基于邻域粗糙模型的高维数据集快速约简算法

刘遵仁[1,2] 吴耿锋[1]

(上海大学计算机工程与科学学院 上海 200072)[1] (青岛大学信息工程学院 青岛 266071)[2]

摘 要 根据粒子群优化算法的思想,给出了求解高维邻域决策表的一个约简算法SPRA。通过采用固有维数的分析方法MLE等,将其估算的维数值作为SPRA算法的初始化参数,提出了高维数据集快速约简算法QSPRA。利用5个UCI标准数据集对该算法进行了验证,结果表明,该算法是有效的、可行的。详细分析了种群规模和迭代次数对结果产生的影响。实验表明,基于核的启发式添加算法思想已经不适合求解高维数据集。

关键词 邻域粗糙模型,决策依赖度,固有维数估算,极大似然估计法,粒子群优化算法,粒子群快速约简算法

中图法分类号 TP391 **文献标识码** A

Quick Reduction Algorithm for High-dimensional Data Sets Based on Neighborhood Rough Set Model

LIU Zun-ren[1,2] WU Geng-feng[1]

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)[1]

(College of Information Engineering, Qingdao University, Qingdao 266071, China)[2]

Abstract According to the particle swarm optimization algorithm's idea, a new algorithm(SPRA) to get a optimal attribute reduction on the high-dimensional neighborhood decision table was proposed. Through the use of intrinsic dimension analysis method, taking the intrinsic dimensionality estimated as the SPRA algorithm's initialization parameter, a quick reduction algorithm(QSPRA) was proposed to deal with the high-dimensional data sets. The algorithm's validity was verified by five high-dimensional data sets from UCI. In the experimental analysis section, the population size and the number of iteration to the influence of the reduction result were also discussed. Moreover, the experiments also show that it is impossible to solve high-dimensional data sets based on kernel-based heuristic algorithm ideas.

Keywords Neighborhood rough set model, Decision-making dependency, Intrinsic dimension estimation, MLE, Particle swarm optimization algorithm, Quick particle swarm reduction algorithm

1 引言

知识表示是人工智能和智能信息处理系统的基本问题。Pawlak经典粗糙集认为知识就是人类对于现实或抽象世界进行分类的能力[1,2]。借助于等价关系(知识)将论域中的对象划分为若干个等价类,即形成了基本概念或者基本信息粒子。作为一种有效的粒度计算模型,Pawlak粗糙集定义在严格的等价关系基础上,只适合处理离散型数据,对于现实中广泛存在的连续型数据却不能直接处理,这严重影响了该方法的实际应用。为了应用该方法处理连续型数据,只能将连续型数据离散化,但这种转换不可避免地带来了信息的损失[3],计算结果的好坏在很大程度上取决于离散化的方法。

Lin在1988年提出了邻域模型的概念,其通过邻域来粒化论域,将邻域理解为基本信息粒子[4];Yao[5]和Wu[6]分别研究了邻域信息系统的代数性质,最近该模型在工程问题上得到了进一步的研究和应用[7]。邻域粗糙模型方法直观,可以直接处理连续型属性值,而无需进行离散化处理,与经典粗糙集相比,省去了离散化的过程,从而拓展了Pawlak粗糙集的应用范围。知识约简是指在保持对论域分类能力不变的情况下,删除冗余的知识。知识约简问题是粗糙集理论的核心问题之一。目前,尚未有一个通用的约简算法,为了找到最优约简(最小约简),且提高效率,普遍采用的方法是基于核的启发式添加算法,这一思路可以有效地解决一大类问题。研究证明,求属性集合的最小约简是NP-难问题,计算量随属性个数呈指数级增长。

在现代社会的许多领域都会遇到对高维数据的分析问题,由于对高维数据的分析十分困苦,并且 高维数据中的信息往往主要集中在若干个低维构造中,因此降维技术是分析高维数据的一个重要手段[8]。20世纪70年代中开始的投影寻踪是一种较好的降维技术,90年代初提出的切片逆回归也是一种降维方式。近几年关于投影寻踪和其他降维方式的讨论较多,然而,真正可用于高维复杂数据的降维方式(例如投影寻踪)计算量都很大,而且往往不能一步到位,需要多次反复。本文旨在基于邻域粗糙模型,针对UCI中属性值为连续型的高维数据集,找到一种快速求解邻域决策表约简的算法。通过实验证明了该算法的有效性,实验发现,基于核的启发式

到稿日期:2011-12-17 返修日期:2012-04-02 本文受国家自然科学基金(61170106)资助。

刘遵仁(1963-),男,博士生,副教授,主要研究方向为粗糙集理论、数据挖掘、智能信息处理,E-mail:liuzunren@163.com。

添加思想，已经不适合高维数据集。

2 度量空间的粒化与近似

2.1 邻域粗糙集的概念

定义 1 设(U,d)是度量空间，其中$U=\{x_1,x_2,\cdots,x_n\}$。

(1)若$x_i\in U,\delta>0$，称

$O(x_i,\delta)=\delta(x_i)=\{x|x\in U\wedge d(x,x_i)<\delta\}$

$S(x_i,\delta)=\delta^+(x_i)=\{x|x\in U\wedge d(x,x_i)\leqslant\delta\}$

分别是以x_i为中心，δ为半径的开球和闭球。

(2)若$P\subset U,\forall x\in P,\exists\delta_x>0$，使得$O(x_i,\delta_x)\subset P$，则称$P$为开集。

(3)包含x_i的任一开集称为x_i的δ-邻域。

开集$\delta(x_i)$也称为由x_i生成的δ邻域信息粒子，$|\delta(x_i)|$反映了粒子颗粒的粗细。从定义可知$\bigcup_{i=1}^{n}\delta(x_i)=U$，即邻域粒子簇构成了$U$的一个覆盖，这是与 Pawlak 粗糙集中的等价关系最明显的不同。当$|\delta(x_i)|=1$时，邻域模型退化为 Pawlak 粗糙集模型，即 Pawlak 粗糙集模型是邻域模型的特例。不同的δ可以产生不同的邻域集$\{\delta(x_i)|i=1,2,\cdots,n\}$。

论域$U=\{x_1,x_2,\cdots,x_n\}$中所有对象的邻域集$\{\delta(x_i)\}$形成了论域的粒化，邻域粒子簇构成了论域中的基本概念，通过这些基本概念，可以逼近论域U中的任意概念。

定义 2 给定一个论域U和U上的一簇邻域集Δ，称二元组$NK=(U,\Delta)$是关于论域U的一个邻域知识库或邻域近似空间。

定义 3 给定邻域近似空间$NK=(U,\Delta)$，$\forall X\subseteq U$和论域U上的一个邻域集$N\in\Delta$，定义信息粒X关于N的下近似和上近似分别为：

$\underline{N}(X)=\{x_i|(\forall x_i\in U)\wedge(\forall\delta(x_i)\in N)\wedge(\delta(x_i)\subseteq X)\}$

$\overline{N}(X)=\{x_i|(\forall x_i\in U)\wedge(\forall\delta(x_i)\in N)\wedge(\delta(x_i)\cap X\neq\emptyset)\}$

集合$bn_N(X)=\overline{N}(X)-\underline{N}(X)$称为$X$的$N$边界域；$pos_N(X)=\underline{N}(X)$称为$X$的$N$正域；$neg_N(X)=U-\overline{N}(X)$称为$X$的$N$负域。

2.2 邻域决策系统

定义 4 称四元组$NDT=(U,C\cup D,V,f)$是一个邻域决策系统，其中U为论域；C和D分别为条件属性集和决策属性集，且$C\cap D=\emptyset,C\neq\emptyset,D\neq\emptyset$；$V$是信息函数$f$的值域。

定义 5 给定一个邻域决策系统$NDT=(U,C\cup D,V,f)$，决策属性D将U划分为m个等价类$X=\{X_1,X_2,\cdots,X_m\}$，$\forall P\subseteq C$，定义

$$\gamma_P(D)=k=\frac{|pos_P(D)|}{|U|}=\frac{|\bigcup_{Q\in X}\underline{P}(Q)|}{|U|}$$

为决策属性D依赖于P邻域粒子的程度，简记为$P\Rightarrow_k D$，$pos_P(D)=\bigcup_{Q\in X}\underline{P}(Q)$是决策$D$的$P$正域，它反映了邻域$U$中所有根据$P$的邻域粒子信息可以准确划分到决策属性$D$的等价类中的对象集合。其中$\underline{P}(Q)=\{x_i|(\forall x_i\in U)\wedge\delta_P(x_i)\subseteq Q\}$。

依赖度性质：

(1)$0\leqslant\gamma_P(D)\leqslant 1$。

(2)当$\gamma_P(D)=1$时，称决策D完全依赖于P邻域，或称D是由P导出的，记为$P\Rightarrow D$。

(3)$\gamma_P(D)=0$时，称决策D完全独立于P邻域粒子，记为$P\not\Rightarrow D$。

定义 6 给定一个邻域决策系统$NDT=(U,C\cup D,V,f)$，$P\subseteq C,\forall a\in P$，如果$\gamma_{P-\{a\}}(D)<\gamma_P(D)$，则称$a$对于$P$是必不可少的；如果$\gamma_{P-\{a\}}(D)=\gamma_P(D)$，则称$a$对于$P$是冗余的。如果$P$中的属性均为必不可少的，则称$P$为独立的。条件属性$a$的重要度为：$sig(a;C,D)=\gamma_C(D)-\gamma_{C-\{a\}}(D)$。

定义 7 给定一个邻域决策系统$NDT=(U,C\cup D,V,f)$，$P\subseteq C$，如果P满足：

(1)P是独立的；

(2)$\gamma_P(D)=\gamma_C(D)$；

则称P是邻域决策系统NDT的一个约简，记为$P\in RED_C(D)$。

定义 8 给定一个邻域决策系统$NDT=(U,C\cup D,V,f)$，系统全部约简的交集为系统的核，即$core_C(D)=\bigcap RED(P)$。

2.3 高维数据集固有维数估算

随着互联网技术等的高速发展，常会碰到高维数据集。随着数据集维数的增加，传统对低维数据集的处理方法的性能急速下降，计算时间呈指数级增长。作为数据之间的相似性度量，在高维空间中的很多情况下，这种概念不复存在，这就给基于高维数据的知识挖掘带来了巨大的困难。实际上，高维数据集通常包含了很多冗余的维，其固有维往往比原始数据集的维数要小得多，因此对高维数据集的处理，可以通过相关的降维方法减少一些不太相关的数据而降低它的维数，然后采用低维数据的处理办法进行处理，因此维数约简技术成为数据挖掘任务的基本技术之一。

目前，针对高维数据集固有维数的估算方法大致有[9]：(1)MLE (Maximum Likelihood Estimator)；(2)GMST(based on the analysis of the Geodesic Minimum Spanning Tree)；(3)PN(based on the analysis of data Packing Numbers)；(4)CD (based on Correlation Dimension)等。

3 基于标准 PSO 算法的决策表快速约简算法 QSPRA

3.1 标准 PSO 算法简介

粒子群优化算法(Particle Swarm Optimization，PSO)最早是在 1995 年由美国 James Kennedy 博士和 Russell Eberhart 博士共同提出的，其基本思想是受他们早期对许多鸟类的群体行为进行建模与仿真研究结果的启发，他们的模型及仿真算法主要利用了生物学家 Frank Heppner 的模型。PSO 是群智能算法(Swarm Intelligence Algorithm)，属于进化计算的范畴。

PSO 算法不像其它进化算法那样对个体使用进化算子，而是将每个个体看作是在n维搜索空间中的一个没有重量和体积的粒子，并在搜索空间中以一定的速度飞行。飞行速度由个体的飞行经验和群体的飞行经验进行动态调整。PSO 算法与其它进化类算法的最大不同点在于：PSO 算法在进化过程中同时保留和利用位置与速度(即位置的变化程度)信息，而其它进化类算法仅保留和利用位置的信息。

标准 PSO 形式如下：

设 $f(X)$为最小化的目标函数，则粒子 i 的当前最好位置由下式确定：

$$P_i(t+1)=\begin{cases}P_i(t), & \text{若 } fitness(X_i(t+1))\geqslant fitness(P_i(t))\\ X_i(t+1), & \text{若 } fitness(X_i(t+1))<fitness(P_i(t))\end{cases}$$

设群体中的粒子数为 s，群体中所有粒子所经历过的最好位置为 $P_g(t)$，称为全局最好位置。

$$P_g(t)\in\{P_0(t),P_1(t),P_2(t),\cdots,P_s(t)\}$$

$$f(P_g(t))=\min\{f(P_0(t)),f(P_1(t)),f(P_2(t)),\cdots,f(P_s(t))\}$$

则标准粒子群优化算法的进化方程可描述为：

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_{1j}(t)[p_{ij}(t)-x_{ij}(t)]+c_2r_{2j}(t)[p_{gj}(t)-x_{ij}(t)] \quad (1)$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1) \quad (2)$$

式中，下标“j”表示粒子的第 j 维，“i”表示粒子 i，t 表示第 t 代，w 称为惯性权重，c_1，c_2 为加速常数，通常在 0～2 间取值，$r_1\sim U(0,1)$，$r_2\sim U(0,1)$为两个相互独立的随机函数。

PSO 具有如下优点：(1)算法简单，搜索速度快、效率高；(2)适合于实值型处理。

缺点：对于离散的优化问题处理不佳，容易陷入局部最优。

3.2 粒子群约简算法 SPRA

基于标准 PSO 算法的思想，依据适应度值，通过迭代，逐渐逼近问题的最优解。

标准粒子群约简算法 SPRA：

输入：邻域决策系统 NDT=(U,C∪D,V,f)。

输出：δ 确定下的条件属性集 C 的最优约简。

具体步骤：

第 1 步　对粒子群的种群规模、迭代次数、随机位置、飞行速度、w、c_1 和 c_2 进行初始化设置。其中位置和飞行速度形如：S={[$\alpha_1,\alpha_2,\cdots,\alpha_i,\cdots,\alpha_n$]|$\alpha_i\in\{0,1\}$，n=length(C)}，对于任意的 α_i 是随机产生的。

第 2 步　计算每个粒子的适应值。

第 3 步　对每个粒子，将其适应值与所经历过的最好位置 P_i 的适应值进行比较，若较好，则将其作为当前的最好位置。

第 4 步　对每个粒子，将其适应值与全局所经历的最好位置 P_g 的适应值进行比较，若较好，则将其作为当前的全局最好位置。

第 5 步　根据进化方程(1)和(2)对粒子的速度和位置进行进化。

第 6 步　如未达到结束条件通常为足够好的适应值或达到一个预设最大迭代数(G_{max})，则返回步骤 2。

算法 SPRA 的几点说明：

(1)粒子的位置和飞行速度表示为向量[$\alpha_1,\alpha_2,\cdots,\alpha_i,\cdots,\alpha_n$]，如果 $\alpha_i=1$，表示选择了属性 C_i，否则相应的属性没有被选择。初始化时，随机产生种群粒子的初态。

(2)适应度函数 $fitness(x_i)=pos_{x_i}(D)$。算法中的粒子最好位置和种群最好位置，在这里指 $|pos_{x_i}(D)|=|U|$ 且满足 $\min\{length(x_i)\}$。说明属性子集 x_i 与原属性集 C 的分类能力完全相同，如果它们又是独立的，则为一个约简。

(3)按进化方程计算时，得到的粒子向量中的元素值需离散化。一般情况下，采用大于 0.5 的离散化为 1，否则取 0 值。

(4)使用本算法前，需要将决策表中的数据归一化，数据取值区间为[0,1]。原因是不同的条件属性常具有不同的单位和不同的变异程度，不同的单位常使系数的实践解释发生困难。另外，不同的条件属性自身具有相差较大的变异时，会使在计算出的欧氏距离中，不同属性值所占的比重大不相同。为了消除量纲影响、属性自身变异大小和数值大小的影响，需将数据标准化。归一化的计算公式为：$y=\frac{x-x_{\min}}{x_{\max}-x_{\min}}$。

3.3 粒子群快速约简算法 QSPRA

标准约简算法 SPRA 中，步骤 1 的粒子位置和飞行速度的初始化可以预先通过固有维数的分析，确定可能的约简所包括的维数大小的上限，从而限制了条件属性的个数。也就是说，在初始化产生每个粒子向量时，随机产生的向量中的 1 的个数等于固有维数的个数。对算法 SPRA 步骤 1 限定粒子向量中 1 的个数，称该改进的算法为粒子群快速约简算法 QSPRA。

4 实验分析

为了验证 SPRA 和 QSPRA 算法的有效性，从 UCI 国际标准数据集中选择了 5 个条件属性维数高的数据集，特别是 Arcene 数据集的条件属性维数高达 10000。这 5 个数据集的条件属性均为连续型数值，如表 1 所列。

表 1　数据集描述

No	数据集	对象数	属性数	类别
1	Ionosphere	351	34	2
2	Sonar	208	60	2
3	Hill-Valley	606	100	2
4	MUSK	476	166	2
5	Arcene_train	100	10000	2

本次实验环境为 CPU：AMD 2.0G，内存：1.5GB；OS：WINDOWS 7 PRO；MATLAB 2010a。

4.1 各种固有维数估算方法的预测

为了使用 QSPRA 算法，需要对所处理的数据集预先估算其固有维数。针对表 1 中的 5 个数据集，采用 MLE、GMST、PN 和 CD 方法，各个数据集的固有维数估算如表 2 所列。

表 2　数据集固有维数估算

No	Data Set	MLE	GMST	PN	CD
1	Ionosphere	13	30	0	1.86
2	Sonar	11	8	0	3.18
3	Hill-Valley	10	2	0.27	0.31
4	MUSK	10	9	0.001	2.55
5	Arcene_train	27	103	0	3.05

现测试算法 QSPRA 的有效性。根据表 2 的各种方法的估算值，对比各个方法所估算的结果，可以采用 MLE 的估算作为初始化种群的上限，即依次取值为：{13,11,10,10,27}。对于前 3 个数据集，设种群规模 $sizepop=150$，迭代次数 $maxgen=20$；对于 MUSK 数据集，设置种群规模 $sizepop=200$，迭代次数 $maxgen=15$；对于 Arcene 数据集，设置种群规模 $sizepop=1000$，迭代次数 $maxgen=10$。

当 $\delta=0$ 时，邻域粗糙模型已经退化到 Pawlak 经典粗糙模型，在其他条件不变的情况下，运行算法 QSPRA 可得到表 1 中的 5 个数据集的最优(最小)约简和运行时间，如表 3 所列。

表 3　最优约简和运行时间(delta=0)

No	数据集	最优约简	运行时间(s)
1	Ionosphere	$\{C_{20}, C_{33}\}$	286.74
2	Sonar	$\{C_{38}\}$	90.96
3	Hill-Valley	$\{C_{83}\}$	801.96
4	MUSK	$\{C_{75}, C_{88}, C_{121}\}$	526.28
5	Arcene_train	$\{C_{880}, C_{5238}\}$	518.43

在 $\delta=0$ 的情况下，算法 QSPRA 收敛的速度如图 1 所示。从图中可以看出，其收敛速度很快，以后一直维持最优约简。

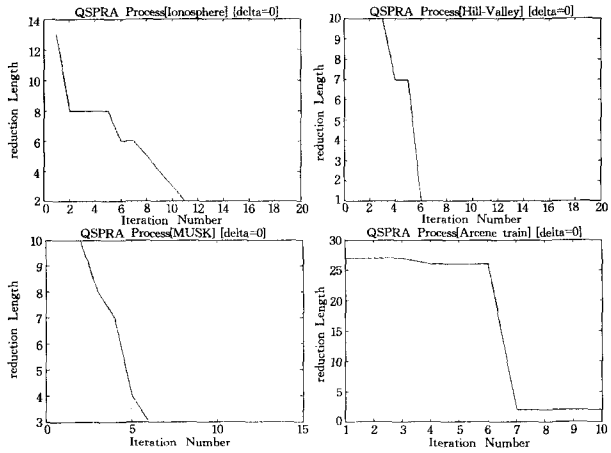

图 1　QSPRA 算法收敛速度示意图

4.2　QSPRA 与 SPRA 算法的对比

为了对比两个算法的运行时间和所产生的结果，在所有条件不变的情况下，测试 SPRA 算法的运行情况，如表 4 所列。

表 4　约简长度和运行时间(delta=0)

No	数据集	约简长度	运行时间(s)
1	Ionosphere	8	361.35
2	Sonar	13	160.94
3	Hill-Valley	25	1864.59
4	MUSK	22	1909.48
5	Arcene_train	—	—

对比表 4 可以看出，无论是所得到的约简长度还是计算时间，SPRA 算法都远远比 QSPRA 算法效率低，并且对于 Arcene 数据集，计算数小时仍然得不到结果。

4.3　种群规模、迭代次数对约简结果的影响

先分析迭代次数对约简的影响。通过对图 1 的分析可以看出，通过引入固有维数参数后，QSPRA 算法基本都可以快速求出决策表的最优约简；从图 2 可以看出，通过足够的迭代，理论上讲，SPRA 算法也可以求出最优约简，但计算量会非常大，很耗时。

再来分析种群规模对产生当前最优约简的影响。在取 $\delta=0.125$ 的情况下，种群规模初始值设为 10，步长为 10，终值为 50，测试数据集为 Ionosphere 和 Sonar，结果如图 3 所示。

从图 3 可以看出，种群规模对产生的结果出现了较大的波动，对高维的 MUSK 和 Arcene 则变化更大。现在从理论上来分析产生这种现象的原因，设条件属性个数为 m，则其组合数为 $\sum_{i=1}^{m} C_m^i = \sum_{i=1}^{m} \frac{m!}{(m-i)!\,i!}$，当 m 值较小时，其组合数也较小；当 m 值很大时，其组合数就是个天文数字了，算法 QSPRA 和 SPRA 第 1 步是随机地从这些组合数中选择 *PopSize* 个粒子，对于较小的组合数集，则选择到“好”的粒子(很快可以逼近最优约简的属性子集)的概率很大；反之，从一个天文数字集中选择到“好”的粒子的概率则很小。这也就是图 3 中数据集产生波动的原因。理论上讲，种群规模 *PopSize* 值越大，则得到“好”粒子的概率也越大，自然所得到的约简越逼近最优约简，但这会加大计算量。

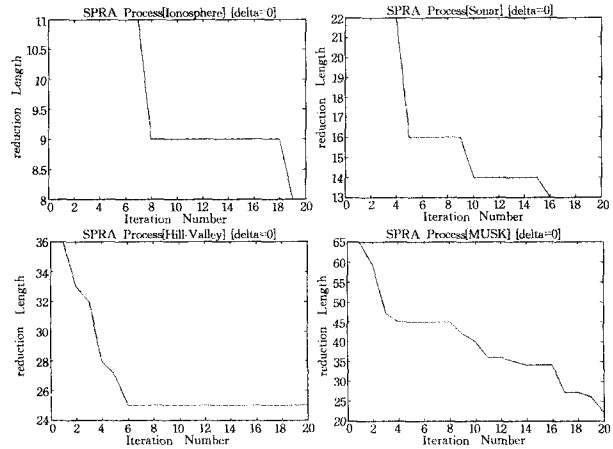

图 2　SPRA 算法收敛速度示意图

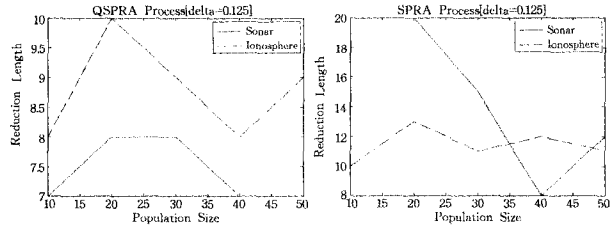

图 3　种群规模与所得约简长度变化图

结束语　本文根据标准 PSO 算法的思想，提出了一个求解高维数据集的约简算法 SPRA，其通过采用固有维数的分析方法，将估算的维数值作为 SPRA 算法的初始化参数，提出了高维数据集快速约简算法 QSPRA。选择 UCI 中的 5 个高维数据集(决策表)，验证了算法的有效性。在实验分析部分，还讨论了种群规模和迭代次数对约简结果的影响。

分析所选的 5 个高维数据集可见，其核均为空，而目前普遍采用的基于核的启发式添加算法思想，对于解决实际工程问题已经失去了理论指导意义。作为经典粗糙集扩展的邻域粗糙模型，将会为解决实际问题提供一个实用的计算模型。

通过实验证明了算法 QSPRA 对于高维的数据集的可行性，但其有时需要运行若干次才可以得到最优约简，因此下一步还需要对算法进行深入的研究。

参 考 文 献

[1] Pawlak Z. Drawing conclusions from data-The rough set way [J]. International Journal of Intelligent Systems, 2001, 16: 3-11

[2] Pawlak Z. Rough Sets-Theoretical Aspects of Reasoning about Data[M]. Dordrecht, Netherlands: Kluwer Academic Publisher, 1991

[3] Jensen R, Shen Q. Semantics-Preserving dimensionality reduction: Rough and fuzzy-rough-based approaches[J]. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(12): 1457-1471

(下转第 317 页)

定义 10 个不同大小作业，构建作业序列，对测试中使用的作业进行多线程算术运算，线程数量可控，每个线程可以将一个内核的利用率提升至 100%，通过线程数量可以控制节点的 CPU 平均利用率。

基于 PBS+Maui 调度运行 5 组测试，基于遗传算法的功耗调度运行一组测试，测试数据如表 2 所列。

5 组 PBS+Maui 测试结果均值与基于遗传算法的功耗调度测试结果对比如图 4 所示。

可以看到，基于遗传算法的功耗调度与 PBS+Maui 的作业调度相比，平均响应时间减少 14.96%，最大响应时间减少 19.51%，作业吞吐率提高 13.29%，运行时间减少 9.85%，节能 9.67%，具有明显的优势。

图 5 为 PBS 选择作业运行时的功耗和资源利用率变化情况，图 6 为基于遗传算法的功耗调度策略选择作业运行时的功耗和资源利用率变化情况。分析两条曲线的相似度，可知采用遗传算法的功耗调度方式能够使资源利用率和功率的变化更加一致，提高了系统资源和能源利用率，系统能随时保持最佳能效比。

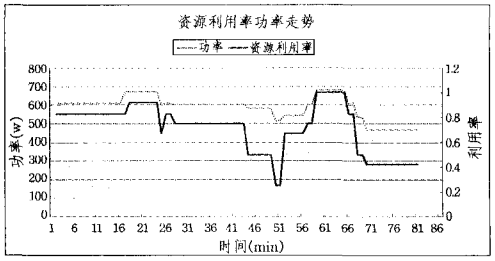

图 5 PBS+Maui 调度作业队列功耗和资源利用率

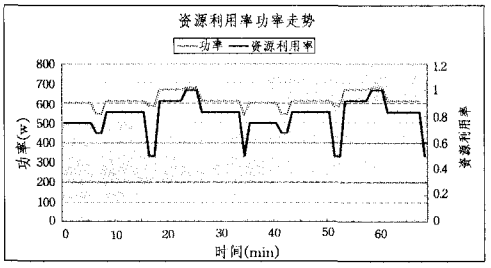

图 6 基于遗传算法的功耗调度作业队列功耗和资源利用率

结束语 本文提出了一种基于自适应功耗管理的高性能计算机作业调度策略，它采用遗传算法作为功耗调度算法，采用作业队列的能效比作为作业调度参数，计算过程中综合考虑了作业类型、预期能效、提交顺序、运行节点、运行时间、系统功耗等因素对最终调度结果的影响。实验证明该策略能有效提高整机能效，与传统作业调度策略相比能节约 9%以上的能耗。

参 考 文 献

[1] Sherwani J, Ali N, Lotia N, et al. Libra: a computational economy-based job scheduling system for clusters[J]. Software-practice and Exoerueice, 2004, 34: 573-590

[2] Weiser M, Welch B, Demers A, et al. Scheduling for reduced CPU energy[C]// Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation. New York: The Advanced Computing Systems Association, 1994: 13-23

[3] 曾宇. 高效能计算机若干关键技术的研究与实现[D]. 北京：中国科学院，2009

[4] Buyya R, et al. High Performance Cluster Computing: Architectures and Systems[M]. Prentice Hall, USA, Volume 1, 1999

[5] Gentzsch W. Sun Grid Engine (SGE): A cluster resource manager[OL]. http://gridengine.sunsource.net/

[6] Platform. Load Sharing Facility (LSF)[OL]. http://www.platform.com/products/wm/LSF/

[7] Systems V. OpenPBS v2.3: The portable batch system software. Veridian Systems, Inc., Mountain View, CA[OL]. http://www.openpbs.org/scheduler.html

[8] Pering T, Burd T, Brodersen R W. The simulation and evaluation of dynamic voltage scaling algorithms[C]// Proceedings of the 1998 International Symposium on Low Power Electronics and Design. Monterey, CA: ACM, 1998: 76-81

[9] Grunwald D, Levis P, Farkas K I, et al. Policies for dynamic clock scheduling[C]// Proceedings of the 4 th Symposium on Operating Systems Design and Implementation. San Diego, 2000

[10] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi objective genetic algorithm: NSGA-II[J]. IEEE Transactions Evolutionary Computation, 2002, 6: 182-197

[11] Bode B, Halstead D M, Kendall R, et al. The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters[C]// Proceedings of 4th Annual Linux Showcase & Conference. Atlanta USA, 2000

(上接第 271 页)

[4] Lin T, Granular Y. Computing on binary relations I: Data mining and neighborhood systems[C]// Skoworn A, Polkowshi L, eds. Proc. of the Rough Sets in Knowledge Discovery. Physica-Verlag, 1998: 107-121

[5] Yao Y Y. Relational interpretation of neighborhood operators and rough set approximation operators[J]. Information Sciences, 1998, 111(198): 239-259

[6] Wu W Z, Zhang W X. Neighborhood operator systems and approximations[J]. Information Sciences, 2002, 144(1-4): 201-217

[7] 胡清华，于达仁，谢宗霞. 基于邻域粒化和粗糙逼近的数值属性约简[J]. 软件学报，2008, 19(3): 640-649

[8] van der Maaten L J P, Postma E O, van den Herik H J. Dimensionality Reduction: A Comparative Review[R]. TiCC TR 2009-005. http://www.uvt.nl/ticc, 2009-10-26

[9] Camastra F. Data Dimensionality Estimation Methods: A survey [J]. Pattern Recognition, 2003, 36(12): 2945-2954