

移动对象数据库预测范围聚集查询技术研究^{*}

吴秋云 廖巍 景宁 李军

(国防科技大学电子科学与工程学院 长沙 410073)

摘要 针对预测范围聚集查询处理技术,提出了一种面向移动对象的聚集 TPR 树索引。聚集 TPR 树索引在 TPR 树中间节点中加入移动对象聚集信息以减少预测范围聚集查询所需要的节点访问代价。并增加了一个建于移动对象标识上的哈希辅助索引结构以支持自底向上的删除搜索算法,具有很好的动态更新性能和并发性。提出了一种 EPRA 查询算法,采用更精确的剪枝搜索准则,大大减少了查询所需要访问的磁盘节点,具有良好的查询性能。

关键词 预测范围,聚集查询,TPR-树,aTPR-树,EPRA 算法

A Novel Prediction Technique for Range Aggregation of Moving Objects

WU Qiu-Yun LIAO Wei JING Ning LI Jun

(College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073)

Abstract To efficiently process predictive range aggregate (PRA) queries, this paper presents a novel aggregate TPR-tree (aTPR-tree) for range aggregation of moving objects. aTPR-tree is based on TPR-tree structure and added with aggregate information in intermediate nodes to reduce the disk accesses of PRA queries. aTPR-tree is supplemented by a hash index on identifier of moving objects, and uses bottom-up delete algorithm, thus having a good update performance and concurrency. Also an Enhanced predictive range aggregate (EPRA) query algorithm which uses a more precise branch and bound searching strategy is developed, thus reducing the disk I/O greatly and having a good performance.

Keywords Predictive range, Aggregate queries, TPR-tree, aTPR-tree, EPRA algorithm

时空聚集 (spatiotemporal aggregation) 查询^[1] 是当前时空数据库研究中新兴的领域之一。它从时空数据集中获取满足空间与时间约束的时空对象集合并应用聚集函数 (如 COUNT, SUM, AVG, MAX, MIN 等 SQL 标准函数) 于此集合以获得相关统计信息。在位置服务及交通控制等应用中,时空数据库管理空间位置随着时间而连续变化的移动对象。大部分研究^[2-4] 通常将这些移动对象在时空数据库中以点对象来进行表示与存储,并假定其具有线性运动特性。服务器收集所有移动对象的运动与位置信息以回答各种预测查询,其中最普遍的一种查询是预测范围聚集查询 (predictive range aggregate query, PRA query)^[5],即根据当前数据库中移动对象的位置与运动信息,返回在未来某个时间窗口范围内落在给定空间区域之内的移动对象数目 (或基于移动对象属性上的总和、平均值、最大值和最小值等) 聚集信息。

目前 PRA 查询算法可以分为两类。一类采用现有的移动对象索引方法获取满足 PRA 查询条件的移动对象集合,然后对此集合利用聚集函数计算其统计信息,我们称之为精确查询。主要包括 TPR-tree^[2] 及其变种 TPR*-tree^[3]、VCI R-tree^[4] 等。这类通用索引方法在处理 PRA 查询时必须搜索至索引树叶节点,因而需要访问过多的索引节点,查询性能非常低下。另外由于动态维护移动对象索引的代价比较高昂,在一些频繁更新如类似数据流查询处理的应用中会由于过多的磁盘 I/O 导致查询性能下降。另外一类方法则是考虑到用户在某些情况下更加关心查询的速度而允许查询结果存在一

定的误差,因此许多学者基于局部均匀等假设前提,利用统计和抽样等方法来进行近似聚集查询^[5-7]。其中 Tao 等人提出的 STH 技术^[6] 对传统直方图进行扩充,将移动对象集合在速度域和空间域中划分成局部均匀分布的桶 (bucket),通过计算出落在聚集查询窗口内每个桶的选择性估计,然后相加便可得到 PRA 的查询估计。Sun 等人^[7] 针对移动对象以数据流形式更新位置的应用,将二维空间划分为规则格网,每个格网内保存落在其中的移动对象聚集信息,并使用 AMH 数据结构来保存 COUNT 聚集函数信息,可以回答历史和预测聚集查询。但是这种方法只适合于瞬态聚集查询,无法回答时间窗口查询。Tao 等人在文^[5] 中提出了 Venn Sampling 方法,采用层次抽样思想根据查询的空间分布对移动对象集合进行偏倚抽样,只有当基于抽样集合的查询估计值与精确值误差超出阈值时,才会更新抽样集合,从而大大提高了聚集查询性能。但是统计和抽样方式并不能精确地表示整个数据集,上述近似算法在处理如 MAX 和 MIN 等聚集查询时并非有效。因此,对于预测范围聚集查询而言,利用新的移动对象索引技术进行精确查询更具有实际的价值。

本文研究面向移动对象的精确预测范围聚集查询问题,通过在 TPR 树中间节点中加入聚集信息,提出了一种新的预测范围聚集查询索引 (aggregate TPR-tree, aTPR-tree) 结构,并采用自底向上的删除搜索算法以提高 aTPR 树索引的动态更新性能和并发性。基于 aTPR 树索引提出了一种增强的 PRA 查询 (Enhanced predictive range aggregate query,

^{*} 国家高技术研究发展计划 863 资助,项目编号:2003A5110。廖巍 博士研究生,主要研究方向为空间数据库、时空数据库等。熊伟 博士研究生,主要研究方向为空间数据库、主动数据库等。陈宏盛 副教授,主要研究方向为数据库技术等。景宁 博士,教授,博士生导师,CCF 会员,主要研究方向为地理信息系统与数据库技术。

EPRA)算法,采用更精确的剪枝搜索准则以减少查询所需要磁盘访问代价。最后通过实验比较了基于 aTPR 树索引的 EPRA 查询算法与现有精确预测范围聚集查询算法的性能。

1 面向移动对象的聚集 TPR-tree 索引

时空对象具有空间和时间尺度(extent),它可以是二维或三维空间中的点、线、面。其空间位置和/或形状随着时间而连续(如汽车运动)或离散(如耕地、森林的变迁)变化。本文中,我们关心的是在二维空间中位置随着时间而连续变化的点对象(移动对象)。在数据库中,通常表示为元组形式 $\langle Loc, Vec, A_1, A_2, \dots, A_n \rangle$,其中 Loc, Vec 分别表示移动对象在参考或更新时刻的空间位置、速度矢量, A_1, A_2, \dots, A_n 则表示移动对象的非空间属性信息。

时空聚集查询从数据库中获取满足时空查询谓词的移动对象元组集合,并返回一个关于集合中移动对象非空间属性的聚集信息(如 COUNT、AVG 等聚集函数)。给定范围查询 $q(q_R, q_T)$,即获取在查询时间窗口 q_T 范围内落在空间区域 q_R 之内的移动对象,假设数据库中移动对象集合为 P ,则时空查询谓词 Sel 可以表示为 $Sel(P) = \{p_i | \exists t \in q_T \text{ such that } p_i(t) \in q_R\}$ 。特别地,若 $q_T = [t_c, t_c + TL]$,其中 t_c 表示范围查询 q 提交时刻,则查询 q 为预测聚集查询。通常而言,查询空间区域 q_R 为静态空间矩形窗口。

定义 1(预测范围聚集查询) 令 P 为元组形式为 $\langle Loc, Vec, A_1, A_2, \dots, A_n \rangle$ 集合,属性 A_i 值域为 D_i ,给定可数聚集函数集合 $Agg = \{f_1, f_2, \dots, f_n\}$,其中 f_i 作用于属性 A_i 上,

对每个函数 $f_i \in Agg$,有 $f_i : D_1 \times D_2 \times \dots \times D_n \rightarrow D_{agg}$, D_{agg} 为聚集函数值域,则查询 $f_i(q, P) = f_i(Sel(P)) = f_i(\{p_i | \exists t \in q_T \text{ such that } p_i(t) \in q_R\})$ 表示一个预测范围聚集查询。不失一般性,我们将 $Agg(q, P)$ 统称为预测范围聚集查询。

aTPR 树索引保留了基本的 TPR 树结构,并对其进行扩充,在中间节点每条记录中加入对应节点子树聚集信息形成聚集 TPR 树,其记录形式为四元组 $\langle MBR, VBR, Agg, ptr \rangle$,其中 MBR, VBR, Agg, ptr 分别表示节点的最小矩形包围框、速度包围框、包含的移动对象聚集信息、节点指针。为了提高 aTPR 树索引的动态更新性能,我们增加了一个建于移动对象标识上的基于磁盘哈希索引结构,通过哈希索引可以直接访问移动对象所在的聚集 TPR 树索引叶节点页面,并采用自底向上的更新算法提高 aTPR 树索引的更新效率。同时为了支持自底向上的更新算法,我们修改聚集 TPR 树索引节点记录形式为 $\langle entry, \dots, entry, parent ptr \rangle$,其中 $entry$ 表示节点包含的子节点记录, $parent ptr$ 则表示指向父节点的物理磁盘地址。与节点页面大小相比, $parent ptr$ 所占用的磁盘空间非常小,对聚集 TPR 树扇出造成的影响可以忽略不计。

如图 1 所示为 aTPR 树索引结构,图中右边是建于移动对象标识上的哈希索引结构,记录形式为二元组 $\langle oid, ptr \rangle$,其中 oid 表示移动对象标识, ptr 指向移动对象记录所在叶节点的磁盘页面偏移。图中左边所示为聚集 TPR 树索引,与标准 TPR 树相比,聚集 TPR 树中间节点记录中增加了一项聚集描述信息(COUNT 函数)。另外,出于简洁考虑中间节点指向父节点的指针未在图中显式给出。

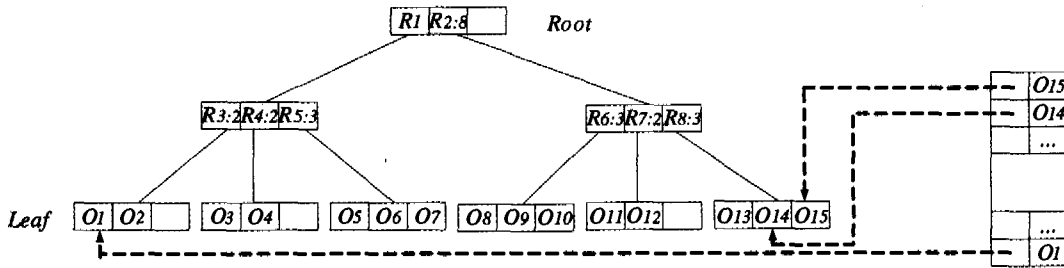


图 1 aTPR 树索引结构

aTPR 树索引构建算法使用类似于 R 树批装载(bulk loading)技术以减少磁盘访问代价。与标准批装载技术不同的是,在聚集 TPR 树构建过程中必须同时修改节点所包含的移动对象聚集信息。aTPR 树索引插入算法采用 TPR* 树的标准插入算法。只是在插入路径中必须修改每个中间节点的聚集信息,同时还必须更新速度桶记录中的聚集信息。

删除移动对象时,很直接的想法是利用 TPR* 树标准删除算法将此移动对象从 aTPR 树索引中删除,但是由于删除搜索路径可能存在多条,从而造成过多的磁盘 I/O。而且在搜索过程中不能即时的修改中间节点聚集信息,而必须保存删除路径以进行聚集信息修改时的二次搜索,会造成多余的磁盘 I/O。因此我们采用自底向上的删除算法,其思想是首先利用基于磁盘的移动对象标识哈希索引直接定位移动对象记录所在的聚集 TPR 树叶节点页面,从中删除此记录并修改叶节点聚集信息,然后利用节点记录中指向父节点的 parentptr 指针逆向回溯到根节点,对回溯路径中的节点依次进行聚集信息和 MBR 的修改。aTPR 树索引更新算法则采用删除-插入机制,首先利用自底向上的删除算法删除此对象所在记录,然后利用插入算法在 aTPR 树索引中插入新的移

动对象记录。

2 增强预测范围聚集查询处理方法(EPRA)

基于 aTPR 树索引结构,一种很直接的 PRA 查询方法是判断在查询时间窗口内,是否存在某时刻使得聚集 TPR 树节点 MBR 完全被包含在查询空间窗口之内。若存在则只需返回索引节点的聚集信息即可;否则必须继续进行搜索。但是采用这种简单的判断准则,查询算法需要访问在查询时间窗口内不同时刻移动对象完全落在查询空间窗口内的索引节点,而实际上这些速度桶和节点不需要进行访问,因而会造成多余的磁盘 I/O。基于这种考虑,我们提出了增强的预测范围聚集查询算法(enhanced predictive range aggregate query, EPRA)。EPRA 算法采用更精确的搜索裁减准则,大大减少了节点访问代价。

定理 1 给定预测范围查询 $q(q_R, q_T)$,其中 $q_T = [t_c, t_c + TL]$,聚集 TPR 树节点 $E(MBR, VBR)$,若存在某个时刻 $t \in [t_c, t_c + TL]$,满足 $MBR_t \subset q_R$,其中 MBR_t 表示 t 时刻节点 MBR 所覆盖的空间域,则预测范围聚集查询 $Agg(q, E)$,等于节点 E 包含的所有移动对象聚集信息。

证明,根据定义 1 直接可证。

定理 2 给定预测范围查询 $q(q_R, q_T)$, 其中 $q_T = [t_c, t_c + TL]$, 聚集 TPR 树节点 $E(MBR, VBR)$, 在 t_c 时刻若 $MBR_k \cap q_R \neq \Phi$, 令 $S = MBR_k - MBR_k \cap q_R$, 若存在某个时刻 $t \in [t_c, t_c + TL]$, 满足 $S_i \subset q_R$, 其中 S_i 表示 S 以速度矢量 VBR 运动时, t 时刻所覆盖的空间域, 则预测范围聚集查询 $Agg(q, buck)$, 等于节点 E 内所有移动对象的聚集信息。

证明: 令 P 为聚集 TPR 树节点 E 中移动对象集合, P_k 为 $MBR_k \cap q_R$ 中包含的移动对象集合, P_s 为 S 中包含的移动对象集合, $P_k \cup P_s = P$, 在 t_c 时刻 $P_k \subset q_R$, 在 t 时刻 $P_s \subset q_R$, 即对节点 E 中任意移动对象 p , 存在某个时刻 $t \in [t_c, t_c + TL]$, $q \subset q_R$, 根据定义 1, 直接返回节点 E 内的聚集信息即可。

在处理预测范围聚集查询时, EPRA 算法在搜索聚集 TPR 树时, 从根节点开始进行深度优先搜索, 对每个访问节点 E , 根据其 MBR 和 VBR 大小可以计算出查询提交时刻 MBR_k 。此时节点 MBR_k 与查询窗口之间的拓扑关系可以分为四种情况: 相交、包含、相离、被包含。显然, 若 MBR_k 包含于查询窗口 q_R 内, 节点 MBR 中所有移动对象落入查询窗口内, 根据定理 1, 只需返回节点聚集信息, 而不必访问节点子树; 若 MBR_k 包含查询窗口 q_R , 查询窗口不能保证包含所有的移动对象, 因此则必须访问节点子树; 若 $MBR_k \cap q_R = \Phi$, 即 MBR_k 与查询窗口 q_R 相离, 则需再判断查询窗口 $[t_c, t_c + TL]$ 范围内存在某个时刻 t , 使得 $MBR_k \subset q_R$, 即 MBR_k 包含在查询窗口 q_R 内, 若存在, 根据定理 1, 则直接返回节点聚集信息, 否则必须搜索节点子树; 若 MBR_k 与查询窗口 q_R 相交, 则根据定理 2, 判断节点剩下区域是否在查询窗口 $[t_c, t_c + TL]$ 范围内某个时刻 t 完全落在查询窗口内, 若存在, 则直接返回节点聚集信息, 否则必须继续搜索节点子树。

算法 1 EPRA 查询处理算法

输入: 预测范围聚集查询 $q(q_R, q_T)$, 聚集 TPR 树索引 T
 输出: $Agg(q)$
 步骤:
 1 初始化 $Agg(q)$, 令 $Agg(q) \rightarrow \emptyset$;
 2 对聚集 TPR 树 T , 从根节点开始进行深度优先遍历, 对每个节点记录 E , 根据 E 中 MBR 和 VBR 计算出查询提交时刻 t_c 时节点矩形包围框所覆盖的空间域 MBR_k
 2.1 若 E 为中间节点, 则进行如下判断
 2.1.1 若 $MBR_k \subset q_R$, 将此节点聚集信息加入到 $Agg(q)$ 中, 令 $Agg(q) \leftarrow Agg(q) + E.Agg$;
 2.1.2 若 $MBR_k \supset q_R$, 对记录 E 节点进行深度优先遍历
 2.1.3 若 $MBR_k \cap q_R \neq \Phi$, 令 $S = MBR_k - MBR_k \cap q_R$, 若 S 为非矩形框, 将其分割为两个矩形框, 对每个矩形框 MBR_i , 令其速度包围框 $VBR_i \leftarrow VBR$, 计算查询时间窗口 $q_T = [t_c, t_c + TL]$ 内任意时刻 t 时 MBR_i 所覆盖的空间域 MBR_i , 对任意 i , 若存在 $t \in q_T$, 满足 $MBR_i \subset q_R$, 则将此节点聚集信息加入到 $Agg(q)$ 中, 令 $Agg(q) \leftarrow Agg(q) + E.Agg$, 否则对记录 E 节点进行深度优先遍历

- 2.1.4 若 $MBR_k \cap q_R = \Phi$, 计算查询时间窗口 $q_T = [t_c, t_c + TL]$ 内任意时刻 t 时 MBR 所覆盖的空间域 MBR_i , 若存在 $t \in q_T$, 满足 $MBR_i \subset q_R$, 则将此节点聚集信息加入到 $Agg(q)$ 中, 令 $Agg(q) \leftarrow Agg(q) + E.Agg$, 否则记录 E 节点进行深度优先遍历
- 2.2 若 E 为叶节点, 则对其所包含的所有移动对象 p
 - 2.2.1 若存在某个时刻 $t \in [t_c, t_c + TL]$, 使得 p 落在查询空间窗口 q_R 范围内, 则将此对象的聚集信息 $p.Agg$ 加入到 $Agg(q)$ 中, 令 $Agg(q) \leftarrow Agg(q) + p.Agg$
 - 2.2.2 否则, 忽略此对象
- 2.3 若遍历完聚集 TPR 树 T , 返回到步骤 2 循环继续处理

3 实验结果与分析

3.1 实验内容与设置

为了评估 aTPR 树索引和基于 aTPR 树索引的 EPRA 查询性能, 我们进行了大量的实验。实验数据与文[3]类似, 采用随机生成, 数据集大小为 100k, 移动对象用点坐标来表示, 均匀分布在 10000×10000 空间域内。移动对象运动用如下方式模拟, 从 LA[Tiger](包含 128k 个二维矩形框) 实际数据集中抽取 5000 个 MBR , 其中心作为移动对象运动目的地, 参考时刻 t_c , 每个移动对象随机选择一个目的地运动, 速度大小在 $[0, 20]$ 之间均匀分布。每个移动对象平均每隔 150 个时间单元更新速度矢量。

我们使用 aTPR 树对实验数据集进行索引, 聚集 TPR 树页面大小设置为 1k, 中间节点扇出为 22, 叶节点中可以保存 41 个移动对象, 聚集 TPR 树高度为 4 层。使用的页面缓存大小为 50k, 即 50 个缓存页面, 并使用最近使用(LUR)缓存替代策略。aTPR 查询产生方式如下, 1) 查询 q 空间窗口大小 $qRlen$ 分别固定为 $100 \times 100, 400 \times 400, 800 \times 800, 1200 \times 1200, 1600 \times 1600$ 。窗口中心点位置在 $(10000 - qRlen) \times (10000 - qRlen)$ 空间域内随机分布; 2) 查询时间窗口范围为 $[T_{in}, T_{in} + TL]$ (T_{in} 为查询提交时刻), 其中 TL 为查询时间窗口大小 $qTlen$, 令 $TL = 20, 40, 60, 80, 100$; 3) PRA 查询为 COUNT 聚集函数查询, 即获取查询窗口范围内的移动对象数目信息。查询性能估计采用回答 10 个 PRA 查询所需要的平均磁盘 I/O 次数来衡量。动态更新性能则用每个时间单元移动对象更新所需要的平均磁盘 I/O 次数来衡量。

实验硬件环境为 Celeron 2.4GHz 的 CPU, 256MB DDR 内存和 7,200 RPM 硬盘。软件环境使用 Windows XP 操作系统和 Visual C++ 6.0 IDE, 测试代码利用 C 语言编写。

3.2 实验结果与分析

我们比较了 aTPR-tree 与 TPR*-tree 在移动对象速度更新及回答预测范围聚集查询时所需要的平均磁盘 I/O 次数。

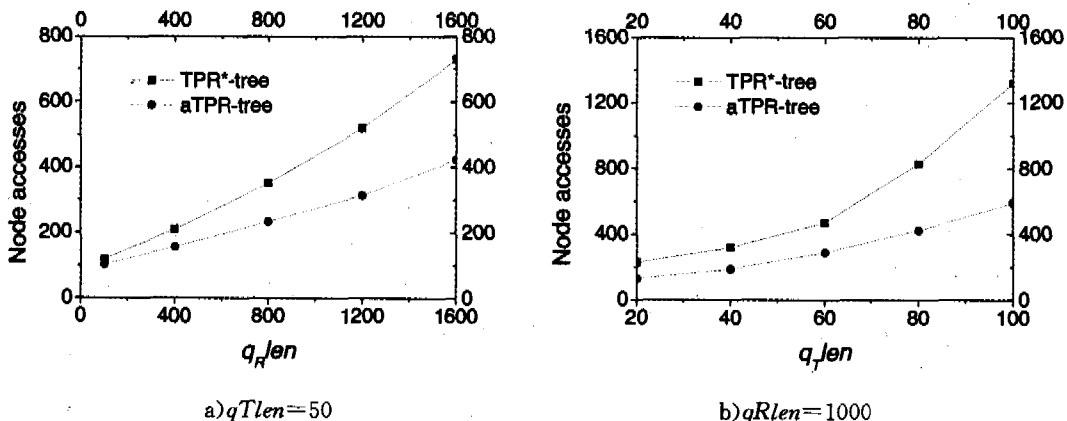


图 2 预测范围聚集查询性能比较

图 2 中 a) 和 b) 所示为分别固定预测范围聚集查询时间窗口 $qTlen = 50$ 及查询空间窗口 $qRlen = 1000$ 时, TPR*-tree 与 aTPR-tree 回答查询所需要的磁盘 I/O 代价。从中可以看出, 基于 aTPR-tree 索引的 EPRA 查询算法具有很好的查询性能。这是由于 TPR*-tree 中间节点并未包含移动对象聚集信息, 回答 PRA 查询必须访问查询空间窗口内的所有叶节点, aTPR-tree 中间节点包含了移动对象聚集信息, 因此可以节省一部分节点访问代价。而且 EPRA 查询算法采用更精确的搜索剪枝准则, 因而具有很好的查询性能。

总结 本文针对移动对象预测范围聚集查询问题, 提出了一种面向移动对象预测范围聚集查询索引 (aggregate TPR-tree, aTPR-tree) 以减少预测范围聚集查询所访问的节点。aTPR 树索引增加了一个建于移动对象标识上的哈希辅助索引结构, 并利用自底向上的更新算法, 具有很好的更新性能和并发性。基于 aTPR 树索引的 EPRA 查询算法大大减少了查询所需要访问的磁盘节点, 且支持多种预测范围聚集查

询, 具有良好的查询性能。

参考文献

- 1 Lo'pez I F V, Snodgrass R T. Spatiotemporal Aggregate Computation: A Survey. In: IEEE TKDE, 2005(2):271~286
- 2 Saltenis S, Jensen C S, et al. Indexing the Positions of Continuously Moving Objects. In: Proc. of the SIGMOD, 2000
- 3 Tao Y, Papadias D, Sun J. The TPR*-Tree: An Optimized Spatio-Temporal Access Method for Predictive Queries. In: VLDB, 2003, 790~801
- 4 Prabhakar S, Xia Y, Kalashnikov D V, et al. Query Indexing and Velocity Constrained Indexing: Scalable Techniques for Continuous Queries on Moving Objects. IEEE Transactions on Computers, 2002
- 5 Tao Yufei, Papadias D, Zhai Jian, et al. Venn Sampling: A Novel Prediction Technique for Moving Objects. In: Proc. ICDE, 2005
- 6 Tao Yufei, Sun Jimeng, Papadias D. Selectivity Estimation for Predictive Spatio-Temporal Queries. In: Proc. ICDE, 2003
- 7 Sun Jimeng, Papadias D, Tao Y, et al. Querying about the Past, the Present, and the Future in Spatio-Temporal Databases. In: VLDB, 2004

(上接第 38 页)

一步研究的问题包括: (1) 通信链路质量的评价机制和邻居节点的选择策略; (2) 拓扑结构的自适应性; (3) 新颖的网络拓扑控制机制; (4) 结合数据收集和移动目标跟踪这两种应用热点, 研究面向具体应用的拓扑控制方案。

综上所述, 无线传感器网络拓扑控制研究的发展趋势是以实际应用为背景, 多种机制相互融合, 强调网络拓扑控制的自适应性和鲁棒性, 在保证网络的连通性和覆盖度^[18]的前提下, 提高网络的通信效率, 最大限度地节省能量, 延长整个网络的生存时间。

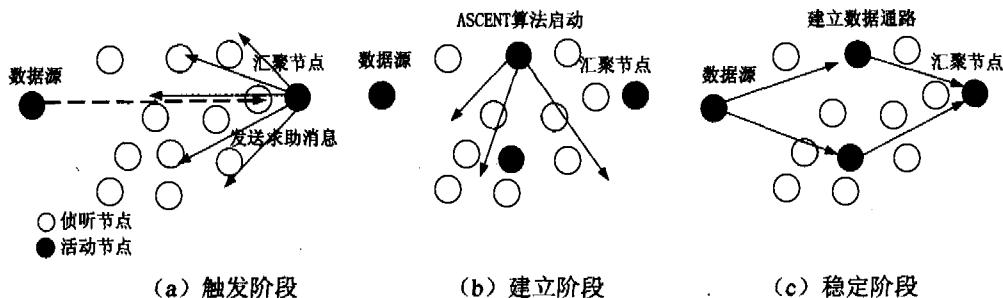


图 4 ASCENT 算法示意图

参考文献

- 1 Akyildiz I F, Su W, Sankarasubramanian Y, et al. A Survey on Sensor Networks. IEEE Communications Magazine, 2002, 4: 102~114
- 2 Santi P. Topology Control in Wireless Ad Hoc and Sensor Networks. ACM Comp Surveys, 2005, 37(2): 164~194
- 3 Amis A, Prakash R, Vuong T, et al. MaxMin D-Cluster Formation in Wireless Ad Hoc Networks. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), Mar. 1999
- 4 Narayanaswamy S, Kawadia V, Sreenivas R S, et al. Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In: Proceedings of the European Wireless Conference, Florence, Italy, 2002. 156~162
- 5 Kawadia V, Kumar P R. Power Control and Clustering in Ad Hoc Networks. INFOCOM 2003, San Francisco, March 30 - April 3, 2003
- 6 Kubisch M, Karl H, Wolisz A, et al. Distributed algorithms for transmission power control in wireless sensor networks. IEEE WCNC 2003, New Orleans, Louisiana, March 2003
- 7 Jaromczyk J W, Toussaint G T. Relative neighborhood graphs and their relatives. In: Proc. IEEE, 1992, 80: 1502~1517
- 8 Li N, Hou J C. Topology control in heterogeneous wireless networks: Problems and solutions. INFOCOM, 2004
- 9 Rodoplu V V, Meng T H. Minimum Energy Mobile Wireless Networks. IEEE J Selected Areas in Communications, August 1999, 17(8): 1333~1344
- 10 Li L, Halpern J Y, Bahl P, et al. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In: Proc. ACM Symposium on Principles of Distributed Computing (PODC), Newport, RI, USA, Aug. 2001. 264~273
- 11 Heinzelman W R, Chandrakasan A, Balakrishnan H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, October 2002, 1(4): 660~670
- 12 Bandyopadhyay S, Coyle E J. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: Proceedings of IEEE INFOCOM'03, April 2003
- 13 Younis O, Fahmy S. Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. In: Proceedings of IEEE INFOCOM, March 2004
- 14 Deb B, Bhatnagar S, Nath B. A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management. [DCS Technical Report]. DCS-TR-441. Rutgers University, May 2001
- 15 Bao L, Garcia-Luna-Aceves J J. Topology Management in Ad Hoc Networks. In: Proc. MobiHoc, 2003. 129~140
- 16 Schurgers C, Tsiatsis V, Ganeriwal S, et al. Topology management for sensor networks: exploiting latency and density. Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, Lausanne, Switzerland, June 09-11, 2002
- 17 Cerpa A, Estrin D. ASCENT: Adaptive Self-Configuring Sensor Networks Topologies. In: Proceedings of IEEE INFOCOM, New York, NY, June 2002
- 18 Wang X, Xing G, Zhang Y, et al. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In: The First ACM Conference on Embedded Networked Sensor Systems (Sensys03), Los Angeles, CA, USA, November 2003