基于广域分布时空轨迹大数据的群体行为模式挖掘方法

杨杰[1,2] 李小平[1] 陈湉[1]

[1]东南大学计算机科学与工程学院，南京 211189

[2]江苏省公安厅数据中心，南京 210024

(ustcawolf@163.com)

A Mining Method for Group Behavior Pattern Based on Distributed Spatio-temporal Trajectory Big Data in WAN

Yang Jie[1,2], Li Xiaoping[1], Chen Tian[1]

[1](School of Computer Science & Engineering, Southeast University, Nanjing 211189)

[2](Data Center, Public Security Bureau of Jiangsu Province, Nanjing 210024)

Abstract For Mining group behavior pattern on distributed spatio-temporal trajectory big data stored in WAN (Wide Area Network), the DPIA (Distributed & Parallel & Incremental ACO) method based on MapReduce and ACO (Ant Colony Optimization) is proposed for clustering, which is divided into a historical total phase and several continuously executing periodic increment phases. Existing clustering results are constantly corrected by the periodic incremental clustering in each period in WAN, which is implemented by MapReduce. The repeated clustering computation and the copy migration on spatio-temporal trajectory are avoided. The computational efficiency is significantly improved without deteriorating the clustering accuracy. Based on the practical data from the traffic monitoring system of Jiangsu, the method is compared with the existing parallel ACO method. The experimental results show the DPIA method achieves better performance.

Key words Group Behavior Pattern; Spatio-temporal trajectory; Incremental clustering; MapReduce

摘要 针对时空轨迹大数据广域网分布存储条件下的群体行为模式挖掘问题，本文基于MapReduce和ACO (Ant Colony Optimization) 算法提出可在广域网环境分布并行增量执行的DPIA（Distributed & Parallel & Incremental ACO）聚类方法。该方法聚类过程分为历史全量阶段和若干周期增量阶段分段持续执行，通过每个周期的增量数据聚类持续修正已有聚类结果，通过 MapReduce 实现每个阶段聚类运算的广域网分布并行执行，避免时空轨迹大数据的重复聚类运算和拷贝迁移，大大提升运算效率，保持聚类结果准确性。通过江苏道路交通监控系统的实际数据比较该方法与已有基于 MapReduce 的并行 ACO 方法，实验结果表明，DPIA 方法具有更好的聚类特性。

关键词 群体行为模式、时空轨迹、增量聚类、MapReduce

中图法分类号 TP391

收稿日期：2015-11-

基金项目：江苏省科技支撑计划（BE2014733）；公安部应用创新计划（2013YYCXJSST044）；江苏省“333 高层次人才培养工程”科研项目资助计划（BRA2013163）

近年来，基于时空轨迹大数据进行群体行为模式挖掘成为研究热点。2010 年，Science 的一项研究成果通过对匿名手机用户通话记录的研究发现，93%的人类行为是可以预测的[1]；2013 年，Nature 的一项研究成果同样基于匿名手机用户通话记录，发现只需对 4 条包含时间和空间的通话记录进行数据分析，就能以 95%的可能性确定打电话者的身份[2]。基于时空轨迹大数据挖掘群体行为模式并对其进行预测，具有广泛的应用场景和重要的实用价值。美国圣克鲁斯和洛杉矶等地警察局[3]，通过分析实体社会及社交网络数据的时间、空间等信息，能够发现易发生犯罪行为的人群并对其行为进行预测，据此预警预防，两地盗窃犯罪率分别下降 19%和 25%；Anastasios 等[4]根据个人或群体行为的时空特征，用监督学习模型预测个人或群体下一时刻的位置；Tang 等[5]提出一种交叉聚类方法，通过车辆轨迹的时间、空间两个特征定位对象，从而发现同行车辆群体。

传统数据挖掘方法在大数据条件下很难直接适用，由此，近年学者们提出了一批以并行计算、抽样计算、增量计算和内存计算等为代表的挖掘算法。Zhao 等[6]提出基于边结构相似度和 MapReduce 的并行化聚类方法，实现对网络节点的聚类，提高聚类运算的精确度和执行效率；Laptev 等[7]采用有放回的抽样方法，通过确定样本分布得到采样分布点，使得进入 MapReduce 的数据量大大减少；Saeed 等[8]通过降维方法提取用户行为时序数据，在对所有数据进行预聚类的基础上，对更新数据进行增量聚类，分别归入已有类或生成新类；Bӧse 等[9]提出基于 MapReduce 的并行增量数据挖掘方法,极大提升大规模流式数据集的挖掘分析性能；Zhang 等[10]提出基于分布内存的 MapReduce 加速方法，极大提高基于 MapReduce 的数据挖掘算法运算性能。

上述方法的主要特点是基于局域网环境物理集中的数据进行并行运算，或通过抽样降维降低数据规模以提高运算效率，或基于传统规模数据进行增量挖掘运算，或以内存为存储资源进行挖掘运算。时空轨迹大数据不仅总量巨大，一定周期内的增量也具有相当规模，以江苏道路交通监控系统采集的车辆轨迹数据为例，其数据总量已达 500 多亿条 100 多 TB，日均增量达 7000 万条 150 多 GB（文本数据记录，不含照片、视频等）。广域网环境数据分布存储条件下，现有方法存在计算复杂度过高、数据集中时间成本过大、数据抽样降维影响结果准确性、大数据增量挖掘考虑不足、内存计算投资较大（某国际知名内存数据库 64GB 单元配置公开报价 180 万人民币）等问题。由此，面向广域网环境分布存储的时空轨迹大数据，本文基于 MapReduce 和 ACO 算法提出可在广域网环境分布并行增量执行的聚类方法 DPIA（Distributed & Parallel & Incremental ACO）。该方法将聚类过程分为若干阶段持续执行，第一阶段基于已有全量数据进行聚类运算（定义为历史全量阶段）；后续阶段基于每个周期增量数据进行聚类运算，修正已有聚类结果或生成新的聚类（定义为周期增量阶段）。在每个阶段，聚类运算都通过 MapReduce 模型分为 Map 运算、Combine 运算、Reduce 运算三个步骤执行，聚类运算复杂度最高的部分通过 Map 运算在数据源端节点本地并行完成，输出结果由 Combine 运算就地完成合并，合并结果传输到中心节点后由 Reduce 运算自适应地生成聚类中心。基于 MapReduce 已有学者提出若干并行 ACO 算法[11,12]，但主要还是基于局域网集中存储的数据进行运算，对广域网环境数据集中的时间成本、增量数据的可持续挖掘等均未做考虑。本文所提 DPIA 方法是一种类流式广域网分布并行增量聚类方法，将计算任务分配到各个地域分布的源端节点，在每个源端节点进行已有历史全量数据和周期增量数据的聚类运算，避免大数据的重复聚类运算和拷贝迁移，大大缩减运算时间，提升运算效率，节约投资成本。

1 基于时空轨迹大数据的群体行为模式挖掘

时空轨迹大数据多采用广域网分布存储架构，主要特点有：数据来源多、增长速度快、物理集中难。

（1）数据来源多：移动通信、道路交通、即时通讯、社交媒体、网上购物等不同类型应用，都成为时空轨迹大数据的来源，且仍在不断持续扩展。

（2）增长速度快：每一类应用都积累了海量时空轨迹大数据，且都快速增长，周期内的增量数据就构成具有一定规模的“大数据”。

（3）物理集中难：各数据源的数据总量、增量均非常庞大，如江苏道路交通监控系统数据总量近千 TB，各源端节点日均存储增量都在 TB 规模，数据向中心节点物理集中的时间成本过大、存储资源投入过高，很难实现物理集中。

由上，已有数据挖掘方法在时空轨迹大数据广域网分布存储条件下很难直接适用。基于时空轨迹大数据进行群体行为模式挖掘，主要是以时间、空间、业务特征等属性为聚类维度，在实现属性维度相同或相似已有全量轨迹数据聚类的基础上，定期基于增量轨迹数据进行二次聚类，动态更新已有类或生成新类。以车辆轨迹数据为例，一条完整的车辆轨迹数据包括车牌号码、通过时间、地点、车行方向、速度、车辆

颜色等若干数据元，如本文作者车辆，江苏道路交通监控系统采集的部分轨迹数据示例如表 1：

表 1 车辆轨迹数据示例

CPHM	TGSJ	TGDD	XSFX	XSSD	CSYS
S032V0	20140801154702	珠江路丹凤街西北	02	21.00	A
S032V0	20140801140000	进香河珠江路东北	02	28.00	C
S032V0	20140801135651	丹凤街大石桥西南	03	17.00	A
S470A5	20140727161644	江东路集庆门东北	01	72.00	J
S470A5	20140727160241	大明路光华路西南	02	19.00	J
S470A5	20140727142652	光华路苜蓿园东南	01	38.00	J

为挖掘群体行为模式，可以以车辆轨迹数据的数据元为属性维度，将不同源端节点采集的轨迹数据全部以 6 元组（车牌号码，通过时间、地点、车行方向、速度、车辆颜色）方式规范化表示，如表 1 中第一条数据记录的 6 元组表示为（S032V0，20140801154702，珠江路丹凤街西北，02，21.00，A），对每一个属性维度按经验值给予不同权重，通过车辆轨迹数据的聚类实现具有相同或相似特征车辆群体及其行为模式的挖掘。

设广域网环境下有$\mathbb{Z}$个地理位置分布广的源端节点$\{S_1,S_2,\cdots,S_{\mathbb{Z}}\}$，$S_i$中包含车牌号码、通过时间、地点、车行方向、速度、车辆颜色等数据元的数据表有T_i个，表示为$\{L_{i,1},L_{i,2},\cdots,L_{i,T_i}\}$，数据表$L_{i,j}$已有数据记录总量为$Q_{i,j}$条，当前周期内增量为$\Delta Q_{i,j}$条，即$\mathbb{Z}$个源端节点已有数据记录共$\sum_{i=1}^{\mathbb{Z}}\sum_{j=1}^{T_i}Q_{i,j}$条，当前周期内增量共$\sum_{i=1}^{\mathbb{Z}}\sum_{j=1}^{T_i}\Delta Q_{i,j}$条。采用按时序周期分阶段执行聚类运算的方法挖掘群体及其行为模式，历史全量阶段是将具有$\sum_{i=1}^{\mathbb{Z}}\sum_{j=1}^{T_i}Q_{i,j}$条数据记录的待聚类数据集合$A$进行划分，使得具有相同或相似属性维度的数据记录归并为同一类，即将A划分为N个子集$A_1,A_2,\cdots,A_N$，使得$\bigcup_{i=1}^{N}A_i=A$且$A_i\cap A_j=\emptyset,\forall i\neq j$。周期增量阶段是将当前周期内具有$\sum_{i=1}^{\mathbb{Z}}\sum_{j=1}^{T_i}\Delta Q_{i,j}$条数据记录的增量集合$\Delta A$，按照给定的策略，将新增的数据记录分别归并到已有类或生成新类，即将ΔA内的数据记录按策略分别划分到已有子集或生成新子集，使得$\bigcup_{i=1}^{N+\Delta N}(A_i\cup\Delta A_i)=A\cup\Delta A$且$(A_i\cup\Delta A_i)\cap(A_j\cup\Delta A_j)=\emptyset,\forall i\neq j$。若所得$(A_i\cup\Delta A_i)$子集内偏离误差不大于给定阈值$\varepsilon_0$，则该子集当前周期增量聚类结束；若所得$(A_i\cup\Delta A_i)$子集内偏离误差大于给定阈值$\varepsilon_0$，则解体该子集，与其他被解体子集及未被归类（孤立点）数据混合后，采用历史全量阶段方法重新进行聚类运算并生成新类。后续新增量周期按照周期增量阶段方法，基于前序周期聚类结果，对新周期内增量数据再次进行增量聚类运算，以此实现聚类结果的持续更新。当聚类运算结束，将每个类中所有车辆轨迹数据记录按照车牌号进行分组统计，定义类中某一车辆（车牌号）的轨迹数据记录数与该车参与聚类运算的轨迹数据记录总数的比值为聚类准确率，当聚类准确率大于给定阈值η时，则认为该车属于该类，类中不同车辆就构成具有相同或相似轨迹的群体，该群体的轨迹共性特征就是其行为模式。

2 广域分布环境的群体行为模式挖掘方法

时空轨迹大数据总量巨大、增长迅速、广域分布存储等特点，使得移动计算大大优于移动数据。据此，本文提出可在广域网环境分布并行增量执行的 DPIA 聚类方法，该方法按时序周期分多个阶段执行，历史全量阶段是对已有全量数据进行聚类，其基本思想是：①将源端节点$S_i(i=1,2,\cdots,\mathbb{Z})$的已有数据切块为$H_i$个数据分块$B_{i,1},B_{i,2},\cdots,B_{i,H_i}$。②采用 ACO 算法对每个数据分块$B_{i,j}$进行 Map 运算，将$B_{i,j}$的所有数据记录按照给定的策略进行归并聚类。③由 Combine 运算将聚类结果合并为中间结果。④所有中间结果传输到中心节点进行跨数据分块的 Reduce 全局聚类运算。⑤如果全局聚类结果收敛或达到最大迭代次数g_{max}，则算法结束并输出所得类；否则，由 Reduce 运算输出比较参数并分发到每个数据分块，从步骤②开始进行下一次迭代。周期增量阶段是对每个周期的增量数据进行聚类，其基本思想是：①将源端节点$S_i(i=1,2,\cdots,\mathbb{Z})$当前周期增量数据切块为$\Delta H_i$个增量数据分块$\Delta B_{i,1},\Delta B_{i,2},\cdots,\Delta B_{i,\Delta H_i}$（通常$\Delta H_i<H_i$）。②采用给定的空间距离计算方法，通过 Map 运算，并行计算每个增量数据分块$\Delta B_{i,j}$内每条数据记录与已有若干类中心之间的空间距离，对所得空间距离不大于给定阈值$\mathbb{R}$的数据记录，按照距离最小原则将其归并到对应类。③采用给定的偏离误差计算方法，按照源端节点S_i当前所有数据记录的类划分，由 Combine 运算并行计算每个类在源端节点S_i的局部偏离误差。④所有局部偏离误差传输到中心节点并由 Reduce 运算按照对应类进行合并，生成每个类跨源端节点的全局偏离误差。⑤对每一个类，如果全局偏离误差不大于给定阈值ε_0，则该类当前周期的增量聚类结束；如果全局偏离误差大于给定阈值ε_0，则解体该类，类内数据记录按照所在源端节点，与该源端节点内其他被解体类的数据记录、未被归类的数据记录混合，按照历史全

量阶段方法重新进行聚类运算。DPIA 方法基本流程如图 1 所示。

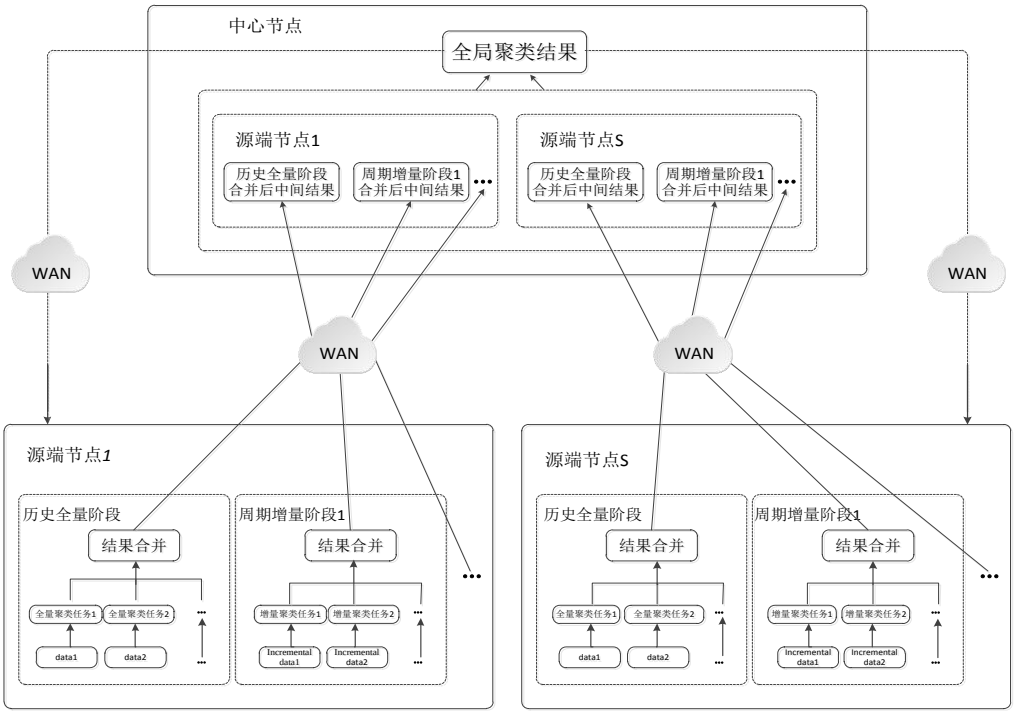

图 1 广域网分布式并行增量聚类流程图

该方法的主要特点是：通过基于已有聚类结果的增量数据聚类，避免已归类历史数据的重复聚类运算，极大缩减增量阶段聚类运算数据规模，大大降低计算复杂度，提高运算效率；通过 MapReduce 模型的 Map 运算、Combine 运算、Reduce 运算，实现聚类运算的广域网分布并行执行，大大减少需要移动的数据量，降低数据移动成本，提高聚类性能。

2.1 DPIA 方法的历史全量阶段聚类运算

历史全量阶段，将每个源端节点当前已有全量数据物理切块为数据分块，一个 Map 运算负责一个数据分块的聚类运算，聚类结果在源端节点本地由 Combine 运算合并为数据量较小的中间结果，不同源端节点中间结果传输到中心节点后由 Reduce 运算合并为最终结果。

2.1.1 历史全量阶段的 Map 运算

对每个数据分块$B_{i,j}$，以时空轨迹数据的数据元为属性维度建立m元组，m为数据元个数，即某一条待聚类数据记录$\vec{R}_k$可表示为$\vec{R}_k=(r_k^1,r_k^2,\cdots,r_k^m)$，其基本思想为：分配$a_{i,j}$只蚂蚁到数据分块$B_{i,j}$，每只蚂蚁初始随机分配一条数据记录$\vec{R}_k$，产生$\vec{R}_k$在半径$\mathbb{R}$（经验值）范围内的邻域记录集合$N(\vec{R}_k,\mathbb{R})$，根据公式（1）计算$\vec{R}_k$与$N(\vec{R}_k,\mathbb{R})$内所有数据记录在$m$个属性维度上的综合相似度。

$$f(\vec{R}_k)=\sum_{\vec{R}_l\in N(\vec{R}_k,\mathbb{R})}[1-\frac{d_{kl}}{\lambda}] \tag{1}$$

λ为相似系数，定义为维度极差（维度最大值与最小值的差）最大值；d_{kl}为待聚类数据记录$\vec{R}_k$与$\vec{R}_l$的空间距离，由于不同数据元属性维度在聚类中作用不同，因此两个待聚类数据记录$\vec{R}_k$与$\vec{R}_l$的空间距离d_{kl}采用加权距离公式（2）表示：

$$d_{kl}=||P(\vec{R}_k-\vec{R}_l)||=\sqrt{\sum_{h=1}^{m}P_h(r_k^h-r_l^h)^2} \tag{2}$$

P_h为属性维度h的权重值，根据经验值和数据准确度确定。依据综合相似度$f(\vec{R}_k)$，可根据公式（3）计算该数据记录$\vec{R}_k$属于$N(\vec{R}_k,\mathbb{R})$类的归并概率。

$$p_{kl}(t)=\frac{\tau_{kl}^{\alpha}(t)f_t^{\beta}(\vec{R}_k)}{\sum_{z\in N(\vec{R}_l,\mathbb{R})}\tau_{zl}^{\alpha}(t)f_t^{\beta}(\vec{R}_z)} \tag{3}$$

α,β为控制参数；$\tau_{kl}(t)$为t时刻$\vec{R}_k$与$\vec{R}_l$路径上的信息素量，由 Combine 运算在迭代中更新。

根据数据记录$\vec{R}_k$的归并概率$p_{kl}(t)$确定$\vec{R}_k$被归并或继续移动：①若$p_{kl}(t)$大于等于给定阈值概率p_0，则该蚂蚁放下$\vec{R}_k$，将其归并到类$N(\vec{R}_k,\mathbb{R})$，保存该蚂蚁携带$\vec{R}_k$已遍历的路径长度和$\vec{R}_k$被放下时的位置坐标，给该蚂蚁重新随机分配另一条数据记录。②若$p_{kl}(t)$小于给定阈值概率p_0，则蚂蚁携带$\vec{R}_k$朝着$p_{kl}(t)$最大的下一路径节点$\vec{R}_l$继续移动；如果达到给定路径

长度或遍历结束也未找到合适的归并类，则直接放下$\vec{R}_k$（该情形定义为放弃），保存该蚂蚁携带$\vec{R}_k$已遍历的路径长度和$\vec{R}_k$被放下时的位置坐标，给该蚂蚁重新随机分配另一条数据记录。③当数据分块$B_{i,j}$内所有待聚类数据记录已由$a_{i,j}$只蚂蚁携带完成遍历时，即所有$|B_{i,j}|$条数据记录都由蚂蚁携带按上述规则被归并或放弃时，返回生成的局部聚类结果。

设 Map 运算的输入键值对为$(<key,\vec{R}_k>,<p,d,s>)$，$key$为$\vec{R}_k$的主键值；$p$为归并概率，$d$为蚂蚁携带数据记录$\vec{R}_k$已遍历路径的长度（$p,d$初始化时均为 0）；$s$为与$d$对应的已遍历路径最新节点的位置坐标，初始化为空；$g$为当前迭代代数；$\tau_{kl}(g)$为完成第$g$次迭代后$\vec{R}_k$到$\vec{R}_l$路径上的信息素值，初始化为 1。$\vec{R}_k$被归并或放弃时，其归并概率为$p_k$，蚂蚁携带$\vec{R}_k$已遍历的路径长度为$d_k$、最新节点位置坐标为$s_k$；$d_g$为完成第$g$次迭代后，所有$\vec{R}_k$被放弃情形下所得$d_k$的最小值，该值为下一次迭代初始输入的比较值。每次迭代运算数据分块$B_{i,j}$的 Map 运算聚类过程描述如下：

算法 1. 历史全量阶段 Map 运算

① $p\leftarrow 0,d\leftarrow 0,s\leftarrow\emptyset,Num\leftarrow 0,\tau_{kl}(0)\leftarrow 1$，给定阈值概率$p_0$；

② While (Num ≤ $|B_{i,j}|$)

③ 根据公式（2）计算$\vec{R}_k$与$N(\vec{R}_k,\mathbb{R})$内所有数据记录的加权距离d_{kl}；

④ 根据公式（1）计算$\vec{R}_k$与$N(\vec{R}_k,\mathbb{R})$内所有数据记录的综合相似度$f(\vec{R}_k)$；

⑤ 读取路径信息素$\tau_{kl}(g-1)$，根据公式（3）计算$\vec{R}_k$属于$N(\vec{R}_k,\mathbb{R})$类的归并概率$p_{kl}(t)$；

⑥ 如果$p_{kl}(t)\geq p_0$，则归并$\vec{R}_k$到$N(\vec{R}_k,\mathbb{R})$类，保存p_k,d_k,s_k，$Num\leftarrow Num+1$，转 Step ⑨；

⑦ 如果$d\geq d_{g-1}$，则放弃$\vec{R}_k$，保存p_k,d_k,s_k，$Num\leftarrow Num+1$，转 Step ⑨；

⑧ 携带$\vec{R}_k$的蚂蚁朝着$p_{kl}(t)$最大的下一路径节点$\vec{R}_l$继续移动，如果蚂蚁携带$\vec{R}_k$访问完所有数据记录，则放弃$\vec{R}_k$，保存p_k,d_k,s_k，转 Step ⑨；否则更新蚂蚁携带$\vec{R}_k$已遍历的路径长度$d\leftarrow d+d_{kl}$，转 Step ②；

⑨ 给蚂蚁随机分配一条尚未归并或放弃的数据记录；

⑩ 输出$B_{i,j}$聚类结果$(<key,\vec{R}_k>,<p_k,d_k,s_k>)$；

⑪ 返回。

2.1.2 历史全量阶段的 Combine 运算

不同数据分块的 Map 运算结果由 Combine 运算在源端节点本地进行合并，可采用多个 Combine 运算并行执行不同数据分块 Map 运算结果的合并。对于所有被归并的数据记录，产生中间结果$(s_k,<\mathbb{N}_{s_k},C_{s_k}>)$；对于所有被放弃的数据记录，产生本次迭代遍历路径长度最小值$d_{i,j}$。Combine 运算同时更新蚂蚁已遍历路径的信息素，在第g代 Map 运算结束后$\vec{R}_k$到$\vec{R}_l$路径上信息素量的更新为

$$\tau_{kl}(g)=(1-\rho)\tau_{kl}(g-1)+\Delta e \qquad (4)$$

$\rho\in(0,1]$为信息素挥发速度，Δe为蚂蚁经过时留下的信息素强度，有蚂蚁经过则为 1，否则为 0。

Combine 运算合并数据分块$B_{i,j}$聚类结果$(<key,\vec{R}_k>,<p_k,d_k,s_k>)$的基本思想为：①对所有归并概率小于$p_0$的数据记录，选取最小$d_k$作为数据分块$B_{i,j}$本次迭代遍历路径长度最小值$d_{i,j}$；②对所有归并概率不小于$p_0$的数据记录，按照被归并时的位置坐标$s_k$进行合并，$s_k$值相同的数据记录归并为同一类，记为类$s_k$。

设$\mathbb{N}_{s_k}$为具有相同s_k值的数据记录数，对于所有属于类s_k的数据记录$\vec{R}_{s_k,k}$，按公式（5）计算类s_k的属性维度值和，并输出为$(s_k,<\mathbb{N}_{s_k},C_{s_k}>)$。

$$C_{s_k}=\sum_{k=1}^{\mathbb{N}_{s_k}}\vec{R}_{s_k,k}=\left(\sum_{k=1}^{\mathbb{N}_{s_k}}r_{s_k,k}^{1},\sum_{k=1}^{\mathbb{N}_{s_k}}r_{s_k,k}^{2},\cdots,\sum_{k=1}^{\mathbb{N}_{s_k}}r_{s_k,k}^{m}\right) \qquad (5)$$

算法 2. 历史全量阶段 Combine 运算

① 如果$p_k<p_0$，选取最小d_k输出为$d_{i,j}$，转 step ③；

② 归并s_k值相同的数据记录，统计$\mathbb{N}_{s_k}$值，按公式（5）计算C_{s_k}，输出$(s_k,<\mathbb{N}_{s_k},C_{s_k}>)$；

③ 更新蚂蚁已遍历路径信息素 $\tau_{kl}(g)$；

④ 返回。

Combine 运算的输出结果仅为两种形式：$(s_k,<\mathbb{N}_{s_k},C_{s_k}>)$和遍历路径长度最小值$d_{i,j}$。经过这一步骤，需经广域网传输的数据量大大降低，若本次迭代数据分块$B_{i,j}$归并为$m_{i,j}$个类，则数据分块$B_{i,j}$需传输到中心节点的数据仅为$m_{i,j}$个中间结果$(s_k,<\mathbb{N}_{s_k},C_{s_k}>)$和 1 个$d_{i,j}$。

2.1.3 历史全量阶段的 Reduce 运算

Reduce 运算在中心节点完成所有数据分块 Combine 运算输出结果$(s_k,<\mathbb{N}_{s_k},C_{s_k}>)$和$d_{i,j}$的二次合并，并生成聚类中心。设第$g$次迭代 Combine 运算的输出结果为$(s_{k,g},<\mathbb{N}_{s_k,g},C_{s_k,g}>)$和$d_{i,j,g}$，Reduce 运算的基本思想是：①根据来自不同数据分块本次迭代的 Combine 运算输出结果，采用公式（2）计算不同数据分块所得聚类中心之间的加权距离，若该距离

小于等于$\mathbb{R}$，则将其合并为同一类$N(s_{k,g},\mathbb{R})$；②根据公式（6）计算第g代全局聚类中心。

$$\bar{C}_{s_{k,g}}=\frac{\sum_{N(s_{k,g},\mathbb{R})}C_{s_{k,g}}}{\sum_{N(s_{k,g},\mathbb{R})}\mathbb{N}_{s_{k,g}}} \tag{6}$$

如果$|\bar{C}_{s_{k,g}}-\bar{C}_{s_{k,g-1}}|\le|\bar{C}_{s_{k,g-1}}-\bar{C}_{s_{k,g-2}}|$，则$\bar{C}_{s_{k,g}}$收敛，结束聚类运算，输出全局聚类中心并分发至各源端节点；如果$\bar{C}_{s_{k,g}}$不收敛，则选取最小$d_{i,j,g}$输出为第g次迭代全局数据聚类遍历路径长度最小值d_g，并分发到各数据分块作为下一次迭代的比较参数。

算法 3. 历史全量阶段 Reduce 运算

① 按公式（2）计算不同数据分块聚类中心$s_{k,g},s_{k',g}$的加权距离$d_{s_{k,g},s_{k',g}}$；

② 若$d_{s_{k,g},s_{k',g}}\le\mathbb{R}$，则$s_{k,g},s_{k',g}$归并为同一类$N(s_{k,g},\mathbb{R})$；

③ 按公式（6）计算当前代g全局聚类中心$\bar{C}_{s_{k,g}}$；

④ 如果$\bar{C}_{s_{k,g}}$收敛，则结束聚类，输出为全局聚类中心，并分发到各源端节点；否则，选取最小$d_{i,j,g}$输出为d_g，并分发到各数据分块；

⑤ 返回。

2.2 DPIA 方法的周期增量阶段聚类运算

在周期增量阶段，同样将每个源端节点当前周期内的增量数据物理切块为增量数据分块，以增量数据分块为单元实现并行聚类运算。其基本思想是：①基于历史全量阶段或上一增量周期的聚类结果，通过 Map 运算，计算增量数据分块内每条数据记录与已有若干聚类中心的空间距离，对所得空间距离符合约束条件的数据记录，按照距离最小原则将其分配到已有类；②在每个源端节点，对当前所有数据（历史全量数据、前序周期增量数据、当期周期增量数据）按照所属类，由 Combine 运算计算每个类在当前源端节点的局部偏离误差；③每个类在不同源端节点的局部偏离误差传输到中心节点，由 Reduce 运算合并为全局偏离误差。偏离误差计算方法如公式（7）：

$$\varepsilon=\frac{\sum_{k=1}^{\mathbb{N}}\sqrt{\sum_{h=1}^{m}(r_k^h-\bar{c}_n^h)^2}}{\mathbb{N}} \tag{7}$$

$\mathbb{N}$为类中数据记录数，$\bar{c}_n^h$为该类类中心第h个属性维度分量值。若某个类的全局偏离误差不大于给定阈值ε_0，则该类增量聚类运算结束；若某个类的全局偏离误差大于给定阈值ε_0，则该类解体，并按照所在源端节点，与其他解体类及未被归类的数据混合后，参照历史全量阶段方法重新进行聚类运算。

2.2.1 周期增量阶段的 Map 运算

设历史全量阶段或前序增量周期聚类运算共输出N个类，即所得聚类中心为$\{\bar{C}_n=\bar{C}_{s_{k,g}}|n=1,2,\cdots,N\}$，周期增量阶段 Map 运算的基本思想为：①对每个增量数据分块$\Delta B_{i,j}$，将块内的数据记录规范化表示为$\Delta\vec{R}_k=(r_k^1,r_k^2,\cdots,r_k^m)$。②根据公式(2)分别计算$\Delta\vec{R}_k$与$N$个已有聚类中心的加权距离，当所得$N$个加权距离都大于给定邻域半径$\mathbb{R}$时，$\Delta\vec{R}_k$作为孤立点抛弃；否则，根据加权距离将$\Delta\vec{R}_k$分配到与其最近的类。

设增量阶段 Map 运算的输入键值对为$(key,\Delta\vec{R}_k)$，key为数据记录$\Delta\vec{R}_k$的主键值，增量数据分块$\Delta B_{i,j}$的 Map 运算聚类过程描述如下：

算法 4. 周期增量阶段 Map 运算

① $Num\leftarrow 0$

② While($Num\le|\Delta B_{i,j}|$)

③ 按公式（2）分别计算$\Delta\vec{R}_k$与N个聚类中心$\bar{C}_1,\bar{C}_2,\cdots,\bar{C}_N$的加权距离$d_{k1},d_{k2}\cdots,d_{kN}$；

④ 如果$\{d_{kn}>\mathbb{R}|n=1,2,\cdots,N\}$成立，则$\Delta\vec{R}_k$被抛弃；否则，根据$d_{k1},d_{k2}\cdots,d_{kN}$的最小值确定$\Delta\vec{R}_k$所属的类$\bar{C}_n$，$Num\leftarrow Num+1$；

⑤ 随机选择另一待聚类数据记录$\Delta\vec{R}_{k'}$，转 step ②；

⑥ 输出增量数据分块$\Delta B_{i,j}$的数据记录聚类结果$(\bar{C}_n,\Delta\vec{R}_k)$；

⑦ 返回。

2.2.2 周期增量阶段的 Combine 运算

在源端节点S_i，按照所属类$\bar{C}_n$合并不同数据分块所有数据记录（包括历史全量阶段数据分块和周期增量阶段数据分块），通过 Combine 运算计算每个类在每个源端节点的局部偏离误差。其基本思想为：设聚类中心$\bar{C}_n$的规范化表示为$\bar{C}_n=(\bar{c}_n^1,\bar{c}_n^2,\cdots,\bar{c}_n^m)$，$\mathbb{N}_{\bar{C}_n,S_i}$为$S_i$中划分到类$\bar{C}_n$的数据记录数，对$S_i$中所有属于类$\bar{C}_n$的数据记录，按公式（8）计算局部偏离误差和，并输出为$\left(\mathbb{N}_{\bar{C}_n,S_i},\varepsilon_{\bar{C}_n,S_i}\right)$。

$$\varepsilon_{\bar{C}_n,S_i}=\sum_{k=1}^{\mathbb{N}_{\bar{C}_n,S_i}}\sqrt{\sum_{h=1}^{m}(r_k^h-\bar{c}_n^h)^2} \tag{8}$$

算法 5. 周期增量阶段 Combine 运算

① 统计源端节点S_i中划分到类$\bar{C}_n$的数据记录数$\mathbb{N}_{\bar{C}_n,S_i}$；

② 按公式（8）计算类$\bar{C}_n$的局部偏离误差和$\varepsilon_{\bar{C}_n,S_i}$；

③ 输出为$\left(\mathbb{N}_{\bar{C}_n,S_i},\varepsilon_{\bar{C}_n,S_i}\right)$；

④ 返回。

经过这一步骤，源端节点S_i需经广域网传输到中心节点的数据仅为N个$\left(\mathbb{N}_{\bar{C}_n,S_i},\varepsilon_{\bar{C}_n,S_i}\right)$数组。

2.2.3 周期增量阶段的 Reduce 运算

Reduce 运算在中心节点运行，负责完成所有源端节点 Combine 运算输出结果$(\mathbb{N}_{\bar{C}_n,S_i},\varepsilon_{\bar{C}_n,S_i})$的合并，生成每个类的全局偏离误差。Reduce 运算的基本思想为：①按公式（9）计算类$\bar{C}_n$的全局偏离误差。

$$\varepsilon_{\bar{C}_n}=\frac{\sum_{i=1}^{\mathbb{Z}}\varepsilon_{\bar{C}_n,S_i}}{\sum_{i=1}^{\mathbb{Z}}\mathbb{N}_{\bar{C}_n,S_i}} \tag{9}$$

②若$\varepsilon_{\bar{C}_n}\le\varepsilon_0$，则类$\bar{C}_n$增量聚类结束；若$\varepsilon_{\bar{C}_n}>\varepsilon_0$，则类$\bar{C}_n$解体，所有解体类的数据记录及孤立点数据记录，按照所在源端节点，重新进行数据切块后，按照历史全量阶段方法重新进行聚类运算。

算法 6. 周期增量阶段 Reduce 运算

① 按公式（9）计算类$\bar{C}_n$的全局偏离误差$\varepsilon_{\bar{C}_n}$；

② 若$\varepsilon_{\bar{C}_n}\le\varepsilon_0$，则类$\bar{C}_n$增量聚类结束；否则，类$\bar{C}_n$解体，每个源端节点重新切块未归类数据，按照历史全量阶段方法重新进行聚类运算；

③ 返回。

3 实验结果分析

实验基于江苏道路交通监控系统采集的车辆轨迹数据，选取常州、南通两个城市子系统为源端节点，南京为中心节点。两个城市距离南京分别为 140 千米、270 千米，采用 200Mbps 专用网络连接，软件环境基于 Hadoop 构建。两个源端节点各采用 2 台服务器，中心节点采用 4 台服务器，配置均为 Intel 5620 2.4GHZ，6 核心，4G 内存，300G 硬盘。实验基于文中所述 6 个数据元建立车辆轨迹数据的属性维度 6 元组（CPHM，TGSJ，TGDD，XSFX，XSSD，CSYS），权重P_h依次为 0.05,0.3,0.3,0.15,0.15,0.05。实验中，历史全量阶段和周期增量阶段的关键参数配置如下：Map 运算为 56 个，Combine 运算和 Reduce 运算各为 4 个，最大迭代次数 5 次，每个 Map 运算内蚂蚁为 3 个，邻域半径$\mathbb{R}$为 0.006，阈值概率p_0为 0.441，相似系数λ根据维度极差值确定。数据分块采用平均分块法，即将投入实验的轨迹数据平均分成 56 个数据分块，每个数据分块对应一个 Map 运算。周期增量阶段关键参数ε_0由历史全量阶段或前序增量周期聚类结果确定：根据当前已有聚类结果，按照公式（7）计算每个类的全局偏离误差，所得值即为下一增量周期每个类对应的偏离误差阈值，由于每一增量周期都要求$\varepsilon_{\bar{C}_n}\le\varepsilon_0$，因此保证了类的收敛性和质量。

实验比较 3 种方法，基于局域网集中存储全量数据的并行聚类（LTPC 方法），基于广域网分布存储全量数据的并行聚类（WTPC 方法），基于广域网分布存储增量数据的并行聚类（DPIA 方法）。实验中，LTPC 方法是将车辆轨迹数据抽取到中心节点后，由中心节点的 4 台服务器并行执行 Map 运算，Map 运算完成后直接执行 Reduce 运算；WTPC 方法是在两个城市的 4 台服务器上，每次均基于节点全量数据，并行执行 Map 运算和 Combine 运算，然后将 Combine 运算结果传输到中心节点，并由中心节点服务器执行 Reduce 运算生成全局聚类结果；DPIA 方法是在 WTPC 方法基础上，将每次基于节点全量数据开展聚类的策略，优化为分阶段的持续增量聚类策略。3 种方法均将 Map 运算、Combine 运算、Reduce 运算平均分配到 4 台服务器。由于 3 种方法都是基于 MapReduce 实现 ACO 算法并行化，因此并行加速能力基本相同，所以重点比较数据量规模增长条件下的结果变化，实验结果如表 2、表 3、表 4：

表 2 LTPC 方法聚类结果

数据量	抽取	Map	Combine	Reduce	总时间	准确率
2×10^7	11.74	21.92	0.00	2.71	36.37	87.92
4×10^7	25.31	47.35	0.00	6.13	78.79	89.04
6×10^7	39.76	83.17	0.00	11.48	134.41	89.86
8×10^7	55.68	112.22	0.00	16.28	184.18	90.43
10×10^7	68.41	145.52	0.00	23.57	237.51	91.56

表 3 WTPC 方法聚类结果

数据量	Map	Combine	Reduce	总时间	准确率
2×10^7	20.36	1.51	1.36	23.23	87.89
4×10^7	45.92	5.44	2.52	53.88	88.72
6×10^7	67.89	7.80	5.10	80.78	89.43
8×10^7	96.57	11.83	7.40	115.81	90.55
10×10^7	129.77	15.21	10.42	155.41	91.61

表 4 DPIA 方法聚类结果

数据量	Map	Combine	Reduce	总时间	准确率
2×10^7	20.36	1.51	1.36	23.23	87.89
4×10^7	8.98	1.66	1.04	34.91	88.54
6×10^7	6.34	1.32	1.00	43.57	89.42
8×10^7	23.16	2.74	1.84	71.31	90.17
10×10^7	27.68	2.78	1.78	103.55	91.25

注：数据量为记录条数，抽取、Map、Combine、Reduce 和总时间的单位为小时，准确率为%。DPIA 方法总时间=前序聚类运算总时间+本周期增量数据的 Map、Combine、Reduce 运算时间。

对比 3 种方法的聚类结果，可以看出，DPIA 方法的 Map 运算时间远低于前两种方法；再对比 Reduce 运算时间和 Combine 运算时间，DPIA 方法同样具有

明显优势。比较 3 种方法的总时间，DPIA 方法由于采用广域网分布式增量机制，其总时间较 WTPC 方法减少 33.4%，较 LTPC 方法减少 56.4%，性能提升非常明显。具体如图 2 所示：

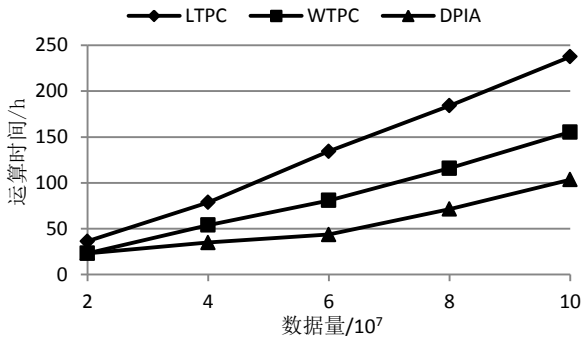

图 2 聚类总时间对比图

再比较 3 种方法的聚类准确率，可以看出，3 种方法聚类准确率在数据量相等情况下大致相同，且都随着数据量的增加逐步提升，说明数据量规模是决定聚类准确率的关键因素，这也从另外一个角度说明通过抽样降维降低数据规模以提升计算效率的方法会牺牲计算准确率，而基于全量数据进行聚类运算是保证准确率的有效方法。具体如图 3 所示：

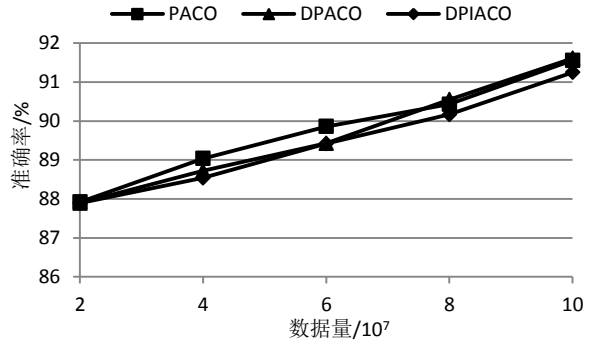

图 3 聚类准确率对比图

再进一步分析 DPIA 方法所得聚类结果，具体如表 5、表 6：

表 5 DPIA 方法聚类结果

数据量	初始类数	结果类数	解体类数
2×10^7	0	6	0
4×10^7	6	9	1
6×10^7	9	11	1
8×10^7	11	18	3
10×10^7	18	23	2

从表 5 可以看出，每一增量周期都有前序周期所得类被解体，且从表 4 和表 5 可以看出，随着数据量和类数量的增加，所得类的聚类准确率同步提升，因此，引入类解体机制保证增量聚类质量是一种可行且有效的方法。

表 6 DPIA 方法所得类特征分析

类别	轨迹数	车辆数	相似度 (%)		
			时间	地点	速度
城市上班族	7150133	53668	70	43	52
跨市移动类	1447609	25814	46	37	39
夜间活动类	483305	36743	65	18	70

表 6 为 DPIA 方法基于 10×10^7 条车辆轨迹数据开展聚类运算所得 23 个类中规模最大的 3 个类，分别为城市上班族类、跨市移动类、夜间活动类，分析其行为模式：

城市上班族类，类中多为城市内本地车辆，轨迹时间集中在白天上下班时间段，行驶速度集中在 30 到 50 km/h。从表 6 可以看出，该类地点相似度相对较小，据此进一步分析，可以发现当前城市规划领域的一些典型特征，如分区域集中化建设的大学城、商贸区、开发区等，使得本地车辆在相对集中的时间段在某些特定区域集中出现，若按区域对该类作进一步细分，很容易划分出若干时间、地点相似度均较高的子类。这些子类也解释了当前城市交通管理领域潮汐式交通的棘手问题，即车辆在相同时间段内通过相同路段以相同方向集中驶向同一区域。根据该类及其子类的行为模式分析，可以给城市规划及交通管理等提出适当建议。

跨市移动类，类中车辆来自多个城市，且不同城市车辆在数量上均没有明显优势；类中车辆轨迹的时间、地点、速度从单项来看呈无序状态，无明显特征；但更进一步，基于时间、地点构建车辆轨迹的时空序列，则表现出明显的城市间有序移动特征，如往返于不同城市的公共汽车，往返于分支机构所在城市与中心机构所在城市的单位小型汽车等。

夜间活动类，类中车辆多来自外地，轨迹时间集中在晚上，行驶速度集中在 60 到 80 km/h，速度较快；地点分布则无明显特征，城郊高速公路、城市主干道、次干道等不同类型道路均有轨迹分布，相似度较低。据此按轨迹所在道路类型进一步细分出若干子类，其中行为模式为“在城市主干道、次干道频繁出现但轨迹时空序列无明显规律性”的子类，就是本文作者所重点关注的特殊车辆群体。

4 结论与展望

避免时空轨迹大数据的重复聚类运算和拷贝迁移，是广域网分布存储条件下提升大数据挖掘应用效能的有效方法。本文基于 MapReduce 和 ACO 算法提出可在广域网环境分布并行增量执行的 DPIA 聚类方法，并应用于基于车辆轨迹大数据的群体行为模式挖掘问题。该方法聚类过程分为历史全量阶段和若干周期增量阶段分段持续执行，通过每个周期的增量聚类运算持续修正已有聚类结果，通过 MapReduce 实现每个阶段聚类运算的广域网分布并行执行，避免广域网环境时空轨迹大数据的重复聚类运算和拷贝迁移，大大提升运算效率，保持聚类结果准确性。目前，该方法已基于生产系统进行测试应用，结果表明，所挖掘出的群体行为模式对实际工作具有很好的指导意义。

测试应用中发现，基于时空轨迹大数据开展面向特定领域应用需要的群体行为模式挖掘，会出现可用轨迹数据数量规模不足的问题，从而导致挖掘结果准确性受到影响。面向特定领域应用需要的时空轨迹数据由于数量较少通常显示为奇异轨迹数据，且单条奇异轨迹数据无法体现行为模式的异常性，必须通过多条奇异轨迹数据的耦合校验与相互印证才有可能实现异常行为模式的检测发现，因此，基于时空轨迹大数据的奇异轨迹数据检测方法、基于奇异轨迹数据耦合校验的异常行为模式发现方法等值得深入研究。

参 考 文 献

[1] Song C, Qu Z, Blumm N, BarabSsi AL. Limits of predictability in human mobility. Science, 2010, 327(5968):1018-1021

[2] Montjoye YA de, Hidalgo CA, Verleysen M, Blondel VD. Unique in the crowd:The privacy bounds of human mobility. Scientific reports, 2013, 3:1376

[3] Predictive Policing Leads to a 19% Reduction in Burglaries in Santa Cruz, CA, http://www.prweb.com/releases/2012/7/prweb9719488.htm

[4] Anastasios N, Salvatore S, Neal L. Mining User Mobility Features for Next Place Prediction in Location-based Services. //Proceedings of the 12th International Conference on Data Mining. Brussels, Belgium , 2012:1038-1043

[5] Tang L A, Zheng Y, Yuan J, et al. On discovery of traveling companions from streaming trajectories. //Proceedings of the 28th International Conference on Data Engineering. Washington, USA, 2012: 186-197

[6] Zhao W Z, Martha VS, Xu X W. PSCAN: A parallel structural clustering algorithm for big networks in MapReduce. //Proceedings of the 27th International Conference on Advanced Information Networking and Applications, Barcelona, Spain, 2013:862—869

[7] Laptev N, Zeng K, Zaniolo C. Very fast estimation for result and accuracy of big data analysis: the EARL system. //Proceedings of the 29th International Conference on Data Engineering, Brisbane, Australia, 2013:1296—1299

[8] Aghabozorgi S, Saybani MR, Wah TY. Incremental clustering of time-series by fuzzy clustering. Journal of Information Science and Engineering, 2012, 28(4): 671-688

[9] Böse JH, Andrzejak A, Högqvist M. Beyond online aggregation: Parallel and incremental data mining with online Map-Reduce. //Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud, New York, USA,ACM Press, 2010, 3

[10] Zhang S B, Han J Z, Liu Z Y, et al. Accelerating MapReduce with Distributed Memory Cache. //Proceedings of the 15th International Conference on Parallel and Distributed Systems, Shenzhen, China, 2009:472-478

[11] Cheng X G, Xiao N F. Parallel Implementation of Dynamic Positive and Negative Feedback ACO with Iterative MapReduce Model. Journal of Information & Computational Science. 2013,10(8): 2359–2370

[12] Yang Y, Ni X H, Wang H J, et al. Parallel implementation of ant-based clustering algorithm based on hadoop. //Proceedings of the 3rd International Conference on Advances in Swarm Intelligence, Shenzhen, China, 2012: 190-197

杨杰，男，1980 年生，博士研究生，研究方向：大数据挖掘分析、并行计算、分布式算法等。

李小平，男，1970 年生，教授，研究方向：云制造、调度理论与算法、服务计算、企业互操作技术。

陈湉，女，1991 年生，硕士研究生，研究方向：复杂调度优化问题。