

结合互信息的多目标属性约简^{*}

范会联¹, 仲元昌²

(1. 长江师范学院 数学与计算机学院, 重庆 408100; 2. 重庆大学 通信工程学院, 重庆 400044)

摘 要: 为了获得决策系统中更好的相对属性约简, 提出一种基于互信息的多目标属性约简算法。该算法首先根据互信息寻找核属性集; 然后以最小属性子集和最大互信息为目标, 定义新的适应度函数, 在粒子运动方程、克隆及自适应变异的作用下进化; 并通过非支配排序及精英保留策略寻找满足目标的 Pareto 最优解。通过 UCI 标准数据集上的对比测试结果表明, 算法能够有效地对决策系统进行约简。

关键词: 粗糙集; 互信息; 核属性; 属性约简

中图分类号: TP18

文献标志码: A

文章编号: 1001-3695(2012)02-0490-03

doi:10.3969/j.issn.1001-3695.2012.02.024

Algorithm of attributes reduction based on multi-objective evolutionary and mutual information

FAN Hui-lian¹, ZHONG Yuan-chang²

(1. College of Mathematics & Computer Science, Yangtze Normal University, Chongqing 408100, China; 2. College of Communication Engineering, Chongqing University, Chongqing 400044, China)

Abstract: In order to obtain better relative attribute reduction in decision systems, this paper proposed an algorithm of attributes reduction based on multi-objective evolutionary and mutual information. The algorithm first searched attribute core set by mutual information, then aiming at the least reduction of attributes sets and maximum mutual information, designed a new adaptive function, redefine motion equation and adaptive mutation operation, searched Pareto optimal solutions based on non-dominated sorting and elitism strategy. Experimental results with UCI data sets show that the algorithm can effectively reduce the decision system and obtain ideal reduction results.

Key words: rough sets; mutual information; core attribute; attribute reduction

粗糙集是处理不精确、不确定或不完全数据的有效工具, 在机器学习、数据挖掘、人工神经网络等领域中得到了广泛应用。而属性约简是粗糙集理论的核心内容之一, 是在保持知识库分类或决策能力不变的条件下, 删除冗余属性^[1]。对于同一个决策系统, 一般可能存在多个约简, 有学者已证明寻找最小约简及所有约简是一个 NP-hard 问题^[2]。很多学者针对不同的需求提出了不同的属性约简属性算法, 主要采用基于差别矩阵、启发式搜索或粒计算等方法寻找最优或次优约简。

本文结合决策表属性的信息熵和互信息^[3,4], 在非支配排序策略的基础上, 设计以最小互信息差和最小属性个数为目标、带精英保留策略的多目标粒子群进化算法求解决策表属性约简问题。

1 粗糙集属性约简

设决策表 $S = \langle U, C \cup D, V, f \rangle$ 。其中: $U = \{u_1, u_2, \dots, u_n\}$ 是有限个样本的集合, 称为论域; C 表示条件属性的非空有限集, D 表示决策属性的非空有限集, $C \cap D = \emptyset$, 令 $A = C \cup D$; $V = \bigcup_{p \in A} V_p$ 是属性值的集合, V_p 是属性 p 的值域; $f: U \times (C \cup D) \rightarrow V$ 是信息函数, 用于确定 U 中每一个对象 u 的属性值, 即

对任一 $q \in A$ 和 $u_i \in U$, 有 $f(u_i, q) \in V_q^{[5]}$ 。

1.1 互信息与核属性

定义 1 对于任意属性子集 $R \subseteq C$, 如果 $IND(R) = \{(u_i, u_j) \in U \times U \mid \forall a \in R, f(u_i, a) = f(u_j, a)\}$, 则称 R 为决策表 S 的一个不可分辨关系, 用 U/R 表示等价关系 $IND(R)$ 的所有等价类。

决策表中, 通过计算条件属性与决策属性之间的互信息, 能知道各条件属性对于决策的重要性。对于决策表 $S = \langle U, C \cup D, V, f \rangle$, 设 $R \subseteq C$, $U/R = \{U_1, U_2, \dots, U_n\}$, $U/D = \{Y_1, Y_2, \dots, Y_j\}$, 则^[6]:

定义 2 属性集 R 的信息熵 $H(R)$ 定义为

$$H(R) = - \sum_{i=1}^n \log P(U_i) \quad (1)$$

其中: $P(X_i) = |U_i|/|U|$, $1 \leq i \leq n$, $|U_i|$ 表示集合 U_i 中属性个数。

定义 3 属性集 R 相对于决策属性 D 的条件熵 $H(D|R)$ 定义为

$$H(D|R) = - \sum_{i=1}^n [P(U_i) \sum_{j=1}^m P(Y_j|U_i) \log P(Y_j|U_i)] \quad (2)$$

其中: $P(Y_j|U_i) = |U_i \cap Y_j|/|U_i|$, $1 \leq i \leq n$, $1 \leq j \leq m$ 。

定义 4 属性集 R 相对于决策属性 D 的互信息 $I(R;D)$ 定义为

收稿日期: 2011-08-04; 修回日期: 2011-09-16 基金项目: 国家科技部创新基金资助项目(10C26215115008); 重庆市教委自然科学基金资助项目(KJ111306)

作者简介: 范会联(1971-), 男, 副教授, 硕士, 主要研究方向为智能信息处理、软件工程(fhlmx@163.com); 仲元昌(1968-), 男, 副教授, 博导, 博士, 主要研究方向为通信与测控系统、无线传感器网络等。

$$I(R;D) = H(D) - H(D|R) \quad (3)$$

定义5 对任意 $a \in C$, 若有 $I(C - \{a\}; D) < I(C; D)$, 则称 a 为决策表的一个核属性。

1.2 属性约简问题模型及编码

定义6 设 $R \subseteq C$, 且 $I(R; D) = I(C; D)$, 若对任意 $r \in R$, 有 $I(R - \{r\}; D) < I(R; D)$, 则称 R 为条件属性集 C 关于决策 D 的一个属性约简。

决策表的最小属性约简是指条件属性 C 的所有属性约简中包含属性个数最少的一个约简^[7]。因此, 求解最小属性约简的问题可以形式上归结为如下以集合为变量的一个非线性优化问题1:

$$\begin{cases} R \subseteq C \\ I(R; D) = I(C; D) \\ \forall a \in R, I(R - \{a\}; D) < I(R; D) \\ \min \text{card}(R) \end{cases}$$

其中: $\text{card}(R)$ 表示集合 R 的基数。进一步, 求问题1可以归结为满足 $R \subseteq C$ 条件下的两目标优化问题2, 即

$$\begin{cases} \min f_1(R) = I(C; D) - I(R; D) \\ \min f_2(R) = \text{card}(R) \end{cases} \quad \text{s. t. } R \subseteq C$$

针对条件属性集合为 $\{c_1, c_2, \dots, c_n\}$ 的决策表, 条件属性组合用向量 $X = \{x_1, x_1, \dots, x_n\} \in \{0, 1\}^n$ 表示, 其中 $x_k (1 \leq k \leq n)$ 取值为1表示该属性组合中含有 c_k 属性, 取值为0表示不含该属性。

2 基于互信息和多目标进化的属性约简

本文提出的结合互信息的多目标属性约简算法, 首先利用互信息求得决策表的核属性集; 然后以核属性集为起点, 基于改进的粒子群进化算法, 以最小属性集和最小互信息差为目标, 通过粒子的自适应进化和精英保留策略, 搜索尽可能满足给定目标的全局 Pareto 最优解。

算法中, 用粒子表示条件属性的组合方案, 每个粒子用一个二进制字符串表示, 第 i 个粒子 p_i 代表第 i 种条件属性组合向量 $X_i = \{x_1, x_1, \dots, x_n\} \in \{0, 1\}^n$, 其中代表核属性集 Core 中的属性位恒置1, 并且在进化过程中这些位不参与变化, 则求解属性约简的多目标粒子群进化算法描述如下:

- 基于互信息寻找核属性集 Core;
- 在核属性集 Core 的基础上, 初始化种群大小为 N 的粒子群, 设初始种群为 $P(0)$, 最大迭代次数 iterMax, 当前迭代次数 t ;
- 克隆 $P(t)$ 并通过 2.2 节自适应变异操作得种群 $Q(t)$;
- 计算种群 $P(t)$ 中粒子分别在 $f_1(R)$ 和 $f_2(R)$ 目标的值;
- 计算种群 $Q(t)$ 中粒子分别在 $f_1(R)$ 和 $f_2(R)$ 目标的值;
- $R(t) = P(t) \cup Q(t)$;
- 对种群 $R(t)$ 进行非劣排序, 得到非劣前端 $F_1, F_2, \dots$;
- 按式(5)计算非劣前端 F_1 中粒子的适应度, 其值最高的粒子作为全局最优粒子 P_g ;
- if ($t \geq \text{iterMax}$), 则转 o);
- 从非劣前端 $F_1, F_2, \dots$ 选择前 N 个粒子重构种群 $P(t)$;
- 对 $P(t)$ 中除 P_g 外的每个粒子 p_i , 从 $P(t)$ 中随机选取不同于 i 的粒子 p_j 作为 p_i 的局部最优粒子 P_i ;
- 按运动方程(8)调整 $P(t)$ 中除 P_g 外的每个粒子的位置, 并根据式(9)形成种群 $P(t+1)$;
- P_g 保留至 $P(t+1)$;
- $t = t + 1$, 转 c);
- 结束。

下面分别描述算法中的几个关键部分。

2.1 基于互信息寻找核属性集

对于决策表条件属性集 C 中的每一个属性 r , 分别计算 $I(C - \{r\}; D)$ 的值, 若该值小于 $I(C; D)$, 则表示属性 r 对于决策属性 D 不可缺少, 故其为核属性。搜索核属性集 Core 的算法 searchCore 描述如下:

```
Core = ∅;
for every r ∈ C
    if (I(C - {r}; D) < I(C; D))
        then Core = Core ∪ {r};
}
```

2.2 克隆粒子的自适应变异

粒子群算法具有很强的全局寻优能力, 而局部搜索能力相对较差。为了增强局部搜索能力, 针对属性约简问题, 算法提出基于互信息的自适应变异策略, 对粒子的适应度进行有目的的微调。第 i 个粒子 p_i 的自适应变异操作定义为对随机选取的不为核属性的某位 j , 执行以下操作:

$$p_{ij} = \begin{cases} 0 & f_1(R_i) = 0 \text{ and } p_{ij} = 1 \\ 1 & f_1(R_i) > 0 \text{ and } p_{ij} = 0 \end{cases} \quad (4)$$

其中: R_i 表示粒子 p_i 代表的条件属性子集; j 是随机产生的 $[0, n]$ 的整数 (若 j 与核属性位相同, 则重新产生); n 是决策表条件属性的个数。

该操作的基本思想是粒子根据各自目标的值, 有目的地变异。当目标 $f_1(R_i)$ 达到最小时, 尝试进一步降低目标 $f_2(R_i)$ 的值; 相反, 则尝试进一步降低 $f_1(R_i)$ 的可行性。所以, 针对目标函数 $f_1(R_i)$, 若大于0, 则随机选取粒子二进制字符串中某个值为0的位, 将其变换为1, 以增加选择的条件属性个数; 相反, 若 $f_1(R_i)$ 等于0, 那么, 为了进一步约简属性, 则随机选取不是核属性且值为1的某位, 将其变换为0。

2.3 非劣排序

对于问题2, 当且仅当 $k=1$ 和2, 都有 $f_k(R_p) \leq f_k(R_q)$, 且存在 $k \in \{1, 2\}$, 有 $f_k(R_p) < f_k(R_q)$, 则称粒子 p 支配粒子 q , 记为 $p \succ q$ 。

粒子群 P 进行非劣排序得到非劣前端 F_i 的过程^[8]如下:

- 粒子群 P 中的每个粒子 p , 能支配 p 的粒子数量记为 N_p , 初始值为0; 粒子 p 支配的其他粒子记为集合 S_p 。
- 对于每个 $p \in P, q \in P$, 且 $p \neq q$, 执行如下操作: 如果 $p \succ q$, 则 $S_p = S_p \cup \{q\}$; 如果 $q \succ p$, 则 $N_p = N_p + 1$ 。
- 将所有 $N_p = 0$ 的粒子 p 放入非劣前端 F_1 中。
- 令 $i = 1$ 。
- 令 Q 为空集, 对于每个 $p \in F_i$, 执行如下操作: 对于每个 $q \in S_p, N_q = N_q - 1$; 如果 $N_q = 0$, 则 $Q = Q \cup \{q\}$ 。
- 如果 Q 不空, 则 $i = i + 1, F_i = Q$, 转 e); 否则, 结束。

2.4 适应度函数及最优粒子的选取

适应度函数是对粒子适应性进行评价的确定性指标, 其形式直接决定着群体的进化行为。对于多目标属性约简问题, 本文对非劣前端 F_1 中粒子采用问题2中的公式计算该集合中各个粒子的适应度值, 用适应度最大的粒子作为全局最优粒子 P_g 。

$$\text{fitness}(R) = \left(1 - \frac{f_2(R)}{n}\right) \times e^{-f_1(R)} \quad (5)$$

其中: n 为条件属性的个数, $f_1(R)$ 和 $f_2(R)$ 是问题2中的两个目标的值。

局部最优粒子的选取方法是: 第 i 个粒子 p_i 的局部最优粒

子 P_i 采取从当前非劣解集中随机选择一个不同于 i 的粒子作为 i 的局部最优。

2.5 粒子运动方程

对于属性约简问题,关键是要解决代表约简属性组合向量的粒子如何在局部最优、全局最优影响下的变化。受文献[9]的启发,新的粒子运动方程定义如下:

$$X_i^{new} = X_i(t) \oplus ((c_1 \times r_1) \cdot P_i(t)) \oplus ((c_2 \times r_2) \cdot P_g(t)) \quad (6)$$

其中:符号“ $\oplus$ ”定义为交叉二元运算; $X_i(t)$ 表示 t 时刻第 i 个粒子即 p_i^t 所对应的向量; $P_i(t)$ 表示到 t 时刻为止,第 i 个粒子 p_i^t 目前找到的最好组合,即局部最优粒子; $P_g(t)$ 表示目前群体找到的最好组合; c_1 和 c_2 为加速因子; r_1 和 r_2 是在 $[0,1]$ 范围内的两个随机数, $c_1 \times r_1$ 和 $c_2 \times r_2$ 的值分别用于控制粒子与局部最优和全局最优交叉时的最大交换区域长度, c_1 和 c_2 值的大小直接关系到从局部最优和全局最优的学习强度。实验中, c_1 的值取为 $[n - \text{card}(\text{Core})]/4$,最小取 2, $c_2 = c_1/2$,其中 n 为条件属性个数, $\text{card}(\text{Core})$ 是核属性集的属性个数。

粒子与局部最优和全局最优交叉方法是取局部或全局最优对应序列 P 中随机产生的某位开始、指定长度的子序列与序列 X 中相同位置的子序列交换。交叉后得到两个新序列,对这两个序列分别进行适应值评估、保留适应值较高的个体。

粒子 p_i^t 在经过式(6)的位置调整后到达新的位置 X_i^{new} ,此时粒子记为 p_i^{new} 。 p_i^{new} 按其 与 p_i^t 的支配关系确定其能否进入下一代参与进化:

$$p_i^{i+1} = \begin{cases} p_i^{new} & \text{若 } p_i^{new} > p_i^t \\ p_i^t & \text{若 } p_i^t > p_i^{new} \\ r > 0.5? p_i^{new} : p_i^t & \text{else} \end{cases} \quad (7)$$

即:若 p_i^{new} 支配粒子 p_i^t ,则 p_i^{new} 进入种群 P_{t+1} ;如果 p_i^t 支配 p_i^{new} ,则 p_i^t 进入种群 $P(t+1)$;如果两者互不支配,则按 50% 的概率从中选一个进入 $P(t+1)$ 。

3 实验与分析

实验从 UCI 数据库(<http://www.ics.uci.edu>)中选取 5 个数据集来测试本文算法,并与文献[1]基于差别矩阵(记为算法 1)和文献[10]基于遗传算法(记为算法 2)的属性约简进行比较。本文算法中,粒子数量 N 设为实验数据集的条件属性个数,最大迭代次数为 100 次。算法 1 和 2 的参数按文献[1, 10]的要求设置。

3.1 实验结果

算法运行 10 次,达到最大迭代次数为止,统计每个算法 10 次运行求得的属性子集所包含的属性个数。本文算法的实验结果如表 1 所示。表中最优约简结果列中的数字表示测试数据集中属性所在列号。

表 2 给出三种算法的实验结果对比。表中数据表示 10 次约简获得的约简属性集中包含的属性个数及对应的次数。比如 $2^{(9)}3^{(1)}$ 表示 10 次实验结果中,约简后属性个数为 2 的有 9 次,为 3 的有 1 次。

3.2 实验结果分析

从实验结果可以看出,当数据集包含的记录数和条件属性较少时,三种算法约简结果差距不大。但随着实例数量及条件属性的增多,对比算法找到最小属性约简的几率逐渐减少,而

本文算法尽管也有所降低,但找到最优结果仍具有较大的概率,说明相比另外两种算法而言,本文算法具有更好的收敛性和稳定性。这得益于四方面的原因:a)本文算法是在核属性集的基础上搜索最小属性约简,排除很多非最小属性约简方案;b)算法对粒子群进行克隆,并通过基于互信息差的自适应变异增加了局部搜索能力;c)基于精英保留的非支配排序策略为算法提供了良好的非劣解集;d)通过加权法从当前非劣解集选出全局最优粒子和从非劣解集中随机选取局部最优粒子的方法为粒子下一次进化指引了方向。

表 1 本文算法约简结果

数据集	条件属性数	记录数	找到的核属性	最优约简结果
Zoo	16	101	{6,13}	{3,4,6,8,13}
Vote	16	435	{1,2,3,9,11,13,16}	{1,2,3,4,9,11,13,15,16}
Soybean _{small}	35	47	$\emptyset$	{4,22}
Dermatology	34	366	$\emptyset$	{1,3,4,16,32,34}
kr-vs-kp	36	3 196	{1,3,4,5,6,7,10,12,13,15,16,17,18,20,21,23,24,25,26,27,28,30,31,33,34,35,36}	{1,3,4,5,6,7,10,11,12,13,15,16,17,18,20,21,22,23,24,25,26,27,28,30,31,33,34,35,36}

表 2 三种算法的实验结果对比

数据集	本文算法	算法 1	算法 2
Zoo	5 ⁽¹⁰⁾	5 ⁽¹⁰⁾	5 ⁽¹⁰⁾
Vote	9 ⁽¹⁰⁾	9 ⁽¹⁰⁾	9 ⁽¹⁰⁾
Soybean _{small}	2 ⁽⁹⁾ 3 ⁽¹⁾	2 ⁽⁵⁾ 3 ⁽³⁾ 4 ⁽²⁾	2 ⁽⁵⁾ 3 ⁽⁴⁾ 4 ⁽¹⁾
Dermatology	6 ⁽⁶⁾ 7 ⁽³⁾ 8 ⁽¹⁾	6 ⁽²⁾ 7 ⁽³⁾ 8 ⁽⁴⁾ 9 ⁽¹⁾	7 ⁽⁵⁾ 8 ⁽³⁾ 9 ⁽²⁾
kr-vs-kp	29 ⁽⁵⁾ 30 ⁽³⁾ 31 ⁽¹⁾ 32 ⁽¹⁾	30 ⁽⁴⁾ 31 ⁽³⁾ 32 ⁽²⁾ 33 ⁽¹⁾	30 ⁽⁴⁾ 31 ⁽⁴⁾ 32 ⁽¹⁾ 33 ⁽¹⁾

4 结束语

本文算法通过基于互信息寻找到决策表的核属性,以核属性为基础初始化粒子群,重新定义粒子运动形式以及最优粒子选取方法,并加入基于互信息的自适应克隆变异机制;同时,结合带精英保留的非支配排序策略搜索决策表的最小属性约简。通过对 UCI 数据库中的多个数据集的测试,表明算法在收敛性及稳定性方面都优于对比算法。

参考文献:

[1] 王加阳,高灿.改进的基于差别矩阵的属性约简算法[J].计算机工程,2009,35(3):66-67.
[2] JENSEN R, SHEN Qiang. New approaches to fuzzy-rough feature selection[J]. IEEE Trans on Fuzzy Systems, 2009, 17(4):824-838.
[3] 付昂,王国胤,胡军.基于信息熵的不完备信息系统属性约简算法[J].重庆邮电大学学报:自然科学版,2008,20(5):586-592.
[4] MIAO Duo-qian, HOU Li-shan. A comparison of rough set methods and representative inductive learning algorithms[J]. Fundamenta Informaticae, 2004, 59(2-3):203-219.
[5] 叶东毅,廖建坤.最小约简问题的一个免疫离散粒子群算法[J].小型微型计算机系统,2008,29(6):1088-1092.
[6] 王国胤,于洪,杨大春.基于条件信息熵的决策表约简[J].计算机学报,2002,25(7):759-766.

循环执行之前计算一次分配方案,此后各轮循环均使用该分配结果。

测试环境:测试系统配置 1 个 IBM BladeCenter H 刀片中心,14 个刀片服务器作为计算节点,每个节点均支持 SMP,配置 2.0 GHz 四核处理器和 4 GB 内存。在本实验中使用该系统中 8 个计算节点。

4.2 计算节点固定、干扰节点不同的实验测试

首先测试使用 8 个计算节点且其中有若干个计算节点受到干扰时的计算时间。如表 1 所示,当使用 8 个计算节点时,干扰节点数为 0 和 8 时,DLB 比 SLB 计算时间略长,这是由于此时系统的各个计算节点负载相同,DLB 与 SLB 任务分配情况相同,但 DLB 需要额外的任务分配计算开销,而 SLB 每轮循环均采用同样的任务分配方案,只在循环执行之前计算一次分配方案,此后的循环均使用该分配结果。因此,当系统负载平衡时,DLB 与 SLB 比较并没有体现出优势。而当干扰节点数为 17 时,此时八个计算节点的负载出现不平衡,DLB 的计算时间均显著少于 SLB。由此说明,在负载不平衡环境中,该负载平衡算法能够有效减少计算时间,提高并行计算效率。

表 1 SLB 和 DLB 在受到干扰情况下的执行时间 /s

干扰节点数	0	1	2	3	4	5	6	7	8
SLB	23.30	25.17	25.41	26.09	25.76	26.08	25.12	25.65	25.41
DLB	23.76	21.07	20.99	21.23	20.74	20.58	20.44	21.60	25.56

4.3 干扰节点固定、计算节点不同的实验测试

为了测试 DLB 在系统出现干扰情况下的可扩展性,固定一个节点作为干扰节点,计算节点由两个增加至八个。如图 3 所示,系统无干扰时 SLB 和 DLB 在单个节点中的计算时间作为串行时间,尽管有一个节点受到干扰,SLB 和 DLB 的加速比均随着计算节点数量的增加呈增长趋势;DLB 与 SLB 加速比的绝对差值随着计算节点数量的增加不断增大。由此可见,在负载不平衡情况下,DLB 比 SLB 具有更好的可扩展性。

并行效率比较如图 4 所示,SLB 和 DLB 的并行效率都随着计算节点数的增加而降低,这是因为随着节点的增加,系统可用的资源总量也随着增加,而计算规模没有增加,因此并行效率均不断降低。但 DLB 的并行效率始终高于 SLB,而且效率始终保持在 0.8 以上。由此说明,在负载不均衡情况下,该动态负载平衡算法比静态负载平衡算法更优越,能够有效提高资源利用率。

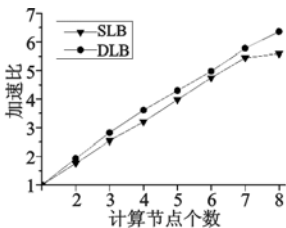


图3 固定1个干扰节点,加速比随计算节点增加的变化

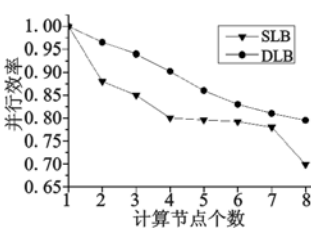


图4 固定1个干扰节点,并行效率随计算节点增加的变化

结合图 3 和 4 说明,该动态负载平衡算法有效地提高了多处理器系统的资源利用率和并行计算性能,而且具有较好的可扩展性。

5 结束语

在并行计算应用中,动态负载平衡确实是提高多处理器系统的资源利用率和并行计算性能的重要途径。本文提出的基于反馈机制、以处理器作业速度为负载指标,根据各处理器作业速度比分配任务的负载平衡算法,有效地解决了 PTM-NL 问题,有效地提高了多处理器系统的资源利用率和并行计算效率,且实现简单、易于理解,具有较好的自适应性和可扩展性。算法的主要缺点是负载变化频繁时效果不佳。例如,当系统负载变化的时间间隔远小于任务分配的时间间隔时,若原本轻载状态的处理器负载加重,原本重载状态的处理器负载变轻,由于该算法并不进行任务迁移,会导致执行时间显著增加。未来的研究将集中于解决该算法在系统负载变化频繁时的负载平衡问题。

参考文献:

[1] YANG Xue-jun, DOU Yong, HU Qing-feng. Progress and challenges in high performance computer technology[J]. Journal of Computer Science and Technology, 2006, 21(5): 674-681.

[2] 蒋江,张民选,廖湘科. 基于多种资源的负载平衡算法的研究[J]. 电子学报, 2002, 30(8): 1148-1152.

[3] ZHENG Geng-bin. Achieving high performance on extremely large parallel machines: performance prediction and load balancing[D]. Urbana: University of Illinois at Urbana Champaign, 2005.

[4] 杨际祥,谭国真,王荣生. 并行与分布式计算动态负载均衡策略综述[J]. 电子学报, 2010, 38(5): 1122-1130.

[5] BRANCO K R L J C, ORDONEZ E D M. Load indices-past, present and future[C]//Proc of International Conference on Hybrid Information Technology. Washington DC: IEEE Computer Society, 2006: 206-214.

[6] 唐丹,金海,张永坤. 集群动态负载平衡系统的性能评价[J]. 计算机学报, 2004, 27(6): 803-811.

[7] KUNZ T. The Influence of different workload descriptions on a heuristic load balancing scheme[J]. IEEE Trans on Software Engineering, 1991, 17(7): 725-730.

[8] 刘滨,石峰,高玉金,等. 多目标迁移机制的动态负载平衡算法[J]. 哈尔滨工程大学学报, 2008, 29(8): 842-845.

[9] ROTH P C, ARNOLD D C, MILLER B P. MRNet: a software-based multicast/reduction network for scalable tools[C]//Proc of ACM/IEEE Conference on Supercomputing. New York: ACM, 2003: 21.

[10] WILKINSON B, ALLEN M. 并行程序设计[M]. 陆鑫达,译. 北京: 机械工业出版社, 2002: 160.

(上接第 492 页)

[7] 陈民铎,张聪睿,罗辞勇. 自适应进化多目标粒子群优化算法[J]. 控制与策略, 2009, 24(12): 1851-1855.

[8] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolu-

tionary Computation, 2002, 6(2): 182-197.

[9] 余伶俐,蔡自兴. 改进混合离散粒子群的多种优化策略算法[J]. 中南大学学报: 自然科学版, 2009, 40(4): 1047-1053.

[10] 张忠林,张军,米伟. 一种具有记忆功能的遗传算法属性约简方法[J]. 计算机应用研究, 2010, 27(1): 96-98.