

# 基于量子粒子群优化的 DAG 并行任务调度研究\*

张 聪, 沈惠璋

(上海交通大学 安泰经济与管理学院, 上海 200052)

**摘 要:** 任务调度是网络并行计算系统的核心问题之一。在有向无环图(DAG)描述问题的基础上,提出了一种进行并行任务调度的量子粒子群优化算法。首先对 DAG 并行任务调度问题作出定义,并给出了优化问题的目标;然后分别讨论了问题的编码表示、解码方案、位置向量的计算方法、离散问题连续化、算法的总体流程等;最后给出算法的仿真实验情况及分析,实验结果表明,该算法有良好的全局寻优性能和快捷的收敛速度,调度效果优于遗传算法和粒子群优化算法。

**关键词:** 任务调度; 量子粒子群优化; 有向无环图

**中图分类号:** TP314      **文献标志码:** A      **文章编号:** 1001-3695(2010)07-2458-04

**doi:**10.3969/j.issn.1001-3695.2010.07.015

## Research on DAG parallel task scheduling problem based on quantum-behaved particle swarm optimization

ZHANG Cong, SHEN Hui-zhang

(Antai College of Economics & Management, Shanghai Jiaotong University, Shanghai 200052, China)

**Abstract:** Task scheduling is one of the important problems in parallel computing system. This paper proposed a quantum-behaved particle swarm optimization algorithm for task scheduling based on directed acyclic graph. First redefined the parallel task scheduling problem and its aim. Then discussed the representation of the encoding, the procedure of the decoding, the computational method of position vector, the continuative of the discrete problem and the structure of the algorithm respectively. In the end, presented the algorithm simulation, experiment result analysis and the conclusions. The simulation results show that this algorithm has better global optimizing ability and more rapid convergence, and it is superior to genetic algorithm and particle swarm optimization algorithm.

**Key words:** task scheduling; quantum-behaved particle swarm optimization(QPSO); directed acyclic graph(DAG)

网络并行计算环境下的任务调度问题是指在一定约束条件下,如何将一组任务分配到多台处理机上执行的组合优化问题,其已被证明是 NP 完全问题,不可能在多项式时间内找到问题的最优解<sup>[1,2]</sup>。目前常见的并行任务调度问题按照任务之间有无数据依赖关系可以划分为独立任务调度和依赖关系任务调度。前者在调度任务时不需要考虑任务之间的数据依赖关系;而后者通常用有向无环图(DAG)表示任务之间的数据依赖关系,在调度过程中满足任务之间的数据依赖关系。依赖关系任务调度的求解优化过程比独立任务调度的要复杂许多,且其适用范围也更广。以 DAG 表示的并行任务模型的研究得到了广泛关注和迅速发展。近年出现的一些启发式算法(如模拟退火算法、遗传算法等)为求解此类 NP 完全问题提供了新的途径<sup>[3-5]</sup>,但是这些算法有些复杂性太高难以实现,有些实现起来太费时,所以有必要寻求更好的算法来解决此问题。

粒子群优化(PSO)算法是由 Kennedy 等人<sup>[6]</sup>提出的一种源于对鸟群捕食行为模拟的进化计算技术,已成为进化计算的一个最吸引人的分支。与遗传算法类似,PSO 是一种基于迭代的优化方法,系统初始化为一组随机解,通过迭代搜寻最优值,

但是在许多实际应用领域,更胜于遗传算法,尤其是在非线性优化问题上。量子粒子群优化(QPSO)算法是在传统的 PSO 基础上提出的一种新型的具有高效率全局搜索能力的进化算法<sup>[7,8]</sup>。它主要是引入量子物理的思想改进了 PSO 的进化方法,即更新粒子位置的方法;在更新粒子位置时重点考虑各个粒子的当前局部最优位置信息和全局最优位置信息。QPSO 具有调整参数少、容易实现、收敛能力强等优势。为适应任务分配问题的求解,本文设计出合适的粒子编码,利用改进的量子粒子群算法求解任务分配问题,并与其他算法相比较。实验结果表明,本文提出的算法可以获得质量更高的解。

### 1 问题描述

本模型的计算系统由一系列异构的处理机组成,需要处理的总任务已分解成一系列子任务。模型的约束条件为:任务执行具有非抢占性,即处理机只有在执行完某个任务之后才能处理另外一个任务;另外这些任务之间具有前驱后继的数据依赖关系,某个子任务只有在它所有的前驱任务处理完毕后才能开始执行。该模型的调度目标就是要使得整个 DAG 图的调度长度最短。

**收稿日期:** 2009-12-22; **修回日期:** 2010-02-13      **基金项目:** 国家自然科学基金资助项目(70671070);高等学校博士学科点专项科研基金资助项目(20070248054)

**作者简介:** 张聪(1975-),男,安徽阜阳人,讲师,博士研究生,主要研究方向为进化计算、数据挖掘(z\_c\_cn@163.com);沈惠璋(1958-),男,天津人,教授,博导,博士,主要研究方向为数据挖掘、网络安全。

为了便于分析问题,可以用下列五元组表述:

$$\Pi = (P, G, \Theta, \Psi, \Omega)$$

其中:

$P = \{P_1, P_2, \dots, P_n\}$  为  $n$  个处理机的集合。

$G$  是子任务集  $T$  的依赖关系图,它通过 DAG 来表示各个子任务间的调度约束关系。 $G = (T, E)$ , 其中  $T = \{T_1, T_2, \dots, T_m\}$  为  $m$  个子任务的集合,一个子任务  $T_i$  就是图  $G$  中的一个节点, $E$  是任务依赖关系图中的有向边集。 $\langle T_i, T_j \rangle \in E (i, j = 1, 2, \dots, m)$ , 则表示在子任务  $T_i$  没有完成之前,任务  $T_j$  不能执行。这时称  $T_i$  为  $T_j$  的一个前驱, $T_j$  为  $T_i$  的一个后继, $E$  可用邻接矩阵存储。

$\Theta$  是一个  $m \times n$  矩阵,其元素  $\theta_{ij}$  表示任务  $T_i$  在处理机  $P_j$  上的执行时间,假设每个任务的执行时间预知 ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ )。

$\Psi$  是一个  $m \times m$  矩阵,其元素  $\psi_{ij}$  表示任务  $T_i$  与  $T_j$  之间的数据传输延时 ( $i, j = 1, 2, \dots, m$ ), 同时假设各处理机间的通信能力是相同的,且忽略网络拥塞,即传输的数据量是惟一影响  $\psi_{ij}$  大小的因素。

$\Omega$  是一个  $m \times n$  的任务分配矩阵,其中  $\omega_{ij} = 1$  表示  $T_i$  分配到处理机  $P_j$  上执行;否则  $\omega_{ij} = 0 (i = 1, 2, \dots, m; j = 1, 2, \dots, n)$ 。

要实现的目标是寻找一个分配调度策略,将  $m$  个子任务分配到  $n$  个处理机上,合理调度各个子任务的执行次序,使得各子任务在满足依赖关系图  $G$  的约束下,整个任务的完成时间最短。现假设某一合法的分配调度  $S$ , 将  $T$  中的  $m$  个子任务分配到  $n$  个处理机上,其中子任务  $T_i$  被分配到处理机  $P_j$  上执行,那么子任务  $T_i$  在处理机  $P_j$  上的执行时间满足以下两式:

$$St(T_i, P_j) = \max_{T_k \in \text{Pred}(T_i)} (Ft(T_k, P_r) + (1 - \omega_{ij})\psi_{ki}) \quad (1)$$

$$Ft(T_i, P_j) = St(T_i, P_j) + \theta_{ij}; i, k = 1, 2, \dots, m; j, r = 1, 2, \dots, n \quad (2)$$

其中: $St(T_i, P_j)$  和  $Ft(T_i, P_j)$  分别表示子任务  $T_i$  在处理机  $P_j$  上的开始执行时刻和结束执行时刻; $\text{Pred}(T_i)$  表示子任务  $T_i$  的前驱节点集合,假设子任务  $T_k \in \text{Pred}(T_i)$  被分配到处理机  $P_r$  上。

根据式(1)(2)迭代计算,可得到所有子任务的结束执行时刻。设  $\Gamma(S)$  为在调度策略  $S$  下完成任务所使用的总时间,那么: $\Gamma(S) = \max(Ft(T_i, P_j)); \forall i = 1, 2, \dots, m; j = 1, 2, \dots, n$ 。

任务调度目标就是  $\min(\Gamma(S)) \forall S$ , 即寻找一个分配调度  $S$ , 使得  $\Gamma(S)$  最小。

鉴于本文主要考虑任务调度问题,在不失问题一般性的情况下,可忽略数据传输延时,即在下文中可假设所有的  $\psi_{ij} = 0$ 。

## 2 算法

### 2.1 PSO 算法

粒子群优化(PSO)算法是一种进化计算方法,是一种基于迭代的优化工具。该算法通过群体中各粒子间的合作与竞争来搜索全局最优解。

系统初始化为一组共  $n$  个随机解,通过迭代搜寻整个群体的最优值。粒子  $i$  的当前位置为  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ , 其飞行速度记为  $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ , 在解空间中追随适应度最优的粒子进行搜索。在每一次迭代中,粒子通过跟踪两个“极值”来更新自己:a) 每个粒子本身所找到的最优解  $pbest$ 。如果粒子当前位置对应的适应度小于  $pbest$  的适应度,则  $pbest$  更新为当

前位置。b) 整个种群从起始到目前所找到的最优解  $gbest$ 。每个粒子按以下两个公式进行动态进化,调整粒子的位置:

$$v_{i,d}(t+1) = wv_{i,d}(t) + c_1 r_{1,d}(t) (pbest_{i,d} - x_{i,d}(t)) + c_2 r_{2,d}(t) (gbest_d(t) - x_{i,d}(t)) \quad (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

其中: $w$  是惯性权重,动态调整惯性权重以平衡收敛的全局性和收敛速度; $c_1$  和  $c_2$  为加速常数,通常在  $0 \sim 2$  取值, $c_1$  调节粒子飞向自身最好位置方向的步长, $c_2$  调节粒子飞向全局最好位置方向的步长; $r_{1,d}(t), r_{2,d}(t) \sim U(0, 1)$ , 且  $d = 1, 2, \dots, n$ 。为了减少在进化过程中粒子离开搜索空间的可能性,粒子的每一维速度被限定在  $[-V_{max}, V_{max}]$  内。

### 2.2 QPSO 算法

Sun 等人从量子力学的角度,通过对粒子收敛行为的研究,基于粒子群算法提出了一种新的算法模型——量子粒子群(QPSO)算法。在该算法中,由于粒子满足聚集态的性质完全不同,使粒子在整个可行解空间中进行搜索寻求全局最优解,因而 QPSO 算法在搜索能力上远远优于所有已开发的 PSO 算法。

QPSO 算法参数个数少,进化方程的形式更加简单,更容易控制。在 QPSO 算法中,每一个粒子必须收敛于各自的随机点  $P_i$ , 粒子按照下面的三式移动:

$$mbest = \frac{1}{m} \sum_{i=1}^m P_i = (\frac{1}{m} \sum_{i=1}^m P_{i1}, \dots, \frac{1}{m} \sum_{i=1}^m P_{ij}) \quad (5)$$

$$PP_{ij} = fP_{ij} + (1-f)P_{gj}, f = \text{rand} \quad (6)$$

$$x_{ij} = PP_{ij} \pm a |mbest_j - x_{ij}| \ln(1/u), u = \text{rand} \quad (7)$$

其中: $mbest$  是粒子群  $pbest$  的中间位置; $P_{ij}$  为粒子本身所找到的最优解  $pbest$ ;  $P_{gj}$  为整个粒子群目前找到的最优解  $gbest$ ;  $PP_{ij}$  为  $P_{ij}$  与  $P_{gj}$  之间的随机点; $a$  为 QPSO 的收缩扩张系数,它是 QPSO 收敛的一个重要参数,第  $t$  次迭代时一般可取

$$a = a_{max} - t(a_{max} - a_{min})/t_{max} \quad (8)$$

其中: $t_{max}$  是迭代的最大次数, $a_{max}$  与  $a_{min}$  分别是最大和最小系数。QPSO 的算法流程如下:

- 迭代次数  $t = 0$ , 对种群的每个粒子的位置向量进行初始化。
- 根据目标函数计算每个粒子的目标函数值。
- 更新每个粒子的新局部最优位置  $P_i$ 。
- 更新粒子群的全局最优位置  $P_g$ 。
- 根据式(5)计算  $mbest$ 。
- 根据式(6)计算每个粒子随机点  $PP_i$ 。
- 根据式(7)(以一定的概率取加或减)更新每个粒子的新位置。
- 令  $t = t + 1$ , 返回到 b), 重新计算,直到终止条件满足。

## 3 基于 QPSO 的 DAG 并行任务调度

### 3.1 编码与解码

任务调度的常见编码包括基于任务的编码、基于操作的编码和基于优先规则的编码等。由于 DAG 并行任务调度的复杂性,采用任一种上述编码形式均无法保证所有解的合法性,这将浪费大量的求解时间。本文设计了一种复合的编码方案:编码长度为  $2m$ , 可描述为两个向量,第一个向量采用基于优先规则的编码方式,为一个包含  $m$  维的向量  $(R_1, R_2, \dots, R_m)$ 。其

中  $R_i$  表示在算法迭代过程中第  $i$  次迭代时发生的冲突利用优先规则  $R_i$  消除。本文选择了五种优先规则,包括最短执行时间(SPT)、最长执行时间(LPT)、最早开工时间(EST)、最早完工时间(EFT)、最晚完工时间(LFT),数字 0、1、2、3、4 分别对应优先规则 SPT、LPT、EST、EFT、LFT。第二个向量是处理机分配向量,即一个包含  $m$  维的向量  $(M_1, M_2, \dots, M_m)$ 。其中  $M_i$  表示编号为  $i$  的子任务被分配到编号为  $(M_i)$  的处理机上执行(所有处理机编号为 0, 1,  $\dots$ ,  $n-1$ )。

在解码过程中,设  $t$  为调度的时间步,PS 为调度列表。其中  $PS_t$  为第  $t$  步调度执行的子任务;TS 为所有前驱已经被调度的子任务所构成的集合。解码算法如下:

- a) 令  $t = 1$ , PS 为空, TS 由所有无前驱的子任务构成。
- b) 由 TS 中所有子任务编码,在处理机分配向量  $(M_1, M_2, \dots, M_m)$  中找到分配给每个子任务的处理机,并在  $\Theta$  中找到具体执行时间。
- c) 依据约束条件和执行时间,得到 TS 中每个子任务对应的指标时间(开工、完工或执行时间),由编码  $R_i$  所对应的优先规则选出一个子任务(如优先规则为最短执行时间,则选 TS 中执行时间最短的子任务,如果有多个子任务符合优先规则,则任选一个),该子任务就是  $PS_t$ ,从 TS 中删除它,并将其加入 PS 的尾部。
- d) 逐个考察  $PS_t$  的后继子任务,如果该子任务无其他前驱,或其他前驱都已被调度执行,则将其加入 TS 中。
- e) 令  $t = t + 1$ ,若  $t < m + 1$ ,则转 b);否则,算法终止。

通过下面示例说明解码过程:

任务的 DAG 如图 1 所示。

优先规则向量:

(0 3 2 1 4 0)

即:(SPT EFT EST LPT LFT SPT)

处理机分配向量:

(0 1 1 0 1 0)

即(P1 P2 P2 P1 P2 P1)

在  $\Theta$  中查到的处理时间:

(2 4 6 5 3 7)

处理时间指 1~6 号子任务在对应处理机上的执行时间。

根据示例数据得到的调度列表 PS 为  $(T_1 T_2 T_4 T_6$

$T_3 T_5)$ ,甘特图如图 2 所示。

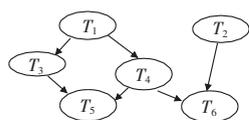


图 1 任务关系图 DAG

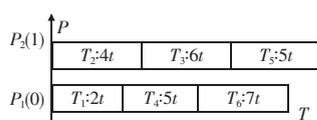


图 2 任务调度的甘特图

由上述编码方式和解码过程可知,本文编码能保证调度的可行性,且码长较短,无冗余,解码复杂性不高。

### 3.2 QPSO 中向量的计算方法

对每个粒子,它的优先规则向量和处理机分配向量可以表示为  $X_{priority}(1..m)$  和  $X_{machine}(1..m)$ ,按式(5)~(7)计算这两个向量。由于前面所述的 QPSO 为连续空间算法,而 DAG 并行任务调度问题为整数规划问题,将离散优化转变成对实数向量的连续优化,具体过程如下:

- a) 将每个向量切断分成若干个子串,各段子串的长度可以相等,也可以不相等,子串形如  $(q_1, q_2, \dots, q_k)$ 。

- b) 从整数组成的子串到实数作一个映射,可表示为

$$r = c \times \sum_{i=1}^k q_i \times b^{k-i} \tag{9}$$

其中: $r$  为映射的实数; $c$  是常数,一般取足够小的实数,本文取值为 0.01; $b$  为基数,对于  $X_{priority}$ ,  $b$  取值为 5,对于  $X_{machine}$ ,  $b$  取值为  $n$ 。

- c) 在计算任务执行总时间前,需将  $r$  转换为子串,即式(9)的逆映射:

$$q_i = \left( \frac{r}{c} - \sum_{j=1}^{i-1} q_j \times b^{k-j} \right) \text{div } b^{k-i} \tag{10}$$

其中:div 为整除,得到的商  $q_i$  为整数,在实际运算时,可用一个循环, $i$  从 1~ $k$  得到子串中所有分量。

例如 9 个子任务的情况,  $X_{priority} = (2\ 4\ 2\ 1\ 4\ 3\ 4\ 1\ 0)$ ,分为三段,各子串长度均为 3。

子串 (242); (143); (410)

变换后得到

$r$  0.72 0.48 1.05

经过迭代后的情况:

逆变换后得到

$r$  0.53 1.12 0.91

子串 (203) (422) (331)

在初始化时,可省掉式(9)的转换过程,直接给粒子位置赋实数。

解决了连续化问题之后,还有一个边界问题,如上例  $r$  的取值为  $[0, 1.24]$ ,如迭代过程中  $z$  的运算结果超出范围时,将  $r$  值取在边界上。若  $r < 0$ ,取值 0;若  $r > 1.24$ ,取值 1.24。

通过上述映射和逆映射,整数规划问题转换为连续优化问题,从而可以利用 QPSO 优化获得高质量的解。

### 3.3 算法流程

- a) 初始化粒子群,根据编码方案设定各粒子的随机位置。
- b) 根据式(10)将每个粒子的实数向量转换为整数向量。
- c) 对每个粒子的整数向量解码后,计算每个粒子的目标函数值。
- d) 更新每个粒子的局部最优值  $P_i$ 。
- e) 更新粒子群的全局最优值  $P_g$ 。
- f) 根据式(5)计算 mbest。
- g) 根据式(6)计算每个粒子随机点  $PP_i$ 。
- h) 根据式(7)更新每个粒子的新位置。
- i) 返回 b) 步,直到满足迭代的次数。

## 4 仿真实验与结果分析

### 4.1 实验参数选取

本文的仿真实验是在 MATLAB 软件上实现的。实验所用 DAG 图随机生成,每个任务节点有 1~4 个前驱与后继,估计运行时间  $\theta_j$  为 1~50 s 的随机数。实验计算了文献[3,4]的算法、PSO 与本文的 QPSO 共四种情况,算法中主要参数:种群大小为 80,终止代数为 1 500,  $a_{max}$  取值 1,  $a_{min}$  取值 0.5;PSO 的惯性权重  $w$  与 QPSO 中的收缩扩张系数  $a$  取值相同,  $c_1$  和  $c_2$  均为 2,编码、解码、连续化与边界问题均使用本文的方案;文献[3]算法的杂交概率为 1.0,变异概率为 0.05;文献[4]算法的内部杂交概率为 0.8,迁移概率为 0.2,演化策略中的参数为  $\mu/\lambda = 5$ 。

### 4.2 计算结果与分析

对于随机生成的同一个 DAG 图,分别用上述四种算法进行计算,记录各算法收敛时得到的最优解的完成时间和收敛时的进化代数。计算结果如表 1 所示。为了消除数据随机性的影响,更好地反映算法的性能,表 1 中的进化代数是 100 次进化的平均收敛代数,完成时间是所有 100 次进化中得到的最优解的平均完成时间。图 3 为四个处理机 100 个子任务情况下四种算法分别进化的静态性能曲线,列出了各算法在不同进化代数时所找到的最优解。表 2 为四种算法在进化中能收敛到其最优解的次数占实验总次数的百分比。

表 1 仿真实验结果

处理机 个数	子任务 个数	完成时间/s				收敛时的进化代数			
		文献[3] 算法	文献[4] 算法	PSO 算法	本文 算法	文献[3] 算法	文献[4] 算法	PSO 算法	本文 算法
2	25	285	271	279	265	39	31	25	29
	50	565	543	551	538	166	131	108	125
	100	1 548	1 429	1 427	1 416	418	339	285	323
4	25	193	186	188	179	53	46	33	38
	50	457	429	428	417	194	168	135	157
	100	1 198	1 136	1 139	1 073	516	468	323	339
8	25	152	141	138	134	73	54	49	52
	50	341	297	292	281	336	217	163	212
	100	1 106	911	923	875	727	621	538	601

表 2 收敛到其已知最优解的次数占进化总次数的百分比 %

各算法	子任务个数 25	子任务个数 50	子任务个数 100
文献[3]算法	87	64	48
文献[4]算法	96	92	81
PSO 算法	92	89	67
本文算法	100	99	96

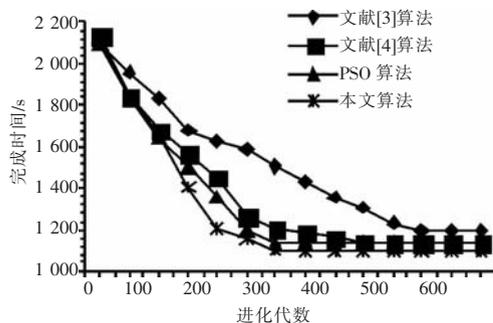


图 3 静态性能曲线(n=4,m=100)

由表 1、2 和算法的静态性能曲线可以得出:

a) 在任务数较多、处理机较多的情况下,PSO 与本文 QP-SO 算法的收敛速度比文献[3]算法快很多,但与文献[4]算法比较时,PSO 算法的收敛速度明显比文献[4]算法快,本文 QP-SO 算法则与文献[4]算法相当;而在任务数少的情况下,除文献[3]算法稍慢,其他算法相差不大。

b) 本文 QPSO 算法能找到的最优解比文献[3,4]算法有

明显的提高,尤其是子任务数较多、处理机数较多时。

c) PSO 与本文 QPSO 算法比较时,发现 QPSO 算法的收敛速度比 PSO 算法慢,但得到的最优解比 PSO 算法好。

这是因为:首先,本文对问题的编码能够覆盖整个解空间,相对来说文献[3,4]的算法只能从一个相对较小的空间内搜索;其次,本文采用了离散空间到连续空间的转换过程,它不仅满足了 QPSO 算法对待解问题的取值要求,还在一定程度上能更好地保护与遗传优良的解片段。另外,PSO 算法收敛过快,而 QPSO 的量子搜索方式对传统的 PSO 算法有了很大的改进,实验证明可防止早熟。

### 5 结束语

基于 DAG 的并行任务调度问题是 NP 难问题,传统的优化算法很难求得全局最优解,虽然已有人将遗传算法应用于此问题,但结果有待进一步改善。本文给出了新的问题定义,对 QPSO 算法作出调整与改进,编码表示采用了适合于任务调度问题的优先规则与处理机分配相结合的形式,并将离散空间优化问题转换为连续空间优化问题,使得 QPSO 有较好的搜索能力。最后通过仿真实验得到的一系列数据,表明了本文的改进 QPSO 算法比遗传算法和 PSO 算法有更好的性能,并有理由认为,合理的编码表示与高效的搜索策略相结合是任务分配调度问题全局寻优的有效途径。

### 参考文献:

[1] GRAY M R, JOHNSON D S. Computers and intractability: a guide to the theory of NP-completeness [M]. New York: W. H. Freeman and Co., 1979.

[2] AHMAD I, KWOK Y K. On parallelizing the multiprocessor scheduling problem [J]. IEEE Trans on Parallel and Distributed Systems, 1999, 10(4): 414-432.

[3] HOU E S H, ANSARI N, HONG Ren. A genetic algorithm for multiprocessor scheduling [J]. IEEE Trans on Parallel and Distributed Systems, 1994, 5(2): 113-120.

[4] 钟求喜, 谢涛, 陈火旺. 基于遗传算法的任务分配与调度 [J]. 计算机研究与发展, 2000, 37(10): 1197-1203.

[5] 张聪, 马义忠. 异构计算系统中基于遗传算法的任务分配与调度 [J]. 微电子学与计算机, 2004, 21(6): 74-78.

[6] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.

[7] SUN Jun, FENG Bin, XU Wen-bo. Particle swarm optimization with particles having quantum behavior [C]// Proc of Congress on Evolutionary Computation. 2004: 325-331.

[8] SUN Jun, XU Wen-bo, FENG Bin. A global search strategy of quantum behaved particle swarm optimization [C]// Proc of IEEE Conference on Cybernetics and Intelligent Systems. 2004: 111-116.

(上接第 2454 页)

[12] ANTTILA P, PAATERO P, TAPPER U, et al. Source identification of bulk wet deposition in Finland by positive matrix factorization [J]. Atmospheric Environment, 1995, 29(14): 1705-1718.

[13] KONDOR R I, LAFFERTY J. Diffusion kernels on graphs and other discrete structures [C]// Proc of the 19th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 2002.

[14] LUSSEAU D, SCHNEIDER K, BOISSEAU O J, et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique

trait? [J]. Behavioral Ecology and Sociobiology, 2003, 54(4): 396-405.

[15] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821-7826.

[16] ROSVALL M, BERGSTROM C T. An information-theoretic framework for resolving community structure in complex networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104(18): 7327-7331.