

文章编号:1001-9081(2014)06-1558-05

doi:10.11772/j.issn.1001-9081.2014.06.1558

无人机中继链路部署算法设计

方斌^{1*}, 陈特放^{1,2}

(1. 中南大学 交通运输工程学院, 长沙 410075; 2. 中南大学 信息科学与工程学院, 长沙 410075)

(*通信作者电子邮箱 aifudi@163.com)

摘要:在存在通信盲区以及无人机(UAV)使用数目受限等情况下,针对如何合理地部署 UAV 并构建通信中继链路,以及扩展 UAV 的数据传输距离的问题,提出改进 Bellman-Ford(IBF)算法和改进 Dijkstra 算法(IDA)两个 UAV 中继链路部署算法。将通信中继链路中 UAV 部署问题形式化表示为全跳最优路径(AHOP)问题,其中 IBF 算法用来迭代生成一组可达记录集,通过逆向访问记录集可生成一组部署方案;IDA 则在每轮迭代过程中改变连接边的权值,并查找能有效减少中继链路跳数的有效方案,得到 UAV 中继部署问题的可行解。仿真结果表明 IBF 和 IDA 能够提供中继链路部署的有效方案,同时它们的时间性能优于 Bellman-Ford 算法。

关键词:无人机;通信中继;全跳最优路径; Bellman-Ford 算法; Dijkstra 算法

中图分类号: TP273 文献标志码:A

Design of relay link deployment algorithms for unmanned aerial vehicles

FANG Bin^{1*}, CHEN Tefang^{1,2}

(1. School of Traffic and Transportation Engineering, Central South University, Changsha Hunan 410075, China;

2. School of Information Science and Engineering, Central South University, Changsha Hunan 410075, China)

Abstract: To get a reasonable deployment and a communication relay link model of Unmanned Aerial Vehicle (UAV), and extend the data transmission distance, the Improved Bellman-Ford (IBF) algorithm and the Improved Dijkstra Algorithm (IDA) were proposed considering communication blind area and limited number of available UAVs. The UAV deployment problem was modeled as a All Hop Optimal Path (AHOP) problem, in which the IBF algorithm was used to generate a set of reachable records, and the solutions were got by accessing the records reversely; Then the IDA algorithm changed the connection weights of edges in each iteration process and found the path which decreased the hops of relay link, hence the feasible solution of UAV relay deployment problem was got. The simulation analysis illustrates that IBF and IDA can provide effective solutions of relay link deployment, and the time performance of the proposed algorithms are superior to Bellman-Ford (BF) algorithm.

Key words: Unmanned Aerial Vehicle (UAV); communication relay; All Hop Optimal Path (AHOP); Bellman-Ford (BF) algorithm; Dijkstra algorithm

0 引言

随着无人机(Unmanned Aerial Vehicle, UAV)技术的不断发展,其应用领域也越来越广泛,一种典型的应用场景是在执行情报侦察任务时,UAV 通过高带宽通信链路,将侦察得到的数据实时传送至基地,为操作员提供信息支持。

UAV 的任务应用需要视距内通信保障,在视距内通信时,链路带宽以及 UAV 的链路通信能力限制了数据的最大传输距离。一种可行的替代方案是使用多个 UAV 构建一个通信中继链路^[1-5],解决 UAV 链路通信距离受限的约束,如图 1 所示。该方案既能保证侦察 UAV 可以灵活地部署在目标周围,又能够有效地将目标情报数据通过通信中继链路传送到基地。

国外关于使用 UAV 作为通信中继的研究开展得较早:Cerasoli^[1]提出射线跟踪法用于评估使用 UAV 在市区作为通信中继的效率;Pinkney 等^[2]在 1996 年就提出使用 UAV 中继

为战场信息传送服务;Omer 等^[6]研究了 UAV 构建通信中继链路中采用人工势场(Artificial Potential Field, APF)法进行威胁避障问题;Ibrahim 等^[7-8]研究了战场环境中存在威胁障碍的情况下,如何使用 UAV 构建通信中继链路。对于使用多个 UAV 构建通信中继链路,本文关注两个目标:1)如何合理地定位 UAV,即将 UAV 部署在有效的位置,使得通信链路的数据传输质量最高;2)如何降低构建通信中继链路的代价,即采用尽量少的 UAV 完成通信链路的构建。上述两个目标可能无法同时满足,例如减少中继链路中相邻 UAV 的距离会提高链路的数据传输质量,但是会增加 UAV 中继的数量。

通常情况下,构建通信中继链路的 UAV 数目是有限制的,需要在给定 UAV 数量上限的约束条件下,合理地在可通信区域部署无人机中继节点,实现构建最优链路通信质量的目标。通过选择不同数目的 UAV,可构建一系列不同数据传输质量的通信链路。文献[3,9]将通信链路构建为帕累托最优链路,即:对于任何一条帕累托最优通信中继链路,如果不

收稿日期:2013-12-03;修回日期:2014-01-05。

作者简介:方斌(1982-),男,安徽合肥人,博士研究生,主要研究方向:多智能体协同控制、路径规划; 陈特放(1957-),男,湖南涟源人,教授,博士,主要研究方向:智能交通、智能通信网络。

增加中继 UAV 的数目,则无法提高通信链路的数据传输质量(即降低链路的传输代价)。

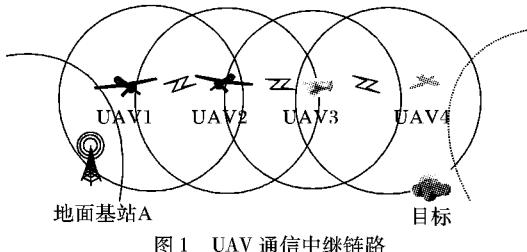


图 1 UAV 通信中继链路

1 UAV 中继定位问题建模

对于中继 UAV 定位问题,给出如下定义:

1) 设已有 UAV 的数目 $|N| \geq 1$, 其中一架 UAV 负责执行情报侦察任务;

2) 中继 UAV 与侦察 UAV 可以安全地部署在区域 $X \subseteq \mathbb{R}^3$, 基地位置 $x_0 \in \mathbb{R}^3 \setminus X$, 侦察目标位置 $x_t \in \mathbb{R}^3 \setminus X$ 。

对于区域 \mathbb{R}^3 中的任意两点 x, x' , 给出两个函数定义:

1) 通信可达性函数 $b(x, x')$, 表示两点之间是否存在通信链路, 返回值为布尔值; 2) 通信代价函数 $c(x, x')$, 表示两点之间建立通信链路需要耗费的代价。对于目标 t , 也有两个函数定义: 1) 侦察可行性函数 $s(x, x_t)$, 表示 x 处的侦察 UAV 是否能够侦察位于 x_t 的目标 t , 其中 $x_t \in \mathbb{R}^3 \setminus X$; 2) 侦察代价函数 $t(x, x_t)$, 表示 UAV 完成对目标 t 侦察任务所要耗费的代价。

设基地 x_0 与目标 x_t 之间存在一条由多架 UAV 组成的长度为 l 的中继链路 c , UAV 的位置是 $\langle x_1, x_2, \dots, x_l \rangle$, 满足 $\{x_1, x_2, \dots, x_l\} \subseteq X, \forall i, 0 \leq i \leq l, b(x_i, x_{i+1})$ 存在并且 $s(x_i, x_t)$ 存在。整条通信中继链的代价定义为: $cost(\langle x_1, x_2, \dots, x_l \rangle) = \sum_{i=0}^{l-1} c(x_i, x_{i+1}) + t(x_l, x_t)$ 。

如果 x_0 与 x_t 之间存在长度为 l 的中继链路 c , 则意味着 x_0 与 x_t 之间不存在中间链路 c' , 其长度 $l' < l$, 并且 $cost(c') < cost(c)$; 否则链路 c 是没有意义的。

中继 UAV 定位问题需要研究的就是对任意长度 l ($1 \leq l \leq N$) 的链路, 找到一条 x_0 与 x_t 之间链路代价最小的链路 c , 或者证明这样的链路不存在。需要注意的是, 符合条件的链路可能不是唯一的。

2 中继链路算法设计

为了降低连续空间中解决中继 UAV 定位问题的难度, 本文的求解过程分为两个步骤: 1) 对求解空间进行离散化处理, 构建离散化的求解空间; 2) 设计改进 BF (Improved Bellman-Ford, IBF) 和改进 Dijkstra (Improved Dijkstra, ID) 算法, 实现中继 UAV 定位最优化。

2.1 求解空间构建

设 $X' \subseteq X$ 是区域 X 离散化后得到的有限子集, 中继 UAV 和侦察 UAV 可部署在 X' 的元素节点上。经过离散化处理, 构建中继节点离散图。对于任一个元素节点 $x \in X' \cup \{x_0, x_t\}$, 对应着离散图中的一个节点, 其中节点 n_0 对应基地位置 x_0 , 节点 n_t 对应目标位置 x_t , N 表示图中所有节点的集合。

对于任意 $x \in X'$ 存在 $n \in N$, 若 $b(x_0, x)$ 存在, 则代价为 $f(x_0, x)$ 的边 $e = (n_0, n)$ 表示基地 x_0 与点 x 之间存在通信链路的概率(代价越高, 概率越低)。类似地, 对于任意 $x, x' \in$

X' , 存在 $n, n' \in N$, 若 $b(x_0, x)$ 存在, 则代价为 $f(x_0, x)$ 的边 $e = (n_0, n)$ 表示基地 x_0 与点 x 之间存在通信链路的概率。

设 E 是上述所有边的集合, 图 $G(N, E)$ 是将连续空间离散化处理后形成的有向图。需要注意的是, 在图 $G(N, E)$ 中, 目标节点 n_t 的出度为 0, 其连接边必须满足侦察可达性约束。图 $G(N, E)$ 的构建主要依赖地形、通信盲区等外部环境因素, 与基地和目标位置的关系不大, 可预先构建。

很多经典算法基于离散图 $G(N, E)$ 解决问题。例如, Dijkstra 算法和 A* 算法可解决求解最短路径问题, 即路径的代价最小。然而, 这些算法的结果是产生一条最短路径, 而本文的目标是生成一组路径, 包括路径代价最小和路径跳数最少, 地面操作员可以根据资源约束和任务需求, 选择不同的方案。文献 [10–11] 将中继 UAV 定位问题看作是全跳最优路径 (All Hop Optimal Path, AHOP) 问题。

对于中继 UAV 定位问题, 有下面的定义:

1) 节点 n_0 表示图的起点, 对应基地位置 x_0 ;

2) 节点 n_t 对应目标位置 x_t ;

3) 对任意一个节点 n, n_- 表示其前驱节点, n_+ 表示其后继节点;

4) $c_{n, n'}$ 表示节点 n 和 n' 之间的代价价值;

5) n_{uavs} 为当前可用的 UAV 数目。

2.2 IBF 算法

BF (Bellman-Ford, BF) 算法是解决 AHOP 问题的典型算法, 其基本思想是: 通过 k 轮迭代, 查找从给定节点出发, 经过 k 跳到达目标的最短路径。但是, 该算法的最大不足是时间复杂度较大, 无法满足解决大规模空间或细粒度离散图的需要。

IBF 算法针对 AHOP 问题, 其主要工作是对于每个节点生成并迭代更新一组可达性记录集合。设节点 n 可达性记录为 $\langle k, g_k, p_k \rangle$, 则表示可以从基地节点 n_0 经过 k 跳, 耗费代价 g_k 到达节点 n , 并且节点 n 的前驱节点是 p_k 。显然, 如果对任意节点 n 都构建了可达性记录 $\langle k, g_k, p_k \rangle$, 则可以通过逆向构建的方法, 依次获取其前驱节点, 直至到达初始节点 n_0 , 从而得到一条从 n_0 到 n 的完整路径。

根据可达性记录的定义, 可知其对应着一条节点路径。例如, 如果节点 n_0 和 n 之间存在一条长度为 l 的路径, 则 n 必有一条可达性记录, 并且 $k = l$ 。依此类推, 如果目标节点 n_t 有一条可达性记录 $\langle k, g_k, p_k \rangle$, 则表明 n_0 和 n_t 之间存在一条长度为 $k - 1$, 代价为 g_k 的中继链路。

IBF 算法首先执行一系列预处理操作, 主要是生成以 n_0 为根节点到各节点 n 的最优路径树。所谓最优路径树是指: 从初始节点 n_0 到目标节点 n 不考虑连接边数, 代价花费最小的一条路径。值 k_{\max}^* 表示树的高度。显然, 不存在深度大于 k_{\max}^* 的最优路径树。 k_{\max}^* 限制了图的搜索深度以及所需中继部署 UAV 的数目。节点集合 N 被分成多个子集合 N_k^* , 其包含了最优路径树中从根节点 n_0 经过 k 跳到达的所有节点 n 的集合, 其中 $0 \leq k \leq k_{\max}^*$ 。

- 1) for each $n \in N$ do $g(n) \leftarrow +\infty$
- 2) $g(n_0) = 0$
- 3) for each $n \in n_{0-}$ do
- 4) $E \leftarrow E \setminus \{(n, n_0)\}$
- 5) $V_1 \leftarrow \{n_0\}$
- 6) for $k = 1, 2, \dots, \min(n_{\text{uavs}} + 1, k_{\max}^* - 1)$ do

```

7)   for each  $n' \in N_k^*$  do
8)     for each  $n \in n'$  do
9)        $E \leftarrow E \setminus \{(n, n')\}$ 
10)     $V_{k+1} \leftarrow N_k^*$ 
11)    for each  $n \in V_k$  do
12)      for each  $n' \in n_+$  do
13)         $c \leftarrow g_{k-1}(n) + c_{n, n'}$ 
14)        if  $c < g(n')$  then
15)           $g(n') \leftarrow c$ 
16)           $g_k(n') \leftarrow c$ 
17)           $p_k(n') \leftarrow n$ 
18)     $V_{k+1} \leftarrow V_{k+1} \cup \{n'\}$ 

```

下面对 IBF 算法作简要分析:

在主算法中, $g(n)$ 表示当前从节点 n_0 到 n 的最优代价值。初始化时, 对所有节点 n , $g(n) = \infty$ 。算法执行过程中, 会构建并使用一系列集合 V_k 。当构建任意一个到达节点 n 且路径深度(路径跳数)为 k 的中继链路 p_n , 则 p_n 必定由两部分构成, 到达节点 n 的前驱节点 n_- 的中继路径 p_{n_-} 以及边 (n_-, n) 。其中 $n_- \in V_k$ 且到达 n_- 的路径深度(路径跳数)为 $k - 1$ 。

第 2) ~ 第 5) 行: 初始化与起始节点 n_0 相关的变量。对于节点 n_0 , 因为其是路径搜索的起始节点, 不存在可达性记录。一般情况下, 置 $g(n_0) = 0$, 表明耗费代价 0 到达该节点(第 2)行)。此外, 由于已经设置 $g(n_0) = 0$, 则指向 n_0 的链路是无法优化 $g(n_0)$ 的值, 也就是在搜索最优路径过程中是不起作用的, 可以直接删除(第 3) ~ 第 4) 行)。第 5) 行设置 $V_1 = \{n_0\}$, 表明对任一个路径长度为 1 的中继链路 p_n , 其必定是由经过 0 跳到达节点 n_0 的路径 p_0 , 以及从 n_0 出发指向目标节点 n 的边 (n_0, n) 两部分组成。

第 6) ~ 第 18) 行: 算法从路径长度 $k \geq 1$ (开始至 $\min(n_{\text{uavs}} + 1, k_{\text{max}}^* - 1)$ 截止, 进行迭代计算。使用值 $\min(n_{\text{uavs}} + 1, k_{\text{max}}^* - 1)$ 作为迭代计算的上限, 包含了三层含义: 1) 最多允许使用 n_{uavs} 架 UAV, 则路径长度的上限不会超过 $n_{\text{uavs}} + 1$, 包括基地节点 n_0 与第一架中继 UAV 之间以及侦察 UAV 与目标节点 n_t 之间的两条路径在内; 2) 不存在路径长度大于 k_{max}^* 的中继链路; 3) 在算法的预处理阶段, 路径长度为 k_{max}^* 的中继链路已经创建。

当第 k 轮计算开始执行时, 对于任意节点 $n' \in N_k^*$, 其可达性记录集在预处理阶段已经全部生成, 并且根据 N_k^* 的定义, 从节点 n_0 到 n' 不存在跳数(hop count)大于 k 的最优路径。在第 k 轮迭代计算中, 以 n' 为终点的边所构成的新路径不可能是从节点 n_0 到 n' 的最优路径。所以, 可将所有以 n' 为终点的边从边集合 E 中删除。但是, 还必须考虑另外一种情况, 即其他节点的最优路径包含经过节点 n' 的边, 所以有 $V_{k+1} \leftarrow N_k^*$ 。

第 11) ~ 第 18) 行: 分析所有可能的中继链路长度为 k 的最优路径。符合条件的中继链路必定由两部分构成: 到达节点 $n \in V_k$ 的路径以及以 n 为起始节点的边 (n, n') 。为了判断所有符合条件的路径是否一定是到达节点 n' 最优路径, 需要计算到达路径代价 c (第 11) ~ 第 13) 行)。如果计算的代价值 c 小于当前的最优代价 $g(n')$, 则对于节点 n' , 创建了一条更优的可达性记录 $(k, g_k(n'), p_k(n'))$ (第 14) ~ 第 17) 行)。

在第 18) 行给出了下一轮迭代计算中使用的节点集合 V_{k+1} 。对于下一轮迭代中需要构建的路径长度为 $k + 1$ 的最优

路径, 其必定是由长度为 k 的到达节点 $n \in V_{k+1}$ 的路径以及以 n 为起始节点的边两部分组成。很显然, V_{k+1} 包括两部分, 一部分是当前 N_k^* 的集合, 另一部分是本轮迭代过程中符合 $c < g(n')$ (第 14) 行) 的节点, 结果如 $V_{k+1} \leftarrow V_{k+1} \cup \{n'\}$ (第 18) 行) 所示。

当算法运行终止后, 会为每一个节点生成一个类似仿真实验部分表 1 的记录表, 通过使用表中提供的信息, 对任意一个节点 n , 可以使用逆向推导的方式, 得到从节点 n 回溯到起始节点 n_0 的最优路径。

2.3 ID 算法

本文提出的第二个算法是 ID 算法。ID 算法基于 Dijkstra 算法, 针对 AHOP 问题使用边代价函数 $c'_{n, n'} = c_{n, n'} + a$ 迭代计算从 n_0 到 n_t 的最短路径 p , 其中 a 的值在每轮迭代过程中增加。计算过程中, 对于任意节点 n , 还给出其路径深度 q_n 以及当前 a 下的最优代价值 y_n 。

```

1)    $a \leftarrow a_0$ 
2)   loop
3)     对所有节点  $n \in n_t$ , 使用代价函数  $c'_{n, n'} = c_{n, n'} + a$ , 计算从  $n_0$  到节点  $n$  的最优路径
4)     对所有节点  $n \in n_t$ , 获取从  $n_0$  到节点  $n$  的路径深度  $q_n$ , 路径代价值  $y_n$  以及路径结果  $p$ 
5)     if  $\text{length}(p) \leq n_{\text{uavs}} + 1$  then return  $p$ 
6)      $H \leftarrow \{(n, n') \in E | q_{n'} > q_n + 1\}$ 
7)     if  $H = \emptyset$  then fail
8)     if  $(n, n') \in H$ , 计算  $\varepsilon_{n, n'} \leftarrow \frac{(y_n + c'_{n, n'}) - y_{n'}}{q_{n'} - (q_n + 1)}$ 
9)      $\varepsilon \leftarrow \min \varepsilon_{n, n'}$ 
10)     $a \leftarrow a + \varepsilon$ 

```

下面对 ID 算法作简要分析:

第 1) 行: 初始化迭代变量 $a = a_0$, 通常情况下, 设 $a_0 = 0$, 保证算法开始时使用原始边代价构建最优路径。第 3 行使用代价函数 $c'_{n, n'} = c_{n, n'} + a$ 构建从 n_0 到节点 n_t 的最优路径 p , 并对任意节点 n , 得到路径深度 q_n (即从根节点沿着最优路径 p 到节点 n 经过的跳数)以及当前 a 下从 n_0 到节点 n 的路径代价值 y_n 。如果得到的路径结果 p 满足 $\text{length}(p) \leq n_{\text{uavs}} + 1$ 的条件, 表明路径 p 是一个满足跳数约束的可行解, 直接返回。

第 6) ~ 第 10) 行: 设计一个合适的改变量 ε , 并在每轮的迭代计算过程中, 增加至变量 a 。 ε 值的选取必须十分合理: 首先 ε 的值必须足够大, 以保证在迭代过程中至少有一条路径能够优化; 其次 ε 的值又必须足够小, 以保证没有可行解遗漏。算法首先分析任一个当前不在最优路径 p 中的节点 $n'(n' \in N \text{ 且 } (n, n') \in E)$ 可以优化当前路径的可能性。很显然, 如果通过边 (n, n') 到达节点 n' , 则路径深度为 $q_n + 1$ 。为保证到达节点 n' 的路径深度能够有效地优化缩减, 必须满足 $q_{n'} > q_n + 1$ 。

如果不存在符合这样条件的边, 则表示当前的路径无法优化缩减, 即不存在可行解, 算法直接终止并返回(第 7) 行)。反之, 对于任意边 $(n, n') \in E$, 当前到达节点 n' 的路径 $p_{n'}$ 被到达节点 n 的路径 p_n 以及 n 和 n' 之间的连接边 (n, n') 取代, 其结果是到达 n' 的路径深度缩减了。此外, 对于当前 y_n 与 $y_{n'}$ 的关系, 有 $y_n + c'_{n, n'} > y_{n'}$, 即当前通过节点 n 到达节点 n' 的路径所耗费的代价更大, 增加的代价值为 $(y_n + c'_{n, n'}) - y_{n'}$ 。该增加值除以缩减的路径深度值 $q_{n'} - (q_n + 1)$, 即得到 $\varepsilon_{n, n'}$ (第 8) 行)。最后, 为保证没有可行解遗漏, 在每轮迭代过程中, a

加上所有 ε_{n_i, n_j} 中的最小值。

2.4 时间复杂度分析

使用传统的 BF 算法解决 AHOP 问题,其算法复杂度为 $O(|N||E|) \leq O(|N|^3)$ 。可以对该算法进行优化,在第 k 轮迭代计算过程中,当节点代价无法减少时直接终止计算。通过优化,算法的时间复杂度可以限制在 $O(k_{\max}^* |E|) \leq O(|N||E|) \leq O(|N|^3)$ 内。

在最坏的情况下,IBF 算法的时间复杂度与 BF 算法是一致的。但是,通过在预处理阶段调用 Dijkstra 算法以及构建并使用一系列集合 V_k ,每轮迭代过程中实际使用的边远小于 $|E|$ 。

对于 ID 算法的时间复杂度,设 T_D 表示节点集合 n_t 的最优路径深度,则 $T_D = \sum_{n \in n_t} q_n$ 。当节点集合 n_t 的拓扑关系如图 2(a) 所示,则除去第一个节点的最优路径深度为 0 之外,其余所有节点的最优路径深度为 1,则 $T_D = |N| - 1$ (其中 $|N|$ 是节点集合 n_t 的个数)。当节点集合 n_t 的拓扑关系如图 2(b) 所示,则每个节点的最优路径深度依次增加, $T_D = |N|(|N| - 1)/2$,所以,对于任意拓扑结构,其节点集合 n_t 的最优路径深度 T_D 满足 $|N| - 1 \leq T_D \leq |N|(|N| - 1)/2$,所以 ID 算法最多执行 $|N|^2$ 次,并且在每次的执行过程中都会调用一次 Dijkstra 算法,其时间复杂度为 $O(|E| + |N| \log |N|)$ 。所以在最坏情况下,算法的时间复杂度是 $O(|N|^2(|E| + |N| \log |N|)) \leq O(|N|^4)$ 。但是,算法执行过程中,只有在某些特定的拓扑结构图的情况下,代价函数才需要迭代执行 $|N|^2$ 次,所以实际执行中,该算法的时间性能要优于 BF 算法。

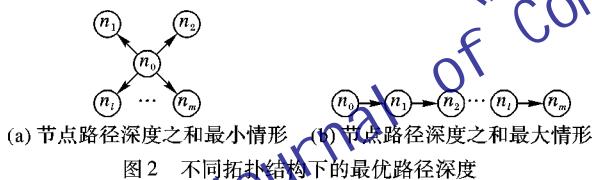


图 2 不同拓扑结构下的最优路径深度

3 仿真实验分析

在 Matlab 2012a 下开展仿真实验,对比分析 BF 算法与 IBF 和 ID 算法的有效性及算法性能。实验硬件环境:Intel Core Duo CPU T2450,内存 2.5 GB。仿真范围设定为 [100 km, 100 km],UAV 之间的最大中继通信距离为 30 km。为了保证可行解存在,设定可以使用的 UAV 数目 $n_{\text{UAV}} = \infty$,同时设置基地所在点始终为 n_1 以对比分析两种算法的性能。图 3 给出了两个算法的执行效果图,图中的黑色梯形区域表示通信盲区,无法在该区域部署中继 UAV。在空间内随机分布 100 个有效点,中继 UAV 可部署在这些点上。连线表示相邻两个节点的距离满足 UAV 通信距离约束,可在两个节点之间构建中继链路。设置基地和侦察目标位置,执行 BF 算法、IBF 和 ID 算法后,生成路径代价最优的 UAV 通信中继链路,如图 3 中黑色圆形节点之间的连接线所示,其中黑色圆形节点(除去首尾两个节点)表示中继 UAV 部署的位置。

IBF 算法执行过程中,为每一个节点生成一组类似表 1 记录集 R 。表 1 中,链路长度即为构建一条从基地节点到节点的中继链路需要的中继 UAV 数目,对应的是使用最少数目的 UAV 构建得到的最优中继通信路径。随着行号的逐渐增

加,使用 UAV 的数目会越来越多,直至最后一行,表示使用可用的最少数目的 UAV,可以构建通信代价最小的中继链路。

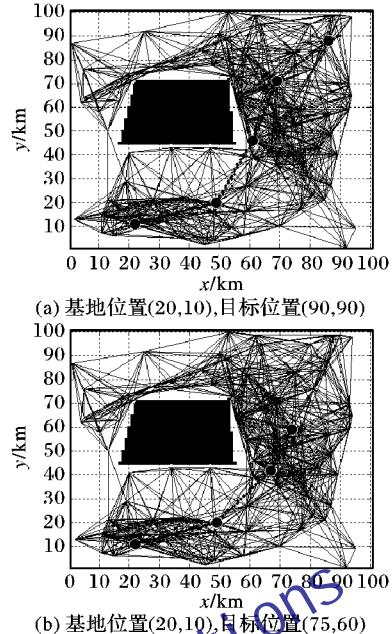


图 3 算法执行结果

表 1 节点 n_0 的记录集 $\langle k, g_k(n), p_k(n) \rangle$

链路长度	链路代价 g_k/km	前驱节点 p_k
4	78.2517	n_{73}
5	70.7240	n_{54}

可用的 UAV 数目发生变化后,可以通过直接访问记录集 R 的方式获得中继 UAV 的部署方案,不需要重新执行大量运算,大大提高了解决问题的灵活性。图 4 给出了构建从基地到目标通信中继链路的两种方案,图中黑色圆形节点表示构建中继链路的节点(包括基地和目标两个首尾节点)。

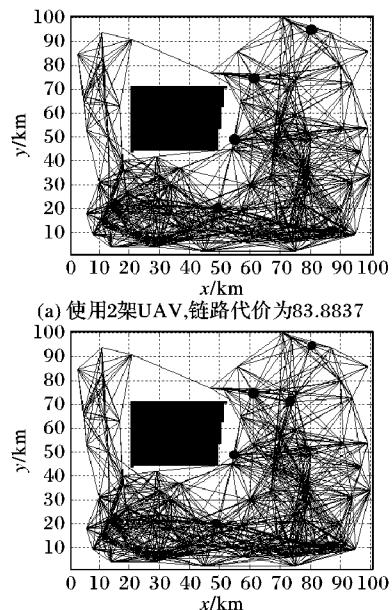


图 4 构建 UAV 中继链路的两种方案

表 2 给出了分别在 100,300,500,1000 个随机部署点下的三个算法性能对比(取算法执行 10 次的平均值),可看出:

随着节点数目的增加,BF 算法的执行时间显著增加。在节点数目较少时,IBF 和 ID 算法由于需要进行预处理的原因,其时间性能劣于 BF 算法;但随着节点数目增加,IBF 和 ID 算法的时间性能与 BF 算法相比,增加幅度较小且优于 BF 算法。

表 2 BF 算法与 IBF 算法、ID 算法性能对比

节点数目 N_{Node}	连接边数 N_{Edge}	执行时间/s		
		BF 算法	IBF 算法	ID 算法
100	1 107	0.095 161	0.230 4	0.353 4
300	9 183	2.734 500	4.658 9	3.775 7
500	25 296	13.679 200	9.641 0	10.520 0
1 000	99 616	176.229 300	113.987 0	108.667 8

4 结语

通过使用多架 UAV 构建通信中继链路以完成侦察数据的远距离传输,是一种解决单架 UAV 通信距离受限的有效方案。本文首先基于 BF 算法提出 IBF 算法,通过花费较小的代价进行预处理,使得 IBF 算法的时间性能较 BF 算法有明显提高;其次,IBF 算法生成了一组记录表,通过直接访问记录表,可以获取任意给定数目下的 UAV 中继链路可行解,提高了解决多架 UAV 构建中继通信链路的灵活性。基于 Dijkstra 算法提出 ID 算法,在每轮迭代过程中通过改变连接边的代价值,查找到达任意目标节点并满足可用 UAV 数目等约束条件的可行解。最后,通过仿真实验证了 IBF 和 ID 算法解决 UAV 中继定位问题的有效性。

参考文献:

- [1] CERASOLI C. An analysis of unmanned airborne vehicle relay coverage in urban environments[C]// MILCOM 2007: Proceedings of the 2007 IEEE Military Communications Conference. Piscataway: IEEE Press, 2007: 1 - 7.
- [2] PINKNEY M F J, HAMPEL D, DIPIERRO S. Unmanned Aerial Vehicle (UAV) communications relay[C]// MILCOM 1996: Proceedings of the 1996 IEEE Military Communications Conference. Piscataway: IEEE Press, 1996: 1: 47 - 51.
- [3] OLEG B, PATRICK D, KAJ H, et al. Optimal placement of UV-

based communications relay nodes [J]. Journal of Global Optimization, 2010, 48(4): 511 - 531.

- [4] CHEN M C, HSIAO P H, KUNG H T, et al. Maximizing throughputs of UAV-relaying networks with the load carry and deliver paradigm[C]// WCNC 2007: Proceedings of the 2007 IEEE Wireless Communications and Networking Conference. Piscataway: IEEE Press, 2007: 4417 - 4424.
- [5] CHEN Y, DONG L, OH J-S. Real-time video relay for UAV traffic surveillance systems through available communication networks [C]// WCNC 2007: Proceedings of the 2007 IEEE Wireless Communications and Networking Conference. Piscataway: IEEE Press, 2007: 2608 - 2612.
- [6] OMER C, IBRAHIM Z. Continuous airborne communication relay approach using unmanned aerial vehicles [J]. Journal of Intelligent & Robotic Systems, 2012, 65(1): 549 - 562.
- [7] IBRAHIM Z, OMER C. Obstacle free airborne communication relay approach by using autonomous unmanned aerial vehicles[EB/OL]. [2013-10-10]. http://www.researchgate.net/profile/Omer_Cetin/publication/234025058_Establishing_Obstacle_and_Collision_Free_Communication_Relay_for_UAVs_with_Artificial_Potential_Fields/file/9fcfd511538f1092bb.pdf.
- [8] JOEL M E, THOMAS W D. Maintaining wireless connectivity constraints for swarms in the presence of obstacles[C]// ICRA 2006: Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Piscataway: IEEE Press, 2006: 946 - 951.
- [9] Gehr D, Luis G. On the directed hop-constrained shortest path problem[J]. Operations Research Letters, 2004, 32(1): 15 - 22.
- [10] ROCH G, ARIEL O. Computing shortest paths for any number of hops [J]. IEEE Transactions on Networking, 2002, 10(5): 613 - 620.
- [11] ANANTARAM B, KEMAL A. Using a hop-constrained model to generate alternative communication network design [J]. Journal of Computing, 1992, 4(2): 192 - 205.
- [12] THOMAS C, CHARLES L, RONALD R. Introduction to algorithms[M]. Cambridge: MIT Press, 1990.
- [13] EUGENE L, LAWLER. Combinatorial optimization: networks and matroids[M]. [S. l.]: Dover Publications, 1976.

(上接第 1557 页)

- [4] LU G, KRISHNAMACHARI B, RAGHAVENDRA C. An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks [C]// Proceedings of the 18th International Parallel and Distributed Processing Symposium. Piscataway: IEEE Press, 224 - 230.
- [5] RHEE I, WARRIOR A, AIA M, et al. Z-MAC: a hybrid MAC for wireless sensor networks[C]// SenSys 2005: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems. New York: ACM Press, 2005: 90 - 101.
- [6] SALAJEGHEH M, SOROUSH H, KALIS A. HYMAC: hybrid TD-MA/FDMA medium access control protocol for wireless sensor networks[C]// PIMRC 2007: Proceedings of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications. Piscataway: IEEE Press, 2007: 1 - 5.
- [7] IEEE Standard 802.15.4. Part 15.4: Wireless Medium Access Control and Physical Layer Specification for Low-Rate Wireless Personal Area Networks (WPANs) [S]. Piscataway: IEEE Press, 2007.

- [8] LIN C Y, CHAN C L, KING C T, et al. A cyclic MAC scheduler for collecting data from heterogeneous sensors [J]. Computer Communications, 2011, 34(13): 1630 - 1644.
- [9] INCEL O D, van HOESEL L, JANSEN P, et al. A multi-channel MAC protocol for wireless sensor networks [J]. Ad Hoc Networks, 2011, 9(1): 73 - 94.
- [10] ZHOU Z, PENG T, CUI J-H, et al. Handling triple hidden terminal problems for multi-channel MAC in long-delay underwater sensor networks[C]// INFOCOM 2010: Proceedings of the 29th Conference on Information Communications. Piscataway: IEEE Press, 2010: 371 - 375.
- [11] LUO T, MOTANI M, SRINIVASAN V. Cooperative asynchronous multichannel MAC: Design, analysis, and implementation [J]. IEEE Transactions on Mobile Computing, 2009, 8(3): 338 - 352.
- [12] SUN Y, GUREWOTZ O, JOHNSON D. A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks [C]// Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. New York: ACM Press, 2008: 456 - 469.