

基于人工蜂群算法的 DV-Hop 定位改进

李牧东 熊伟 郭龙

(空军工程大学电讯工程学院 西安 710077)

摘要 针对无线传感器网络无需测距依赖的 DV-Hop 定位算法节点定位精度不高的问题,将鲁棒性强、收敛速度快且全局寻优性能优异的人工蜂群算法引入到 DV-Hop 算法的设计中,提出了一种 ABDV-Hop(Artificial Bee Colony DV-Hop)算法。该算法在传统 DV-Hop 算法的基础上,利用节点间的距离和锚节点的位置信息,在 DV-Hop 算法的最后阶段,通过建立目标优化函数,实现对未知节点坐标的估计。仿真结果表明,与传统 DV-Hop 算法相比,在不增加传感器节点的硬件开销的基础上,改进算法能有效降低定位误差。

关键词 无线传感器网络,定位,DV-Hop 算法,人工蜂群算法

中图分类号 TP393 文献标识码 A

Improvement of DV-Hop Localization Based on Artificial Bee Colony Algorithm

LI Mu-dong XIONG Wei GUO Long

(Institute of Telecommunication Engineering, Air Force Engineering University, Xi'an 710077, China)

Abstract With regard to the problem that the typical range-free localization algorithm of DV-Hop for wireless sensor networks has low localization accuracy, the artificial bee colony algorithm with good robust, high convergence speed and outstanding performance on solving global optimization problems was applied to the design of DV-Hop algorithm and an improved algorithm named as ABDV-Hop(Artificial Bee Colony DV-Hop) was proposed. Based on the original DV-Hop algorithm, the improved algorithm uses the information of distance between the nodes as well as the location of beacon nodes. Through establishing the optimization function, the location of unknown nodes is estimated at the final stage of the algorithm. The results show that the proposed method can significantly reduce positioning error compared with the original DV-Hop algorithm without increasing the hardware overhead of sensor nodes.

Keywords Wireless sensor networks(WSNs), Localization, DV-Hop algorithm, Artificial bee colony

1 引言

无线传感器网络作为当前国际上备受关注的研究热点,可广泛地应用于工业生产、环境监控、军事侦察等多个领域^[1]。无线传感器网络节点的定位技术则是实现上述应用的重要基础^[2]。出于对传感器网络成本、功耗以及寿命等因素的考虑,现阶段对无需测距的定位技术研究较多。

DV-Hop 算法是目前研究和应用最为广泛的无需测距定位算法,但该算法的定位精度较低^[3]。针对这一问题,国内外学者已对 DV-Hop 算法进行了部分改进,如文献[4]提出了对平均每条距离进行加权处理后再进行未知节点坐标的计算;文献[5]将差分进化算法引入到 DV-Hop 算法的定位阶段以求得未知节点的位置;文献[6]提出了利用虚拟锚节点的方法来增加锚节点的数量,同时通过信号强度的等级对跳数进行筛选处理;文献[7]将多个锚节点估算的平均跳距信息进行加权处理后代替传统算法的平均跳距。

本文针对 DV-Hop 算法定位精度不高这一缺陷,根据其定位过程,结合人工蜂群算法的基本原理,提出了一种基于人

工蜂群算法的 DV-Hop 改进方案。通过仿真实验,比较了两种算法在不同节点个数、不同网络连通度、不同通信半径及不同邻居锚节点数的条件下的定位误差。结果表明,改进后的算法的定位精度较原算法具有明显的提高。

2 DV-Hop 定位算法

Niculescu 等人利用距离矢量路由的原理提出了 DV-Hop 算法,具体的定位过程可以分为以下 3 个阶段^[3]:

第一阶段 最小跳数的计算。每个锚节点向通信范围内的邻居节点广播自身的位置信息,其中包括跳数(初始化为 0)、锚节点标识以及坐标。接收节点则记录到每个锚节点的最小跳数,同时忽略来自同一个锚节点的较大跳数信息,而后将跳数值加 1 转发给邻居节点。

第二阶段 未知节点与锚节点间距离的计算。每个锚节点根据所记录的其它锚节点的坐标信息和跳数,通过下式估计出平均每跳距离:

$$Hopsize_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (1)$$

到稿日期:2012-03-01 返修日期:2012-06-14

李牧东(1987-),男,硕士生,主要研究方向为无线传感器网络定位技术,E-mail:lmd422@163.com;熊伟(1965-),男,博士,副教授,主要研究方向为无线传感器网络、EDA;郭龙(1987-),男,硕士生,主要研究方向为无线传感器网络覆盖技术。

式中, $(x_i, y_i), (x_j, y_j)$ 是锚节点 i, j 的坐标, h_{ij} 是锚节点 i, j 之间的跳数。锚节点将所计算的平均每跳距离广播至整个网络中, 未知节点仅记录接收到的第一个平均跳距信息, 并将其转发至邻居节点。未知节点接收到平均跳距后, 根据所记录的跳数信息, 通过式(2)计算其到锚节点的距离。

$$d_i = \text{Hopsize}_i \times h_i \quad (2)$$

第三阶段 未知节点坐标的计算。未知节点利用记录的到各个锚节点的距离, 通过三边法或极大似然估计法计算未知节点的坐标, 式(2)中, d_i 还可用下式表示:

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 = d_2^2 \\ \vdots \\ (x-x_m)^2 + (y-y_m)^2 = d_m^2 \end{cases} \quad (3)$$

式中, (x, y) 为未知节点的坐标, $(x_1, y_1) \cdots (x_m, y_m)$ 为该未知节点所记录的锚节点的坐标。令式(3)前面的 $(m-1)$ 个方程依次减去最后一个方程得到:

$$\begin{cases} x_1^2 - x_m^2 - 2(x_1 - x_m)x + y_1^2 - y_m^2 - 2(y_1 - y_m)y = d_1^2 - d_m^2 \\ x_2^2 - x_m^2 - 2(x_2 - x_m)x + y_2^2 - y_m^2 - 2(y_2 - y_m)y = d_2^2 - d_m^2 \\ \vdots \\ x_{m-1}^2 - x_m^2 - 2(x_{m-1} - x_m)x + y_{m-1}^2 - y_m^2 - 2(y_{m-1} - y_m)y = d_{m-1}^2 - d_m^2 \end{cases} \quad (4)$$

上式的线性表达式为 $AX=b$, 其中 $X=(x, y)^T$,

$$A = \begin{bmatrix} 2(x_1 - x_m) & 2(y_1 - y_m) \\ 2(x_2 - x_m) & 2(y_2 - y_m) \\ \vdots & \vdots \\ 2(x_{m-1} - x_m) & 2(y_{m-1} - y_m) \end{bmatrix} \quad (5)$$

$$b = \begin{bmatrix} x_1^2 - x_m^2 + y_1^2 - y_m^2 + d_m^2 - d_1^2 \\ x_2^2 - x_m^2 + y_2^2 - y_m^2 + d_m^2 - d_2^2 \\ \vdots \\ x_{m-1}^2 - x_m^2 + y_{m-1}^2 - y_m^2 + d_m^2 - d_{m-1}^2 \end{bmatrix} \quad (6)$$

则未知节点坐标可用标准的最小均方差估计方法得出:

$$\hat{X} = (A^T A)^{-1} A^T b \quad (7)$$

从上述分析可知, 在计算未知节点坐标时, 估计的 d_i 值由于环境、通信等因素的影响必然存在较大误差, 从而影响了算法整体的定位精度。

3 采用人工蜂群算法的 DV-Hop 改进

3.1 人工蜂群算法

人工蜂群算法 (Artificial Bee Colony, ABC) 是一种基于蜜蜂群体智能的优化算法, 通过模拟蜂群依各自分工不同智能采蜜, 交换蜜源信息而得出最优解^[8]。该算法由于有劳动分工和协作机制, 因此相比于粒子群算法和差分进化算法具有更强的灵活性与适应性^[9]。

在 ABC 算法中, 将蜂群分为 3 个部分: 采蜜蜂、跟随蜂和侦查蜂, 其中采蜜蜂和跟随蜂各占蜂群的一半, 并且每个蜜源在同一时间内只有一个采蜜蜂工作, 也就是说蜜源的数量与采蜜蜂的数量相等, 用 S 表示。放弃所采蜜源的采蜜蜂转变为侦查蜂。蜂群算法的寻优求解过程可以概括为: 采蜜蜂根

据它们所记录的蜜源位置在其邻域内确定另外一个蜜源, 之后将蜜源信息共享给跟随蜂, 跟随蜂通过贪婪机制选择一个蜜源, 而后根据所选择的蜜源在其邻域内搜索另外一个蜜源, 依次循环, 最终寻得最优解。具体步骤如下:

步骤 1 初始化蜂群。随机生成初始种群 S , 均匀分布于寻优空间, 其中采蜜蜂与跟随蜂的数量相等且为 $N_c = N_g = 0.5S$, 采蜜蜂群体 $Z = \{X_1, X_2, \dots, X_{N_c}\}$, 采蜜蜂个体为 $X = \{X_{i,1}, X_{i,2}, \dots, X_{i,M}\}$, 其中 $i=1, 2, \dots, N_c, M$ 为问题解的维数, X_i 的各个分量由式(8)产生:

$$X_{i,j} = X_{i,j,\min} + \text{rand}(0, 1)(X_{i,j,\max} - X_{i,j,\min}) \quad (8)$$

式中, $X_{i,j,\min}, X_{i,j,\max}$ 分别表示 X_i 的第 j 个分量的下限和上限。

步骤 2 收益度 e_i 的计算。收益度通过式(9)求得:

$$e_i = \begin{cases} 1 + f_i, & f_i < 0 \\ \frac{1}{1 + f_i}, & f_i \geq 0 \end{cases} \quad (9)$$

式中, f_i 为目标函数。采蜜蜂通过式(10)产生新的采蜜蜂 $X_{i,1}$ (其中 $i \neq s$):

$$X_{i,1} = X_i + \text{rand}(0, 1)(X_i - X_s) \quad (10)$$

步骤 3 计算新产生采蜜蜂个体的收益度, 根据式(11)分配跟随蜂的数量, 并由式(12)进行蜂群的更新:

$$P_i = e_i / \sum_{i=1}^{N_c} e_i \quad (11)$$

$$X_{i,2} = X_{i1} + \text{rand}(0, 1)(X_{i1} - X_{s1}) \quad (12)$$

步骤 4 依据贪婪机制选择新蜂群。通过计算新产生的蜂群个体的收益度并与之前产生蜂群个体的收益度进行比较, 选择收益度较大的个体组成新的蜂群。

步骤 5 侦查蜂的生成。通过上一步后, 对于收益度没有变化且达到限定次数 $limit$ 的采蜜蜂个体, 按照式(13)转变为侦查蜂 X_w , 其各个分量为:

$$X_{w,j} = 2(X_{i,j,\max} - X_{i,j,\min})(0.5 - \text{rand}(0, 1)) \quad (13)$$

步骤 6 不满足收敛条件时, 返回第 2 步, 直至循环结束。

3.2 改进定位算法的计算过程

人工蜂群算法在寻优求解过程中的关键问题是适应度函数的设定, 蜂群算法的每个个体通过计算适应度值比较收益率来选择个体的全局最优解。文献^[10]提出了一种约束适应度函数的方法来求得适应度值。本文将这一思想应用于人工蜂群算法的适应度函数设置上。

通过对 DV-Hop 算法定位过程的分析可知, 在算法的第三阶段测量锚节点与未知节点间距离时, 由于外界因素及内部测量的误差使得 d_i 误差较大, 则式(3)可变为:

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = r_1^2 + \epsilon_1 \\ (x-x_2)^2 + (y-y_2)^2 = r_2^2 + \epsilon_2 \\ \vdots \\ (x-x_m)^2 + (y-y_m)^2 = r_m^2 + \epsilon_m \end{cases} \quad (14)$$

式中, $\epsilon_1, \epsilon_2, \dots, \epsilon_m$ 为存在的误差范围, r_1, r_2, \dots, r_m 为未知节点与相应锚节点之间的真实距离, 则 $-\epsilon_i \leq r_i - d_i \leq \epsilon_i$ 。则对于未知节点坐标 (x, y) , 应满足下式:

$$\begin{cases} d_1^2 - \epsilon_1^2 \leq (x-x_1)^2 + (y-y_1)^2 \leq d_1^2 + \epsilon_1^2 \\ d_2^2 - \epsilon_2^2 \leq (x-x_2)^2 + (y-y_2)^2 \leq d_2^2 + \epsilon_2^2 \\ \vdots \\ d_m^2 - \epsilon_m^2 \leq (x-x_m)^2 + (y-y_m)^2 \leq d_m^2 + \epsilon_m^2 \end{cases} \quad (15)$$

求解 (x, y) , 使得式(16)取得最小值:

$$fit(x, y) = \sum_{i=1}^m (\sqrt{(x-x_i)^2 + (y-y_i)^2} - d_i) \quad (16)$$

则 $fit(x, y)$ 为适应度函数。由上式可知, 所得出的未知节点坐标受误差的影响将会减为最小, 从而可以提高坐标计算的准确度。

综上所述, 本文提出的基于人工蜂群算法的 ABDV-Hop (Artificial Bee Colony DV-Hop) 定位算法流程如图 1 所示。

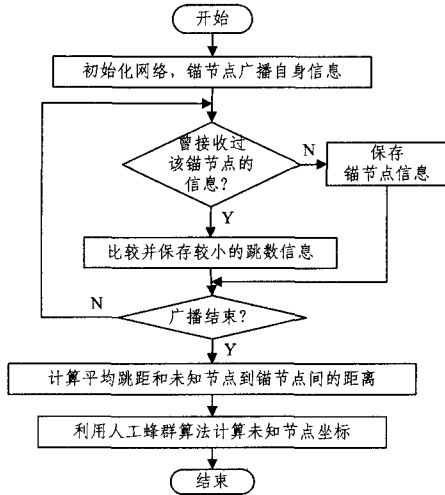


图 1 ABDV-Hop 算法定位流程

4 仿真实验及结果分析

为了验证改进算法的性能, 采用 Matlab 进行仿真。设节点随机分布在边长为 200m 的方形区域内, 其中锚节点坐标已知。ABDV-Hop 算法参数设置如下: 蜂群数 $S=50$, 循环次数为 100, 采蜜蜂和跟随蜂的数量 $N_c = N_g = 25$, 搜索范围 $0 \leq x \leq 200\text{m}, 0 \leq y \leq 200\text{m}$, 收敛条件设为 $|fit(x, y)| \leq 10^{-4}$, $limit$ 为 20 次, 维数 $M=2$ 。本文参考文献[11]提出的算法定位性能的评价指标并加以修正, 提出以基于 k 次仿真结果统计的归一化的平均定位误差作为本文实验的评价指标, 即:

$$error = \frac{\sum_{i=1}^k \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}}{kn_c \cdot R} \quad (17)$$

式中, $k=100$ 为实验次数, R 为节点的通信半径, n_c 为可定位的节点个数, (x_i, y_i) 为未知节点的真实坐标, (x_i', y_i') 为该未知节点的估计坐标。

4.1 不同连通度条件下定位结果的比较

若节点之间可以直接通信, 则称该节点之间是连通的。网络的连通度在一定程度上反映了通信半径与节点数量的关系, 因此连通度的变化可以通过控制节点的通信半径或节点个数来实现。

表 1 列出在 100 次仿真实验的基础上不同通信半径条件下所对应的网络平均连通度, 设节点数为 200 个, 其中锚节点为 20 个。在表 1 所列参数的设置下, 对两种算法归一化的平均定位误差进行了比较, 如图 2 所示。

表 1 不同通信半径下网络的平均连通度

通信半径/m	网络平均连通度	通信半径/m	网络平均连通度
20	5.7384	40	20.9294
25	8.7267	45	25.8908
30	12.3185	50	31.2206
35	16.4434	60	42.5818

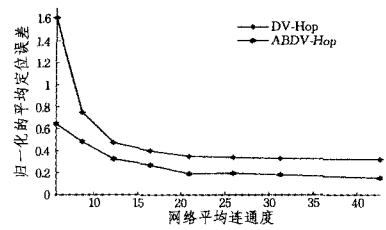


图 2 两种算法在不同网络平均连通度下的定位误差

从图 2 中可以看出, 在网络平均连通度较低时, 一些未知节点由于缺乏足够的参考节点信息, 导致定位误差较大; 随着网络平均连通度的增加, 两种算法的定位误差均有明显下降, 并且逐渐趋于稳定, 说明了在网络连通度达到一定值后, 该参数对误差的影响逐渐减小。同时可以看出, 在连通度较小 (小于 10) 时, 本文的改进算法的平均定位误差可以降至 45% 左右, 较 DV-Hop 算法有明显改善, 而本文算法较 DV-Hop 算法定位误差平均减小了 26.55%。

4.2 不同邻锚节点数条件下定位结果的比较

未知节点的平均邻居锚节点个数的大小在定位过程中对定位结果有一定的影响。

表 2 列出在 100 次仿真实验的不同锚节点个数时所对应的平均邻锚节点数, 设通信半径为 40m, 节点总数为 200 个。邻居锚节点数反映了锚节点个数与节点总数的关系, 因此可以通过改变锚节点个数或节点总数来控制平均邻居锚节点的数量。两种算法在表 2 所列的参数设置下, 归一化的平均定位误差如图 3 所示。

表 2 不同节点数条件下的平均邻锚节点数

锚节点个数	邻锚节点数	节点个数	邻锚节点数
5	0.5223	25	2.6184
10	1.0450	30	3.1253
15	1.5654	40	4.2090
20	2.0964	50	5.1985

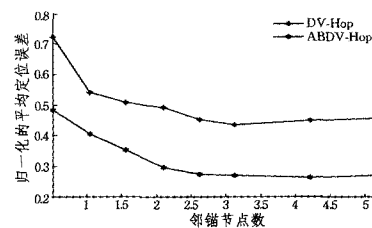


图 3 两种算法在不同邻锚节点数下的定位误差

从图 3 中可以看出, 两种算法在邻锚节点数变化的过程中, 平均定位误差逐渐减小, 但当大于 3.5 左右时, 定位误差有上升的趋势, 说明了邻锚节点数在一定范围内对减小定位误差有积极的影响, 但该参数取值不应过大。相比于传统 DV-Hop 算法, 本文算法定位误差平均减小了 18.02%。

4.3 不同通信半径条件下定位结果的比较

在仿真区域内分别随机分布 150 个和 300 个节点, 其中锚节点为 20 个, 将本文改进算法与传统 DV-Hop 算法在通信半径从 25m 到 80m 变化过程中归一化的平均定位误差进行比较, 实验结果如图 4 所示。

从图 4 的结果可知, 两种算法在通信半径增大的过程中, 定位误差逐渐减小, 并最终趋于稳定; 虽然两种算法分别在上述两种情况下最终的定位误差相差不大, 但在 150 个节点的条件, 两种算法的定位误差变化幅度较大, 而在 300 个节点的条件小, 平均定位误差减小的幅度相对平稳。相比于传统

的 DV-Hop 算法,本文算法在两种条件下定位误差平均减小了 19.2%和 11.76%。

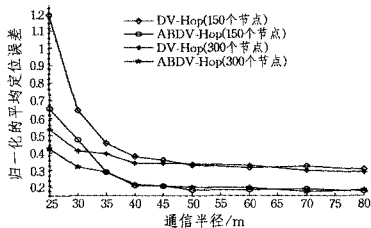


图 4 两种算法在不同通信半径下的定位误差

4.4 不同节点数条件下定位结果的比较

设初始锚节点个数分别为 20 和节点总数的 10%,通信半径为 40m,在仿真区域内随机布置从 50 到 400 变化的节点总个数,比较本文算法与 DV-Hop 算法归一化的平均定位误差,结果如图 5 所示。

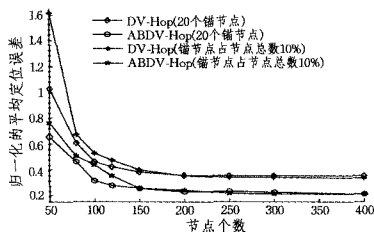


图 5 两种不同算法在节点数不同的定位误差

从图 5 可知,随着节点个数的增加,两种算法的平均定位误差均逐渐减小并趋于稳定。当锚节点个数固定为 20 时,相比于节点总数 10%的锚节点,在节点数小于 200 时,定位误差较小,但随着节点数的增多,两种算法在占节点总数 10%的条件下,定位误差逐渐小于在锚节点固定为 20 个的情况。两种不同条件下,本文算法相比于 DV-Hop 算法定位误差分别平均减小了 16.18%和 20.7%。

结束语 本文通过对传统 DV-Hop 算法定位过程的分析,针对其定位阶段由于环境、测距误差等因素导致采用最小二乘法计算未知节点坐标时存在较大误差的问题,在分析人工蜂群算法原理的基础上,将其替代最小二乘法用于定位阶

段,提出了基于人工蜂群算法的 DV-Hop 算法。该算法控制参数少、搜索精度高,且运行稳定,通过确定适应度函数,可以较快地寻求最优解。仿真实验表明,相比于传统 DV-Hop 算法,ABDV-Hop 算法在不同网络平均连通度、邻锚节点数量、通信半径及节点个数的情况下明显提高了定位精度。

参考文献

- [1] 尚志军,曾鹏,于海斌. 无线传感器网络节点定位问题[J]. 计算机科学,2004,31(10):35-38
- [2] 张佳,吴延海,石峰,等. 基于 DV-Hop 的无线传感器网络定位算法[J]. 计算机应用,2010,30(2):323-326
- [3] Nicolescu D, Nath B. Positioning in ad hoc networks[J]. Journal of Telecommunication System,2003,22:667-280
- [4] Lu Qing-ling, Bai Meng-liang, Zhang Wei, et al. A kind of improved DV-Hop Algorithm[C]//Proc. of the 2nd International Conference on Intelligent Control and Information Processing, Harbin, China; IEEE Press, 2011:867-869
- [5] Li Wen-wen, Zhou Wu-neng. Genetic algorithm-base localization algorithm for wireless sensor network[C]//Proc. of 2011 Seventh International Conference on Natural Computation. Shanghai, China, 2011:2096-2099
- [6] 姜韵,程良伦. 采用虚拟锚节点的高精度 VAD-Hop 定位算法[J]. 传感技术学报,2011,24(07):1048-1051
- [7] 王新生,赵衍静,李海涛. 基于 DV-Hop 定位算法的改进研究[J]. 计算机科学,2011,38(2):76-78
- [8] Karaboga D, Basturk B. A powerfull and efficient algorithm for numerical function optimization; artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3):459-471
- [9] Basu B, MAhanti G K. A competitive study of modified particle swarm optimization, differential evolution and artificial bees colony optimization in synthesis of circular array[C]//Proc. of Power, Control and Embedded System(ICPCES). 2010:1-5
- [10] 欧阳丹彤,何金胜,白洪涛. 一种约束粒子群优化的无线传感器网络节点定位算法[J]. 计算机科学,2011,38(7):46-50
- [11] 林金朝,陈晓冰,刘海波. 基于平均跳距修正的无线传感器网络节点迭代定位算法[J]. 通信学报,2009,30(10):107-113

(上接第 21 页)

的协同过滤推荐算法的性能瓶颈问题提供了一种可行的解决方案。

参考文献

- [1] 许海玲,吴潇,李晓东. 互联网推荐技术比较研究[J]. 软件学报, 2009,20(2):350-362
- [2] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009,42(8):30-37
- [3] Bell R M, Koren Y. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights[C]//Proc of the 7th IEEE International Conference on Data Mining. Omaha NE, USA; IEEE, 2007:43-52
- [4] Takacs G, Pilyasz I, Nemeth B, et al. Matrix Factorization and Neighbor Based Algorithms the Netflix Prize Problem[C]//Proceedings of the 2008 ACM conference on Recommender systems. Lausanne, Switzerland; ACM, 2008:267-274
- [5] 杨阳,向阳,熊磊. 基于矩阵分解与用户近邻模型的协同过滤推荐算法[J]. 计算机应用,2012,32(2):395-398
- [6] Zhou Y, Wilkinson D, Schreiber R, et al. Large-Scale Parallel

Collaborative Filtering for the Netflix Prize[C]//Proc of the 4th international conference on Algorithmic Aspects in Information and Management. 2008

- [7] Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters[J]. Communication of the ACM 50th anniversary issue, 2008, 51(1):107-113
- [8] Hadoop. Open-source software for reliable, scalable, distributed computing[EB/OL]. <http://hadoop.apache.org/>, 2011
- [9] Mahout. Scalable machine learning and data mining[EB/OL]. <http://mahout.apache.org>, 2011
- [10] Takacs G, Pilyasz I, Nemeth B, et al. Investigation of Various Matrix Factorization Methods for Large Recommender Systems [C]//Proc of the IEEE International Conference on Data Mining Workshops. IEEE, 2008:553-562
- [11] Pilyasz I, Zibriczky D, Tik D. Fast ALS-based Matrix Factorization for Explicit and Implicit Feedback Datasets[C]//Proceedings of the fourth ACM conference on Recommender systems. New York; ACM, 2010:71-78
- [12] 陈国良. 并行计算——结构、算法、编程(修订版)[M]. 北京:高等教育出版社, 2003