基于 $a+bi$ 型联系数的不确定网格静态调度算法*)

黄德才[1] 张丽君[1] 赵克勤[2]

(浙江工业大学信息工程学院 杭州 310032)[1] (诸暨市联系数学研究所 诸暨 311811)[2]

摘 要 任务调度算法是计算网格任务管理系统中的核心问题。由于网格环境中存在大量的不确定因素,导致传统网格调度算法和调度系统不能在开放、异构和动态的真实网格环境中有效运行。利用一种新的软计算方法——集对分析联系数研究和处理网格调度中的综合不确定性问题。在简单介绍集对分析概念和应用情况基础上,引入联系数概念、运算规律和全序关系,分别提出了基于联系数的不确定网格静态调度算法 CBU_Min-min、CBU_Max-min 和 CBU_Surferage,并进行了数值仿真实验研究。理论和实验研究表明,这些算法能较好地描述网格任务预期执行时间的动态性和不确定性,并使传统网格调度方法成为其特例,在动态和不确定网格环境中有良好的理论和实际应用价值。

关键词 计算网格,不确定性,任务调度,调度算法

Static Scheduling Algorithms Based on Connective-number of Type $a+bi$ for Uncertain Computing Grid

HUANG De-Cai[1] ZHANG Li-Jun[1] ZHAO Ke-Qin[2]

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310032)[1]

(Zhuji Institute of Connective Mathematics, Zhuji 311811)[2]

Abstract Job scheduling algorithms are kernel technique in task management system of computing grid. Because the dynamic and uncertainty exist in grid environment, the traditional job scheduling algorithms cannot be applied effectively in the real open, heterogeneous and dynamic grid environment, and let the job scheduling system do not work well. Using connective number of Set Pair Analysis(SPA), a new soft computation method to express & process the synthetic uncertainty in task scheduling of the computing grid. After introducing SPA and its application briefly, present the definition of connective number, operational rules and total order relation suitable for computing grid scheduling, three static scheduling algorithms, CBU_Min-min, CBU_Max- min and CBU_Surferage, are presented for the uncertain computing grid. Theory analysis and numerical experiment illustrate that these algorithms can express the dynamics and uncertainty of expected time to compute of tasks in the computing grid environment, they are the generalization of traditional grid scheduling algorithms, and there are high value in theory and application in the dynamic and uncertain grid environment.

Keywords Computing grid, Uncertainty, Task scheduling, Scheduling algorithm

1 引言

网格技术已成为下一代互联网应用的关键技术,也是当今计算机领域的前沿研究课题。虽然有计算网格、数据网格、存储网格、知识网格和服务网格等多种类型,但不论什么样的网格,任务调度系统都是其发挥潜在性能和优势所共有的核心系统,且任务管理、任务调度和资源管理是其必备的三种基本功能[1,2],而任务调度模型及其优化算法则是网格调度必须解决的核心问题和关键技术。调度算法的好坏,将直接影响网格的信息传递效率、资源利用率、网格可靠性和应用程序的执行时间等[3]。虽然对网格任务调度算法的研究已取得不少成果[3~7],但这些算法却都没有考虑网格环境的动态性和不确定性,导致当前的任务调度系统难以在开放、异构和动态的真实网格环境中有效运行。

网格系统与传统并行高性能计算的最大不同之处在于其动态性,且如何处理这种动态性被认为是网格的"十大"问题之一[8]。由于网格环境中存在大量的不确定因素,比如资源的共享性、资源的随时进入和退出、通信带宽和延迟差异性等都导致计算资源的性能动态变化,正如文[9]所指出的那样,网络流量和资源的当前负载都是动态变化的,它们反映出的结果就直接影响任务的执行速度,即任务的预期执行时间是不确定的。因此,文[9]提出具有模糊处理时间的人工免疫网格任务调度算法,文[10]利用随机方法来预测任务的执行时间。然而,由于网格的广域、异构和动态性,网格任务调度中的不确定性已经不是单纯的模糊或随机不确定性,而是由随机、模糊、不确知、中介和突发等不确定性导致的综合不确定性问题,任何单一的不确定性方法都难于真实描述和表达这种综合不确定性问题。正是在这一点上,由我国学者赵克勤

*)基金项目:浙江省自然科学基金资助项目(Y105118&Y105109)。**黄德才** 工学博士,教授,博士生导师,主要研究方向有网格调度、人工智能、图像处理、数据挖掘等;**张丽君** 硕士研究生,主要研究方向有网格调度和数据挖掘;**赵克勤** 高级工程师,理学学士,主要研究集对分析、联系数学及其应用等。

提出的集对分析及其联系数[11]在表示由随机、模糊、不确知、中介和突发等不确定性导致的综合不确定问题方面有独到之处，并在网络计划、气象预报、不完备信息系统和产品设计等许多领域得到应用[12~16]。

本文利用集对分析[11](Set Pair Analysis，简记SPA，又称为联系数学)这一新的数学工具来研究和处理网格调度的这种综合不确定性问题。在简单介绍集对分析概念和应用情况基础上，引入联系数概念、运算规律和全序关系，并借鉴传统网格任务静态调度算法 Min-min、Max-min、Sufferage 的思想，分别提出了基于联系数的不确定网格静态调度算法 CBU Min-min、CBU Max-min、CBU Sur- ferage(CBU 即为 Connective-number Based Uncertain 的首字母)，并进行了数值仿真实验研究。理论和实验研究表明，这些算法能较好地描述网格任务预期执行时间的动态性和不确定性，符合客观实际，并使传统网格调度方法成为其特例，不仅在动态和不确定网格环境中有良好的理论和实际应用价值，还将成为网格任务调度理论建模的一个新的研究方法。

2 联系数运算及全序关系

2.1 联系数运算

集对分析是一种新的软计算方法[16]，而联系数则是集对分析中的重要概念之一[11]。引进联系数的最初目的是为了应用上的方便，其理论意义则在于拓广了数的概念。一方面它把可确定数与所在范围的数与值联系起来，另一方面它把宏观层次上的确定量和微观层次上的不确定量联系起来。虽然联系数 $u=a+bi+cj$ 的表达形式是统一的，但 a,b,c 和 i,j 的语义可根据所研究问题的实际背景具体定义和解释。因此，本文根据计算网格中任务预期执行时间(ETC，Expected Time to Compute)的不确定性特点，重新定义网格调度问题中的联系数为当 $c=0$ 时的 $a+bi$ 型联系数。

定义 1 称 $u=a+bi$ 为网格调度问题中表示某个任务预期执行时间的联系数，其中 a 表示该任务在通常情况下的预期执行时间，b 为网格环境不确定性引起的时间波动，i 在区间$[-1,1]$内不确定取值。特别当 $a=b=0$ 时，记联系数 $u=0$。

比如，某项任务在一个节点上计算完成通常情况下需要100s，但由于该节点上并发执行的其它任务退出或新任务的进入，则最多可能提前 31s 或延迟 31s 完成，因此可以用联系数 $u=100+31i$ 来表示该任务在这种不确定情况影响下的预期执行时间。

下面引入联系数的四则运算规则。

定义 2[12] 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数，则它们之和是一个联系数 $u=a+bi$，记作 u_1+u_2，其中 $a=a_1+a_2,b=b_1+b_2$。

从定义 2 可以看出，联系数的加法运算满足交换律和结合律。下面，我们引进联系数的减法运算。

定义 3[12] 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数，则它们之差是一个联系数 $u=a+bi$，记作 u_1-u_2，其中 $a=a_1-a_2,b=b_1-b_2$。

由定义 1，定义 2 和定义 3 容易得出如下推论。

推论 1 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数，则 $u_1=u_2$ 的充分必要条件是 $a_1=a_2$，$b_1=b_2$。

定义 4 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数，则它们之积是一个联系数 $u=a+bi$，记作 $u_1\times u_2$，其中 $a=a_1a_2+b_1b_2,b=a_1b_2+a_2b_1$。

定义 5 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数，若存在一个联系数 $u=a+bi$，使得 $u_1=u\times u_2$，则称 u 是 u_1 除以 u_2 的商，记作 $u=u_1\times u_2$。这时称 u_1 能够除以 u_2，否则称 u_1 不能够除以 u_2。

此外，我们容易证明如下定理。

定理 1 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数，则 u_1 能够除以 u_2 的充分必要条件是以下矩阵 M 是非奇异的。

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

且 u_1 除以 u_2 的商 $u=a+bi$ 中的 a,b 为以下线性方程组的解。

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

此外，我们可以证明联系数乘法的一些性质，如交换律，结合律等。限于篇幅不予赘述。

2.2 调度问题与全序关系

粗粒度元任务的调度问题是网格任务调度中最基本的调度问题，许多其它的调度模型都是由它修正和推广而来。它假设有 n 项独立任务 $T=\{t_1,t_2,\cdots,t_n\}$ 和 m 个计算资源节点 $M=\{M_1,M_2,\cdots,M_m\}$，任务 t_i 在节点 M_j 上的预期执行时间是 $E(i,j)$，调度目标是极小化 makespan。由于这个看似简单且无约束的单目标调度问题也是一个 NP-Hard 问题，因此，人们提出了许多启发式算法，如著名的静态调度算法 Max-min、Min-min 和 Sufferage 算法等。根据以元任务调度问题的定义，本文讨论基于联系数的不确定网格调度问题叙述如下：

设有 n 项独立任务 $T=\{t_1,t_2,\cdots,t_n\}$ 和 m 个计算机资源节点 $M=\{M_1,M_2,\cdots,M_m\}$，任务 t_p 在节点 M_q 上的预期执行时间 $E(p,q)$ 是不确定的，可用联系数表示为 $E(p,q)=a_{pq}+b_{pq}i$，并构成一个 $n\times m$ 的 ETC 矩阵 $(E(p,q))_{n\times m}$。任务调度的目标是把 T 中任务安排到 M 中的节点上执行，使其总的执行时间(makespan)尽可能地“小”。

由于每个任务的预期执行时间是由联系数表达的不确定量，且根据联系数的运算法则，以上调度问题的总执行时间也是联系数，因此，如何比较联系数的“大”、“小”，即定义联系数的全序关系是建立基于联系数的不确定网格调度模型的关键。

因为联系数表达的是不确定量，一般可以根据实际应用问题的不同需要，定义不同的序关系。文[11]根据联系数中 a,b 的比值 a/b 是小于 1、大于 1 或是等于 1 的不同情况定义了联系数的同异反态势关系，但这种关系仅是偏序而非全序关系。文[12]根据不确定网络计划的需要，对联系数集合通过定义主关键联系数的概念给出了联系数的一种偏序关系，并在网络计划关键路径识别方面得到应用，但仍然不是全序关系，致使大多数联系数仍然无法进行比较。

下面我们结合网格调度的实际需要，给出联系数的两种全序关系。

定义 6 设 $u_1=a_1+b_1i,u_2=a_2+b_2i$ 是两个联系数。如果 $a_1<a_2$，或 $a_1=a_2$，且 $b_1<b_2$，称 u_1 小于 u_2，记作 $u_1<_pu_2$。

显然，若用 R_c 表示所有联系数的集合，则以上定义的小于关系 $<_p$ 是 R_c 上的全序关系，它首先对确定性部分 a_1 和 a_2 进比较，因为确定部分执行时间是相对确定的，其优先级高于不确定部分；如果 $a_1=a_2$，则对不确定部分 b_1 和 b_2 进行

比较。这个全序关系是一种最坏情形的全序关系，也称为悲观估计下的全序关系，因为它对不确定性部分进行比较时以 $b_1<b_2$ 确定 $u_1<_p u_2$。

根据定义 6 我们容易证明如下定理。

定理 2 设 R_c 为所有联系数的集合，则对于任意 $u_1, u_2 \in R_c$ 有，以下三个结论中有且恰有一个成立。

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

下面定义一种乐观估计下的全序关系。

定义 7 设 $u_1=a_1+b_1 i, u_2=a_2+b_2 i$ 是两个联系数。如果 $a_1<a_2$，或 $a_1=a_2$，且 $b_1>b_2$，称 u_1 小于 u_2，记作 $u_1<_o u_2$。

类似定理 2，我们可得如下定理 3。

定理 3 设 R_c 为所有联系数的集合，则对于任意 u_1，$u_2 \in R_c$，以下三个结论中有且恰有一个成立。

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

在本文以下部分讨论中，当提到联系数或所表示的时间进行比较时的“最小”、“最大”、“小于”、“大于”或“最短”、“最长”等，都是指在全序关系$<_p$ 或$<_o$ 意义下的“最小”、“最大”等概念。而在一个算法中究竟选择哪一种全序关系，完全取决于对系统状态的宏观估计。如果根据最近网格系统运行的实际情况得知，实际运行的时间通常比静态调度的时间长，则宜选用悲观估计的序关系，反之就应该选用乐观估计的序关系。

3 静态调度算法

3.1 参数定义

(1)M_q：既表示编号为 q 的计算资源(节点)，也表示该节点上所分配任务的集合，且按照任务分配的先后次序存放。

(2)S_q：节点 M_q 在完成已分配给它的所有任务后，预期可执行另外一个新任务的开始时间；

(3)$E(p,q)$：当任务 t_p 分配给可用节点 M_q 的预期执行时间，形成的矩阵称为 ETC 矩阵；

(4)$C(p,q)$：若现在将新任务 t_p 分配给可用节点 M_q 的预期完成时间，$C(p,q)=S_q+E(p,q)$；

(5)CT：任务与资源匹配形成的二元组〈任务，资源〉集合，也称任务-资源对集合；

(6)Assigned_q：在一次循环中，节点 M_q 是否已经分配一个新任务，取值 True 或 Flase；

3.2 静态调度算法

(1)**算法 1** CBU_Max-min 算法

输入：任务集 T，节点集 M 和基于联系数的 ETC 矩阵；
输出：任务与资源的映射方案 $M_1, M_2, \cdots, M_m$；
Step0. for M 中的每一个节点 M_q
　　$M_q=\phi$, $S_q=0$
　endfor
Step1. Repeat
　　$CT=\Phi$
Step2.　for T 中每一个作业 t_p
Step3.　　for M 中每一个节点 M_q
　　$C(p,q)=S_q+E(p,q)$;
Step4.　endfor
　　找出使任务 t_p 完成时间“最短”的资源 M_q，$CT=CT\cup\{\langle t_p, M_q\rangle\}$
Step5.　endfor.
Step6.　$C_{\max}=\max\{C(p,q)\mid\langle t_p,M_q\rangle\in CT\}$
　　$M^{\max}=\{M_k\mid C(p,k)=C_{\max}$且$\langle t_p,M_k\rangle\in CT\}$
Step7.　在 $M^{\max}$ 中任选节点 M_k 及其对应的$\langle t_r,M_k\rangle$，令
　　$M_k=M_k\cup\{t_r\}$，$T=T-\{t_r\}$
　　$S_k=S_k+E(r,k)$
Step8. Until(T=ϕ)

(2)**算法 2** CBU_Min-min 算法

输入：任务集 T，节点集 M 和基于联系数的 ETC 矩阵；
输出：任务与资源的映射方案 $M_1, M_2, \cdots, M_m$；
Step0. for M 中的每一个节点 M_q
　　$M_q=\phi$, $S_q=0$
　endfor
Step1. Repeat
　　$CT=\Phi$
Step2.　for T 中每一个作业 t_p
Step3.　　for M 中每一个节点 M_q
　　$C(p,q)=S_q+E(p,q)$;
Step4.　endfor
　　找出使任务 t_p 完成时间“最短”的资源 M_q，$CT=CT\cup\{\langle t_p,M_q\rangle\}$
Step5.　endfor
Step6.　$C_{\min}=\min\{C(p,q)\mid\langle t_p,M_q\rangle\in CT\}$
　　$M^{\min}=\{M_k\mid C(p,k)=C_{\min}$且$\langle t_p,M_k\rangle\in CT\}$
Step7.　$M^{\min}$中任选一个节点 M_k 及其对应的$\langle t_r,M_k\rangle$
　　$M_k=M_k\cup\{t_r\}$，$T=T-\{t_r\}$
　　$S_k=S_k+E(r,k)$
Step8. Until(T=ϕ)

(3)**算法 3**：CBU_Sufferage

输入：任务集 T，节点集 M 和基于联系数的 ETC 矩阵；
输出：任务与资源的映射方案 $M_1, M_2, \cdots, M_m$；
Step0. for M 中的每一个节点 M_q
　　$S_q=0$, $\text{Assigned}_q=$False, $M_q=\phi$
　endfor
Step1. Repeat
Step2.　for T 中的每一个作业 t_p
Step3.　　for M 中每一个节点 M_q
Step4.　　$C(p,q)=S_q+E(p,q)$;
Step5.　endfor
Step6.　for M 中每一个节点 M_q
Step7.　　找出使 t_p 完成时间“最短”的资源 M_r 和对应的完成时间 $C(p,r)$;
　　找出使 t_p 完成时间“次短”的资源 M_s 和对应的完成时间 $C(p,s)$;
　　$SuffV_p=C(p,s)-C(p,r)$
Step8.　endfor
Step9.　if 任务 t_p 对应的节点 M_q 的 $\text{Assigned}_q=$False
　　$M_q=M_q\cup\{t_p\}$，$T=T-\{t_p\}$;
　　$\text{Assigned}_q=$True;
Step11.　else if t_k 是本次循环已分配给节点 M_q 的任务且 $SuffV_p$“小于”$SuffV_k$，
　　$M_q=M_q\cup\{t_p\}-\{t_k\}$
　　$T=T\cup\{t_k\}-\{t_p\}$;
　　endif.
　endfor.
Step12.　for M 中的每一个节点 M_q
　　$S_q=\sum_{t_p\in M_q}E(p,q)$
　　$\text{Assigned}_q=$False
　endfor
Step13. Until(T=ϕ)

显然，如果算法 1、算法 2 和算法 3 在 Step0 中令参数 $S_q\neq 0$，则它们就成为批模式下的在线调度算法。

4 算例与仿真分析

4.1 计算实例

为说明算法的计算过程并对调度结果进行预测分析，下面给出一个简单的例子。

例 1 设计算资源数 $m=3$，任务数 $n=5$，每个任务在各个计算节点上的预期不确定执行时间矩阵 ETC 如下，试在联系数的悲观序关系条件下用 CBU_Min-min 算法完成其任务调度。

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

解：根据 CBU_Min-min 算法的计算过程，可知每次循环完成一个任务的指派，其具体过程如下：

1. 由于$(S_1,S_2,S_3)=(0,0,0)$，而各任务的“最小”完成时间集合 $CT=\{C(1,2)=100+22i, C(2,3)=99+7i, C(3,3)=98+3i, C(4,1)=100+5i, C(5,1)=97+16i\}$，由定义 6 和算法，将任务 $t5$ 安排到节点 $M1$，$(S_1,S_2,S_3)=(97+16i, 0, 0)$。

2. 对未安排的 t_1, t_2, t_3, t_4，计算任务的最小完成时间集合 $CT=\{C(1,2)=100+22i, C(2,3)=99+7i, C(3,3)=98+3i, C(4,3)=101+6i\}$，同理可知应将任务 t_3 安排到节点 M_3 执行，得 $(S_1, S_2, S_3)=(97+16i, 0, 98+3i)$；

3. 对未安排的 t_1, t_2, t_4，计算任务的最小完成时间集合 $CT=\{C(1,2)=100+22i, C(2,2)=99+9i, C(4,2)=102+8i\}$，同理将任务 t_2 安排到节点 M_2 执行，得 $(S_1, S_2, S_3)=(97+16i, 99+9i, 98+3i)$；

4. 对未安排的 t_1, t_4，计算任务的最小完成时间集合 $CT=\{C(1,1)=198+48i, C(4,1)=197+21i\}$，同理将任务 t_4 安到节点 M_1 执行，得 $(S_1, S_2, S_3)=(197+21i, 99+9i, 98+3i)$；

5. 对未安排的 t_1，其最小完成时间为 $C(1,2)=199+31i$，因此将任务其安排到节点 M_2 执行，得 $(S_1, S_2, S_3)=(197+21i, 199+31i, 98+3i)$。

按照定义 6 可知，该系统的 makespan $=199+31i$。这个调度结果说明，在正常情况下，所有任务可以在 199 单位时间内完成。如果系统中多数节点上并发执行的其它任务退出或新任务的进入，或者受其它不确定因素的影响，则最多可能提前 31 或延迟 31 个单位时间完成，但具体的完成时间应该根据不确定量 i 的取值来进行预测和分析。比如，如果系统最近一段时间运行都比较慢，则可以选择悲观估计的最坏情形，即取 $i=1$，这时系统的 makespan $=230$；也可以采用“面向式子”的取值法[11]，取 $i=31/(199+31)\approx 0.1348$，因此，在 i 取 0.1348 这种情形的悲观估计条件下，系统的 makespan $=203.1788$。关于 i 的取值还有动态取值法、概率取值法、专家取值法和综合取值法等[10]。在实际应用中一般要根据网格系统运行的实际状况来选择合适的取值方法，并对系统执行任务的时间进行预测和分析，也正是在这一点上，基于联系数的网格任务调度算法更能体现网格的动态性和不确定性。

4.2 仿真分析

1. 数据产生方法

仿真实验考察 20 个计算资源组成的网格系统对 40～200 个独立任务构成的任务集合调度情况。

任务在计算资源上的预期执行时间 ETC 根据文[8]中的方法来生成 ETC 矩阵中的 a_{pq}，然后根据 a_{pq} 值的大小，利用均匀分布产生 b_{pq}。文[8]中的方法需要输入 4 个参数，设控制数据均值变化的参数 $\mu_{task}=100$，$\mu_{mach}=100$，另外的两个参数 V_{task} 和 V_{mach} 是用来控制任务和机器一致性的，其值越高，任务和机器的异构性越强，将二者取值 $V_{task}=0.1$ 或 $V_{task}=0.6$，$V_{mach}=0.1$ 或 $V_{mach}=0.6$，这样使得机器和任务分别达到较高或较底的异构性，可用来测试异构性的高低对调度是否有大的影响。

2. 实验结果与性能分析

本文采用悲观估计的偏序关系情形下的任务的平均 makespan 值对改进后的新启发式算法进行综合评价，每个绘图采用的是 50 次实验的平均值。

在实验结果中，机器数取 20，用字母 m 表示；同时，用符号‘V_t’表示‘V_{task}’，用符号‘V_m’表示‘V_{mach}’。

因为仿真比较发现 CBU-Max-min 算法的性能比 CBU-Min-min 算法及 CBU-Sufferage 算法的性能要差很多，所以我们没有将 CBU-Max-min 算法的结果放在图中进行比较。

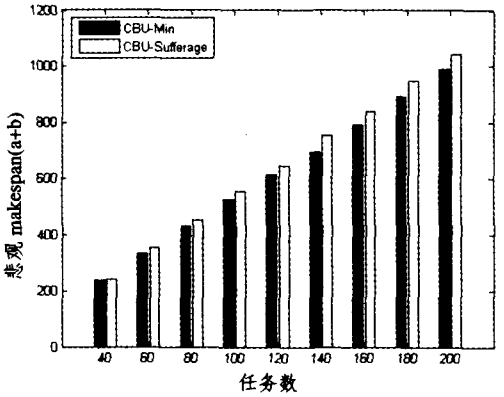

图 1 $m=20, v_t=0.1, v_m=0.1$

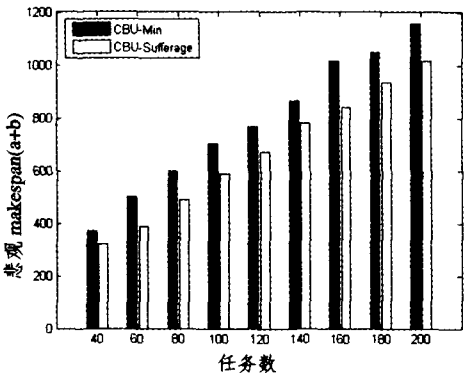

图 2 $m=20, v_t=0.6, v_m=0.1$

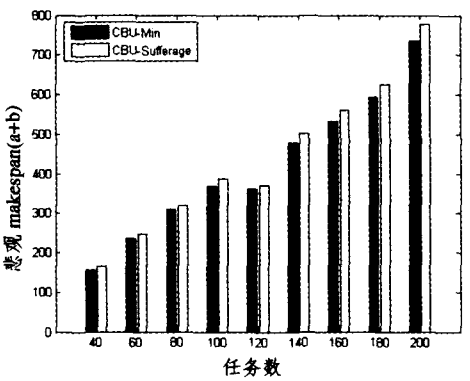

图 3 $m=20, v_t=0.1, v_m=0.6$

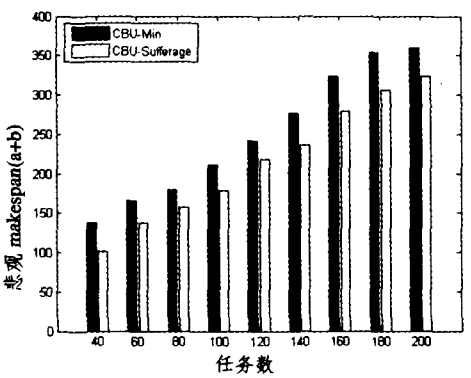

图 4 $m=20, v_t=0.6, v_m=0.6$

从图 1、图 3 可以看出，在非一致的 ETC 矩阵条件下，当

（下转第 179 页）

本实例是用来比较文本证据权 WET 和 χ^2 统计量两种特征选择算法在特定环境下的效果。在分词接口实现正向匹配算法，在特征选择接口实现两种方案，在分类模块中选择 Naïve Bayes 算法，利用宏平均精度来展示分类精度，预料库利用复旦大学提供的测试语料库，采用 10-fold cross validation 的方法。

表 1 分类测试方法比较

分词算法	特征选择算法	文本分类算法
正向最大匹配法	文本证据权 WET	Naïve Bayes
	χ^2 统计量	

下表是比较结果。

表 2 分类测试结果比较

特征选择算法	score	point
χ^2 统计量 选取 40%特征	79.87%	88.19
WET 选取 40%特征	82.27%	100.90
WET 选取 10%特征	75.15%	67.68

结果显示，在选择相同数目特征的情况下，文本证据权 WET 算法的特征选择算法相对于 χ^2 统计量算法表现了较好的分类精度。同时，文本证据权 WET 算法在选取特征比较少的情况下，分类结果下降较明显。

总结与展望 本文介绍了一个中文文本自动分类评估体系，用于对中文文本自动分类过程中使用到的分词算法、特征选择算法以及分类算法进行综合评价，并实现了一个中文文本自动分类研究的开放平台，供人们实现和比较各种算法。利用该平台，得出了若干已有中文文本自动分类算法的实验结果，评价了不同特征选择算法和不同参数对分类的影响。

目前中文方面并没有一个公开的、相对标准的语料库，所以本评估体系的下一步方向就是利用自动下载工具对权威网站进行定期下载，建立一个标准的语料库，并实现语料库的动态更新。

参 考 文 献

1 Yang Yiming, Liu Xin. A Re-Examination of Text Categorization Methods. In: 22nd Annual International SIGIR. 1999. 42～49
2 刘延章，余义芳. 近五年来网络信息分类组织研究的现状及其展望. 情报学报，2004，23(2)
3 张春霞，郝天勇. 汉语自动分词的研究现状与困难. 系统方针学报，2005(1)
4 孙茂松，邹嘉彦. 汉语自动分词研究评述. 当代语言学，2001(1)：22～32
5 代六玲，黄河燕，陈肇雄. 中文文本分类中特征抽取方法的比较研究. 中文信息学报，2004，18(1)：26～32
6 Wang Yi, Wang Xiao-Jing. A new approach to feature selection in text classification, Machine Learning and Cybernetics, 2005. In: Proceedings of 2005 International Conference on, Aug. 2005, 6：3814～3819
7 李凡，鲁明羽，陆玉昌. 关于文本特征抽取新方法的研究. 清华大学学报(自然科学版)，2001，41(7)：98～101
8 Sebastiani F. Machine Learning in Automated Text Categorization. In: 18th International Conference on Computational Linguistics (COLING'00), Nancy, France, July 2000
9 Jain G, Ginwala A, Aslandogan Y A. An approach to text classification using dimensionality reduction and combination of classifiers, Information Reuse and Integration, 2004. IRI 2004. In: Proceedings of the 2004 IEEE International Conference on, Nov. 2004. 564～569
10 Wang Baoyi, Zhang Shaomin. A Novel Text Classification Algorithm Based on Naïve Bayes and KL-Divergence, Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. In: Sixth International Conference on, Dec. 2005. 913～915

(上接第 129 页)

任务的异构性比较低时，CBU-Min-min 算法的性能比 CBU-Sufferage 算法的性能要好(这一点与传统方法认为 Sufferage 算法总是比 Min-min 算法好的结论不一样)，但差距不是很大；从图 2 和图 4 可以看出，当任务的异构性较高时，CBU-Min-min 算法的性能没有 CBU-Sufferage 算法的性能好，且二者的性能差异较大。

总结 由于网格的广域、异构和动态特性，网格任务调度存在随机、模糊、不确知、中介和突发等多种不确定因素，任何单纯的不确定性方法都难于真实描述和表达这种综合不确定性问题。本文利用集对分析联系数来研究和处理网格调度的这种综合不确定性问题，并借鉴传统网格任务静态调度算法提出了相应的不确定网格静态调度新算法。研究结果表明，这些算法能较好地描述网格任务预期执行时间的动态性和不确定性，不仅在动态和不确定网格环境中有良好的理论和实际应用价值，还将成为网格任务调度理论建模的一个新的研究方法。

参 考 文 献

1 Foster I, Kesselman C 著. 金海，袁平鹏，石柯译. 网格计算[M](第 2 版). 北京：电子工业出版社，2004
2 Foster I, Kesselman C. The Anatomy of the Grid: Enabling Scalable Virtual Organizations[J]. International Journal of Supercomputer Applications, 2001, 15(3): 200～222
3 罗红，蔡德俊，邓智群，王晓东. 网格计算中任务调度研究综述[J]. 计算机应用研究，2005(5)：16～19
4 Brauny T D, Siegely H J, Becky N, et al. A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems[C]. In: Proceedings of the 8th Heterogeneous Computing Workshop (HCW'99), Apr. 1999. 15～29
5 陈志刚，刘安丰，熊策，张连明. 一种有效负载均衡的网格 Web 服务体系结构模型[J]. 计算机学报，2005，28(4)：458～466
6 刘安丰，陈志刚，陆静波，张连明. 网格环境中一种有效的 Web 服务资源组织机制[J]. 计算机研究与发展，2004，41(12)：2141～2147
7 张伟哲，刘欣然，云晓春，等. 信任驱动网格作业调度算法[J]. 通信学报，2006，27(2)：73～79
8 Schopf J M, Nitzberg B. Grids: Top Ten Questions. Scientific Programming, special issue on Grid Computing, 2002, 10(2): 103～111
9 李季，钟将，吴中福. 具有模糊处理时间的网格任务调度免疫算法[J]. 计算机科学，2006，33(2)：35～38
10 赵克勤. 集对分析及其初步应用[M]. 杭州：浙江科学技术版社，2000
11 黄德才，赵克勤. 用联系数描述和处理网络计划中的不确定性[J]. 系统工程学报，1999，14(2)：112～117
12 薛根元，王国强. 不确定性理论集对分析在预报模型建立中的应用研究[J]. 气象学报，2003，61(5)：592～599
13 黄兵，周献中. 不完备信息系统中基于联系度的粗集模型拓展[J]. 系统工程理论与实践，2004，24(1)：88～72
14 李志辉，夏少云，查建中. 基于案例推理的同异反产品设计及其应用[J]. 计算机辅助设计与图形学学报，2003，15(11)：1397～1403
15 蒋云良，徐从富. 集对分析理论及其应用研究进展[J]. 计算机科学，2006，33(1)：205～209
16 Ali S, Siegel H J, Maheswaran M, Hensgen D, Ali S. Representing Task and Machine Heterogeneities for Heterogeneous Computing Systems[J]. Tamkang Journal of Science and Engr, 2000, 3(3): 195～207