

# 基于 Voronoi 图的移动单元节点调度算法

于瑞云<sup>1</sup>, 王兴伟<sup>2</sup>

(1. 东北大学计算中心, 沈阳 110004; 2. 东北大学信息科学与工程学院, 沈阳 110004)

**摘要:** 在稀疏无线传感器网络中, 移动单元节点常被用于数据采集和转发。基于 Voronoi 图设计一条尽可能短的移动单元节点数据采集路径。在该路径中, 移动单元节点被调度去访问一个 Voronoi 节点子集, 在给定通信半径内, 该节点子集能覆盖所有传感器节点。仿真实验结果表明, 通过连接 Voronoi 节点子集而形成的优化路径能有效缩短移动单元节点的行进路径长度。

**关键词:** 稀疏无线传感器网络; Voronoi 图; 移动单元节点; 数据采集

## Voronoi Diagram-based Mobile Element Schedule Algorithm

YU Rui-yun<sup>1</sup>, WANG Xing-wei<sup>2</sup>

(1. Computing Center, Northeastern University, Shenyang 110004;

2. College of Information Science and Engineering, Northeastern University, Shenyang 110004)

**【Abstract】** Mobile Element(ME) are usually exploited for collecting and relaying data in sparse wireless sensor networks. This paper proposes the Voronoi Diagram-based Mobile Element Schedule(VDMES) algorithm to construct the shortest possible path for ME data collection. ME are scheduled to visit a small subset of Voronoi vertices, which exactly covers all sensor nodes in a given transmission radius. Simulation experimental result shows that the path concatenating the Voronoi vertices is much shorter than that formed by regular sensor nodes.

**【Key words】** sparse wireless sensor networks; Voronoi diagram; Mobile Element(ME); data collection

### 1 概述

无线传感器网络通常是密集分布的, 但在一些实际应用中, 由于地理条件的限制或应用需求的不同, 传感器节点是稀疏布置的。在稀疏无线传感器网络中, 端到端路径的缺失、高时延、网络分割等特点给协议算法设计带来了很大的挑战。

文献[1]利用网络节点的随机移动性完成数据在稀疏网络不同部分之间的交换。文献[2]通过主动控制节点的运动轨迹进行高效地数据传递。文献[3-4]采用了不同于文献[1-2]的方法, 在稀疏传感器网络架构之上, 叠加了一个由移动中继节点组成的消息中继层, 履行连通网络、消息传递、数据采集等功能。与传感器网络节点相比, 移动中继节点具有更强的处理能力、存储能力和移动性, 并且能量不受限。

上述文献主要是调度移动中继节点进行有效地信息查询、消息交换等, 本文提出了一个基于 Voronoi 图的移动单元节点调度算法(Voronoi Diagram-based Mobile Element Schedule, VDMES), 目标是为移动单元节点(Mobile Element, ME)设计一条最优的数据采集路径, 使 ME 能以最小时延从全网采集传感器节点感知的数据。该算法的核心思想是构建一个最小的 Voronoi 映射节点集, 在给定通信半径内, 实现传感器节点集到 Voronoi 映射节点集的映射。然后, 在此映射节点集的基础上为 ME 构建一条旅行销售商(TSP)路径。

本文在构建 Voronoi 映射节点集时, 采用了虚拟节点迭代插入(iterative virtual node insertion)技术和贪婪节点筛选(greedy selection)算法, 仿真实验结果表明, 与普通的 ME 调度算法相比较, VDMES 算法有效地缩短了 ME 行进路径的长度。

### 2 Voronoi图和Delaunay三角剖分

假设  $S = \{p_1, p_2, \dots, p_i, \dots, p_n\}$  是欧几里德平面的一个散点(site)集。Voronoi 图围绕每个散点将整个平面分割成多个区域(Voronoi 区域), 使在围绕散点  $p_i$  的 Voronoi 区域  $V(p_i)$  中的任意点  $x$  满足条件:

$$V(p_i) = \{x : |p_i - x| \leq |p_j - x|, \forall j \neq i\} \quad (1)$$

所有散点的 Voronoi 区域集合构成散点集  $S$  的 Voronoi 图  $V(S)$ 。在平面中给定 12 个散点, 相应的 Voronoi 图如图 1 所示。

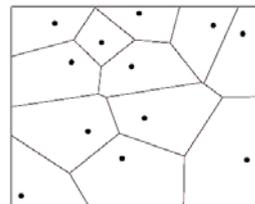


图 1 Voronoi 图

Delaunay 三角剖分是把平面散点集合剖分成不均匀的三角形网格, 每个三角形都以散点为顶点, 在 Delaunay 三角网中任一个三角形的外接圆范围内都不会存在散点集合中的其

**基金项目:** 国家“863”计划基金资助项目(2006AA01Z214); 国家自然科学基金资助项目(606073159, 70671020)

**作者简介:** 于瑞云(1974-), 男, 讲师, 主研方向: 无线传感器网络; 王兴伟, 教授

**收稿日期:** 2009-06-05 **E-mail:** yuryu@cc.neu.edu.cn

他点。

散点集  $S$  的 Delaunay 三角剖分记作  $DT(S)$ ,  $DT(S)$  与  $V(S)$  互为对偶图, 两者之间的对偶关系如图 2 所示。

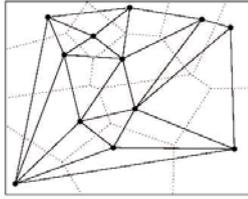


图 2 Voronoi 图的对偶关系

### 3 基于Voronoi图的移动单元节点调度

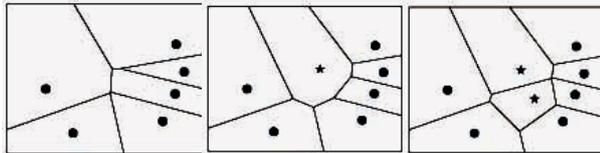
#### 3.1 Voronoi映射节点集的构建

在 Voronoi 图中, 一个 Voronoi 节点与至少 3 个最近散点等距。将网络中的传感器节点看作散点集  $S$ , 如果一个 Voronoi 节点  $v$  到最近散点的距离  $r_v$  小于通信半径  $\gamma$ , 则可以让 ME 访问  $v$  来收集在通信半径范围内的传感器节点的数据。

VDMES 算法的目的就是寻找一个最小的 Voronoi 映射节点集  $T$  在通信半径  $\gamma$  内覆盖  $S$  中的所有散点, 并调度 ME 沿着由集合  $T$  中节点生成的最短路径去收集全网的数据。

然而, 由于在稀疏传感器网络中,  $r_v$  很可能大于  $\gamma$ , 因此  $S$  中的某些散点不能被任何 Voronoi 节点覆盖, 本文提出一个虚拟散点(virtual site)的概念。虚拟散点是指平面中的一个几何点, 在计算 Voronoi 图时被当作一个真正的散点。

在散点集  $S$  中插入虚拟散点将产生一个更密集的 Voronoi 图, 图 3 显示了在散点集中插入一个和 2 个虚拟散点后 Voronoi 图的变化情况。



(a)初始 Voronoi 图 (b)插入 1 个虚拟散点 (c)插入 2 个虚拟散点

图 3 插入虚拟散点对 Voronoi 图的影响

通过在散点集中增加适当的虚拟散点, 用贪婪算法选取合适的 Voronoi 映射节点, 构建一个尽可能小的 Voronoi 映射节点集, 具体算法如下(集合  $T$  为构建的 Voronoi 映射节点集):

#### 算法 1 Voronoi 映射节点集构造算法(MVVSC)

```

将  $S$  中的散点按 X 坐标排序, 生成有序散点集  $S = \{p_1, p_2, \dots, p_i, \dots, p_n\}$ 
 $S' \leftarrow S$ 
for  $i = 1$  to  $n$  do
    添加散点  $p_i$ , Delaunay 三角化图中受影响部分, 将新生成三角形的结构存入  $DT(S')$ 
    for all 在  $DT(S')$  中的三角形 do
        if 三角形外接圆包含  $p_i$  then 删除此三角形 end if
    end for
end for
计算循环得到的 Delaunay 三角网的对偶图, 就是 Voronoi 图  $V(S')$ , Voronoi 节点集为  $R = \{v_1, v_2, \dots, v_j, \dots, v_n\}$ 
 $\gamma \leftarrow$  网络通信半径
 $T \leftarrow \Phi$ 
调用函数 VoroMapping() (见算法 2)
 $Q \leftarrow R$ 

```

```

 $R \leftarrow \Phi$ 
while  $S \neq \Phi$  do
    从  $Q$  中选取节点  $u$ , 使  $u$  距离  $S$  中未被覆盖节点的最短距离值最小
     $Q \leftarrow Q - u$ 
     $S' \leftarrow S' \cup u$ 
    重新三角化图中受影响的部分, 更新集合  $DT(S')$  和  $V(S')$ , 将新增 Voronoi 节点存入集合  $R$ 
    调用函数 VoroMapping()
     $Q \leftarrow Q \cup R$ 
     $R \leftarrow \Phi$ 
end while

```

#### 算法 2 VoroMapping() 函数

输入  $R, S, T, \gamma$

输出  $S, T$

```

 $U \leftarrow \Phi$ 
for  $j = 1$  to  $|R|$  do
    将  $S$  中与 Voronoi 节点  $v_j$  的距离小于  $\gamma$  的散点存入  $S_j$ 
     $U \leftarrow U \cup S_j$ 
end for
repeat
    选取  $S_j$ , 使  $|S_j \cap U|$  最大化
     $U \leftarrow U - S_j$ 
     $S \leftarrow S - S_j$ 
     $R \leftarrow R - v_j$ 
     $T \leftarrow T \cup v_j$ 
until ( $U$  中的所有子集皆为空)

```

#### 3.2 移动单元节点调度

根据移动单元节点的数量, 本文将 VDMES 算法划分为 2 类: 单汇聚节点调度算法(sVDMES)和多汇聚节点调度算法(mVDMES)。

##### 3.2.1 单汇聚节点调度算法

如果只有一个 ME, 节点调度实际上就是一个 TSP 问题。为便于实现, 采用最近邻居算法来计算移动汇聚节点的 TSP 路径, 同时为了获得近似最优的 TSP 路径, sVDMES 算法选择每个 Voronoi 节点作为起始节点计算一条 TSP 路径, 最后从中选取最短的路径。具体实现如下:

#### 算法 3 单汇聚节点调度算法(sVDMES)

```

for  $i = 1$  to  $|T|$  do
     $P_i \leftarrow \Phi$ 
     $P_i \leftarrow P_i \cup v_i$  //  $v_i \in T$ 
    while  $|P_i| < |T|$  do
         $r \leftarrow P_i$  中新增的节点
        if  $s \in T$  and  $s \notin P_i$  and  $s$  是  $T$  中距  $r$  最近的点 then
             $P_i \leftarrow P_i \cup s$ 
        end if
    end while
end for
返回具有最短路径长度的有序集合  $P_i$ 

```

##### 3.2.2 多汇聚节点调度算法

当网络中有多个移动单元节点负责数据采集时, 汇聚节点调度实际上是一个多旅行销售商问题(MTSP)或车辆路问题(VRP)。由于版面所限, 本文对此问题暂不作讨论, 具体可参阅文献[5-6]。

### 4 性能评价

本节通过实验仿真分析 sVDMES 算法的性能, 并与普通汇聚节点调度算法(PlainMES)进行比较。PlainMES 调度 ME

沿着连接传感器节点的 TSP 路径收集网络数据。

#### 4.1 仿真环境

本文的仿真环境为 Matlab R2006a, 由于在稀疏无线传感器网络中, 节点之间无法正常通信, 因此算法采用集中式的方法实现。假设移动单元节点 ME 掌握整个网络的拓扑情况 (包括节点的 ID 和位置等信息), ME 根据这些信息集中式地计算 Voronoi 映射节点集, 并在此基础上进行优化路径构建, 然后进行运动调度。

仿真网络模型在  $1\ 000 \times 1\ 000\ \text{m}^2$  的场地中随机生成, 假定传感器网络节点是静止的, 节点的通信半径  $\gamma$  初始设定为 100 m, 并且可以在 60 m~150 m 之间随意调节; ME 与普通传感器网络节点具有相同的通信半径, 但 ME 具有足够的计算能力、存储空间, 并且能量不受限。

#### 4.2 参数影响

##### 4.2.1 节点数量的影响

在场地中分别随机布置 10, 20, ..., 100 个传感器节点, 在每种配置下, 以不同的拓扑结构运行 10 次 MVVSC 算法。

表 1 列出了传感器节点数量 ( $sNum$ )、虚拟散点平均数量 ( $vsNum$ ) 和 Voronoi 映射节点平均数量 ( $vvNum$ ) 之间的关系。

表 1 节点数量对 MVVSC 算法的影响

$sNum$	$vsNum$	$vvNum$	$sNum$	$vsNum$	$vvNum$
10	13.0	8.1	60	2.9	22.1
20	11.1	12.4	70	2.8	23.9
30	11.8	16.3	80	2.9	26.1
40	8.2	20.0	90	1.2	27.6
50	5.7	20.9	100	0.5	28.2

可见, Voronoi 映射节点平均数量要远远低于传感器节点的数量, 这说明较少的 Voronoi 节点可以在给定通信半径  $\gamma$  内覆盖较多的传感器节点。所以 ME 不需要访问每个传感器所在的点, 而只需要访问 Voronoi 映射节点就能达到收集全网数据的目的。

当网络变得稀疏时, MVVSC 需要更多的虚拟散点。

##### 4.2.2 通信半径的影响

在 4.2.1 节中仿真配置的基础上, 将传感器节点数量固定在 50, 网络通信半径  $\gamma$  从 60 m 调整到 150 m, 仿真结果如表 2 所示。

表 2 通信半径对 MVVSC 算法的影响

$\gamma$	$vsNum$	$vvNum$	$\gamma$	$vsNum$	$vvNum$
60	32.0	31.3	110	3.8	18.8
70	21.7	28.0	120	2.5	17.5
80	13.9	25.5	130	1.3	16.0
90	9.4	23.6	140	1.1	14.8
100	5.7	20.9	150	0.9	13.3

由于节点数量是固定的, 因此当通信半径增大时, 网络会变得相对密集, 虚拟散点的数量会变少, Voronoi 节点的数量也会随之减少。

#### 4.3 sVDMES算法和PlainMES算法的比较分析

用算法 3 计算 sVDMES 算法的 TSP 路径长度, 同时用相似的算法为 PlainMES 算法计算 TSP 路径长度 (传感器节点构成的散点集的 TSP 路径长度)。

图 4 显示了 sVDMES 算法在缩短路径长度这一指标上的优越性能, 在 3.2.1 节的仿真配置下, sVDMES 的平均路径长度最多可以缩短到 PlainMES 算法的 60% 以下, 如图 5 所示。

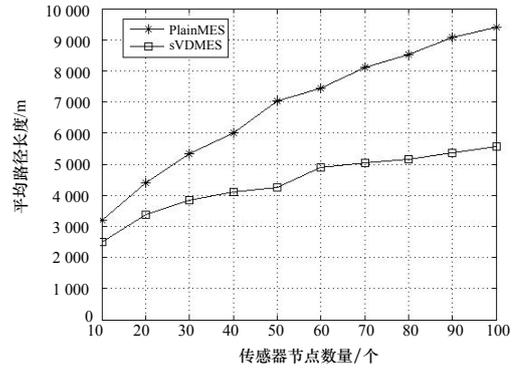


图 4 在不同传感器节点数量下的平均路径长度

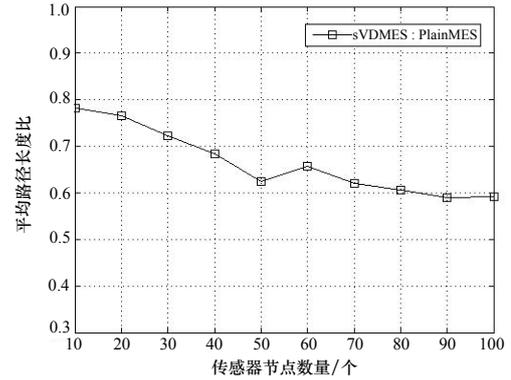


图 5 sVDMES 算法和 PlainMES 算法的平均路径长度比率

图 6 显示了在 4.2.2 节的仿真配置下, VDMES 算法和 PlainMES 算法的平均路径长度。在 PlainMES 算法中, 因为 ME 是被调度去访问每个传感器所在的点, 所以通信半径的增减不会改变 PlainMES 算法中的平均路径长度。在 sVDMES 算法中, 当通信半径增加时, 更少的 Voronoi 节点就可以覆盖传感器节点所形成的散点集, 因此平均 TSP 路径长度会随之减小。

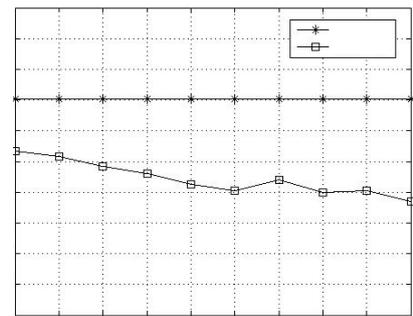


图 6 在不同通信半径下的平均路径长度

## 5 结束语

本文提出一个基于 Voronoi 图的移动单元节点调度算法, 用来在稀疏传感器网络中为采集数据的移动单元节点构建一条尽可能短的路径。该算法通过迭代虚拟散点插入过程生成一个 Voronoi 映射节点集合, 在给定通信半径范围内, 这个集合恰好覆盖传感器节点形成的散点集。贪婪 Voronoi 节点筛选技术使得到的 Voronoi 映射节点集尽可能小, 保证了 sVDMES 算法在缩短路径长度方面具有良好的性能。

(下转第 96 页)