

基于改进粒子群算法的 P2P 流媒体数据调度策略

陈光喜^{1*}, 李振兴¹, 刘卓军²

(1. 桂林电子科技大学 广西可信软件重点实验室, 广西 桂林 541004; 2. 中国科学院 数学与系统科学研究院, 北京 100190)

(* 通信作者电子邮箱 chgx@guet.edu.cn)

摘要:根据 P2P 流媒体数据调度特点, 改进了粒子群优化算法, 并提出适用于离散粒子群算法(MDPSOA)的数字串编码方式。调度策略引入资源紧急度以及资源稀缺度来选择调度数据片, 然后用改进离散粒子群算法来进行节点寻优, 找出最优调度节点集。最后通过实验仿真算法收敛性、数据调度时间、网络利用带宽和节点负载平衡, 从而验证调度策略的可行性和有效性。

关键词:对等网络; 流媒体; 数据调度; 粒子群算法; 编码

中图分类号: TP393 **文献标志码:** A

Data scheduling strategy in P2P streaming system based on improved particle swarm optimization algorithm

CHEN Guangxi^{1*}, LI Zhenxing¹, LIU Zhuojun²

(1. Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin Guangxi 541004, China;

2. Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Data scheduling strategy in Peer-to-Peer (P2P) media streaming is the key research of the P2P media streaming system. A Particle Swarm Optimization (PSO) algorithm was modified according to P2P streaming data scheduling features and the style of digital encoding string for the algorithm was proposed in this paper. The data scheduling strategy to choose the data chunk took account of resource urgency and scarcity degree. The modified discrete particle swarm algorithm was used to choose the peers to get the optimal scheduling peers set. In order to verify the feasibility and effectiveness of the algorithm, experiments were done to simulate the convergence of the algorithm, the scheduling time, the P2P network uplink bandwidth utilization and the load balancing of peers.

Key words: Peer-to-Peer (P2P); media streaming; data scheduling; Particle Swarm Optimization (PSO) algorithm; encoding

0 引言

近年来,随着互联网技术的发展,对等网络(Peer-to-Peer, P2P)流媒体应用越来越普遍,成为研究的热点。P2P 流媒体数据服务,是一项扩展性好,性价比高的流媒体服务。P2P 网络中每个节点可以向邻居节点请求缺失的数据片,同时也可以充当其他节点的资源提供者,它可以避免传统的 C/S 模式服务器负载过大、网络利用率不高、不能满足流媒体实时性播放等缺点。

P2P 流媒体应用大多数据量大,服务时间长,对数据有严格的时序要求。P2P 流媒体数据调度研究是使请求服务的节点找到为他提供服务的最佳邻居节点,即从哪个节点调哪个数据片,使得系统服务能力强,启动时延小,吞吐量大,播放质量好。实际上,P2P 流媒体数据调度是一个数据片请求任务在多邻居节点的分配问题,是个 NP 问题^[1],一般通过启发式算法找到近优解。如何解决好从哪个节点调度哪片数据能够为 P2P 流媒体系统提供高质量的服务,保障网络中视频资源的完整性,减少用户的平均等待时间等,是 P2P 流媒体数据调度的研究内容。

1 相关工作

P2P 流媒体数据调度问题研究,对 P2P 流媒体发展起着至关重要的作用。目前,很多流媒体系统中都采用随机数据调度算法,很多研究者通过各种方法探求最优 P2P 流媒体数据调度方式。在文献[2]中,考虑到系统含有某些数据片的节点很少,有可能导致这些数据片不能在播放期限内到达,因而采用调度算法先调度发送节点少的那些数据片。文献[3]描述了一种根据带宽加权的轮询调度算法,而且讨论了对持续性和启动时延的影响。以上算法都是随机调度算法,没有完全考虑到网络服务质量和节点的负载平衡,且没有对整个网络性能进行评估。

文献[4]中提出基于遗传算法的数据调度算法,首次将生物智能算法引入到数据调度问题中,算法通过仿真验证性能优于随机算法;文献[5]中采用粒子群算法解决数据调度问题,仿真性能优于遗传算法以及爬山法,但算法很容易陷入局部最优。这些算法都是假设资源节点含有媒体所有的数据片,且是顺序选择数据片,没有考虑到资源的时间紧急度和资源稀缺度。

本文借鉴粒子群算法的基本寻优思想,以及数据调度的

收稿日期:2012-10-16;修回日期:2012-12-18。

基金项目:广西可信软件重点实验室开放基金资助项目(kx201213);中国科学院数学机械化重点实验室开放基金资助项目。

作者简介:陈光喜(1971-),男,四川金堂人,教授,博士,主要研究方向:智能软件与算法、信息安全、符号计算;李振兴(1988-),男,湖南常德人,硕士,主要研究方向:智能算法、网络;刘卓军(1958-),男,山东即墨人,教授,博士,主要研究方向:符号计算、计算机数学。

特点,将解决连续性问题的基本粒子群算法改进为适用于流媒体数据调度离散粒子群算法(Modified Discrete Particle Swarm Optimization Algorithm, MDP SOA)^[6-8],并且在选择调度数据片时考虑到数据片的时间紧急度和资源稀缺度。通过改进的离散粒子群算法寻找最优解空间,即确定数据片调度的资源节点集。

2 改进离散粒子群算法的数据调度策略

2.1 数据调度问题描述

数据调度问题^[9-11]是P2P流媒体需要解决的关键问题之一,P2P流媒体系统由一个跟踪服务器和一个媒体服务器(包含有媒体文件所有数据片)以及多个超级节点和若干普通节点组成。每个普通节点都有一个唯一标志ID,每个节点与其邻居节点之间有着频繁的数据交换,节点根据自己缓存映射信息,应用数据调度策略可以知道向邻居节点请求的数据块信息。新节点加入到系统时,需要向跟踪服务器注册,并向跟踪服务器查询所请求内容的信息,跟踪服务器会根据新加入节点的位置,指定合适的超级节点,由超级节点选择合适的在线节点作为其邻居,超级节点管理的普通节点一般为50~100个。节点在传输数据的过程中,定期与邻居节点和超级节点交换邻居视图,从而保证视图信息的不断更新。一旦某个邻居失效,超级节点会选择新的节点作为其邻居节点,并建立连接,保证有足够多的邻居节点进行数据交换。P2P流媒体系统模型如图1所示。

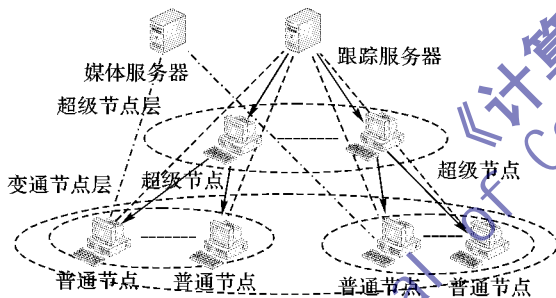


图1 P2P流媒体系统模型

P2P流媒体服务器是将一个完整的流媒体文件划分 N 个数据片,按播放顺序给数据片编号,编号越靠近播放点,播放紧急度越高。系统启动时,需要预加载数据片,当缓冲区缓存数据达到一定预设时,开始播放媒体数据。在P2P网络中,每个节点维护一个缓存映射表BM,用于与邻居节点交换数据。系统维护三个窗口,分别为调度窗口、紧急窗口和预测窗口。其中:调度窗口大小设定为一次调度的数据片集大小 W_r ,紧急窗口 U_r 大小设定为 $W_r/3$,预测窗口大小设定为 $3W_r$ 。调度时在预测窗口内选择数据片,紧急窗口内数据片是必须在本次调度周期内要调度的。在缓存映射表BM中0表示缺失该数据片,1表示缓存中已存在该数据片。

一次完整的数据调度就是根据数据片的资源稀缺度以及紧急度,选择优先级高的数据片,然后根据网络带宽等因素,选择合适节点调度数据集,数据调度模型如图2所示。图2中: R 节点为数据请求节点,需请求的数据集 $\{1,2,3,4\}$, S_1, S_2, S_3 为资源节点。 S_1 缓存有数据片 $\{1,3\}$,带宽为4; S_2 缓存有数据片 $\{1,4\}$,带宽为2; S_3 缓存有数据片 $\{2,4\}$,带宽为1。调度算法是根据节点能力信息,确定一个节点集 $\{S_1, S_3, S_1, S_2\}$,使整个系统调度周期短。

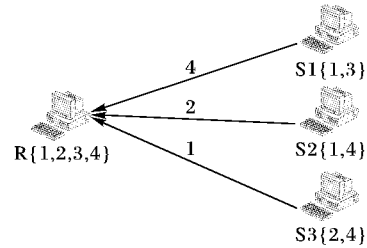


图2 数据调度模型

本文调度策略先根据优先级来确定请求数据片集合,数据片的优先级需要考虑到流媒体播放紧急性以及资源的稀缺性,播放紧急性是资源包相对于本地播放的紧急程度,稀缺性是指在资源在邻居节点中稀缺程度。数据片的紧急性根据调度窗口大小、数据片播放时间及当前播放时间计算,根据资源片序号与当前播放序号比较,序号越接近播放点意味着播放越紧急,紧急度可以表示为式(1):

$$Ur_i = W_r / (IP_i - IP_{play}) \quad (1)$$

其中: W_r 表示调度窗口的大小, IP_i 表示为请求数据包序号, IP_{play} 表示当前播放数据包序号。 Ur_i 越大表示数据片越紧急。

为了保证资源在系统中的扩散率,需要考虑到邻居节点中资源的占有率,可以通过超级节点的位图信息统计资源数量,稀缺度可以表示为式(2):

$$Ra_i = K / (1 + \sum_{k \in \text{nbrview}} BM_k(i)) \quad (2)$$

其中: K 表示超级节点管理的邻居节点个数,一般取值为50~100; $BM_k(i)$ 表示邻居节点 k 是否拥有 i 数据片,1表示拥有,0表示没有; Ra_i 越大表示资源越稀缺。

在根据优先级选择数据片的过程中,由于 Ra_i 和 Ur_i 的值为同一个数量级,相差不大。为了保证媒体播放的连续性以及节点的贡献率,需要在紧急性和资源稀缺性之间达到一个平衡,优先级由紧急度和稀缺度确定。系统根据优先级选定所需调度的数据片,优先级计算可以表示为式(3):

$$Por_i = aUr_i + (1-a)Ra_i \quad (3)$$

其中: a 是一个平衡权值,称为平衡因子,取值范围为 $[0,1]$,用于平衡优先级中数据片的紧急度和稀缺度。在一次数据调度过程中,系统根据优先级高低选择需要调度的数据片,将紧急窗口内缓存中未缓存的数据片加入调度数据集 M ,然后根据 Por_i 大小选择未缓存数据片加入到调度数据集 M 。

在P2P流媒体系统数据调度中,系统根据网络带宽将节点分为高性能节点、普通节点和低性能节点。对于高性能节点,平衡因子取值为0.2,优先调度稀缺性资源,保证系统吞吐量;对于普通节点,平衡因子为0.5,综合紧急性和稀缺性;对于低性能节点,平衡因子为0.8,优先调度紧急性资源,保证播放连续性。

2.2 评估函数

流媒体数据调度过程中,主要考虑因素是数据片的调度时间,因此可以把调度时间作为调度策略评价参数。一次数据调度时间是以最后一块数据传送到请求节点结束。在数据调度算法中,就是求解最优节点集,是调度时间最短,可以先通过式(4)计算每个节点的调度时间集:

$$T_i = Ssize_i / b_i \quad (4)$$

其中: T_i 是第 i 个节点的数据传输时间, $Ssize_i$ 是一次调度中资源节点 i 需要传输数据片的大小, b_i 是资源节点 i 的带宽。

在实际的数据调度过程中,需要考虑到由于网络拥塞原

因产生的端到端之间的时延,以及节点的负载情况,因此,一次数据调度时间为式(5)所示:

$$T_s = \sum_{i=1}^k (T_i + (L_i - 1)T_{wi} + D_i) \quad (5)$$

其中: T_s 为一次调度总时间; T_i 为资源节点传输时间; L_i 为资源节点负载,即等待传输数据片数量; T_{wi} 为节点 i 的处理时延; D_i 为节点 i 的传输时延。

2.3 数据片编码

传统的粒子群算法在解决连续问题上体现很好的优势,而流媒体数据调度问题属于离散型问题求解,采用粒子群算法寻优思想解决流媒体数据调度这一具体问题,一个非常关键的工作就是如何对问题进行直观、简单的编码。在流媒体数据调度过程中,由于每个数据片只处理一次,可以通过数字串来表示数据调度编码。若一次调度集合有 8 个数据片,有 3 个资源节点,调度任务如表 1 所示。

表 1 数据片调度任务

数据片	资源节点	数据片	资源节点
0	1	1	1
3	3	4	1
5	2	6	2
2	3	7	2

由于采用优先级选择数据片,导致数据片可能不连续,用一个数组 $M[i]$ 保存数据片,另外请求节点与超级节点及邻居节点交换信息,获得一个二维数组 $P[i][j]$,表示含有数据片 $M[i]$ 的第 j 个资源节点,用于粒子变换和变异操作,编码如表 2 所示。

表 2 数据编码

索引号	数据片	资源节点	索引号	数据片	资源节点
0	0	1	4	1	1
1	3	3	5	4	1
2	5	2	6	6	2
3	2	3	7	7	2

可以简化表 3 编码形式为只含有节点的子串形式,记为 $X = [1, 3, 2, 3, 1, 1, 2, 2]$ 。这种编码形式简便,直观,便于离散粒子群算法计算,而且不需要复杂的解码过程,能够很好将离散粒子群优化算法的寻优思想应用到 P2P 流媒体数据调度中。

2.4 改进离散粒子群算法

基本的粒子群优化(Particle Swarm Optimization, PSO)算法要求每个个体在进化时维护两个向量,分别为速度向量 $v_i = [v_i^1, v_i^2, \dots, v_i^D]$ 和位置向量 $x_i = [x_i^1, x_i^2, \dots, x_i^D]$,其中 i 表示粒子编号, D 表示求解问题的维数。粒子速度决定了其运动的方向和速率,而位置体现了粒子所代表的解在解空间中的位置,是评估解质量的基础。同时每个粒子还各自维护一个自身的历史最优位置向量($pbest$),以及全局最优向量($gbest$),这个全局最优向量引导粒子向该全局最优区域收敛。基本粒子群优化算法主要是通过速度和位置更新公式(式(6)和(7))不断进化而得到近优解。

$$v_i^{k+1} = wv_i^k + c_1 rand() (pbest_i - x_i^k) + c_2 rand() (gbest - x_i^k) \quad (6)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (7)$$

其中: c_1, c_2 称为粒子学习因子,使粒子具有向群体中优秀个体学习能力; $rand()$ 为 $[0, 1]$ 的随机数,用来保存个体的多样性。

本文数据调度任务采用的编码方式是基于任务节点的数字串,编码长度取决于调度窗口的大小。结合数据调度的特点,本文改进了粒子的位置更新公式,在式(8)中,先将粒子自身的最优位置 $pbest_i$ 作用于当前粒子位置,然后根据全局最优位置 $gbest$ 对粒子位置进行调整。

$$\begin{cases} X_i = S(\chi \cdot M_a^b(X_i), pbest_i) \\ X_i = S(X_i), gbest \end{cases} \quad (8)$$

其中: χ 取值区间为 $[0, 1]$; $M_a^b(X_i)$ 是变异函数,主要是将向量 X_i 中 a 分量的值更新为 b 。为了保证算法执行过程中,粒子种群的多样性以及在迭代后期仍可以具有进化能力,故以概率 χ 变异,式中 $\chi \cdot M_a^b(X_i)$ 含义可以表示为:

$$\chi \cdot M_a^b(X_i) = \begin{cases} M_a^b(X_i), & \rho \leq \chi \\ X_i, & \text{其他} \end{cases} \quad (9)$$

其中 $\rho \in (0, 1)$ 由算法随机产生。

$S(X_i, pbest_i), S(X_i, gbest)$ 函数主要是用于粒子学习 $pbest_i, gbest$ 。通过随机方法,把 m 个资源节点集 SN 随机分成两个子集 SN_1, SN_2 , 若 X_i 的 j 维分量 $X_{i,j} \in SN_2$, 则根据 $pbest_{i,j}$ 更新 $X_{i,j}$ 。公式表示为:

$$S(X_{i,j}, pbest_{i,j}) = \begin{cases} pbest_{i,j}, & X_{i,j} \in SN_2 \\ X_{i,j}, & \text{其他} \end{cases} \quad (10)$$

$$S(X_{i,j}, gbest_j) = \begin{cases} gbest_j, & X_{i,j} \in SN_2 \\ X_{i,j}, & \text{其他} \end{cases} \quad (11)$$

在数据调度问题中,寻求最优的调度节点集,就是使用改进离散粒子群算法求解 X ,使式(5)最小,因此离散粒子群算法个体适应度评价函数可以表示为:

$$F(X_i) = \min T_s \quad (12)$$

2.5 离散粒子群算法数据调度实现

本文描述的调度算法策略执行过程如下:

1) 节点根据自己的数据片缓存信息,把紧急窗口内缺失的数据片作为一次必须调度的数据片,加入调度窗口;然后根据数据片优先级排序,选择高优先级数据片,加入调度窗口,把调度窗口内数据作为一次调度的数据片。

2) 节点通过与邻居节点和超级节点通信,获取整个邻居节点位图信息 BM,统计出含有某数据片的邻居节点信息。

3) 通过离散粒子群算法(MDPSOA)寻求最优节点集合。作为一次最优调度策略,MDPSOA 执行过程如下描述:

步骤 1 初始化粒子群算法参数,如终止条件、变异概率。

步骤 2 随机初始化粒子种群,历史最优向量($pbest$)和全局最优向量。

步骤 3 根据个体适应度函数评价个体,更新 $pbest$ 和 $gbest$,且根据式(8)更新位置。

步骤 4 判断是否达到终止条件,没有则跳到步骤 3 继续执行;否则终止算法。

4) 若某数据片没有邻居节点缓存,则直接从媒体服务器上调度。

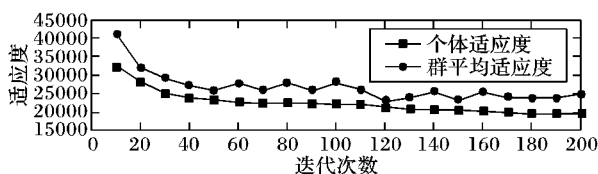
本文实验仿真过程中,初始化粒子种群数为 30,变异概率为 0.6,终止条件为 200 次或者 $gbest$ 连续 10 次不更新。

3 实验仿真

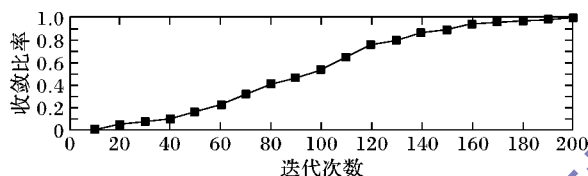
在仿真过程中,使用 Peersim 的 cycle-driven 模式,在 P2P

网络中,资源服务器将一个流媒体文件划分为 N 块,每块大小为 32 KB,仿真随机选取 50 个资源节点,一次调度 50 个数据片,每个数据片以 50% 的概率随机分布到 50 个资源节点上。每个资源节点的上行带宽随机为 30 Kb/s ~ 1 Mb/s,网络时延随机为 30 ~ 130 ms,资源节点的处理等待时间随机设为 50 ~ 100 ms,假设每个节点下行带宽无限制。

验证 MDPSOA 的收敛性,实验随机选取一个调度周期,进行 200 次迭代,得到粒子适应度值变化以及种群适应度值变化趋势图,然后再选取 200 个周期,统计每个周期达到终止条件的迭代次数,计算收敛比率,统计数据如图 3 所示。MDPSOA 的个体适应度值和种群适应度值迭代 120 次达到稳定值,由收敛比率图可以看出 120 代收敛比率为 0.8,由此可以看出 MDPSOA 一般在迭代 120 次就能得到近优解,比基本粒子群算法需要迭代 300 ~ 500 次收敛效果更好,能有效提高算法的执行效率,以及算法的可靠性。



(a) 适应度函数随迭代次数的变化



(b) 收敛比率随迭代次数的变化

图3 MDPSOA 收敛性

图 4 是 MDPSOA 和随机算法的调度时间的趋势图,调度过程中,选取 40 个调度周期,计算每次调度时间,在图中可以看到随机算法(Random Algorithm, RA)调度时间比较长,且各调度周期时间波动比较大,模拟退火算法(Simulated Annealing Algorithm, SAA)调度时间比较稳定,且比随机算法调度时间短很多,而本文 MDPSOA 所用调度时间最短,各个周期调度时间也非常稳定,远优于模拟退火算法和随机算法。MDPSOA 根据节点上行带宽,找到最优节点,从而保证数据片能够在播放时间之前到达,保证资源在网络中的快速扩散,同时也保证了流媒体播放质量。

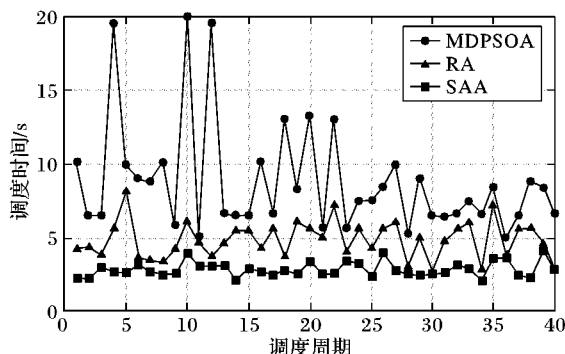


图4 调度时间趋势

为了使 P2P 流媒体系统充分利用网络带宽,增加了对上行利用带宽仿真,为了增加网络扰动,依次向网络中增加资源节点,计算网络的平均上行带宽,如图 5 所示。随着资源节点的不增加,MDPSOA 调度过程中实际利用网络带宽不断提高直到稳定,SAA 调度过程中实际利用网络带宽也不断增高,但实际带宽没有 MDPSOA 高,RA 的实际利用网络带宽没有

多大变化。MDPSOA 能够充分利用网络资源,提高 P2P 流媒体系统的服务能力。

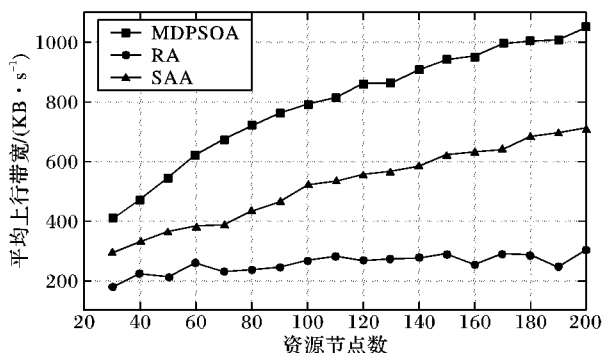


图5 资源节点数与网络带宽利用关系

对于 P2P 流媒体系统,资源节点的负载平衡也是一个很重要的参数,P2P 流媒体网络中应该尽可能多地利用资源节点,仿真数据如图 5 所示。RA 节点平衡率要比 SAA 高,而 SAA 的节点平衡率高于 MDPSOA,主要是 MDPSOA 用组合寻优尽量选择最优的节点调度,忽略了些性能差的节点,但总体来说,资源节点负载平衡性还较好,在后续的研究工作中,将引入一种机制,提高 MDPSOA 的负载平衡性。

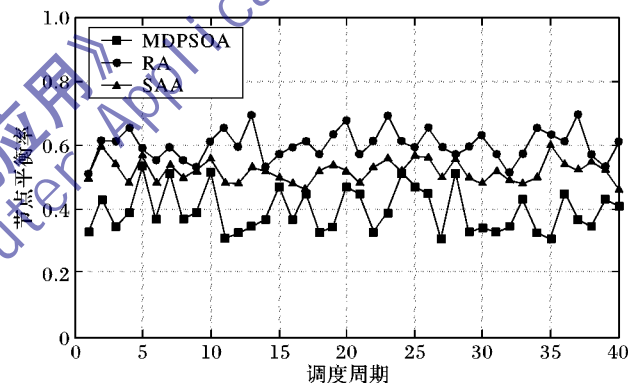


图6 节点负载平衡率

4 结语

本文利用资源播放紧急度和网络中资源稀缺度来选择调度数据片,然后改进离散粒子群算法来进行节点寻优,以此来提高流媒体播放质量,有效利用 P2P 网络中节点资源的带宽。减少用户平均等待时间,在后续的研究工作中,将研究引进一直机制,尽量提高资源节点的平衡性,优化 P2P 流媒体系统数据调度策略。

参考文献:

- [1] BRUCKER P, JURISCH B, SIEVERS B. A branch and bound algorithm for job-shop scheduling problem[J]. Discrete Applied Mathematics, 1994, 49(1/2/3): 105 ~ 127.
- [2] ZHANG X, LIU J, LI B. CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming[C]// Proceedings of IEEE INFOCOM 2005. Washington, DC: IEEE Computer Society, 2005: 2102 ~ 2111.
- [3] AGARWAL V, REJEIE R. A daptive multi-source streaming in heterogeneous peer-to-peer networks[C]// Multimedia Computing and Networking 2005. San Jose: [s. n.], 2005: 13 ~ 25.
- [4] 张立芳. 基于遗传算法的 P2P 流媒体数据调度策略研究[J]. 计算机与数字工程, 2009, 37(4): 31 ~ 33.
- [5] 朱金平. 采用改进粒子群的 P2P 流媒体数据调度算法[J]. 科学技术与工程, 2010, 16(10): 4040 ~ 4042.

(下转第 938 页)

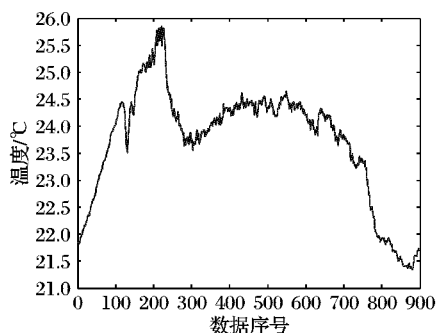


图5 实际网络采集数据

表1 重构时间及误差对比

算法	重构时间/s	均方误差
BP	1.887	4.63E-005
CoSaMP	0.096	2.51E-004
本文算法	0.035	7.11E-005

为在不同数据传输量情况下比较本文算法与其他算法,本部分实验采用前一部分实验类似的方法,变换 M 的取值,使之在 0~400 递增,对于每种不同测量数重复实验 100 次,如果信号重构的 MSE 小于 $2E-04$,则视为重构成功。重构的成功率为成功次数与实验次数之比,实验得到的成功率曲线如图 6 所示。结果表明,对于相同的测量数,本文算法可获得与经典算法类似的重构准确度。

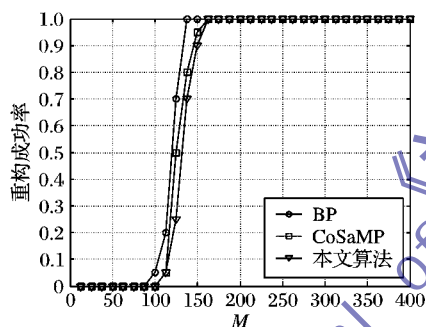


图6 网络数据重构成功率对比

4 结语

利用无线传感器网络节点间的相关性对网络数据进行压缩,减少数据传输能耗是无线传感器网络的关键技术。压缩感知理论为这一问题提供了较好的选择。然而传统网络数据的压缩感知恢复算法复杂度较高,难以满足无线传感器网络数据恢复的实时性要求。本文提出的基于二次规划的无线传感器网络数据恢复算法,将压缩感知问题中的欠定线性方程组求解转换为二次规划问题,然后采用阿米霍步长准则设计求解算法,通过求解为此规划问题对网络数据进行恢复。相

对于传统分布式压缩感知理论而言,本文算法可明显降低网络数据恢复算法的计算复杂度,同时保证数据恢复的准确度,从而大大提高了网络数据恢复的实时性。

参考文献:

- [1] AKYILDIZ I F, SU W, SANKARASUBRAMANIAM Y. A survey on sensor networks [J]. IEEE Communications Magazine, 2002, 40 (8): 102-114.
- [2] 李建中, 李金宝, 石胜飞. 传感器网络及其数据管理的概念、问题与进展[J]. 软件学报, 2003, 14(10): 1717-1727.
- [3] DONOHO D. Compressed sensing [J]. IEEE Transactions on Information Theory, 2006, 52(4): 1289-1306.
- [4] CANDES E, ROMBERG J, TAO T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information [J]. IEEE Transactions on Information Theory, 2006, 52 (4): 489-509.
- [5] 焦李成, 杨淑媛, 刘芳, 等. 压缩感知回顾与展望[J]. 电子学报, 2011, 20 (7): 1651-1662.
- [6] 石光明, 刘丹华, 高大化, 等. 压缩感知理论及其研究进展[J]. 电子学报, 2009, 37(5): 1070-1081.
- [7] CHEN S, DONOHO D, SAUNDERS M. Atomic decomposition by basis pursuit [J]. SIAM Journal on Scientific Computing, 1999, 20 (1): 33-61.
- [8] KIM S, KOH K, LUSTIG M, et al. An interior-point method for large-scale ℓ_1 regularized least squares [J]. IEEE Journal of Selected Topics in Signal Processing, 2007, 1(4): 606-617.
- [9] TROPP J, GILLBERT A. Signal recovery from random measurements via orthogonal matching pursuit [J]. IEEE Transactions on Information Theory, 2007, 53(12): 4655-4666.
- [10] DONOHO D, TSAIG Y, JEAN-LUC S. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit [R]. Stanford, California, USA: Stanford University, Department of Statistics, 2006.
- [11] DAI W, MILENKOVIC O. Subspace pursuit for compressive sensing: Closing the gap between performance and complexity [J]. IEEE Transactions on Information Theory, 2009, 55(5): 2230-2249.
- [12] NEEDELL D, TROPP J. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples [J]. Applied and Computational Harmonic Analysis, 2009, 26(3): 301-321.
- [13] BERTSEKAS D. Nonlinear programming [M]. 2nd ed. Boston: Athena Scientific, 1999.
- [14] ROCKAFELLAR R. Convex analysis [M]. Princeton, NJ, USA: Princeton University Press, 1970.
- [15] BIRGIN E, MARTNEZ J, RAVDAN M. Nonmonotone spectral projected gradient methods on convex sets [J]. SIAM Journal on Optimization, 2000, 10(4): 1196-1211.
- [16] Intel berkeley research lab. Intel lab data[DB/OL]. [2012-10-01]. <http://db.csail.mit.edu/labdata/labdata.html>.

(上接第934页)

- [6] 陈品, 潘全科. 求解独立任务调度的离散粒子群优化算法[J]. 计算机工程, 2008, 34(6): 214-215.
- [7] 刘建华, 杨荣华, 孙水华. 离散二进制粒子群算法分析[J]. 南京大学学报: 自然科学版, 2011, 47(5): 935-514.
- [8] CHEN G M, JIA J Y, HAN Q. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm[J]. Journal of Xi'an Jiaotong University, 2006, 40(1): 1039-1042.
- [9] LI W, ZHENG Q, WANG S. Context-aware adaptive data scheduling algorithm for P2P streaming system [C]// CECNet 2012: Pro-

ceedings of the 2nd International Conference on Consumer Electronics, Communications and Network. Piscataway, NJ: IEEE Press, 2012: 946-952.

- [10] HSU C-H, HEFEEDA M. Quality-aware segment transmission scheduling in peer-to-peer streaming systems[C]// Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems. New York: ACM, 2010: 169-180.
- [11] 郭远威, 许雪梅, 张键洋. P2P 流媒体的数据调度算法[J]. 计算机应用, 2012, 32(4): 935-937.