

DTN 网络中 Ferry 节点的 MSSL 路由算法研究

章 韵^{1,2}, 魏 鹏², 王汝传², 王绍棣²

(1. 苏州大学 计算机科学与技术学院, 江苏 苏州 215006;

2. 南京邮电大学 计算机学院, 江苏 南京 210003)

摘 要:在节点移动规律无法预知、连通性较差的延迟容忍网络中,针对 DTN 网络的 Ferry 节点自定位方案,能够有效提高网络中数据信息采集效率。该方案能够在节点移动过程中根据一个修正因子不断对 Ferry 的移动节点坐标进行调整,最后使坐标收敛到一个可用的范围。通过概率路由算法对 Ferry 节点的移动路线优化,使得 Ferry 节点能够在有限时间内最大限度地收集信息。实验证明,该方案通过合适的修正因子调整节点坐标,使路由算法具有良好的稳定性,在节点信息传输率、公平性和延迟性方面有良好的表现。

关键词:延迟容忍网络; Ferry 路由设计; 移动空间自定位

中图分类号:TP393.04

文献标识码:A

文章编号:1673-629X(2009)05-0107-04

Research on Ferry Node MSSL Routing Algorithms in DTN

ZHANG Yun^{1,2}, WEI Peng², WANG Ru-chuan², WANG Shao-di²

(1. Computer Science and Technology School, Suzhou University, Suzhou 215006, China;

2. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: A scheme of Ferry self-location is introduced to address the location of Ferry in advance in the routing algorithm. In Delay Tolerant Network, it is hard to know in advance how a node move and the poor network connectivity. The algorithm can improve the efficiency of data collection effectively. In this scheme, all coordinates of Ferry are adjusted gradually by adjusting parameter while the node is moving. By TSP algorithm to optimize the mobile path of Ferry node, it can make the maximum collection of information. The simulation shows that the scheme is able to find a proper value of adjusting parameter to make the routing algorithm stable, well performance in transfer rate, fairness and delay among the nodes.

Key words: delay tolerant network; ferry routing; mobil space self local

0 引 言

DTN(Delay Tolerant Network)^[1]是一种新型的网络体系结构,主要是指数据传输延迟较大的网络环境。在一个典型的 DTN 场景中,会存在频繁的连接中断的情况,使得网络不通或者分裂为部分联通在一起的网络分块,即出现网络分割,从而不能保证在数据包转发的过程中节点对之间的端到端的连接路径,如星际间的互联网、某些稀疏的传感器网络。这一类的网络需

要对原有的网络协议和体系结构进行修改才能正常工作。

目前,对于 DTN 的研究还处于不成熟阶段,很多机制有待研究和完善,2003 年 Kevin Fall 提出 DTN 后, Sushant Jain, Kevin Fall 和 Rabin Patra^[2]提出了 DTN 中的单播路由方法,随后的研究人员相继提出了其单播路由策略。目前的路由策略可以分成两大类:洪泛策略(flooding)和转发策略(forwarding)。洪泛策略把报文的多个拷贝传送到邻居节点,并逐步扩散。如基于这种模式的传染病算法,只要是节点相遇就交换信息,那么信息就可以最终实现端到端的传递。最早的 DTN 路由方法都归于这一类,之后的很多算法也大多是在此基础上结合对拓扑结构的了解实现的改进算法。转发策略是通过网络拓扑结构选择最佳路径,在最佳路径经过的节点上转发报文,不需要复制报文,但要求节点之间交换路由信息,会造成控制信息的延迟和能量的消耗。

收稿日期:2009-03-03

基金项目:国家自然科学基金(60573141;60773041);国家高科技 863 项目(2006AA01Z201;2006AA01Z219)

作者简介:章 韵(1963-),男,安徽芜湖人,副教授,博士研究生,研究方向为计算机技术在通信网络中的应用等;王汝传,教授,博士生导师,研究方向为计算机软件、计算机网络和网络、对等计算、信息安全、无线传感器网络、移动代理和虚拟现实技术等;王绍棣,教授,博士生导师,研究方向为计算机网络、信息安全和移动代理技术等。

对于节点比较稀疏的 DTN 网络,为了减少节点的能量消耗,更多的是引入特殊的 Ferry 节点来采集普通节点的信息^[3,4]。Ferry 节点是 DTN 中的一种特殊节点,在网络中搜集和传递信息,具有高速运动、大容量缓存和充足能量等特点,普通节点可以不用考虑信息路由和转发问题从而节约能量。结合 Jeremie Leguay 等人提出的基于“移动空间”^[5]模型的概率路由算法,提出了 Ferry 节点移动空间自定位路由算法 MSSL(Mobile Space Self Local),这种算法是转发策略的延伸。它不用精确地知道网络拓扑结构,而是利用节点在区域空间中出现的概率作为节点坐标,用概率的方法反映了节点在物理空间中的运动规律。Ferry 节点收集普通节点在空间中出现的概率,自己确定遍历路由。只要数据在网络中随机分布存在的时间足够长,数据最终都能被 Ferry 节点采集到,如图 1 所示。

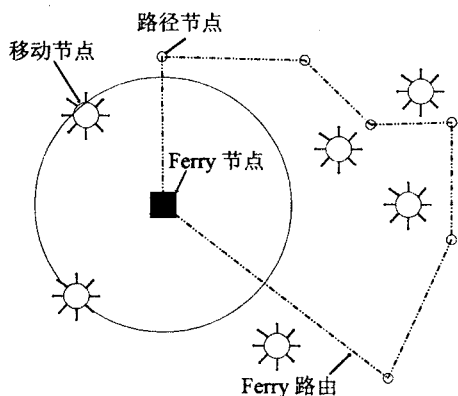


图 1 Ferry 传输信息模式图例

1 Ferry 的 MSSL 路由设计

路由算法设计是 DTN 网络研究中的主要问题。选择节点出现概率最大的空间,只要节点在 Ferry 通信范围内等待足够的时间或者 Ferry 遍历足够多的次数,就一定能接触到节点。当等待时间无限大的时候,接触概率为 1。由于 Ferry 节点不可能等待无限长的时间,因此不能保证节点之间一定能够接触,但是只要 Ferry 等待有限长的时间,就存在一定的概率 $p(0 < p < 1)$ 可以接触到这些节点。可以设想 Ferry 遍历 $1/p$ 次,就一定能接触到节点^[6]。根据这一假设,设计了可以使 Ferry 根据节点在空间出现的概率,自己确定移动路径的路由算法 MSSL。

Ferry 的 MSSL 路由设计由两部分组成。第一步,找出 Ferry 的路径节点, Ferry 在路径节点之间移动,就能够以最大的概率和节点接触;第二步,将这些路径节点按照最短移动距离排序,组成一个最短路径,使得 Ferry 能够在现有条件下以最短时间遍历一次。

假设一个有限的二维空间 S ,所有的节点和 Ferry

均在 S 中移动。把节点活动的物理空间共分成 N 个区域,称为“移动空间”。“移动空间”实际是一个高维的虚拟空间,其坐标不是节点的物理坐标,而是用概率的方法反映了节点在物理空间中的运动规律^[7]。用每个节点 i 在“移动空间”中的第 j 个位置上出现的概率表示其坐标 $p_i(j)(1 \leq i \leq M, 0 \leq j \leq N)$ 。Ferry 中保存一个 $M \times N$ 的矩阵 L ,第 i 行表示当前节点 i 的坐标, M 是节点总数。另外有一个 $\text{Tud}[M]$ 数组,表示矩阵中第 i 行数据(即第 i 个节点的坐标)最后更新的时间。

1.1 选择合适的路径节点

假设 Ferry 的通信范围能够覆盖所划分的小区域,只有在区域的中心的时候才能最大限度的和其它节点通信,所以每个区域的中心点 s 为备选路径节点,

令 $P = \frac{1}{M} \sum_{j=1}^M p_j$,表示所有节点在区域 j 中出现概率的加权平均, p 越大表明在这个区域中节点出现的概率就越大,选出一个概率门限值 P , p 大于 P 的空间中心 s 作为路径节点。

通过节点移动仿真,发现如果节点 i 在一个方形区域中随机移动,那么在接下来的 t 秒中,节点到达某一区域的概率是一个指数分布函数。

$$p = 1 - \exp(-t_w/\gamma_{s,r}) \quad (1)$$

在这里 $\gamma_{s,r}$ 是 Ferry 到达每个路径节点的平均到达时间,由划分区域的半径 r 的大小确定, t_w 为等待时间,用 Ferry 的通信半径作为区域划分的半径,以区域的中心作为候选的路径节点,那么以 p 为横轴, T/p 为纵轴,可以得到其对应关系,如图 2 所示。这是一个指数正弦曲线,在某段等待时间上,接触概率的增加比等待时间增加的要快(p 增加的时候 T/p 减小)。同时,有一段时间接触概率的增加比等待时间的增加要慢,意味着 T/p 对于某些 p 的值是增加的。这表明对于 p

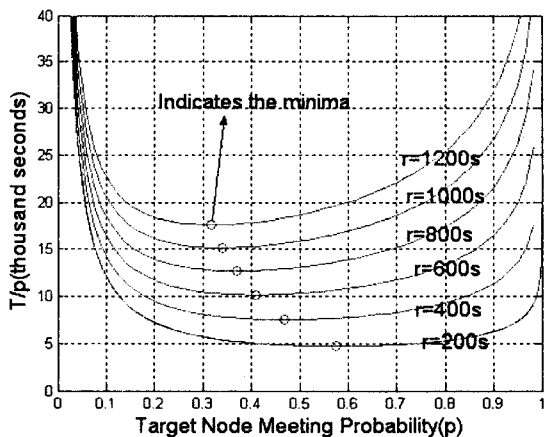


图 2 T/p 时间指数分布图

($n = 15$, γ 是平均到达时间)

来说 T/p 是一个凸型的,从中可以找到一个 p ,使得比例最小。

假设有 E 个路径节点,假设移动中的瞬时接触概率为 0,这种情况在大的空间中更接近真实情况。Ferry 在路径节点的等待时间也相同。在这种情况下, T_w 是 n 倍的等待时间 t_w 。那么 $T = (T_j + nt_w)$, 并且 $p = 1 - \exp(-t_w/\gamma_{s,r})$, 所以

$$\left(\frac{T}{p}\right) = \frac{T_j + nt_w}{p} = \frac{T_j - \gamma \ln(1-p)}{p} \quad (2)$$

通过公式(2)很容易计算出 p 的最小值。图2表明虽然函数图形是凸型的,但是图形的底部是相当平稳的。即使不能够很精确地找到最优的选择来确定概率 p ,也是很容易找到相近的值,只要选择的 p 不是太大或者太小就能做出合理的选择^[6]。

1.2 最短遍历路径节点

把 p_i 大于 P 的区域中心 s 选作路径节点后, Ferry 需要最短的时间遍历所有的路径节点,假设 Ferry 移动速度是一定的,那么只要选择最短遍历路径,遍历时间就是最短的。这个问题实际上是旅行家问题,是一个 NP 难度问题,在节点比较少的情况下,可以在较短的时间算出比较精确的结果^[8],但是如果节点数量巨大,就不能在有限的时间内算出精确的结果,可以选择现有的关于旅行家问题的近似算法来处理,比如最近邻居算法(nearest neighbor algorithm)来解决这个问题,算法如下:

- (1) 任意选择一个路径节点为起始节点,访问之;
- (2) 选择与最近一次访问的节点距离最短的未访问过的节点访问之,重复这一过程直到所有的路径节点都被访问;
- (3) 返回开始的节点。

在上面描述的近似算法中,下一个被访问的节点总是距离最短的未访问过的节点。当 Ferry 和节点相遇后,节点坐标(概率)进行更新, Ferry 遍历完一次以后,会重新计算下一次遍历的路径节点。遍历次数少的情况下,路径节点几乎不变,但是当遍历的次数多了,节点的坐标就会发生比较大的变化,从而使得路径节点发生改变, Ferry 能够更好地适应网络环境的变化。

当 Ferry 和节点 i 相遇时,数据更新。Ferry 的 $Tud[i]$ 与节点的 Tud 比较,如发现 Ferry 节点中矩阵的第 i 行数据已经过时了,使用 i 的坐标更新 Ferry 中对应的节点 i 的坐标,并且更新 Ferry 中对应的 $Tud[i]$,同时 Ferry 采集节点的信息。如果发现 Ferry 节点中矩阵第 i 行数据比节点新,则使用 Ferry 中对应的节点 i 的坐标更新 i 的坐标,并且更新节点 i 的 Tud ,不传递数据。

路径节点的坐标是由节点在各个位置出现的概率决定的。如果节点移动到某一个位置,可以认为已经影响到了节点在各个位置出现的概率,从而必须对 Ferry 中存储的矩阵 L 以及对应的 $Tud[M]$ 进行一次更新,这个称之为位置改变引起的更新,或者称作自定位。例如节点 i 到了每个位置 j ,则需要根据 j 的值使用公式(3),(4)更新一下节点 a 的坐标。

$$p_i(j) = p_i(j)_{old} + (1 - p_i(j)_{old}) \times \theta \quad (3)$$

$$p_i(k) = p_i(k)_{old} \times (1 - \theta) \quad (j \neq k) \quad (4)$$

公式(3)中 $p_i(j)$ 表示行向量 p_i 中的第 j 个分量, $p_i(j)_{old}$ 表示更新前的数据,这个公式增加了第 j 个分量,表示节点 i 在 j 这个位置出现的概率增大。公式(4)减少其他位置相应的坐标分量 $p_i(j) (j \neq k)$, θ 是修正因子, $0 < \theta < 1$, 这个参数决定自定位收敛的快慢及稳定性。经过验证可知, θ 越小收敛效果就越好,取 0.005 或者 0.0005 比较理想^[8]。

2 模拟仿真以及性能分析

假设仿真空间是 $500m \times 500m$, 分成五行五列共 25 个小的区域,稀疏地分布着 15 个随机移动的普通节点,最大移动速度为 $1m/s$ 。仿真时间是 1500s, 10ms 为一个时间单位, Ferry 的移动速度是 $10m/s$ 。通常,更高的 Ferry 速度能够导致更小的接触间隔时间,得到更好的表现。在后面将讨论 Ferry 的速度对于系统的影响。通过仿真,从 Ferry 和节点的接触频率、公平性、信息的丢失率以及路径节点选取的有效性几个方面进行了数据统计,并分别和以下几种路由算法进行了比较。

2.1 其他的路由算法

其他的路由算法如下:

(1) 随机路径节点(RWP): RWP 中所有的节点的参数都一样,每一个节点按照任意的路径节点在 $500m \times 500m$ 区域中移动。经过计算和统计,发现在 RWP 模式下,得到的结果很不理想。

(2) 区域中心路径节点(RCWP): 在同样节点参数条件下,将移动节点移动范围的中心作为路径节点,并且按照 TSP 的算法把路径节点最短化。经过统计, RCWP 模式下要比 RWP 模式要好。

在算法中,只是把 Ferry 当作节点之间通信的中介,仅考虑各种 Ferry 路由算法而不用考虑其他方面的影响。如果允许节点之间直接通信,就会影响 Ferry 和节点之间的接触,从而对信息丢失率、延迟等造成影响。但是在稀疏环境下当节点和 Ferry 相遇的时候,通信范围内存在其他节点的可能性不大。

2.2 性能比较

(1) 节点和 Ferry 接触的频率。

设计路由的最终目的就是尽可能地让 Ferry 和普通节点相遇。在这个标准下,评价 Ferry 和所有节点的接触频率,如图 3 所示。

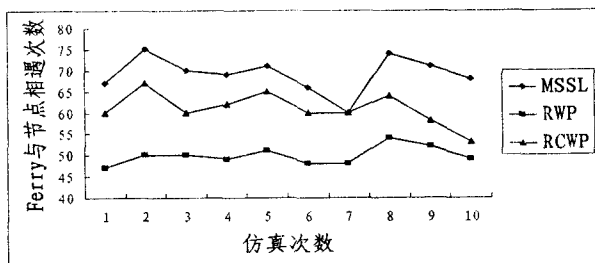


图 3 Ferry 与节点每次遍历相遇的次数

(2) Ferry 与节点之间接触的公平性。

公平性是设计 Ferry 与节点之间接触路由的很重要的一个方面。如果不考虑公平性,可以设计一种路由,使得 Ferry 与小部分节点保持较高的接触概率,而不考虑与其他节点的接触。但是这样会造成其他节点信息长时间不能传递。所以应该在保证公平的前提下,尽力提高 Ferry 与节点的接触概率。如果在单位时间内和节点 i 接触的次数是 ω_i ,公式 $f(\omega_1, \dots, \omega_n)$

$$= \frac{(\sum_{i=1}^n \omega_i)^2}{n \sum_{i=1}^n \omega_i^2} \text{ 表示 Ferry 与节点之间接触的公平性。当}$$

其值大小接近 1 时,说明很公平,反之,其值越小于 1,越不公平。

从表 1 可以知道,在 Ferry 与节点都是完全自由移动的情况下,公平性比较理想,但是单位时间内 Ferry 与节点的接触次数就不太令人满意。而 MSSL 方案,在公平性保证的前提下,能够提高 Ferry 每次遍历接触节点的次数。

表 1 Ferry 与节点接触的公平性

Ferry 移动的方式	普通节点的移动方式	
	自由移动	小区域中移动
RWP	0.94	0.92
RCWP	0.90	0.87
MSSL	0.89	0.86

(3) 信息延迟和丢失。

延迟是指从信息产生开始,到传送到目的节点的时间,由在源节点产生的延迟和 Ferry 传送的延迟组成。通过仿真数据,可以看到 Ferry 与各个普通节点的平均接触次数,如表 2 所示。

通过表 2 的数据可以看出,Ferry 在 RWP 模式下与一个普通节点相遇,平均要 41s,在 RCWP 模式下与一个普通节点相遇平均要 35s,而在 MSSL 模式下与一

个普通节点相遇平均只要 22s。而对于某一次具体的仿真,最坏的情况下,可以得到在 MSSL 模式下和一个节点相遇最长时间需要 240s,在这种情况下,节点信息仍然能够保证不会丢失。

表 2 Ferry 与普通节点平均接触次数

节点	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RWP	21	23	21	24	22	25	24	21	22	25	20	24	23	20	20
RCWP	28	29	30	30	29	31	37	35	35	28	36	25	27	29	33
MSSL	40	43	41	45	51	51	43	45	54	54	48	49	47	47	44

(4) 路径节点效率。

通过仿真结果可见,除了初始化时,Ferry 需要遍历所有的候选路径节点,以后节点的概率坐标变的比较均匀,可以发现,通过几次遍历后,Ferry 选择的路径节点稳定在 4 个或者 5 个,并且在每个路径节点都会与普通节点相遇,不会出现 Ferry 在路径节点走空的情况。

3 结束语

延迟容忍网络 DTN 是一种连通性较差,数据传输延迟较大的网络。根据 DTN 的特点,对于 Ferry 移动路由设计了一个新的框架,使得 Ferry 和普通节点之间不需要事先联机通信,通过 MSSL 路由算法有效地寻找节点,实现数据信息的采集。

MSSL 路由算法由两部分组成:一个是通过计算,挑选路径节点;另一个是 Ferry 在这些路径节点间的最短路径遍历。Ferry 与所有节点有一个接触的最小概率值,通过设计路由算法和合理优化,在同等条件下,能够最大限度的收集信息。模拟仿真证明:采用 MSSL 路由算法,在节点的信息传输率、公平性和路径节点效率等几个方面,得到的结果均优于其它的一些路由算法。

参考文献:

- [1] Fall K. A delay-tolerant network architecture for challenged internets[C]// In Proceedings of ACM SIGCOMM. [s.l.]: [s.n.], 2003: 24-27.
- [2] Jain S, Fall K, Patra R. Routing in a delay tolerant network [C]// In Proceedings of ACM SIGCOMM. [s.l.]: ACM Press, 2004: 145-158.
- [3] Zhao W. A message ferrying approach for data delivery in sparse mobile Ad Hoc networks[C]// Proc. 5th ACM Int'l. Symp. Mobile Ad Hoc Net. and Comp. [s.l.]: ACM Press, 2004: 187-198.
- [4] Zhao W, Ammar M, Zegura E. Multicasting in delay tolerant networks: Semantic models and routing algorithms[C]// In-

(下转第 118 页)

现高效、高速的构件功能检索过程。同时,研究过程中也发现,基于本体的构件检索,后续工作还有很多,例如本体表示汉化能力,以及对复杂数据结构如 Functor 的推理等。

构件功能检索演示系统 Powered by yw		
检索		
检索结果		
构件信息		
	检索项: 构件功能	检索值: register
构件名称	构件版本	主要功能
RegisterInstance42	v2.0	http://www.owl-ontologies.com/ComponentOntology.owl#register
RegisterInstance23	v1.0	http://www.owl-ontologies.com/ComponentOntology.owl#register
RegisterInstance32	v2.0	http://www.owl-ontologies.com/ComponentOntology.owl#register
RegisterInstance31	v2.0	http://www.owl-ontologies.com/ComponentOntology.owl#register
RegisterInstance21	v1.0	http://www.owl-ontologies.com/ComponentOntology.owl#register
RegisterAndLoginInstance4	v3.0	http://www.owl-ontologies.com/ComponentOntology.owl#register http://www.owl-ontologies.com/ComponentOntology.owl#login
RegisterInstance22	v1.0	http://www.owl-ontologies.com/ComponentOntology.owl#register

图 4 具有 register 功能的构件列表

参考文献:

- [1] Gruber T R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing[J]. Information Journal of Human Computer Studies, 1995, 43: 907-928.
- [2] Gibb F, McCartan C, O'Donnell R, et al. The Integration of Information Retrieval Techniques within a Software Reuse Environment[J]. Journal of Information Science, 2000, 26(4): 520-539.
- [3] 徐正权. 软件复用方法与技术[M]. 武汉: 华中理工大学出版社, 1998.
- [4] Musen M A. Dimensions of Knowledge Sharing and Reuse[J]. Computers and Biomedical Research, 1992, 25: 435-467.
- [5] Stuckenschmidt H. Ontology-Based Information Sharing in Weakly Structured Environments [D]. Amsterdam: AI Department, Vrije Universiteit Amsterdam, 2002.
- [6] Doan A, Madhavan J, Domingos P, et al. Learning to map between ontologies on the Semantic Web [C]//In 11th International World Wide Web Conference. Honolulu, USA: [s. n.], 2002.
- [7] Baader F, Sattler U. Description logics with aggregates and concrete domains [J]. Information Systems, 2003, 28(8): 979-1004.
- [8] Zoe L. Web data retrieval and extraction[J]. Data and Knowledge Engineering, 2003, 44(3): 347-367.
- [9] 李选如, 何洁月. 语义集成: 本体映射方法研究[J]. 计算机技术与发展, 2007, 17(2): 127-130.
- [10] 黄烟波, 张红宇, 李建华, 等. 本体映射方法研究[J]. 计算机工程与应用, 2005(18): 33-35.

(上接第 110 页)

- [1] Proceedings of the SIGCOMM Workshop on Delay Tolerant Networking. [s. l.]: [s. n.], 2005.
- [2] Leguay J, Friedman T, Conan V. DTN Routing in a Mobility Pattern Space[C]//In proc. ACM SIGCOMM 05 Workshop on Delay Tolerant Networking and Related Topics (WDTN-05). [s. l.]: [s. n.], 2005.
- [3] Mukarram M, Tariq B, Ammar M H, et al. Message ferry route design for sparse ad hoc networks with mobile nodes[C]//Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing. [s. l.]: [s. n.], 2006: 37-48.
- [4] 王行甫. 一种 DTN 节点自定位方案及其性能分析[J]. 中国科学院研究生院学报, 2008, 25(3): 367-371.
- [5] Cook W, Applegate D, Bixby R, et al. Concorde: A code for solution of Travelling Salesman problem [J/OL]. 2005. http://www.tsp.gatech.edu/.

(上接第 113 页)

- [1] trieval. [s. l.]: ACM Press, 1996: 76-80.
- [2] 贺玲, 吴玲达, 蔡益朝. 数据挖掘中的聚类算法综述[J]. 计算机应用研究, 2007(1): 10-13.
- [3] 谢立宏. 面向高维数据的聚类算法研究[D]. 武汉: 武汉大学, 2002: 10-12, 29-30.
- [4] Makhoul J, Roucos S, Gish H. Vector quantization in speech coding[J]. Proc. of the IEEE, 1985, 7: 1551-1556.
- [5] Karyapis G, Han E H, Kumar V. Chameleon: A hierarchical clustering algorithm using dynamic modeling[J]. IEEE Computer, 1999, 32(8): 68-71.
- [6] 李雄飞, 李军. 数据挖掘与知识发现[M]. 北京: 高等教育出版社, 2003: 11-19.
- [7] 毛国君. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2005: 171-175.
- [8] 朱明. 数据挖掘[M]. 合肥: 中国科学技术大学出版社, 2002: 22-32.
- [9] 李秀芳, 李志成. 基于数据挖掘的聚类算法研究[J]. 计算机技术与自动化, 2006(3): 1-2.
- [10] 段明秀, 杨路明. 对层次聚类算法的改进[J]. 湖南理工大学学报, 2008(2): 1-2.