

基于动态规划的矿井无线再编程最优基站选址算法

韩江洪^{1,2}, 段章领¹, 卫星^{1,2}, 陆阳^{1,2}, 张建军^{1,2}

(1.合肥工业大学计算机与信息学院, 安徽 合肥 230009; 2.安全关键工业测控技术教育部工程研究中心, 安徽 合肥 230009)

摘要: 广泛应用于安全关键系统的大规模无线网络需动态更新节点程序, 采用无线再编程方式进行程序更新, 具有高效性及灵活性。研究在节点均匀分布的矿井无线传感器网络环境下, 如何选取最优的基站使整个网络再编程代价最小。首先由网络节点程序更新的接力覆盖过程推导出基站选址模型, 其次将其转化为求解连通简单图任意节点不循环最长最短通路的最小值问题, 最后给出基于动态规划的最优算法, 包括最优子结构和重叠子问题。数值结果表明, 与其他最优算法相比, 该算法在网络复杂度急剧增大时仍保持平稳的时间消耗。

关键词: 无线再编程; 基站选址; 动态规划; 矿井无线传感器网络

中图分类号: TP301

文献标识码: A

Optimal base station selection based on dynamic programming for reprogramming in mine wireless network

HAN Jiang-hong^{1,2}, DUAN Zhang-ling¹, WEI Xing^{1,2}, LU Yang^{1,2}, ZHANG Jian-jun^{1,2}

(1. School of Computer and Information, Hefei University of Technology, Hefei 230009, China;

2. Engineering Research Center of Safety-Critical Industry Measure and Control Technology of Ministry of Education, Hefei 230009, China)

Abstract: The nodes of large-scale wireless network which widely used in safety-critical systems often update new code, the way of wireless reprogramming is efficiency and flexibility. The problem of selecting the optimal base station when nodes uniformly distributed in mine wireless network environment and all other nodes were reprogrammed efficiency. Firstly, the model of base station selecting was established by covering process of reprogramming. Then the problem was transformed into solving the minimum of longest non-cyclic path between node and base station in undirected circulant graph. Finally, the optimal algorithm based on dynamic programming which included optimal substructure and overlapping subproblems was proposed. Numerical analysis and experimental results show that the time consumption of proposed algorithm is more stable than other optimal algorithm when the complexity of network is increased sharply.

Key words: wireless reprogramming, base station location, dynamic programming, mine wireless sensor network

1 引言

近些年, 大型矿难事故频发造成了重大的生命财产损失, 井下生产运输越来越向着无人化、智能化的方向发展。矿井无线传感器网络能够全天候、不间断地监测井下生产环境并将数据实时传输至地面监控中心, 对矿井下的安全生产起到了至关重要的作用。但井下环境复杂多变, 需定期对节点的

软件进行更新维护以满足地面监控中心的监控需求, 矿井无线传感器网络因节点数目较多、分布范围较广且井下环境恶劣等原因无法以传统人工手动方式进行软件更新, 只能采用无线再编程^[1,2]的方式进行软件更新。无线再编程是指节点因环境变更以及功能更新等原因, 采用无线的方式对所有节点进行内部程序更新的过程^[3,4]。在多跳无线网络中, 再编程代码由基站发送给周围节点, 再由周围节点

收稿日期: 2016-05-29; 修回日期: 2017-01-04

基金项目: 国家重点研发计划专项基金资助项目 (No.2016YFC0801804); 国家自然科学基金资助项目 (No.61370088); 国家国际科技合作专项基金资助项目 (No.2014DFB10060)

Foundation Items: The National Key Research Development Program of China (No.2016YFC0801804), The National Natural Science Foundation of China (No.61370088), The National Science and Technology Cooperation Project (No.2014DFB10060)

依次向其他节点传播。现有再编程传播协议大都针对二维无约束平面网络,但矿井无线传感器网络拓扑受巷道结构限制较大,其节点传播通路完全取决于巷道方向^[5],使传统的再编程路由策略及基站选择策略都不再适用。此外,软件更新代码量比普通监控数据要大,再编程一般采用长码的方式进行广播传输占用较大的带宽资源,若再编程时间过长影响节点监控数据的通信质量,会对矿井生产安全造成一定影响。因此,针对矿井无线传感器网络等安全关键检测领域研究一种快速而高效的无线再编程机制成为一个新的挑战。

目前,已有的无线再编程的高层协议研究集中于以能量均衡为目的的传输路由选择,如 MNP (multihop network reprogramming) 协议^[6]的路由选择策略是相同通信半径下覆盖需要更新节点数目最多的节点作为转发节点,从而提高能量利用效率。DTPC(dynamic transmission power control) 算法^[7]以贪婪思想选择转发节点,虽然加入了能量均衡策略但无法达到网络寿命的全局最优,ACDP(adaptive code dissemination protocol) 是一种自适应的代码分发协议^[8],基本思想是通过选择剩余能量较多的节点作为路由转发节点以均衡节点的负载,达到最大化网络生存周期的目的。上述再编程路由协议均针对的是二维无约束网络,在路由受井下巷道约束的矿井无线传感器网络中失去效率,再编程是一种下行的数据通信方式在路由限制的情况下,基站作为再编程代码传输的初始点,其位置的选取对再编程的总代价有着重要的影响,因此,如何选择最佳的基站位置对降低整个再编程过程的时间及能量开销具有重要意义。

在目前的基站选址领域,二维无约束拓扑网络下的基站选址问题是近年来无线网络基站选址的研究热点,其被证明是混合整数非线性规划 NP-hard 问题^[9]。文献[10,11]提出在节点数据流量较为均衡的无线传感器网络中,所有节点的数据流量被汇集到基站,此时选择网络拓扑结构的重心作为基站的佳位置,但是该方法只适用于节点数据流量均衡的典型无约束无线传感器网络;ABS(anycast base-station selection) 算法^[12]是一种针对普通最大化网络寿命的启发式基站选址算法,得到的基站位置为近似最优。OBSP(optimal base station placement) 算法^[13]以节点为中心运用不同半径的圆对网络区域进行划分,最终给出基站的最优选址区域,该算

法虽然降低了选址的复杂度,但只能近似最优解。面向多层、移动、多基站等特殊无线网络基站选址应用场合,文献[14]提出了一种针对双层网络拓扑的基站最优选址算法,使网络的生存寿命最长。文献[15,16]提出了一种可移动基站的佳选址算法。文献[17]提出了将网络划分为不同的子网,每个子网再独立进行最优选址的多基站的佳选址算法。文献[18]提出了将网络最小包围圆的圆心作为佳位置的基站选址算法。文献[19]提出了一种给定拓扑结构下的逐一分解编程(LSP, lexicographical stratified programming) 模型,再基于完整的分层序列和网格搜索的方法来获得基站佳可能位置,该方法只能获得基站的佳区域,无法获得准确位置,上述基站选址研究的都是节点路由方向不受约束且数据只进行上行传输的网络拓扑,目前,针对节点路由受方向限制且数据下行传播的无线网络拓扑的基站选址研究较少。

本文研究在节点路由受限且数据以下行方式传播的矿井无线再编程过程中,如何选取最优的再编程基站使其整体时间代价最小,并进一步将该问题转化为求解无向循环图任意节点不循环最长通路的最小值模型。针对该问题传统算法时间复杂度过高、较大网络规模下无法实施,本文利用链表在数据搜索中的优势,将已确定的网络拓扑以链表的形式进行存储,同时,运用动态规划^[20]将任意节点不循环最长通路的问题分解为若干相同的重叠子问题,进一步确定求解子问题的最优递归子结构,再结合节点等距离分布的特点,自底向上递归求解各个子问题,最终给出了一种新的基于动态规划的再编程基站选址(OBSS, optimal base-station selection) 算法。

2 问题模型及分析

2.1 矿井无线再编程过程

矿井无线再编程过程如图 1 所示。

1) 在给定的矿井拓扑区域内等间距^[18]分布若干同构 AP 节点,节点间以无线多跳的方式传输代码,待再编程代码接收完毕,系统软重启后再继续任务,任意一个 AP 节点所在位置都能够成为基站的候选位置。

2) 选取某一个合适的 AP 节点位置作为基站所在位置,则该 AP 节点第一个拥有更新代码且作为整个再编程过程的发起者。

3) 基站确定以后,选择合适的通信半径将更新代码传输到其相邻节点,基站相邻节点收到更新代

码后再以相同的方式将更新代码传播给相邻节点，如此以接力的方式一轮一轮地进行再编程，直到所有节点都已经收到更新代码。

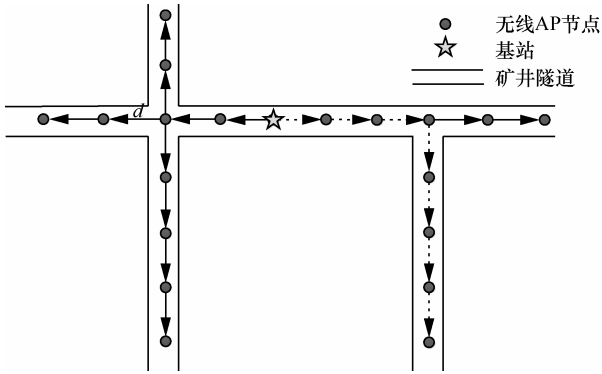


图 1 矿井无线再编程过程

根据上述再编程过程，可以看出在给定矿井拓扑及 AP 节点分布的情况下，每一轮的节点更新在软重启过程中都伴随一定的时间消耗，因此，整个再编程时间代价则取决于初始发送节点到最后一轮节点更新所需的轮数（如图 1 中虚线箭头所示），而初始发送节点的选择由基站所在位置决定，最终可以得到基站的选址直接决定着整个矿井再编程的时间消耗。

2.2 选址模型

把每个矿井 AP 节点视为顶点，相邻 AP 节点的无线连接关系视为边，各边的权相同且为单位 1；由于不存在平行回路（任意 2 个相邻顶点之间存在多条边）和环（任意顶点无自身到自身的边），矿井 AP 节点拓扑可用连通的简单图 $G(V, A)$ 表示，其中， V 表示顶点集合， A 表示顶点的邻接矩阵，即

$$A = (a_{v_i, v_j})_{n \times n} \quad (1)$$

其中，若顶点 v_i 与顶点 v_j 间有边直接相连，则 $a_{v_i, v_j} = 1$ ，若顶点 v_i 与顶点 v_j 间无边直接相连，则 $a_{v_i, v_j} = 0$ 。

定义 1 定义 $w_{(i,j)}^{k_{i,j}}$ 为图 G 中从起点为 v_i ，终点为 v_j 的且长度是 $k_{i,j}$ 的最短通路集合，计算式为

$$w_{(i,j)}^{k_{i,j}} = v_{(i,j)}^0 v_{(i,j)}^1 \cdots v_{(i,j)}^{m-1} v_{(i,j)}^m, 1 \leq m \leq n-1 \quad (2)$$

其中， $v_{(i,j)}^s \neq v_{(i,j)}^{s+1}$ 且 $a_{v_{(i,j)}^s, v_{(i,j)}^{s+1}} = 1, v_{(i,j)}^0 = v_i, v_{(i,j)}^m = v_j$ ，通路长度 $k_{i,j}$ 的表达式为

$$k_{i,j} = \sum_{i=0}^{m-1} a_{v_{(i,j)}^i, v_{(i,j)}^{i+1}} \quad (3)$$

其中， $\forall a_{v_{(i,j)}^s, v_{(i,j)}^{s+1}} = 1$ 。

令 $W_{(i,j)}$ 表示起点 v_i 到终点 v_j 的所有通路集合

$$W_{(i,j)} = \{w_{(i,j)}^{k_{i,j}^1}, \dots, w_{(i,j)}^{k_{i,j}^h}\}, l \leq h \leq n-1 \quad (4)$$

其中， $w_{(i,j)}^{k_{i,j}^h}$ 表示从起点为 v_i ，终点为 v_j 的长度为 $k_{i,j}^h$ 通路集合， h 表示通路集合 $W_{(i,j)}$ 有 h 种通路长度，由图的连通性知， $\forall i \neq j$ 且 $v_i, v_j \in V, W_{(i,j)} \neq \emptyset$ 。

令 $d_{i,j}$ 表示起点为 v_i ，终点为 v_j 的最短通路长度，即

$$d_{i,j} = \min_{i=1}^h k_{i,j}^i, w_{(i,j)}^{k_{i,j}^i} \in W_{(i,j)} \quad (5)$$

令 $R(i, j)$ 表示起点为 v_i ，终点为 v_j 的最短通路集合，即

$$R(i, j) = w_{(i,j)}^{d_{i,j}}, w_{(i,j)}^{d_{i,j}} \in W_{(i,j)} \quad (6)$$

定义 2 v_{j^*} 代表图 G 中起点为 v_i ，使最短通路长度 $d_{i,j}$ 最长的顶点，即

$$v_{j^*} = \arg \max \{d_{i,j} \mid j \in V - \{i\}\} \quad (7)$$

定理 1 起点 v_i 确定，若存在其所对应的最短通路长度 $d_{i,j}$ ，则对任意最短通路长度 $d_{i,g}$ ， $g \in V - \{v_i, v_{j^*}\}$ 有 $d_{i,g} \leq d_{i,j}$ ，证明见附录。

根据定理 1 及上述定义，结合选址问题，要从所有顶点位置中选出基站 v_{i^*} ，使其相对应的最短通路长度最短，即

$$v_{i^*} = \arg \min \{d_{i,j^*} \mid i \in V\} \quad (8)$$

最终整个问题的优化表达式为

$$\min_{v_i \in V} d_{i,j^*}$$

s.t. 式(2)~式(7)

2.3 模型分析

上述的选址模型是一个典型的最小最大问题的优化模型，要得到问题的最小值 d_{i^*, j^*} ，需要先得到每个顶点作为基站时的最远通路长度 d_{i, j^*} ；而要得到 d_{i, j^*} 必须求出整个网络图 G 的距离矩阵 $D = (d_{i,j})_{n \times n}$ ， d_{i, j^*} 即为距离矩阵 D 中每行或每列中所对应的最大元素。

3 算法设计与分析

经过上述模型分析，将问题转化为求解已知网络拓扑结构下的距离矩阵 $D = (d_{i,j})_{n \times n}$ 的问题，现有

求解距离矩阵的算法有 Dijkstra 算法和 Floyd-Warshall 算法, 但是上述算法求解距离矩阵的时间复杂度较高^[21], 本文结合井下网络拓扑的特点, 提出了一种基于多层链表结构的动态规划距离矩阵的求解算法。

3.1 算法设计

3.1.1 递归解结构

图 G 的拓扑结构确定后, 其距离矩阵也随之确定, 如图 2 所示, 节点之间的最短通路长度 $d_{i,j}$ 即为距离矩阵中的距离元素。由最短通路的性质得到定理 2。

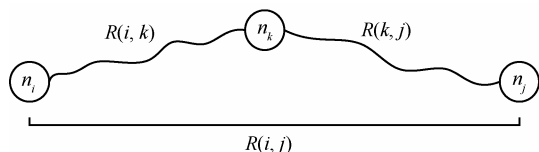


图 2 节点 n_i 与 n_j 最短通路

定理 2 若 $R(i, j)$ 为节点 n_i 与 n_j 之间的最短通路, 节点 n_k 为通路 $R(i, j)$ 上除 n_i 与 n_j 之外的节点, 将 $R(i, j)$ 以 n_k 为分界分成 $R(i, k)$ 和 $R(k, j)$, 则 $R(i, k)$ 为节点 n_i 与 n_k 之间的最短通路, $R(k, j)$ 为节点 n_k 与 n_j 之间的最短通路, 证明见附录。

由定理 2 可以得到, 若 $R(i, j)$ 为节点 n_i 与 n_j 之间的最短通路, 节点 n_k 为通路 $R(i, j)$ 上除 n_i 与 n_j 之外的节点, 则有

$$d_{i,j} = d_{i,k} + d_{k,j} \quad (9)$$

由定理 2 可知, 可以将求解一条最短通路长度的问题分解为求该通路上若干个子最短通路长度的问题, 可以将式(9)改写为

$$d_{i,j}^{(k)} = d_{i,k}^{(k-1)} + d_{k,j}^{(k-1)} \quad (10)$$

该问题的初始解即为当子问题最后分解到 2 个相邻节点, 根据模型条件可知相邻节点之间的距离为 1, 因此可以得到完整的递归表达式

$$d_{i,j}^{(k)} = \begin{cases} 1, & k = 0 \\ d_{i,k}^{(k-1)} + d_{k,j}^{(k-1)}, & k \geq 1 \end{cases} \quad (11)$$

但是图 G 中可能会出现如图 3 所示的环路情况, 此时 2 个节点之间会出现多条通路。

多条通路就会造成递归式(11)中 $d_{i,j}^{(k)}$ 出现多个值的情况, 选取最小值作为最终的通路长度, 因此, 要对式(11)做出相应修正。

$$d_{i,j}^{(k)} = \begin{cases} 1, & k = 0 \\ \min(d_{i,k}^{(k-1)} + d_{k,j}^{(k-1)}), & k \geq 1 \end{cases} \quad (12)$$

式(12)即为距离矩阵求解的递归表达式。

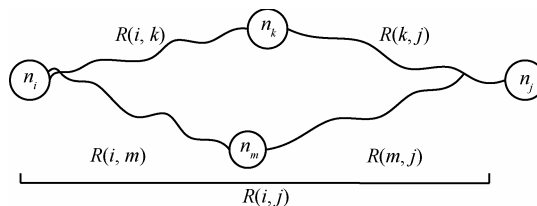


图 3 节点间存在多条通路

3.1.2 最优求解过程

距离矩阵确定以后, 需要从初始条件开始对其自底向上进行递归求解, 传统算法 (如 Floyd 算法) 从初始邻接矩阵开始进行递归迭代求解, 但是节点多通路长度之间的比较工作增加了算法的复杂度, OBSS 算法为了降低多通路长度比较的复杂度, 首先引进了链表结构, 如图 4 所示。

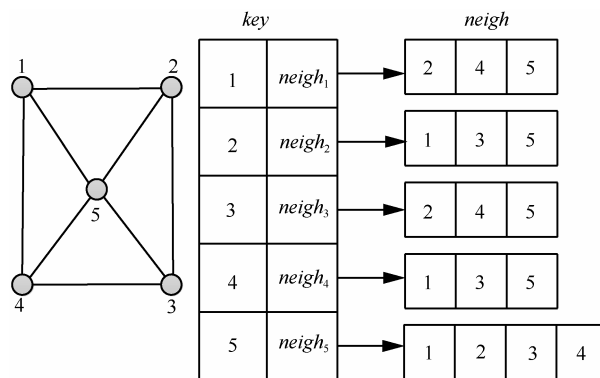


图 4 链表结构

图 4 中链表结构主要由 2 部分数组组成, 第一部分数组主要保存节点的 ID 号及其邻居节点所在数组的指针 $neigh_i$, 第二部分是每个节点其邻居节点的数组 $neigh[i]$ 。

根据上述数据结构对距离矩阵进行自底向上的求解, 为了避免循环回路造成的多通路长度比较问题, 引入轮次 r , 即一轮一轮计算距离矩阵中的元素值, 第一轮计算相邻节点之间的距离, 当 $r=1$ 时, 显然有 $d_{i,neigh_i[j]} = 1$, 其中, $neigh_i[j]$ 表示与节点 i 相邻的节点, 当 $r=2$ 时, 有 $d_{i,neigh_{neigh_i[j]}} = 1$, 为了便于表达, 将 $neigh_{neigh_i[j]}$ 改写成 $neigh_i^{(2)}[j]$, 用来表示节点 i 相邻节点的相邻节点, 以此类推, 当 $r=k$ 时, 则有

$$d_{i,neigh_i^{(k)}[j]} = k \quad (13)$$

为了解决节点间多通路的比较问题，首先将初始矩阵 $A' = (a_{i,j})_{n \times n}$ 置为零矩阵，规定当 $i = j$ 时， $a_{i,j} = 0$ ，依照式(13)计算各元素的值，在将计算出的元素 $d_{i,j}$ 写入距离矩阵之前先判断相对应的元素 $a_{i,j}$ 是否为 0，若 $a_{i,j} = 0$ ，则 $a_{i,j} = d_{i,j}$ ，若 $a_{i,j} \neq 0$ ，则 $a_{i,j}$ 保持当前数值。

当出现上述的两点之间的多通路情况，如图 5 所示，节点 n_i 、 n_j 和 n_k 组成一条回路，且 $d_{i,j} = d_{i,k} = k$ ，当 $r = k$ 时，可以得到 $d_{i,j} = d_{i,k} = r$ ，并将其写入到距离矩阵中，若此时 $a_{i,j} = 0$ ，则 $a_{i,j} = k$ ，当 $r = k + 1$ 轮时，由于节点 n_j 是 n_i 的邻居节点，由式(13)可以得到 $d_{i,j} = k + 1$ ，但是此时 $a_{i,j} = k$ ，因此，保持当前数值，从而避免了多通路长度之间的比较问题。

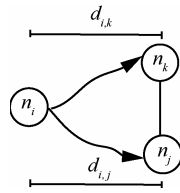


图 5 n_i 和 n_j 之间存在回路的情况

3.1.3 基站选取算法

结合式(13)和图 4 的链表结构，给出完整的距离矩阵求解算法步骤。

1) 给出所有元素为 0 的距离矩阵 $A' = (a_{i,j})_{n \times n}$ ，规定当 $i = j$ 时， $a_{i,j} = 0$ ，不能被更改。

2) 轮次 $r = r + 1$ 依次递增，按照式(13)依次计算各节点最短通路距离 $d_{i,j}$ 的值，并依照如下的语句进行写入条件判断。

```
function Judgment (n,n,k) {
for r=1 to n do //轮数依次增加
for i=1 to n do
for j=1 to k do
if  $neigh_i^{(r)}[j] \neq i$  &&  $a_{i,neigh_i^{(r)}[j]} == 0$ 
then
 $a_{i,neigh_i^{(r)}[j]} = r$ ; //对其进行赋值
end
end
end }
```

3) 直到整个初始矩阵 $A = (a_{i,j})_{n \times n}$ 除对角线之外的所有元素被写入，此时的矩阵即为距离矩阵

$D_{n \times n}$ 。

4) 由距离矩阵 $D_{n \times n}$ 筛选出每行的最大值 d_{i,j^*} ，再从 d_{i,j^*} 选出最小的作为基站的最佳位置。

整个 OBSS 算法的伪代码如算法 1 所示。

算法 1 最佳基站选择算法

输入 $A = (a_{i,j})_{n \times n}$ //初始距离矩阵;

L //图 G 的链表;

输出 i^* //最终的基站位置;

初始化 r //算法的轮次;

$d_{i,j}$ // n_i 到 n_j 的最短通路距离;

$S[i] = 0$ //存放 d_{i,j^*} 的数组;

function Judgment (n,n,k) //得到距离矩阵;

for $i = 1$ to n do

for $j = 1$ to n do

if $S[i] > a_{i,j}$ then

$S[i] = a_{i,j}$; //筛选每一行的最大值

end

end

$i^* \leftarrow 1, S_{\min} = S[1]$;

for $i = 2$ to n do

if $S[i] \leq S_{\min}$ then

$S_{\min} = S[i], i^* \leftarrow i$; //选出最佳基站

end

return i^*

3.2 算法分析

在 OBSS 算法中，第一轮当 $r = 1$ 时，对节点 n_i 来说相邻节点距离计算的复杂度与其相邻节点的数目相等 $|neigh_i[j]|$ ；由于矿井拓扑结构的限制，每个节点相邻节点数目上限是一个很小的常数 k ，则有 $|neigh_i[j]| \leq k$ ，经过 n 轮计算后最坏的时间代价为 $n|neigh_i[j]| \leq kn$ ；一个节点的最坏时间代价为 kn ，则对 n 个节点来说其最坏时间代价为 $\sum_{i=1}^n n|neigh_i[j]| \leq kn^2$ ， $S[i]$ 求解算法的最坏复杂度为 $O(n^2)$ ， $S[i]$ 得到后最佳基站的选取复杂度为 $O(n)$ ，因而 OBSS 的算法最坏的复杂度为 $O(kn^2 + n^2 + n)$ ，由于 k 是不随 n 规模增大而变化的常数（井下大巷途经同一路口的岔道数量一般小于 8），由此得出 OBSS 的算法复杂度为 $O(n^2)$ 。

若使用 Dijkstra 算法进行求解须求得任意 2 个节点之间的最短距离，Dijkstra 本身的算法复杂度为

$O(n^2)$ ，任意 2 个节点的选择数目为 $C_n^2 = \frac{n(n-1)}{2}$ ，其复杂度也为 $O(n^2)$ ，如果采用该种方式来求解，其算法复杂度可表述为 $O(\frac{n(n-1)n^2}{2})$ ，最终的算法复杂度为 $O(n^4)$ ，这样的算法复杂度在网络规模较大节点数目较多的情况下难以适用。

Floyd-Warshall 算法也是采用动态规划的思想避免了 Dijkstra 中节点间距离的重复计算，但是 Floyd-Warshall 算法在处理节点间多通路问题时，采用逐个比较的方法造成了其算法复杂度需要 3 层循环，每层的算法复杂度为 $O(n)$ ，所以，其求解距离矩阵的算法复杂度为 $O(n^3)$ ，求解 $S[i]$ 的过程也可以嵌在距离矩阵的求解过程中，从 $S[i]$ 中选取其中最小的 i^* ，其时间复杂度为 n ，因而算法复杂度可表示为 $O(n^3 + n)$ ，最终整个算法的时间复杂度为 $O(n^3)$ 。

4 仿真分析

本文主要针对算法的性能进行仿真测试，因此，选择了对数据具有极强的处理能力的 Matlab，本文使用的版本是 Matlab 2013b，CPU 采用的是 Intel 酷睿 i3-390M 处理器，主频 2.6 GHz，首先依照上述的场景需求进行仿真参数的设定，假定 AP 节点等距离分布在 4 km×5 km 的仿真区域内，节点通信半径确定，其仿真参数如表 1 所示。

参数	取值范围	取值依据
网络覆盖区域	4 km×5 km	参考中型矿井的规模
节点间距	300 m	参考井下 AP 间距
通信半径	300 m	参考井下 AP 通信距离
节点数目	10~200	视具体网络规模而定
可选最佳基站位置	1~200	视具体拓扑而定
最终基站位置	1	从可用基站中任选一个

仿真参数确定以后，列举了不同拓扑复杂度的网络拓扑结构，并分别运行基于 Dijkstra 算法和 Floyd-Warshall 算法与 OBSS 算法对不同结构的网路进行最佳基站位置的计算，然后对 3 种算法进行性能分析比较。

如图 6~图 8 所示，随着节点数目以及网络拓扑复杂度的逐步提高，3 种算法在时间性能上有不同体现。图 6(a)所示的 58 个节点组成只有简单的 2 个环路，从图 6(b)可以看出，3 种算法的时间消

耗差别不大，都在较短的时间内计算出最佳基站可选的位置。如图 7(a)所示，当网络中的节点数目增大

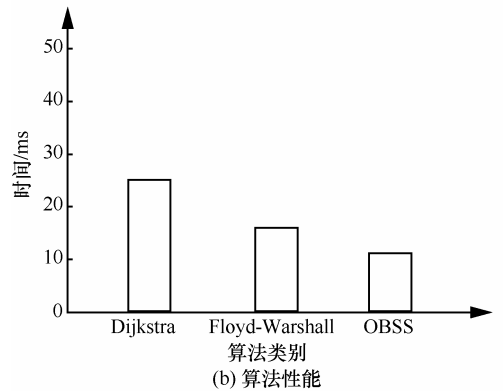
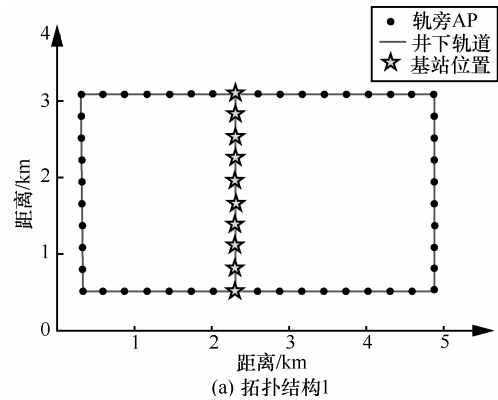


图 6 拓扑结构 1 及算法性能

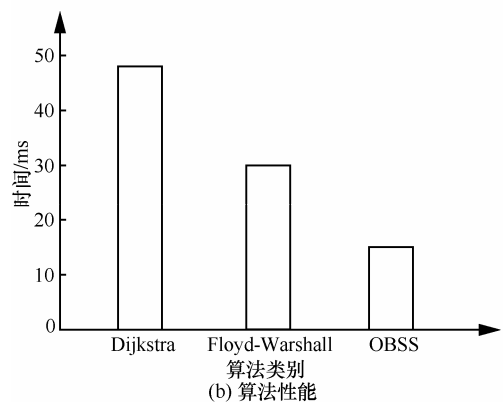
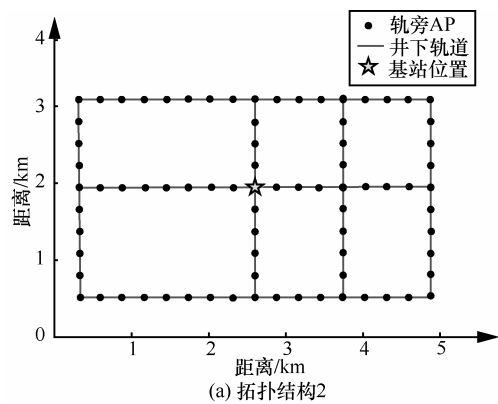


图 7 拓扑结构 2 及算法性能

到 96 个且环路数目逐渐增大时,从图 7(b)可以看出, Dijkstra 算法时间消耗较图 6(b)增大了近一倍, Floyd-Warshall 算法也增大了近一倍, OBSS 算法仅增加了不到 30%。如图 8(a)所示,当节点数目增加到 131 个且结构更加复杂时,从图 8(b)可以看出, Dijkstra 算法时间消耗较图 6(b)增大了近 10 倍, Floyd-Warshall 算法也增大了近 6 倍, OBSS 算法增大了 3 倍多。

经过与另外 2 种典型的算法比较可以看出, OBSS 算法在面对网络复杂度依次增大的情况下,在时间消耗方面具备较为稳定的性能。

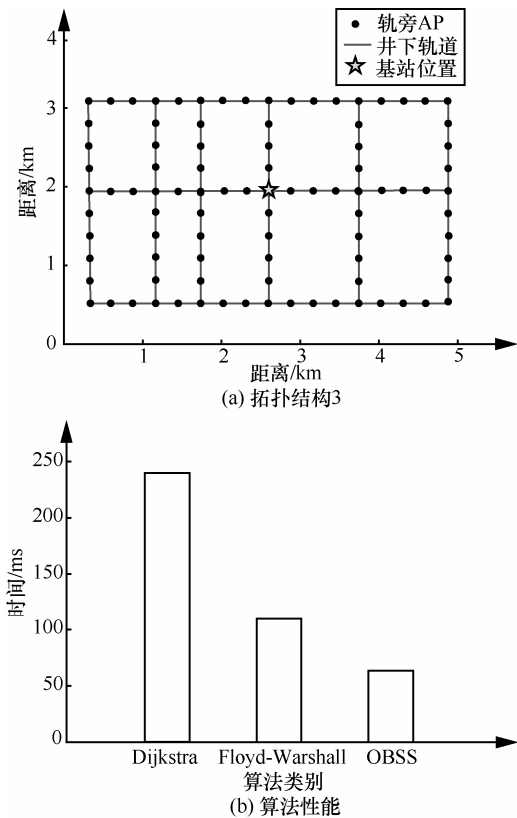


图 8 拓扑结构 3 及算法性能

从上述仿真结果可以看出, OBSS 算法在处理等间距分布的特殊网络下具有优越的稳定性和高效性。但上述结果只能反映不同节点、不同复杂度对算法性能的影响。为了更全面地验证 OBSS 算法面对不同等级复杂结构等间距分布网络的稳定性,将网络中所有节点的度之和除以顶点数来表示网络的复杂度,则有

$$\theta = \frac{\sum_{i=1}^n b_i}{n}, \theta \geq 1 \tag{14}$$

其中, b_i 表示节点 n_i 的度, θ 表示网络的复杂度。

选择 100 个节点,通过调整 θ 来比较节点数目相同的情况下,不同网络复杂度对各算法性能的影响,结果如图 9 所示。

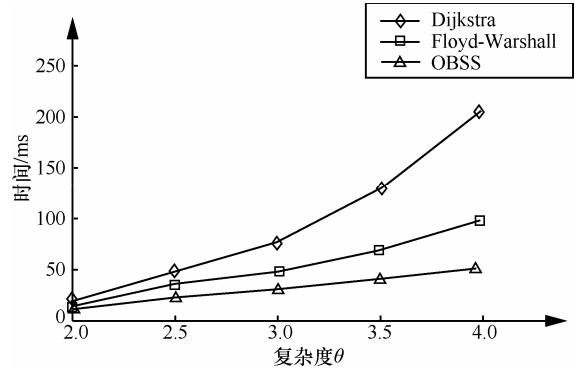


图 9 100 个节点不同网络复杂度下算法性能比较

为了验证算法在实际应用场景中是否适用,本文从实际矿井隧道地图上截取了 $8\text{ km} \times 11\text{ km}$ 的矩形区域的 205 个 AP 节点,如图 10 所示,其拓扑结构不规则且有较多的循环回路和不循环支路,能够较好地验证 3 种算法的性能,3 种算法在该拓扑下的时间性能如图 11 所示。

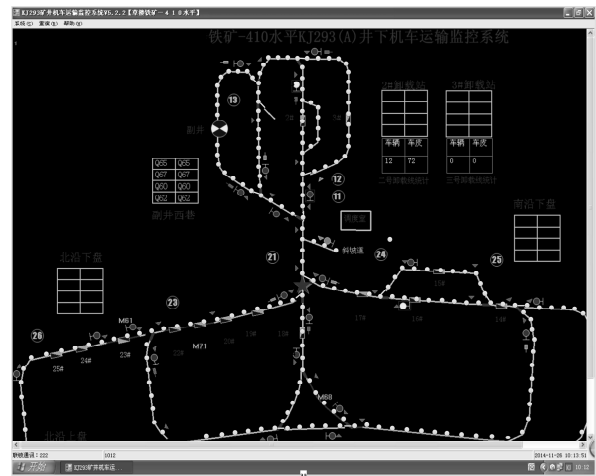


图 10 真实矿井结构下的算法应用

由图 11 中的数据可以看出, Dijkstra 算法的时间消耗较图 8(b)增大了近 12 倍, Floyd-Warshall 算法也增大了 9 倍, OBSS 的时间代价只增大了 4 倍且总时间消耗远低于前 2 种算法,结合之前所描述的 3 种算法的特性分析可知, Dijkstra 算法和 Floyd-Warshall 算法在判断循环回路中的最短路径时会消耗大量时间,矿井结构中一般循环回路较多会对其性能产生较大影响,但 OBSS 算法因为其链

表结构的优势并结合动态规划的思想能够在判断循环回路时节省大量时间,因此,无论是规则还是不规则的拓扑结构,OBSS 算法都能保证其较好的时间性能。

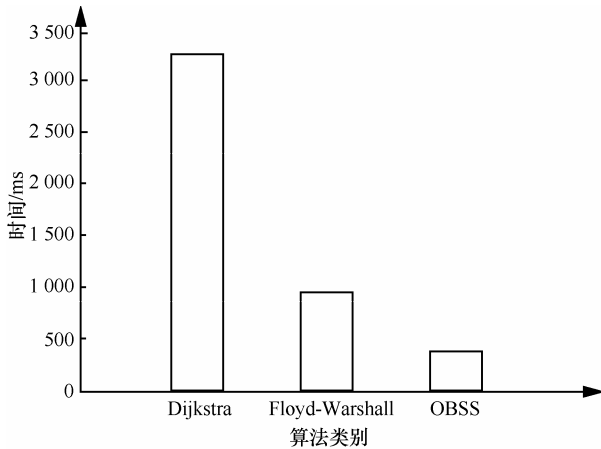


图 11 真实矿井结构下的 3 种算法的时间性能比较

经过上述的仿真测试以及在实际的矿井拓扑中的测试结果,表明 OBSS 算法对节点等距离分布这一特殊网络表现出较好的时间性能。

5 结束语

本文对矿井网络无线再编程基站选址问题进行研究,将其转化为求解连通简单图任意节点不循环最长最短通路的最小值问题,建立了最小最大模型,提出了一种新的基于动态规划的最佳基站选择算法。

经过理论分析和仿真验证,结果表明 OBSS 能够在较短的时间消耗下,准确定位最佳基站的位置,并且经过比较其时间消耗和稳定性优于 Floyd-Warshall 算法和 Dijkstra 算法。

下一步工作将研究在大型网络规模,一个基站无法满足时间性能的条件下,如何选择多个基站的位置使整个网络的无线再编程时间最短。

附录 定理 1 和定理 2 的证明

定理 1 证明 假设存在节点 $g \in V - \{v_i, v_{j^*}\}$ 及最短通路长度 $d_{i,g}$ 使 $d_{i,g} > d_{i,j^*}$, 则根据定义 2, 节点 g 即为使 v_i 最短通路长度最长的节点,也即 $g = j^*$, 与假设矛盾,定理得证。

定理 2 证明 假设 $R'(i,k)$ 为节点 n_i 与 n_k 之间比 $R(i,k)$ 更短的通路,则 $|R'(i,k)| < |R(i,k)|$, 那么将通路 $R'(i,k)$ 与通路 $R(k,j)$ 进行首尾拼接组成新的通路 $R'(i,j)$,

则 $|R'(i,j)| = |R'(i,k)| + |R(k,j)|$, 而 n_i 与 n_j 之间的最短通路 $|R(i,j)| = |R(i,k)| + |R(k,j)|$, 由 $|R'(i,k)| < |R(i,k)|$ 得到 $|R'(i,j)| < |R(i,j)|$, 与前提条件矛盾,因此, $R(i,k)$ 为节点 n_i 与 n_k 之间的最短通路。同理可证 $R(k,j)$ 为节点 n_k 与 n_j 之间的最短通路,则定理得证。

参考文献:

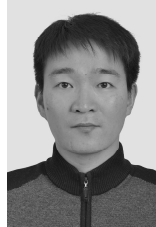
- [1] NANDU P, SHEKOKAR N. An enhanced authentication mechanism to secure re-programming in WSN[C]//International Conference on Advanced Computing Technologies and Applications. Mumbai, INDIA, 2015: 397-406.
- [2] SHIM K A, KYUNG-AH. (SDRP)-D-2: Secure implementations of distributed reprogramming protocol for wireless sensor networks[J]. Ad Hoc Networks, 2014, 19(8):1-8.
- [3] 卫星, 张芳荣, 段章领, 等. 基于模拟退火的无线传感节点再编程策略[J]. 仪器仪表学报, 2015, 36(3):538-544.
WEI X, ZHANG F R, DUAN Z L, et al. Simulated annealing based reprogramming scheme of wireless sensor nodes[J]. Chinese Journal of Scientific Instrument, 2015, 36(3):538-544.
- [4] 方效林, 高宏, 李建中. 一种基站可移动传感器网络再编程协议[J]. 电子学报, 2013, 41(5): 1007-1011.
FANG X L, GAO H, LI J Z. MovPro: data dissemination for reprogramming in wireless sensor network with mobile sink[J]. Acta Electronica Sinica, 2013, 41(5): 1007-1011.
- [5] 王珂. 矿井无线传感器网络节点部署关键技术研究[D]. 中国矿业大学, 2011:49-62.
WANG K. Research on key technologies of mine wireless sensor networks node deployment[D]. China University of Mining and Technology, 2011:49-62.
- [6] PRADIP D, LIU Y H, SAJAL K D. Energy-efficient reprogramming of a swarm of mobile sensors [J]. IEEE Transaction on Mobile Computing, 2010, 9(5): 703-718.
- [7] SEUNGKU K, DOO S E. Dynamic transmission power control for wireless sensor network reprogramming[C]//The 17th Asia-Pacific Conference on Communications (APCC). 2011:145-150.
- [8] DONG C C, YU F Q. An efficient network reprogramming protocol for wireless sensor networks[J]. Computer Communications, 2015, 55(10): 41-50.
- [9] GAREY M R, JOHNSON D S. Computers and intractability: a guide to the theory of NP-completeness[M]. New York: Freeman, 1979: 245-248.
- [10] J LUO, J P HUBAUX. Joint mobility and routing for lifetime elongation in wireless sensor networks[C]//The 24th Annual Joint Conference on Computer and Communications Societies (INFOCOM). 2005: 1735-1746.
- [11] WANG J, URRIZA P, HAN Y, et al. Weighted centroid localization algorithm: theoretical analysis and distributed implementation[J]. IEEE Transaction on Wireless Communication, 2011, 10(10): 3403-3413.
- [12] HOU Y T, SHI Y, SHERALI H D. Optimal base station selection for anycast routing in wireless sensor networks[J]. IEEE Transactions on Vehicular Technology, 2006, 55(3): 813-821.
- [13] SHI Y, HOU Y T. Optimal base station placement in wireless sensor network[J]. ACM Transactions on Sensor Networks, 2009, 5(4): 2855-2869.
- [14] JIAN P P, LIN C Y, HOU T. Optimal base-station locations in two-tiered wireless sensor networks[J]. IEEE Transactions on Mobile

Computing,2005,4(5):458-473.

- [15] LIANG W, LUO J, XU X. Prolonging network lifetime via a controlled mobile sink in wireless sensor networks[C]//GLOCOM. Miami Florida,USA,2010:1-6.
- [16] THANIGAIVELU K, MURUGAN K. Impact of sink mobility on network performance in wireless sensor networks[C]//NETCOM. Chennai ,India,2009:7-11.
- [17] SLAMA I, JOUABER B. ZEGHLACHE D. Energy efficient scheme for large scale wireless sensor networks with multiple sinks[C]//WCNC. Las Vegas, NV, USA, 2008: 2367-2372.
- [18] PAUL B, MATIN M A. Optimal geometrical sink location estimation for two-tiered wireless sensor networks[J]. IET Wireless Sensor Systems, 2011, 1(2): 74-84.
- [19] YAN J, YU K G. Base station placement for wireless sensor network positioning system via lexicographical stratified programming[J]. KSII Transactions on Internet and Information Systems, 2015, 9(11): 4453-4468.
- [20] 张化光, 张欣, 罗艳红. 自适应动态规划综述[J]. 自动化学报, 2013, 39(4): 304-310.
ZHANG H G, ZHANG X, LUO Y H. An overview of research on adaptive dynamic programming[J]. Acta Automatica Sinica, 2013, 39(4): 304-310.
- [21] THOMAS H, CORMEN C E L, RONALD L R. 算法导论[M]. 北京: 机械工业出版社, 2007: 386-390.
THOMAS H, CORMEN C E L, RONALD L R. Introduction to algorithms[M]. Beijing: China Machine Press, 2007: 386-390.



段章领 (1990-), 男, 安徽颍上人, 合肥工业大学博士生, 主要研究方向为无线网络优化、安全关键通信协议。



卫星 (1980-), 男, 安徽肥西人, 合肥工业大学副教授、硕士生导师, 主要研究方向为物联网工程、离散事件动态系统。



陆阳 (1967-), 男, 安徽合肥人, 合肥工业大学教授、博士生导师, 主要研究方向为安全关键工业测控技术、离散事件控制系统理论与应用、自组织网络理论与应用、二进神经网络理论与应用。

作者简介:



韩江洪 (1954-), 男, 安徽庐江人, 合肥工业大学教授、博士生导师, 主要研究方向为网络与通信、嵌入式系统、无线网络等。



张建军 (1963-), 男, 浙江宁波人, 合肥工业大学教授、硕士生导师, 主要研究方向为汽车电子、物联网工程、计算机集成制造。