max} = \max\{C(p,q) \mid <t_p, M_q> \in CT\}$
  $M^{max} = \{M_k \mid C(p,k) = C_{max} \text{ and } <t_p, M_k> \in CT\}$
Step7. Choose one $<t_r, M_k>$ from $M^{max}$, Let $M_k = M_k \cup \{t_r\}$, $T = T - \{t_r\}$
  $S_k = S_k + E(r,k)$

⑶ Algorithm BUD_Sufferage

The input and output are the same as that of Algorithm OUD_OLB.

Step0. For every $M_q \in M$
  Assigned$_q$=False, $M_q = \phi$
  endfor
Repeat the following steps until($T = \phi$)
Step2. for every $t_p \in T$
Step3. for every $M_q \in M$
Step4. $C(p,q) = S_q + E(p,q)$;
Step5. endfor
Step6. for every $M_q \in M$
  Find the minimum earliest completion time of $t_p$, and the corresponding resource $M_q$ and completion $C(p,r)$
Step7. Find the 2nd minimum earliest completion time of $t_p$, and the corresponding resource $M_s$ and completion $C(p,s)$
  $SuffV_p = C(p,s) - C(p,r)$
Step8. endfor
Step9. if Assigned$_q$=False of $<t_p, M_q>$
  $M_q = M_q \cup \{t_p\}$, $T = T - \{t_p\}$;
  Assigned$_q$=True.

Step11. else if $t_k$ is already assigned to $M_q$ in this loop and $SuffV_p$ is less than $SuffV_k$, let

$M_q=M_q\cup\{t_p\}-\{t_k\}$

$T=T\cup\{t_k\}-\{t_p\}$;

endif.

endfor.

Step12. for every $M_q \in M$

$$S_q=\sum_{t_p\in M_q}E(p,q)$$

$Assigned_q$=False

endfor

## V. Example and Numerical Analysis

### A. Examples

**Example** 1. Let the number of resources m=3, the number of tasks n=5, and

$$ETC=\begin{pmatrix}101+32i & 100+22i & 102+17i\\ 100+3i & 99+9i & 99+7i\\ 101+12i & 99+6i & 98+3i\\ 100+5i & 102+8i & 101+6i\\ 97+16i & 101+20i & 97+19i\end{pmatrix}$$

We are going to use algorithm BUD_ Min-min to get the scheduling result with the linear order "$<_p$".

***Solution***: According to of algorithm BUD_Min-min, each step of the loop assigns one task to a machine, here we let $(S_1,S_2,S_3)=\mathbf{(0,0,0)}$.

(1) Because $(S_1,S_2,S_3)=(0,0,0)$, and the set CT= $\{C(t_1,M_2)=100+22i,\ C(t_2,M_3)=99+7i,\ C(t_3,M_3)=98+3i,\ C(t_4,M_1)=100+5i, C(t_5,M_1)=\boldsymbol{97+16i}\}$, where $<t_p, M_q>$ means that $t_p$ has the minimum earliest completion time on the corresponding resource $M_q$ under the meaning of the linear order "$<_p$". The minimum binary connection number in CT is $C(t_5,M_1)$ $\boldsymbol{=97+16i}$. Thus we assign $t_5$ to $M_1$, and get $(S_1,S_2,S_3)$ =( 97+16$i$, 0, 0).

(2) For the rest tasks $t_1,t_2,t_3,t_4$, we get the set CT=$\{C(t_1,M_2)=100+22i,\ C(t_2,M_3)=99+7i,\ C(t_3,M_3)\ \boldsymbol{=98+3i},\ C(t_4,M_3)=101+6i\}$. According to the same reason of CBU_Min-min, we assign $t_3$ to $M_3$,and get $(S_1,S_2,S_3)$=(97+16$i$, 0, 98+3$i$);

(3) For the rest tasks $t_1,t_2,t_4$, we get the set CT=$\{C(t_1,M_2)=100+22i, C(t_2,M_2)\boldsymbol{=99+9i},\ C(t_4,M_2)=102+8i\}$. We assign $t_2$ to $M_2$, and then get $(S_1,S_2,S_3)$=(97+16$i$, 99+9$i$, 98+3$i$).

(4) For the rest tasks $t_1,t_4$, we get the set CT=$\{C(t_1,M_1)=198+48i,\ C(t_4,M_1)\boldsymbol{=197+21i}\}$, and we assign $t_4$ to $M_1$,thus get $(S_1,S_2,S_3)$=(197+21$i$, 99+9$i$, 98+3$i$);

(5) For the rest task $t_1$, we get the set CT=$\{C(t_1,M_2)=199+31i\}$, and we assign $t_1$ to $M_2$, and get $(S_1,S_2,S_3)$=(197+21$i$, 199+31$i$, 98+3$i$).

According to definition 6, the scheduling result is makespan=199+31$i$. It tells us that all tasks will be completed within 199s at normal condition. If some resources get more concurrent tasks or some concurrent tasks on some machine cancelled, or affected by other uncertain factors, all tasks to be completed will delay 31s or save 31s at most. But how long the delay or saving is depends on the forecasting value of uncertain coefficient $i$. For example, if the grid system is overload and heavily slows down lately, we can set $i$=1 to forecast the makespan, namely let makespan =230. If the grid system slow down a little lately, we can let $i$=31/(199+31) ≈0.1348 to get the estimate makespan=203.1788. In this way the value of $i$ of the binary connection number 199+31$i$ itself can be called "formula oriented estimate"[12]. There are other methods to estimate the value of $i$, such as "estimate step by step", "estimate by experts", "estimate by synthetic method" etc[12]. In practice, we ought to choose a suitable estimate method to forecast and analysis the scheduling makespan according to the state of grid system. Just on this occasion, the scheduling algorithms based on the binary connection number have some distinct advantages to represent and process the uncertainty and dynamics of computing grid.

### B. Numerical analysis

#### *(1) Generating experiment data*

In experiments, we assume grid system with 20 computing resources, namely m=20, and the number of dependent tasks is between 40～200.

The expected execute time in ETC matrix is generated by the way of paper [17], firstly we generate $a_{pq}$, and $b_{pq}\in[0,a_{pq}]$ randomly in the uniform distribution. The method of paper [17] needs 4 input parameters. We set $\mu_{task}$=100, $\mu_{mach}$=100, the other two parameters $V_{task}$ and $V_{mach}$ represent the consistent of matrix ETC and heterogeneous of machine respectively. Then we set $V_{task}$= 0.1 or $V_{task}$=0.6, $V_{mach}$=0.1 or $V_{mach}$=0.6.

#### *(2) Analysis of experiment*

- Batch Dynamic Case

We use the linear order "$<_p$" and the average makespan of 50 examples as the evaluating criterion. The experiment results are showing from figure 1 to figure 4, where $V_t$ denotes $V_{task}$, $V_m$ denotes $V_{mach}$.

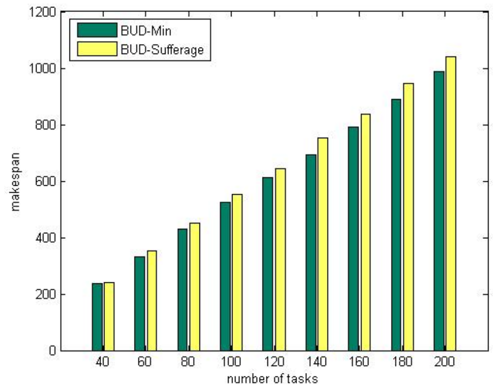


Fig.1 m=20, $v_t$=0.1, $v_m$=0.1

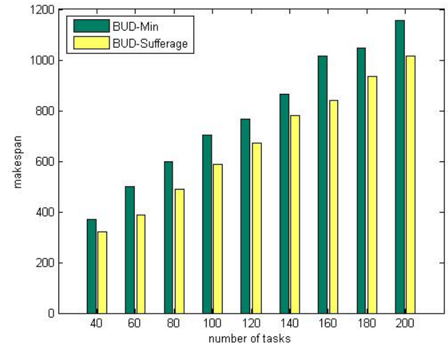

Fig.2 m=20, $v_t$=0.6, $v_m$=0.1

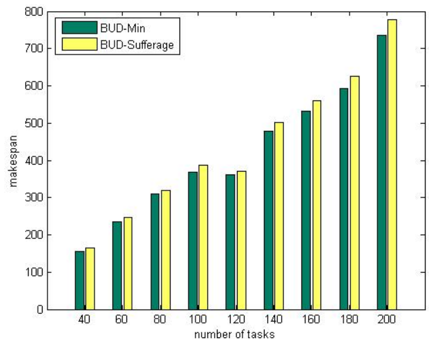

Fig.3 m=20, $v_t$=0.1, $v_m$=0.6

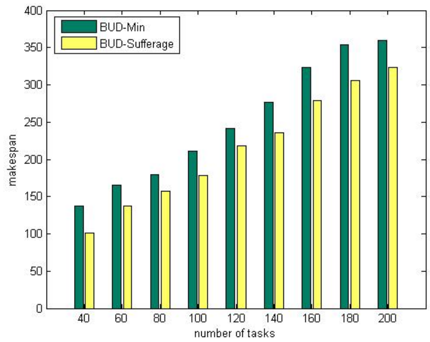

Fig.4 m=20, $v_t$=0.6, $v_m$=0.6

Because we find out that the performance of algorithm BUD-Max-min is extremely bad compared with the algorithm BUD_Min-min and BUD_Sufferage, we do not show the experiment results of BUD_Max-min in figure 1 to figure 4.

From figure 1 and figure 3 we can see that the performance of algorithm BUD-Min-min is better than that of algorithm BUD_Sufferage in the case of consistent matrix ETC and low/high machine heterogeneity, this is different from the traditional view that algorithm Sufferage is always better than algorithm Min-min. From figure 2 and figure 4 we can see that the performance of algorithm BUD_Sufferage is better than that of BUD_Min-min in the case of inconsistent matrix ETC and low/high machine heterogeneity.

(2) Online Dynamic Case

We use the linear order "$<_p$" and the average makespan of 50 examples as the evaluating criterion. The experiment results of online dynamic case are showing from figure 5 to figure 6, where $V_t$ denotes $V_{task}$, $V_m$ denotes $V_{mach}$.

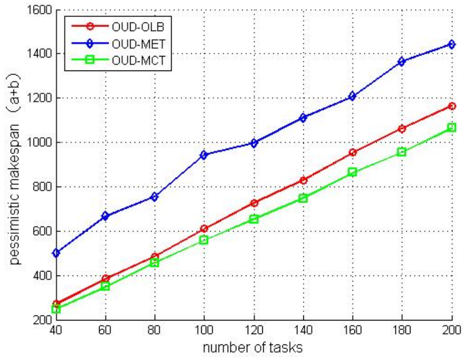

Fig.5 m=20,$v_t$=0.1,$v_m$=0.1

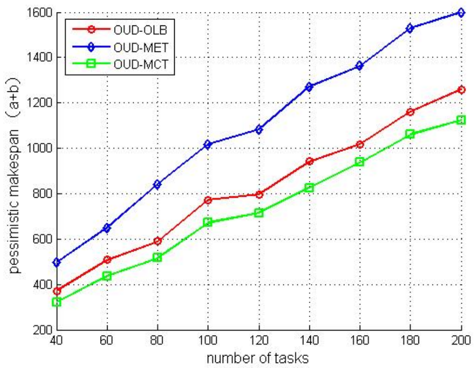

Fig.6 m=20,$v_t$=0.6,$v_m$=0.1

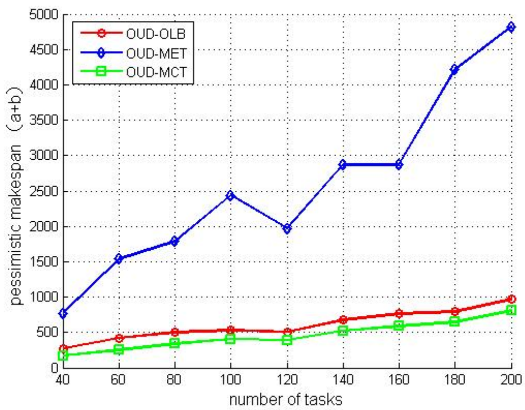

Fig.7 m=20,$v_t$=0.1,$v_m$=0.6

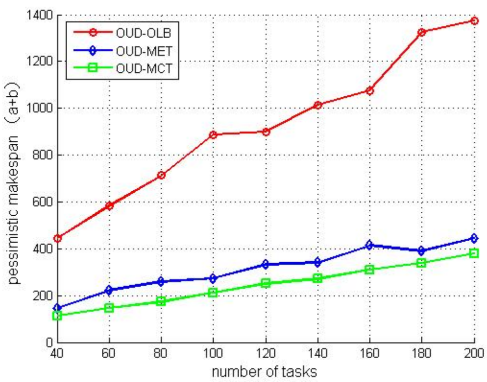

Fig.8 m=20,$v_t$=0.6,$v_m$=0.6

From figure 5 to figure 8 of online dynamic case, we can see that the performance of algorithm BUD-MCT is the best, and algorithm BUD-MET is better than algorithm BUD_OLB only in the case of high task heterogeneity and high machine heterogeneity. The performance of algorithm BUD_OLB is nearly equal to that of algorithm BUD_OLB in the case of low task heterogeneity and low machine heterogeneity. If task heterogeneity and machine heterogeneity are both higher, the performance of BUD-MET is nearly equal to that of algorithm BUD-MCT.

## VI. Conclusion

Uncertainty in the task scheduling of computing grid is imposed by fuzzy, random, indeterminate-known uncertainty and unexpected incidents etc. Using only one of fuzzy or random theory cannot represent and process well this synthetic uncertainty. In this paper, we use binary connection number of SPA to represent and process this synthetic uncertainty. Borrowing the idea of traditional grid scheduling algorithm such as Min-min etc, we put forward three online uncertain dynamic scheduling algorithms, OUD_OLB, OUD_MET, OUD_MCT, and three batch uncertain dynamic scheduling algorithms BUD_Min-min, BUD_Min-max, BUD_Surferage for the uncertain computing grid. Theoretical analysis and numerical experiment illustrate that these algorithms can represent the dynamics and uncertainty of expected execution time of tasks in the computing grid environment. They are the generalization of traditional grid scheduling algorithms, and possess high value in theory and application in the uncertain grid environment. It is a new method to modeling and designing task scheduling algorithm in uncertain computing grid environment.

**Acknowledgments.** The authors gratefully acknowledge the support of this work by Zhejiang Provincial Natural Science Foundation of China (Grant no.Y105118).

## References

[1] I. Foster and C. Kesselman. Grid 2:Blueprint for a New Computing Infrastructure. 2nd edition, Morgan Kaufmann Publishers Inc., 2003.
[2] I. Foster and C. Kesselman, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 2001, 15(3): 200~222
[3] Luo Hong, Mu De-jun, Deng Zhi-qun,et al. A Review of Task Scheduling for Grid Computing**.** Application Research of Computers,2005,22(5):16-19
[4] Tracy D. Brauny, Howard Jay Siegely, Noah Becky, et al A Comparison Study of Dynamic Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems [C]. Proceedings of the 8th Heterogeneous Computing Workshop (HCW'99), pp.15-29, Apr. 1999.
[5] CHEN Zhi-gang, LIU AN-feng, XIONG Ce, et al. An Effective Loading -Balancing Framework for Grid Web Service. Chinese Journal of Computers, 2005, 28(4): 458-466.
[6] LIU An-feng, CHEN Zhi-gang, LU Jing-bo, et al. An Effective Organizing Mechanism for Web Services Resource in Grids. Journal of Computer Research And Development, 2004,41(12):2141-2147
[7] ZHANG Wei-zhe, LIU Xin-ran, YUN Xiao-chun, et al. Trust-driven task scheduling heuristics for computing grid. Journal on Communications, 2006, 27(2):73-79
[8] J. M. Schopf and B. Nitzberg. Grids: Top Ten Questions. Scientific Programming, special issue on Grid Computing, 2002, 10(2):103~111
[9] LI Ji, ZHONG Jiang, WU Zhong-Fu. An Artificial Immune Algorithm for Task Scheduling in Grid Environment with Fuzzy Processing Time. COMPUTER SCIENCE, 2006, 33(2):35-38
[10] Michael A. Iverson, Füsun Özgüner, and Lee C. Potter. Statistical Prediction of Task Execution Times Through Analytic Benchmarking for Scheduling in a Heterogeneous Environment. IEEE Transactions on Computers,1999,48(12): 1374～1379
[11] ZHAO Ke-qin XUAN Ai-li． Set Pair Theory-A New Theory Method of Non-Define and Its Applications, Systems engineering,1996,(1):18-24
[12] HUANG De-cai, ZHAO ke-qin, Lu Yao-zhong. Forecasting and Controlling Method of the Time Limit for a Project of Identical-Discrepancy-Contrary Network Planning. Systems Engineering and Electronics, 2001,23 (5):24-27
[13] XUE Gen-yuan,WANG Guo-qiang. The Application Research of the Theory of Uncertainty-Set Pair Analysis in Establishment of Weather Forcast Models. ACTA METEOROLOGICA SINICA, 2003, 61(5): 592-599
[14] Ye Yue-xiang, Mi Zhong-chun, Wang Hong–yu, et al. Set-pair-analysis-based method for multiple attributes decision-making with intervals. Systems Engineering and Electronics,2006,28(9): 1344-1347
[15] LI Zhi-hui,XIA Shao-yun, CHA Jianzhong. CBR- Based Same-Indefinite-Contrary Product Design and Application. Journal of Computer-Aided Design & Computer, 2003, 15(11):1397-1403
[16] JIANG Yun-Liang, XU Cong-Fu.Advances in Set Pair Analysis Theory and its Applications. Computer Science, 2006, 33(1):205-209
[17] ZHAO Ke-qin. The theoretical basis and basic algorithm of binary connection a+b*i* and its application in AI. CAAI Transaction on Intelligent System.2008,3(6):476-486
[18] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali. Representing Task and Machine Heterogeneities for Heterogeneous Computing Systems. Tamkang Journal of Science and Engr, 2000,3(3): 195-207

**Huang Decai** is a professor of computer science and Technology at Zhejiang University of Technology. His major interests include computer algorithm, data mining and decision theory and methodology.

**YUAN Yuan** is a Doctoral student at New Jersey Institute of Technology, Newark, New Jersey, USA. His major interests include computing grid & algorithms, image processing.

**ZHANGLi-jun** is a graduate student at Zhejiang University of Technology. Her major interests include computing grid & scheduling algorithms.

**ZHAO Ke-qin** is a professor at the institute of Zhuji connection mathematics. His major is SPA and its applications.