

## 基于 PSO 算法的医疗大数据任务调度策略

胡超<sup>1</sup>, 彭军<sup>2</sup>, 于文涛<sup>2</sup>

(1. 中南大学 公共卫生学院 信息与网络中心, 湖南 长沙 410083; 2. 中南大学 信息科学与工程学院, 湖南 长沙 410075)

**摘要:** 在医疗信息共享平台下, 选取一种合适的任务调度策略完成医疗数据查询任务在各医院内的调度分配, 是医疗大数据处理所需解决的重要问题。为了保证任务调度时间最短和成本最低, 提出一种改进的粒子群算法。该算法构造了时间最优和成本最优双适应度函数, 自适应地调整粒子速度更新的惯性权重, 加快搜寻最优粒子的速度, 并求解出最合理的数据查询任务调度方案, 最大限度地提高医疗信息共享平台中医疗数据查询的效率。实验结果验证了所提出算法的有效性。

**关键词:** 医疗大数据; 任务调度; 粒子群

中图分类号: TP391

文献标识码: A

文章编号: 1000-436X(2014)Z1-0065-07

## PSO based task scheduling for medical big data

HU Chao<sup>1</sup>, PENG Jun<sup>2</sup>, YU Wen-tao<sup>2</sup>

(1. School of Public Health, Information and Network Center, Central South University, Changsha 410083, China;

2. School of Information Science and Engineering, Central South University, Changsha 410075, China;

**Abstract:** How to select a suitable task scheduling strategy to accomplish the task of medical data query in scheduling and allocation inside each hospital is an important problem demanded to be dealt with in medical big data processing. In order to content the optimal medical data corresponding time and optimal cost considered in task scheduling, a improved particle swarm algorithm was proposed. The algorithm constructs the dual fitness function of optimal time and optimal cost to adjusted the inertia weight of the update of particle velocity adaptively, fasten the speed of optimal particle searching, and find out the most reasonable task scheduling scheme of data query, maximize the efficiency of medical data query in medical information sharing platform. Experiment results demonstrate the effectiveness of the proposed algorithm.

**Key words:** medical big data; task scheduling; particle swarm

### 1 引言

如何在医疗信息共享平台上查询及高效利用医疗大数据是医疗信息化发展所面临的重要问题之一<sup>[1]</sup>。通过医疗信息共享平台实现跨地域、跨时域的医疗信息管理与服务, 可以解决贫困偏远地区群众参与度低、高水平的医务人员匮乏、医疗数据共享困难等问题, 促进全球医学科学的快速发展<sup>[2]</sup>。医疗信息共享平台中各家医院医疗信息系统及数据存在高度的异构性<sup>[3]</sup>, 并且不同系统具有不同的

数据处理能力, 如何依据医疗信息系统的特点, 合理地医疗数据查询任务进行调度, 及时、准确、高效地从各个医院获取患者的诊疗信息和健康档案信息, 是医疗大数据任务调度所面临的一个研究重点与难点。

医疗信息共享平台下, 用户在问诊及开展科研工作往往需要查询和收集来自各家医院的医疗数据<sup>[4]</sup>, 而这样的查询和收集任务是通过任务平台拆分为子任务后, 分配到部署在医院内的前置机最终完成对医疗数据的实时提取。如何依据查询任务

收稿日期: 2014-10-15

基金项目: 国家自然科学基金资助项目(61379111); 湖南省科技计划基金资助项目(2013FJ4066)

**Foundation Items:** the National Natural Science Foundation of China(61379111); Hunan Provincial Science and Technology Program of China(2013FJ4066)

的规模及前置机的数据处理能力和使用成本,对医院内的查询任务进行调度分配,最终决定了用户查询医疗数据的响应时间和成本。因此,在医疗信息共享平台下,选取一种合适的任务调度策略,完成医疗数据查询任务在各医院内的调度分配,是医疗大数据处理所需解决的重要问题。

通常,传统的调度算法可分为表调度算法、聚类调度算法等几类。以 DLS (dynamic level scheduling) 算法、MCP (modified critical-path) 算法为代表的表调度算法<sup>[5]</sup>按照优先级对任务集进行降序排列,并依据优先级依次为任务安排具有最早完成时间的计算资源,具有时间复杂度较低的优点,但此类算法仅将任务完成时间作为性能指标,适用范围有限。以 DSC (dominant sequence clustering) 和 DCP (dynamic critical path) 为代表的聚类调度算法<sup>[6]</sup>的主要思想是将 2 个数据传输量大的计算任务聚合到同一个计算单元执行,通过降低传输时间达到降低任务执行时间的目标,其执行时间取决于每一步执行关键路径任务聚类操作的时间,通常只能处理特定应用领域中的特定需求。

考虑到医疗信息平台中医疗大数据的分布式特性,上述的 2 种传统的调度算法难以很好地解决其中大规模医疗数据查询的调度问题。在传统调度算法的研究基础上,新兴的启发式调度算法能很好地弥补传统调度算法的不足。如 Li<sup>[7]</sup>提出了一种具有双适应度的遗传算法 (double fitness genetic algorithm), 此算法不但能针对大规模调度任务寻找完成时间最优的调度策略,并且能保证任务平均完成时间也较短;其他常用于大数据环境下调度任务的启发式算法还有粒子群<sup>[8]</sup> (particle swarm optimization) 算法,蚁群<sup>[9]</sup> (ant colony optimization) 算法等。该类算法有着实现容易、收敛精度高的优点<sup>[10,11]</sup>,并可以依据实际的任务设定对应的性能指标,可以很好地应用于需要处理大数据的环境。

几类启发式调度算法中,蚁群算法更适用于路径规划的问题,而遗传算法在应对大数据及复杂问题时并不能很好地保证计算时间,且容易出现稳定差的问题。综合比较几类算法,粒子群算法通过合理地选取粒子种群的规模,依据实际的要求设定优化目标及对应的适应度函数,可以快速而高效地完成大规模任务调度的最优求解<sup>[12]</sup>,是一种能够很好地解决医疗信息共享平台中医疗大数据查询任务调度问题的算法。

如何在传统粒子群算法的基础上进一步提升算法的寻优能力及收敛速度,是将该算法很好地运用于医疗信息共享平台中数据查询中的关键。本文在现有的研究基础上,提出一种多适应度的自适应粒子群算法解决医疗信息共享平台中大规模的医疗数据查询问题。在考虑执行时间和执行成本的前提下,利用自适应法动态的调整粒子速度更新的惯性权重,进一步改良粒子群算法的寻优能力,根据医院中实际的前置机部署情况,快速求解出最合理的数据查询任务调度方案,最大限度地提高医疗信息共享平台中医疗数据查询的效率。实验部分通过与传统的粒子群算法进行比较,进一步验证所提出算法的有效性。

## 2 问题描述

在医疗数据查询系统中,每个医院下设有多个前置机,但每个前置机对应的数据处理能力不同,而且每个前置机单位时间任务运行的成本也不同,如何依据医院各前置机的执行能力合理地分配资源进行医疗数据查询是医疗信息共享平台建立的关键问题之一。

在本文中,构建了一个基于大数据的医疗数据查询系统,如图 1 所示。数据查询任务包括以下 3 个步骤。

首先,为了处理来自用户的数据查询请求,医疗信息共享平台通常将收到的请求任务划分为多个子任务,分配到对应的各医院数据服务中心执行。大数据环境中子任务的划分大部分采用 Google 提出的 Map/Reduce 编程模型。然后,子任务通过医院内的前置机调用存储在医院医疗信息服务器中的医疗数据。最后,各前置机将查询到的数据反馈到医疗信息共享平台,并由医疗信息共享平台将最终查询结果汇总后发回给用户。

在由  $N$  个医疗数据资源和  $M$  个医生用户所组成的大数据查询任务中,医疗数据查询问题可以转化为一个标准的任务调度问题,即在考虑数据查询时间和任务执行成本的前提下求解最合理的资源分配方式。通常,通过全局遍历法可以获得最优的任务调度结果,但是其算法复杂度为  $N!$ ,对于大规模数据数学上不可解。考虑到该问题的 NP 难特点,结合启发式算法的低求解复杂度特性,本文采用了多适应度的自适应粒子群算法对该问题进行求解。

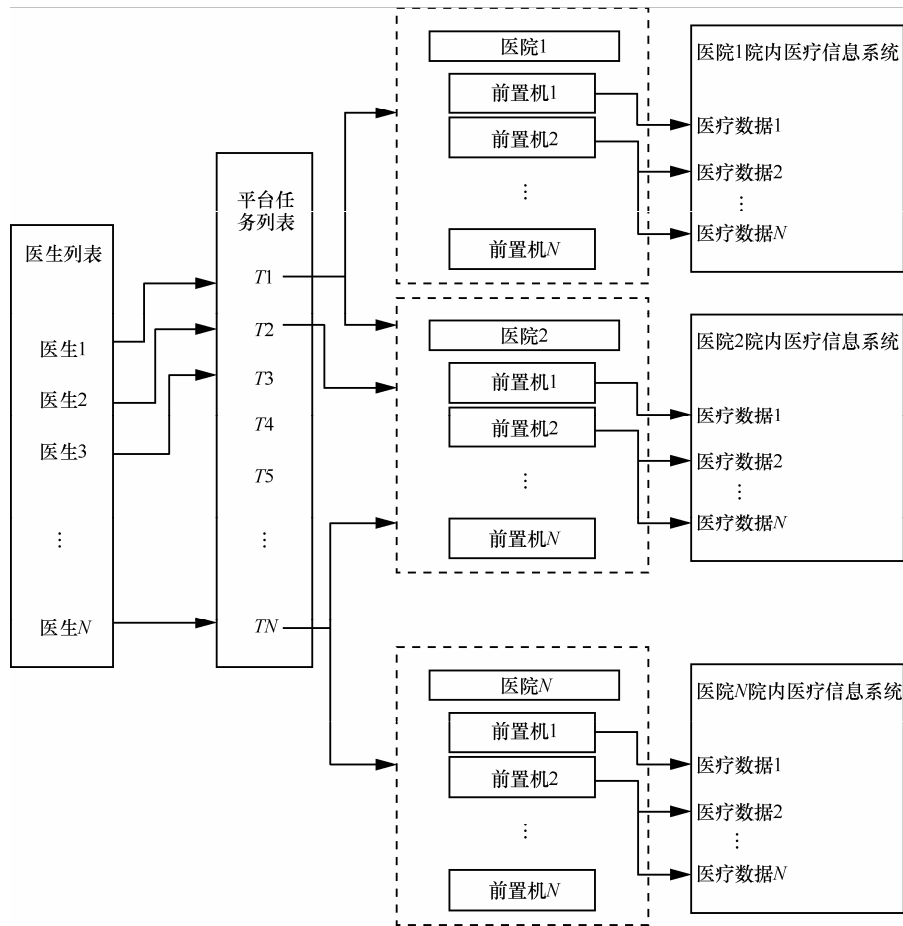


图 1 医疗大数据查询任务结构框架

### 3 基于双适应度函数与自适应权重粒子群算法的任务调度

粒子群 (PSO, particle swarm optimization) 算法是一种具有自适应搜索能力的种群优化算法，大量的实验研究表明群体中个体之间信息的社会共享对整体的进化有很大的帮助，而这也正是粒子群算法的核心思想所在，其在解决任务调度等问题上有着许多优势。粒子群算法通过预先初始化粒子的位置与速度，在设定好的适应度函数的协助下，寻找每一时刻粒子最佳的位置与整个种群的最优位置，并同时更新下一次迭代后的位置与速度，直到达到最大迭代次数后输出最优解。由于医疗服务调度需要考虑总完成时间与成本，本文结合二者建立双适应度函数，并且利用自适应权重的方法改进粒子群算法。

#### 3.1 数据查询任务的建立

每一种针对医院内前置机的医疗数据查询任务调度策略均可视为一个粒子。建立查询任务首先

要针对粒子进行编码，编码方式有多种，既可以运用直接编码的方式，即基于要解决的问题来直接编码粒子的速度与位置；也可以利用间接编码方式。考虑到实际医疗数据查询中查询任务可划分为子任务分配到医院中的各前置机，本文采用任务的间接编码方式，即编码每个子任务所占用的资源，并且子任务的数量决定了编码的长度，由此可知，每一个编码后的粒子实际上都与一个医疗数据查询任务调度策略相对应。

假设医疗信息共享平台在某一时刻有若干个医疗数据查询任务请求，每个医疗数据查询任务依据各个医院划分为多个对应于单个医院的较小的查询子任务。分配到某个医院的查询子任务依据查询数据的内容及类别再次被细分为多个三级查询子任务，由医院内的各前置机分别执行。

设定某一医院内前置机的数量也即资源的数量为  $M$ ，划分后的子任务总数为  $U$ ，并且与前置机数量  $M$  满足关系  $U > M$ 。本文假设某医院内所需完成的数据查询请求子任务总数  $U=9$ 。在这里假设一

个粒子潜在调度策略编码(3,1,3,2,1,2,1,2,3),粒子的编码如表 1 所示。子任务与资源对(x, y)表示把子任务 x 分配到前置机 y 上进行。

对粒子编码之后还需要解码相应的粒子,从而获得前置机上的子任务的分布状况。针对表 1 中的粒子对其进行解码后的结果如表 2 所示,由此可知子任务 {2, 5, 7} 被分配到前置机 1 上来运行,与此同时子任务序列 {4, 6, 8} 被分配到前置机 2 上来处理,子任务序列 {1, 3, 9} 被分用到前置机 3 上执行。

表 1 粒子编码表举例

子任务序号	计算资源序号
1	3
2	1
3	3
4	2
5	1
6	2
7	1
8	2
9	3

表 2 粒子解码表举例

计算资源序号	子任务序号
1	2 5 7
2	4 6 8
3	1 3 9

记录任务在各前置机上运行的时间由 ETC (expect time to complete)矩阵来完成,其中子任务 i 在计算资源 j 上的运行时间用 ETC[i, j]来表示;而采用 RCU(resource cost per unit)数组代表各个前置机上任务运行单位时间所需的成本,其中 RCU[r]代表第 r 个前置机上单位时间内任务运行所需的成本。

依据 ETC 矩阵以及解码后生成的序列,能够将各个前置机执行完所有分配给它的子任务所需的时间,即任务完成总时间 SFT 定义为

$$SFT = \max_{r=1}^M \sum_{i=1}^l M(r, m) \quad (1)$$

M(r, m)表示第 m 个子任务在第 r 个前置机上运行所需要的时间,而分配到此前置机上的子任务的个数为 l。则第 r 个前置机上执行其所有子任务需要的时间为

$$S_{\text{time}}(r) = \sum_{i=1}^l M(r, m) \quad (2)$$

完成所有任务的总成本为

$$S_{\text{cost}}(t) = \sum_{r=1}^M S_{\text{time}}(r) \times RCU(r) \quad (3)$$

### 3.2 粒子群的初始化

假设粒子群的规模大小为 P, 根据前文定义的某医院院内前置机的数目为 M, 划分后的子任务的总数量为 U, 对种群的初始化描述为: 首先由系统随机生成 P 个粒子, 用向量  $x_i$  代表第 i 个粒子的位置, 即  $x_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}\}$ , ( $1 \leq n \leq U, 1 \leq i \leq P$ ) 其中  $x_{ij}$  代表第 j 个子任务被分配到前置机  $x_i$  上执行, 并且  $x_{ij}$  范围在 1 与 M 之间, 用向量  $v_i = \{v_{i1}, v_{i2}, v_{i3}, \dots, v_{in}\}$  表示粒子的速度, 满足 ( $1 \leq n \leq U, 1 \leq i \leq P$ ), 且其速度范围为 -M 与 M 之间。初始化的位置为 [1, M] 之间的随机整数, 初始速度则是  $[-(M-1), (M-1)]$  之间的随机整数。

### 3.3 基于查询任务完成时间和成本的双适应度函数的建立

依据某个适应度函数, 粒子群算法来选择下一代进化, 进而去搜索最佳的解决方案, 在算法的收敛速度以及所求解好坏方面, 它起到了决定性的作用, 因而需要合理地选取适应度函数。在任务调度中, 总任务完成时间往往被用来作为调度目标, 然而在实际的调度过程中, 调度任务完成所需的成本在很大程度上决定了调度策略的优劣。因此综合考虑医院内各前置机的使用成本及数据处理能力, 本文定义了如下的 2 个适应度函数: 医疗数据查询任务完成时间作为第一适应度函数, 而医疗数据查询任务完成成本作为第二适应度函数。其中定义医疗数据查询任务完成时间的适应度函数为

$$f_i(i) = \frac{1}{SFT_i}, 1 \leq i \leq P \quad (4)$$

其中,  $SFT_i$  代表第 i 个粒子的数据查询请求任务完成时间, 它的值由式(1)计算得出。定义医疗数据查询任务完成成本的适应度函数为

$$f_c(i) = \frac{1}{S_{\text{cost}}(i)}, 1 \leq i \leq P \quad (5)$$

考虑二者在各个适应度函数中所占的权重, 得到总适应度函数如下

$$f = q_1 F_t(i) + q_2 F_c(i) \quad (6)$$

其中,  $q_1$  和  $q_2$  分别为各自适应度函数的权重。由式(6)可知, 医疗数据查询任务完成时间越小以及完成成本越小的粒子, 其适应度值越高, 这样的粒子可以被以较大概率优先选择。借助这种选择方式, 能够确保医疗数据查询任务完成时间较少以及完成成本较少的粒子被包含在粒子种群中。

### 3.4 基于惯性权重改进的位置与速度更新下的粒子群算法

粒子群算法中的每一个粒子都有一个由被优化的函数决定的适应值, 每个粒子还有一个速度决定它们的距离和方向, 即通过速度得到下一个时刻粒子的位置。每一次更新后, 粒子们可以得到目前为止自己所发现的最好位置( $pbest$ )和现在的位置, 以及整个群体的最好位置( $gbest$ )。若当前的位置可以使粒子获得更大的适应值, 则当前位置被用来更新为其所经历的最佳位置。

本文中定义  $pbest_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \dots, pbest_{in})$  为第  $i$  个粒子所经历的最佳位置, 定义  $gbest = (gbest_1, gbest_2, gbest_3, \dots, gbest_n)$  为整个种群所经历的最佳位置, 可以求得其位置公式如下

$$pbest_i = \begin{cases} pbest_i(t), & f_i(x_i(t+1)) < f_i(pbest_i(t)) \\ pbest_i(t), & f_i(x_i(t+1)) = f_i(pbest_i(t)) \cap \\ & f_c(x_i(t+1)) \leq f_c(pbest_i(t)) \\ x_i(t+1), & f_i(x_i(t+1)) = f_i(pbest_i(t)) \cap \\ & f_c(x_i(t+1)) > f_c(pbest_i(t)) \\ x_i(t+1), & f_i(x_i(t+1)) > f_i(pbest_i(t)) \end{cases} \quad (7)$$

$$f_i \max(t) = \text{get max} \{f_i(pbest_1(t)), f_i(pbest_2(t)), f_i(pbest_3(t)), \dots, f_i(pbest_p(t))\} \quad (8)$$

其中,  $f_i \max(t) \in \{pbest_1, pbest_2, pbest_3, \dots, pbest_p\}$ , 式(8)表示基于时间的适应度函数最大值。这样, 得到最优的全局粒子为

$$gbest(t) = \begin{cases} f_i \max(t), & \text{length}(f_i \max(t)) = 1 \\ \max \{f_c(f_i \max(t)[1]), f_c(f_i \max(t)[2]), \dots, \\ f_c(f_i \max(t)[n])\}, & \text{length}(f_i \max(t)) > 1 \end{cases} \quad (9)$$

上式表明当有多个粒子具有相同的时间适应度值时, 选择具有最佳成本适应度值的粒子作为全局最优粒子的判断规则。

在找到局部最优值和全局最优值 2 个最优值后, 粒子根据如下的公式来更新自己的速度

$$v_i(t+1) = w \times v_i(t) + c_1 \times r_1 \times (pbest_i(t) - x_i(t)) + c_2 \times r_2 \times (gbest(t) - x_i(t)) \quad (10)$$

其中,  $x_i(t+1) = x_i(t) + v_i(t)$ ,  $x_i(t)$  与  $v_i(t)$  分别为更新后的位置与速度,  $w$  为惯性因子,  $c_1, c_2$  为正的学习因子,  $r_1$  和  $r_2$  为 0 到 1 之间均匀分布的随机数。在一次次进化过程中, 用预先设定的范围来限制粒子的速度与位置, 并且不断地更新  $pbest$  与  $gbest$ , 最终的全局最优解则是最后生成的  $gbest$ 。

本文为了平衡 PSO 算法的全局搜索能力和局部搜索能力, 进一步采用自适应权重法, 得到非线性的动态权重系数公式。其表达式如下

$$w = \begin{cases} w_{\min} - \frac{(w_{\max} - w_{\min}) * (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leq f_{\text{avg}} \\ w_{\max}, & f > f_{\text{avg}} \end{cases}$$

$$f_{\text{avg}} = \frac{\sum_{i=1}^P f_i}{N}, f_{\min} = \min\{f_1, f_2, \dots, f_N\} \quad (11)$$

其中,  $w_{\max}, w_{\min}$  分别表示权重  $w$  的最大值和最小值, 其值中分别取为 0.9 和 0.4。  $f$  表示粒子当前的目标函数,  $f_{\text{avg}}$  和  $f_{\min}$  分别为当前所有微粒的平均目标值和最小目标值。上式中, 惯性权重随着微粒的目标函数值而自动改变, 因此称为自适应权重。当各个微粒的目标趋于一致或者趋于局部最优时, 将使  $w$  减小, 同时对于目标函数值优于平均目标函数的微粒。其对应的  $w$  较小, 从而保护该微粒。反之当目标函数值差于平均目标值的微粒。  $w$  增大, 使微粒向较好搜索区域靠拢。

基于改进适应度函数与自适应权重的粒子群算法具体步骤如下。

**步骤 1** 建立云环境下的信息查找任务, 将查询策略等价于粒子群的各个粒子。

**步骤 2** 对粒子群进行初始化, 包括设置粒子数  $P$ 、初始位置  $x_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}\}$  与速度  $v_i = \{v_{i1}, v_{i2}, v_{i3}, \dots, v_{in}\}$ , 并对各个子任务进行编码, 子任务总数  $U$  与前置机数  $M$  满足  $U > M$ 。

**步骤 3** 计算各个粒子的目标函数, 得到基于总成本与总完成时间的适应度函数  $f_i(t)$  与  $f_c(i)$ , 合成总适应度函数  $f = q_1 F_i(i) + q_2 F_c(i)$ , 寻找当前的个体极值  $pbest_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \dots, pbest_{in})$  和全局极值  $gbest = (gbest_1, gbest_2, gbest_3, \dots, gbest_n)$ 。

**步骤 4** 通过非线性动态权重式(11)更新速度

和位置。

**步骤 5** 检查是否达到最大迭代次数，若达到则输出最优解，若未达到，则返回步骤 3。

**步骤 6** 结束。

基于改进适应度函数与自适应权重的粒子群算法具体流程如图 2 所示。

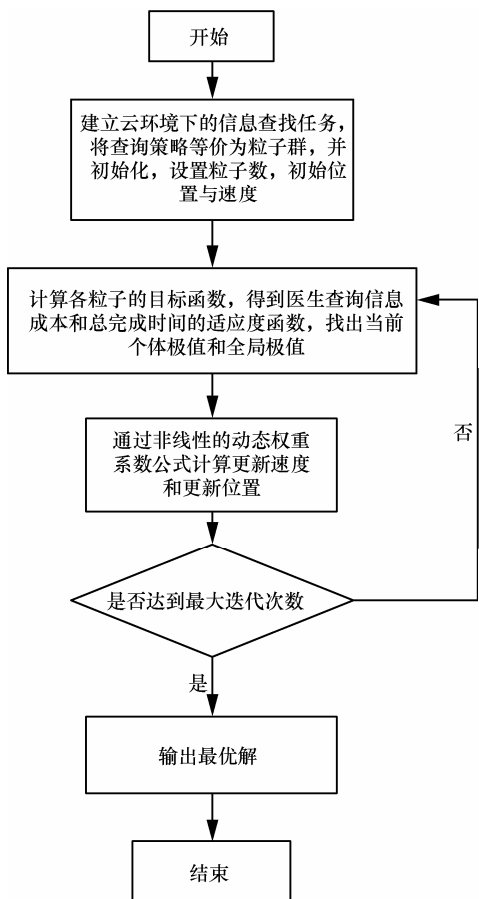


图 2 基于改进适应度函数与自适应权重的粒子群算法流程

## 4 仿真结果与分析

### 4.1 仿真参数设置

本文通过计算出 ETC 矩阵和 RCU 数组，针对医疗数据查询任务调度，分别用传统 PSO 算法和基于自适应权重的 PSO 算法进行了仿真测试，通过多次仿真实验，可以在较短时间内获取最优解。其具体的参数如下。

粒子种群规模  $P$  设为 100，前置机数量  $M$  设置为 50，任务数  $N$  设为 50，未使用自适应权重的惯性因子设置为 0.9， $w_{max}$  设置为 0.9， $w_{min}$  设置为 0.4，学习因子  $c_1$  设置为 2， $c_2$  设置为 2，最大迭代次数  $t_{max}$  设置为 150。

### 4.2 仿真结果与性能分析

本文依照上一节所得到的仿真参数，得到规模为 100，前置机为 50，总任务数为 50 情况下的医生利用前置机查询信息的总完成时间和总完成成本仿真。其仿真结果分别如图 3 和图 4 所示。

如图 3 和图 4 所示，当迭代次数小于 20 时，一般的粒子群算法下医生查询医疗数据所用的总完成时间与完成成本和改进粒子群算法下相差不多，但是随着迭代次数的逐渐增加，如图 3 和图 4 所示，改进粒子群算法相对传统粒子群算法下医生查询信息的总完成时间上快了约 10 s，成本也少了约 50。由此可见，改进粒子群算法的总完成时间与总完成成本相对于一般粒子群算法来讲出现了明显减少。

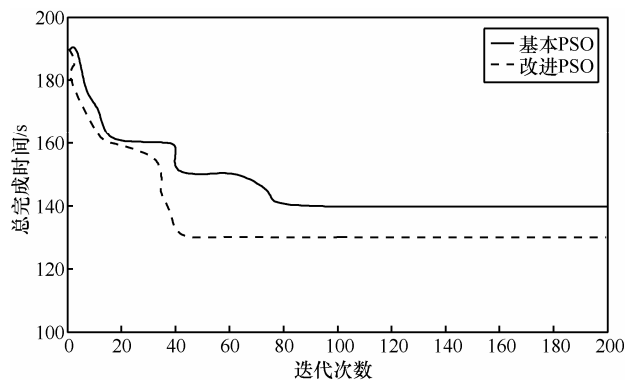


图 3 子任务总数  $U=9$  时任务的总完成时间

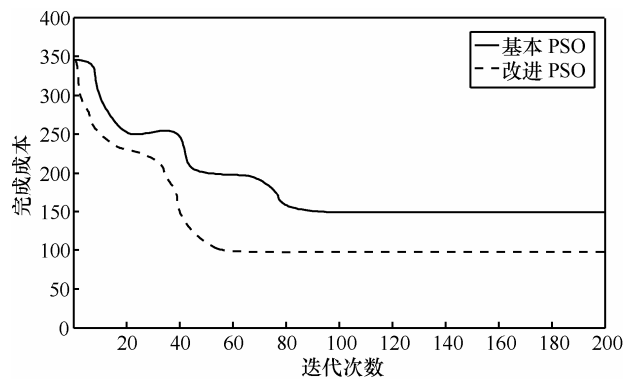


图 4 子任务总数成本  $U=9$  时总任务的总完成成本

以上结果表明，如果只将医生查询信息的总完成时间这一指标作为调度目标，而没有考虑总任务完成成本这一要素，一般粒子群算法在迭代的过程中丢失了一些潜在的优良粒子，这导致在云环境中的迭代过程中使查询的结果过早地收敛于局部最优解中，而改进的粒子群算法将时间与成本作为调度目标在云环境中实现了缩短总任务完成时间的

同时兼顾成本最小的原则。

由此可见, 本文采用的基于自适应权重的粒子群算法明显优于传统的粒子群算法, 在完成任务调度问题是有良好的优化效果, 在医疗系统中, 医生查询病人信息相对较频繁, 采用本文提出的算法, 可以提高系统的收敛速度, 明显地改善了工作效率, 节约时间。

## 5 结束语

本文综合考虑医院院内各前置机的数据处理能力及使用成本, 提出了一种双适应度的自适应权重粒子群算法, 该算法能使医疗信息查询总时间和总成本均能实现最优化。通过实时调整粒子群算法的惯性权重, 进一步提升了粒子群算法的寻优能力及收敛速度, 并保证所求解可以兼顾医疗数据查询时间及查询成本最优。在实验中通过 ETC 矩阵及 RCU 数组分别模拟医院院内各前置机的任务完成时间及使用成本, 分别用传统的粒子群算法及所提出的双适应度自适应权重粒子群算法对大规模医疗数据查询任务进行了最优调度求解, 实验结果表明, 相较于传统的粒子群算法, 所提出的算法不仅使求解最优医疗数据查询任务调度方案的时间进一步缩短, 并提高了粒子群算法的寻优能力, 使所求的最优调度方案具有更短的完成时间及更少的完成成本, 提升了医疗信息共享平台中用户查询医疗数据的效率, 节约了前置机的使用成本。

## 参考文献:

- [1] TRAVIS B, ALLAN S. The inevitable application of big data to health care[J]. The Journal of the American Medicine Association, 2013, 309(13): 1351-1352.
- [2] LANG T. Advancing global health research through digital technology and sharing data[J]. Science, 2011, 331(6018): 714-717.
- [3] HAUX R. Medical informatics: past, present, future[J]. International Journal of Medical Informatics, 2010, 79(9): 599-610.
- [4] 张振, 周毅, 杜守洪等. 医疗大数据及其面临的机遇与挑战[J]. 医学信息学杂志, 2014, 35(6): 2-8.  
ZHANG Z, ZHOU Y, DU S H, *et al.* Medical big data and the fadng opportunities and challenges[J]. Journal of Medical Informatics, 2014, 35(6): 2-8.
- [5] RAMESHKUMAR K, AMALARETHINAM D G. Applying non-traditional optimization techniques to task scheduling in grid computing-an overview[J]. Int J Res Rev Comput, 2010, 4(1): 33-38.
- [6] KAUR N, AULAKH T S, CHEEMA R S. Comparison of workflow scheduling algorithms in cloud computing[J]. Int J Adv Comput, 2011, (10)2: 81-86.
- [7] JIAYIN L I, QIU M, MING Z, *et al.* Online optimization for scheduling preemptible tasks on IaaS cloud systems[J]. Journal of Parallel and Distributed Computing, 2012, 72(2): 666-677.
- [8] JEYARANI R, NAGAVENI N, RAM R V. Self adaptive particle swarm optimization for efficient virtual machine provisioning in cloud[J]. International Journal of Intelligent Information Technologies, 2011, 7(2): 25-44.
- [9] MERKLE D, MIDDENDORF M, SCHMECK H. Ant colony optimization for resource-constrained project scheduling[J]. IEEE Trans Evol Comput, 2002, 6(4): 333-346.
- [10] 刘志雄, 梁华. 粒子群算法中随机数参数的设置与实验分析[J]. 控制理论与应用, 2010, 27(11): 1489-1496.  
LIU Z X, LIANG H. Parameter setting and experimental analysis of the random number in particle swarm optimization algorithm[J]. Control Theory & Applications, 2010, 27(11): 1489-1496.
- [11] 段海滨, 马冠军, 王道波等. 一种求解连续空间优化问题的改进蚁群算法[J]. 系统仿真学报, 2007, 19(5): 974-977.  
DUAN H B, MA G J, WANG D B, *et al.* Improved ant colony algorithm for solving continuous space optimization problems[J]. Journal of System Simulation, 2007, 19(5): 974-977.
- [12] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization[J]. Swarm intelligence, 2007, 1(1): 33-57.

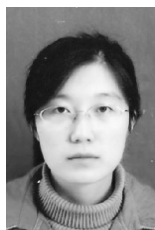
## 作者简介:



**胡超** (1980-), 男, 湖南长沙人, 中南大学网络中心博士生, 主要研究方向为网络管理、教育信息化、区域医疗信息化等。



**彭军** [通信作者] (1967-), 女, 湖南长沙人, 中南大学教授、博士生导师, 主要研究方向为多智能体系统的体系结构与协同模型、网络协作通信系统的优化等。  
E-mail: 8186895@qq.com。



**于文涛** (1981-), 女, 山东烟台人, 中南大学博士生, 主要研究方向为多机器人导航、多智能体系统协作策略设计等。