

无人机自组网基于加权的稳定分簇算法

梅家栋, 南建国

(空军工程大学 航空工程学院, 西安 710038)

摘要: 相较于传统移动自组网, 无人机自组网具有节点移动速度更快、分布范围更广、网络规模更大的显著特征, 而分簇结构能实现网络的有效控制和管理, 提高网络的扩展性。为了提高分簇结构的稳定性, 提出了一种基于加权的稳定分簇算法, 该算法基于现有加权分簇算法, 从分簇测度指标的选取、指标权重因子分配、分簇稳定保持策略三个方面进行改进。仿真结果表明, 该算法不仅能合理控制分簇数量, 有效提高网络的扩展性, 而且能够延长簇首保持时间和降低成员节点的簇间切换次数, 提高分簇的稳定性。

关键词: 无人机自组网; 分簇算法; 测度指标; 加权; 稳定性

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2021)11-037-3411-06

doi:10.19734/j.issn.1001-3695.2021.04.0145

Weighted stable clustering algorithm for flying Ad hoc network

Mei Jiadong, Nan Jianguo

(College of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China)

Abstract: Compared with traditional mobile Ad hoc network, flying Ad hoc network has the characteristics of faster node movement, wider distribution and larger network scale. The clustering structure can realize the effective control and management of the network, and improve the scalability of the network. In order to improve the stability of clustering structure, this paper proposed a weighted stable clustering algorithm. Based on the existing weighted clustering algorithm, the algorithm improved from three aspects: the selection of clustering measure index, the allocation of index weight factor, and the strategy of clustering stability maintenance. Simulation results show that the algorithm can not only reasonably control the number of clusters, effectively improve the scalability of the network, but also extend the holding time of cluster heads and reduce the number of inter cluster handoffs of member nodes, and improve the stability of clustering.

Key words: FANETs; clustering algorithm; measurement index; weighting; stability

0 引言

随着战场信息化保障能力的增强和无人机的进一步武器化, 无人机在作战任务中的作用日益凸显, 提高无人机集群协同作战能力, 是信息化、网络化战争的需要。相较于单无人机系统, 无人机集群能在复杂多变的战场环境中更好、更有效地完成作战任务^[1,2]。而无人机集群作战的基础是建立稳定可靠的通信网络, 为无人机节点间的信息资源共享提供保障, 实现任务协同, 充分发挥集群优势, 提高系统总体效能。移动 Ad hoc 网络具有无中心、组网高度灵活、拓扑动态可变、抗毁性强等特点, 可有效满足无人机集群组网通信需求^[3]。无人机集群通过采用移动 Ad hoc 网络的通信架构, 形成了无人机自组网 (flying Ad hoc network, FANET)。相较于传统移动 Ad hoc 网络, 无人机自组网具有节点移动性更强、网络拓扑结构高动态变化等特点^[4,5]。同时, 为了充分发挥集群优势, 无人机自组网中的节点数量通常较多、分布范围更广、网络规模更大。针对无人机网络的上述特点, 分簇策略^[6,7]是提高无人机自组网性能的有效手段。

通过将网络中的节点分成若干个簇, 每个簇都可以看做是一个相对较稳定的小型子网络^[8]。与平面结构相比, 分簇的网络结构能够减轻网络局部的拓扑结构变化对整个网络的影响。在路由的计算和生成过程中, 只需要部分节点参与, 有效减少了路由和控制开销。此外, 分簇结构能实现对网络的有效

控制和管理, 提高了网络的扩展性^[9]。

文献[10~12]详细介绍了一些传统的分簇算法, 包括最小 ID (lowest ID) 分簇算法和最大连接度 (max degree) 分簇算法等。最小 ID 算法的优点在于计算简单、分群速度快, 但是该算法未考虑节点的移动特性, 在高动态的网络场景下, 簇结构易发生分解和重构。最大连接度算法选择邻居节点中具有最大节点度的节点作为簇首, 因此网络中分簇数少, 降低了端到端时延。但是该算法存在着簇内成员数多、通信资源分配不均衡、簇结构稳定性较差的问题。文献[13]提出了加权分簇算法 (weighted clustering algorithm, WCA), 通过综合考虑节点的节点度、距离和移动性等因素, 使得簇首选取更加合理, 场景的适应性更强。基于加权分簇的上述优点, 国内外学者提出了多种改进型加权分簇算法, 如文献[14]提出了一种自适应安全加权分簇算法, 旨在解决分簇过程中恶意节点选举为簇首节点存在的安全问题, 但对于当网络处于高动态条件下, 关于节点移动性的分簇指标选取以及分簇保持策略等方面的问题没有进行改进, 并不适用于无人自组织网络。文献[15]在大规模网络环境下, 提出了一种稳定的分簇算法, 但在推导节点相对运动关系时, 基于某一节点位于坐标原点保持静止的假设, 条件较为理想, 且没有考虑分簇大小对网络性能的影响。文献[16]针对无人机集群组网需要, 提出了一种多参数加权算法, 改进了最大速度相似度指标, 并提出自适应节点度差, 但是对于最佳节点度指标, 该算法存在计算复杂的问题, 且采用固

收稿日期: 2021-04-29; 修回日期: 2021-06-07

作者简介: 梅家栋 (1997-), 男, 河南新蔡人, 硕士研究生, 主要研究方向为无人机集群组网通信路由协议 (1158292019@qq.com); 南建国 (1969-), 男, 陕西西安人, 教授, 硕士, 主要研究方向为计算机网络与通信。

定权值的方法对各分簇指标加权组合,不能很好地应对高动态的网络环境。文献[17]提出了一种基于 AHP-熵权的分簇算法,通过计算组合权重,增加权重的可靠性,但是该方法只是两种赋权方法简单累加的结果,依然存在一定的缺陷。

基于相关文献研究,提出了一种基于加权的稳定分簇算法(weighted stable clustering algorithm, WSCA),针对加权分簇算法的三个关键性问题进行了改进:a)在分簇测度指标选取方面,为了体现节点的分布差异,提出了连接可靠性因子修正的平均距离测度指标,改进了节点相对移动性指标,推导了节点不同运动关系下的链路保持时间;b)在确定参数指标权值方面,为充分体现指标的区分度和重要性,通过有效结合主客观赋权的优势,提出了基于极大熵准则的指标赋权方法,根据网络环境和节点状态动态分配权值;c)在分簇稳定保持策略方面引入了预备簇首,并设置其现任簇首权值差的动态阈值,合理调整簇首更换的触发条件,提高了分簇的稳定性。

1 分簇测度指标

1.1 连接可靠性因子修正的平均距离测度指标

在 GPS 等地理位置服务系统支持下,节点可获得自身地理位置坐标,则在时刻 t 节点 i 的位置信息为 $(x_i(t), y_i(t))$, 节点 k 的坐标信息为 $(x_k(t), y_k(t))$, 则节点 i 和 k 之间的距离可表示为

$$d_{i,k}(t) = \sqrt{(x_i(t) - x_k(t))^2 + (y_i(t) - y_k(t))^2} \quad (1)$$

设节点的传输半径是 R , 则其无线通信覆盖范围是一个以该节点为圆心, 以 R 为半径的圆。分布在圆内的节点, 认为是该节点的邻居节点, 设邻居节点集合为 $S[i]$ 。

$$S[i] = \bigcup_{k \in V, k \neq i} \{k | d_{ik} < R\} \quad (2)$$

其中: V 表示网络中的节点集合; d_{ik} 为成员节点 k 与簇首节点 i 之间的距离。

簇首节点 i 的节点度 deg_i 表示为

$$deg_i = |S[i]| = \sum_{k \in V, k \neq i} \{d_{ik} < R\} \quad (3)$$

对于某一节点的邻居节点集合成员, 在文献[13, 17]的簇首选取算法中, 通常是将邻居节点间距的平均值作为簇首选取的距离值度量, 即

$$\bar{d}_i = \frac{1}{deg_i} \sum_{k \in S[i]} d_{ik} \quad (4)$$

在进行簇首选取时, 节点的平均邻居节点距离越小, 一定程度上说明节点的移动对簇的稳定性带来的影响越小。但是平均邻居节点距离并不能体现出节点的具体分布情况, 部分位于通信边界(如图1中的区域2)的簇成员节点通信质量低, 报文丢失率显著增大, 文献[18, 19]给出了详细的公式推导证明。此外, 位于簇首节点边缘的节点会因为微小的移动, 就移出了簇首的通信范围, 影响簇的稳定性, 节点的簇间转移频率增大。

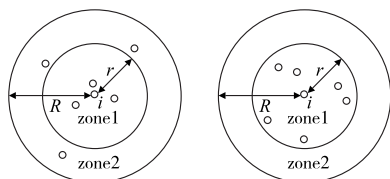


图1 平均距离相同时的两种节点分布

Fig.1 Two node distributions with the same average distance

因此, 可以采用对簇结构分层处理的方式, 区别出位于通信边界的成员节点, 同时可对分簇测度中的平均距离指标进行修正。通过式(5), 可以定量描述簇成员节点分布情况对连接可靠性的影响, 表示为

$$\begin{cases} p_{rel}(k) = 1 & 0 < d_{ik} \leq r \\ p_{rel}(k) = \alpha \left(\frac{R - d_{ik}}{R - r} \right) & r < d_{ik} < R \end{cases} \quad (5)$$

其中: $p_{rel}(k)$ 表示簇成员 k 与簇首间的连接可靠性概率分布函数; d_{ik} 表示簇首节点 i 与成员节点 k 之间的距离; α 是一个可调参量, 取值为 $0 < \alpha < 1$ 。由各个簇成员节点的 $p_{rel}(k)$, 定义簇首节点的连接可靠性函数为

$$CR_i = \frac{N_{zone1}}{deg_i} \cdot \left\{ \prod_{k=1}^{N_{zone1}} p_{rel}(k | d_{ik} \in (0, r]) \right\}^{\frac{1}{N_{zone1}}} + \frac{N_{zone2}}{deg_i} \cdot \left\{ \prod_{k=N_{zone1}+1}^{deg_i} p_{rel}(k | d_{ik} \in (r, R)) \right\}^{\frac{1}{N_{zone2}}} \quad (6)$$

将式(5)代入式(6)可得

$$CR_i = \frac{N_{zone1}}{deg_i} + \frac{deg_i - N_{zone1}}{deg_i} \cdot \left\{ \prod_{k=N_{zone1}+1}^{deg_i} \alpha \left(\frac{R - d_{ik}}{R - r} \right) \right\}^{\frac{1}{deg_i - N_{zone1}}} \quad (7)$$

通过 CR_i 得到修正后的平均距离指标 \bar{d}_i^* 为

$$\bar{d}_i^* = \bar{d}_i \cdot (1 - CR_i) \quad (8)$$

其中: \bar{d}_i^* 越小说明簇首节点与成员节点连接越可靠, 簇的整体稳定性越高。通过 \bar{d} 反映了分簇测度中距离特性, 通过 CR_i 则可以在一定程度上反映成员节点在簇中的分布特性。

1.2 节点移动性指标

设节点 j 在周期性发送分簇消息的时间 T 内, 节点 j 测量 K 次位置信息, 即检测频率为 $\tau = T/K$ 。在时间 $k\tau$ 内节点的坐标变化可表示为

$$\begin{cases} \Delta x_j(k) = x_j(t + k\tau) - x_j(t) \\ \Delta y_j(k) = y_j(t + k\tau) - y_j(t) \end{cases} \quad (9)$$

其中: $k = 1, 2, \dots, K$ 。由节点 j 的地理位置的变化可以推导出节点间隔 $k\tau$ 可推导出移动速度 $v_j(t + k\tau)$ 和移动方向 $\theta_j(t + k\tau)$ 为

$$v_j(t + k\tau) = \sqrt{\Delta x_j(k)^2 + \Delta y_j(k)^2} / k\tau \quad (10)$$

$$\theta_j(t + k\tau) = \begin{cases} \arctan \left| \frac{\Delta y(k)}{\Delta x(k)} \right| \cdot \text{sgn}(\Delta y(k)) & \Delta x(k) > 0 \\ \frac{\pi}{2} \cdot \text{sgn}(\Delta y(k)) & \Delta x(k) = 0 \\ (\pi - \arctan \left| \frac{\Delta y(k)}{\Delta x(k)} \right|) \cdot \text{sgn}(\Delta y(k)) & \Delta x(k) < 0 \end{cases} \quad (11)$$

在检测时间 T 内, 可以求得节点 j 的移动速度和移动方向分别表示为

$$v_j(t + T) = \frac{1}{K} \sum_{k=1}^K v_j(t + k\tau) \quad (12)$$

$$\theta_j(t + T) = \frac{1}{K} \sum_{k=1}^K \theta_j(t + k\tau) \quad (13)$$

结合式(12)(13)中的移动速度和方向, 则在时刻 t , 节点 j 的速度矢量可以表示为

$$\mathbf{v}_j(t) = (v_j(t) \cos \theta_j(t), v_j(t) \sin \theta_j(t)) \quad (14)$$

定义节点 j 与 i 之间的相对运动性参数为

$$M_{ij} = \frac{|\mathbf{v}_i - \mathbf{v}_j|}{|\mathbf{v}_i| + |\mathbf{v}_j|} \quad (15)$$

节点 i 的邻居节点集合为 $S[i]$, 对于集合中的每一个节点求相对运动性参数, 并求其均值后可以得到节点 i 的移动性指标 M_i 为

$$M_i = \frac{\sum_{k=1}^{deg_i} M_{ik}}{deg_i} \quad (16)$$

1.3 不同场景下链路保持时间的计算

图2表示为两节点先靠近后远离的场景, 假设在当前时刻, 一跳通信范围内的节点 i 和 j 以速度 $\mathbf{v}_i = (v_i \cos \alpha, v_i \sin \alpha)$ 和

$v_j = (v_j \cos \beta, v_j \sin \beta)$ 运动, 节点 j 相对节点 i , 沿着直线 AB 的方向运动, 先靠近节点 i 后远离, 到达 B 点后, 离开节点 i 的通信范围, 两节点之间的链路断开, 则节点 j 在 i 的通信范围运动过程中, 通过 CB 的时间即为两节点间的链路保持时间。

通过 i 和 j 的速度可以求得两节点的相对运动速度为 Δv_{ij} , 表示为

$$\Delta v_{ij} = v_j - v_i \quad (17)$$

节点 i 和 j 的相对速度 Δv_{ij} 用向量坐标可以表示为

$$\Delta v_{ij} = (v_i \cos \alpha - v_j \cos \beta, v_i \sin \alpha - v_j \sin \beta) \quad (18)$$

在数学关系上, Δv_{ij} 可作为直线 AB 的方向向量, 由于 $DE \perp AB$, 所以直线 DE 的向量 n 可表示为直线 AB 的法向量, 则有

$$\Delta v_{ij} \cdot n = 0 \quad (19)$$

其中: $n = (-v_i \sin \alpha + v_j \sin \beta, v_i \cos \alpha - v_j \cos \beta)$ 。由向量的运算性质可得, 线段 DE 的长度可表示为

$$\overline{DE} = \frac{|n \cdot \overrightarrow{DC}|}{|n|} \quad (20)$$

在得到线段 DE 的长度后, 根据几何关系, 利用勾股定理可以计算 EB 和 CE 的长度为

$$\overline{EB} = \sqrt{\overline{DB}^2 - \overline{DE}^2} = \sqrt{R^2 - \overline{DE}^2} \quad (21)$$

$$\overline{CE} = \sqrt{\overline{DC}^2 - \overline{DE}^2} = \sqrt{d_{ij}^2 - \overline{DE}^2} \quad (22)$$

在节点先靠近后远离的场景, 节点 i 与 j 之间的链路保持时间为

$$T_{ij} = \frac{\overline{CB}}{|\Delta v_{ij}|} = \frac{\overline{CE} + \overline{EB}}{|\Delta v_{ij}|} \quad (23)$$

图 3 表示两节点远离的场景, 节点 j 相对节点 i , 沿着直线 AB 向远离节点 j 方向运动, 到达 B 点后, 离开节点 i 的通信范围, 两节点之间的链路断开, 则节点 j 在 i 的通信范围运动过程中, 通过 CB 的时间即为两节点间的链路保持时间 T_{ij} , 表示为

$$T_{ij} = \frac{\overline{CB}}{|\Delta v_{ij}|} = \frac{\overline{EB} - \overline{EC}}{|\Delta v_{ij}|} \quad (24)$$

对节点 i 的邻节点的链路保持时间求均值后, 可得到 i 平均链路保持时间为

$$T_i = \frac{\sum_{i=1}^{deg_i} T_{ij}}{deg_i} \quad (25)$$

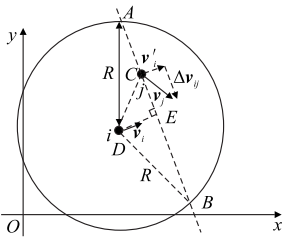


图2 节点先靠近后远离的运动场景
Fig.2 Motion scene where the nodes are close first and then far away

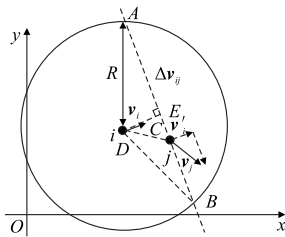


图3 节点远离的运动场景
Fig.3 Moving scene with nodes far away from each other

1.4 理想节点度差

由于网络带宽资源的有限性, 进行分簇时, 在考虑节点移动和分布状态的同时, 还需要考虑簇的规模大小。文献[20]给出了理想情况下, 网络中单个节点吞吐量表达式为

$$R_{per} = \Theta\left(\frac{B}{\sqrt{N}}\right) \quad (26)$$

其中: N 为网络中节点个数; B 为通信带宽; $\Theta()$ 表示为一个正相关函数。由式(26)可知, 随着节点数的增大, 网络吞吐量降低。对于单个簇而言, 簇内节点数过多会降低分簇的吞吐量。同时, 较大的分簇规模会导致簇首节点的负载很大, 易出现节

点的拥塞, 成为网络通信的瓶颈, 而当簇的规模较小时, 又会使得网络带宽资源利用率降低。对于整个网络而言, 随着簇数量增大, 簇间的通信开销也会增加。此外, 节点度的大小可以反映节点的分布情况, 节点度越大, 则该节点越处于网络局部的中心位置。文献[20]给出了最佳分簇个数为

$$N_{CH} = \frac{B_1}{B_2} \sqrt{N} \quad (27)$$

则最佳簇大小为

$$\delta = \frac{N}{N_{CH}} = \frac{B_2}{B_1} \sqrt{N} \quad (28)$$

其中: B_1 为簇内带宽; B_2 为簇间带宽。则簇首节点的最佳节点度为

$$\beta = \delta - 1 \quad (29)$$

本文中网络节点是同构节点, 则 $B_1 = B_2$, 所以 $\delta = \sqrt{N}$ 。因此, 将节点度与理想节点度的差值设置分簇指标, 对于节点 i 则为

$$D_i = |deg_i - \beta| \quad (30)$$

1.5 分簇联合测度指标

在计算得到节点 i 的各项分簇指标后, 综合考虑各项参数, 通过对各项指标加权后得到节点 i 的分簇联合测度指标为

$$W_i = \omega_1 \bar{d}_i^* + \omega_2 M_i + \omega_3 T_i + \omega_4 D_i \quad (31)$$

式(31)中各个指标的加权系数的取值为 $\omega_m \in [0, 1]$, 且满足 $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$ 。 ω_m 的大小分配体现指标的重要程度, 其取值需适应网络环境, 具有可调性。

2 基于极大熵准则的指标赋权方法

在得到分簇测度指标后, 另一个需要解决的问题是确定各指标的权值。为了能同时体现到各项指标的区分度和重要性, 本文采用极大熵准则实现各测度指标的动态赋值, 具体实现过程如下:

设节点 a 与其邻居节点总数为 n 个, 节点集合记做 $Q^a = \{Q_1^a, Q_2^a, \dots, Q_n^a\}$, 分簇测度指标数量为 m , 记做 $P = \{P_1, P_2, \dots, P_m\}$ 。邻居节点 Q_i^a 的第 j 个测度指标值 P_j 表示为 e_{ij} , 因此 n 个节点的 $m \times n$ 个测度指标可构成评价矩阵 $E^a = [e_{ij}]_{m \times n}$ 。

$$u = \frac{e - e_{\min}}{e_{\max} - e_{\min}} \quad (32)$$

$$u = \frac{e_{\max} - e}{e_{\max} - e_{\min}} \quad (33)$$

选取本地节点集合的各项指标, 分别通过式(32)(33)对正向指标和逆向指标进行归一化处理。在本文确定的指标 \bar{d}_i^* , M_i 和 D_i 的值越小代表性能越好, 所以是逆向指标。 T_i 的值越大代表性能越好, 是正向指标。进而, 得到 n 个节点的 $m \times n$ 个测度指标可构成评价矩阵 $U^a = [u_{ij}]_{m \times n}$ 。

设虚拟基点节点为 Q_{n+1}^a , 分别 U^a 取矩阵第 i 行 ($i = 1, 2, \dots, m$) 测度指标值的最大值为 Q_{n+1}^a 的第 i 行测度指标值 u_{i*} , 即

$$u_{i*} = \max\{u_{ik}\} \quad (34)$$

从而可得到虚拟基准节点的测度指标向量为 $(u_{1*}, u_{2*}, \dots, u_{m*})^T$, 将其作为第 $n+1$ 列补充到矩阵 U^a 中, 可得到扩展矩阵 U_*^a 为

$$U_*^a = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} & u_{1*} \\ u_{21} & u_{22} & \cdots & u_{2n} & u_{2*} \\ \vdots & \vdots & & \vdots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mn} & u_{m*} \end{bmatrix} \quad (35)$$

将 U^a 中每一行元素与虚拟基点节点对应值比较,得到当量化处理后的矩阵记做 R^a 。

$$R^a = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} & 1 \\ r_{21} & r_{22} & \cdots & r_{2n} & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} & 1 \end{bmatrix} \quad (36)$$

节点分簇目标函数为

$$W(Q_i^a) = \omega_1 r_{i1} + \omega_2 r_{i2} + \cdots + \omega_m r_{im} \quad (37)$$

根据基点节点的定义,当满足各评价节点与基点节点的指标值加权方差最小时,所求的权重才最合理。同时,各评价指标的权值在满足主客观约束条件下,再满足熵极大,才能保证各指标权重的客观公平性。建立目标规划方程为

$$\begin{cases} \min \delta \sum_{i=1}^n \sum_{j=1}^m \omega_j (1 - r_{ij})^2 + (1 - \delta) \sum_{j=1}^m \omega_j \ln \omega_j \\ \text{s. t. } \sum_{j=1}^m \omega_j = 1, \omega_j \geq 0, j = 1, 2, \cdots, m \end{cases} \quad (38)$$

其中:参数 $0 \leq \delta \leq 1$ 。通过构建拉格朗日函数,并求解后可以得到

$$\omega_j = \frac{\exp(-[1 + \delta \sum_{i=1}^n (1 - r_{ij})^2 / (1 - \delta)])}{\sum_{j=1}^m \exp(-[1 + \delta \sum_{i=1}^n (1 - r_{ij})^2 / (1 - \delta)])} \quad (39)$$

在保证客观公平性的同时,可以根据主观约束条件,让权值的计算体现出各项指标的重要性。在高动态环境下,认为平均链路连接时间 T_i 的重要程度最高。结合式(31)可认为,对应权值 ω_3 最大,即为

$$\omega_3 = \max\{\omega_m\} \quad m = 1, 2, 3, 4 \quad (40)$$

主观约束条件可根据网络的实际特点来确定,结合式(39),在计算过程中可调整参数 δ 的取值,求解满足主观约束条件的权值解。在得到各指标的计算权值后,代入到式(31),得到联合测度指标 W 的值,作为分簇的依据。

3 分簇算法的实现

3.1 簇首的选择

簇首选择基于联合测度指标,通过计算局部区域节点的联合测度指标后,具有最大值的节点被选为簇首。通过联合测度指标的计算过程可知,选举的簇首是为了最大程度上保证分簇的稳定性。

a) 在进行簇的初始化过程时,网络中的节点通过交互 HELLO 消息,建立邻居节点列表,得到节点的节点度,并根据广播消息中携带的关于邻居节点参数信息,计算自身节点的联合测度指标值 W 。

b) 节点通过新一轮的广播消息,将自身的 W 值告知邻居节点,收到消息包的节点通过本地比较决定自身角色,联合测度指标最大的节点被选举为簇首。

c) 簇首节点通过广播 CH 消息,对簇首地位进行声明。邻居节点在收到 CH 消息包后,则向簇首发送加入请求消息(request to join, RTJ),成为簇内成员节点。如果一个节点收到了多个簇首声明信息,则通过比较后,选择具有最大联合测度指标值 W 的簇首所在簇加入。

d) 若某一节点在等待一段时间后,且 W 值比其大的邻居节点都已经确定自身角色后,没有收到簇头声明消息 CH,则该节点成为簇首。

3.2 网关节点的选择

当某一节点同时收到多个簇首发送的 CH 消息时,选择 W 值最大簇首所在簇加入,同时作为网关节点辅助完成簇间

通信。

3.3 稳定分簇保持策略

3.3.1 节点的加入与离开

新节点(包括网络新加入的节点和脱离原簇的节点)周期性广播入簇请求消息,当邻近簇首节点收到此消息后,则发送入簇接收消息,“握手”完成后,实现新节点的加入。如果同一节点收到多个簇首发送的入簇接收消息,则通过读取消息中携带的 W 值后,选择 W 值最大的簇加入,成为该簇的成员。

当节点无法收到簇首的消息,但能收到其他簇内成员节点的消息时,说明该节点与簇首节点相距两跳。此时,该节点首先探寻是否存在相距一跳的邻居簇首,如果存在则加入邻簇。如果不存在,节点成为附属簇员,通过其他簇内成员与簇首进行通信。若附属簇员监听到一跳范围内的其他簇首,则加入新簇。当该节点与原簇首节点相距 3 跳时,且不存在可加入簇时,节点自身宣告成为新的簇首。

通过引入附属簇员,减少孤立通信节点的产生,相较于单纯的单跳簇结构,也使得分簇规模更加灵活。

3.3.2 簇首更换策略

预备簇首的选取:在完成簇首选举后,将权重值第二的节点暂时确定为预备簇首节点,同时,在簇首节点保持有效的时间内,周期性轮询簇内成员,获取邻节点的距离、速度等状态信息,并计算自身的权值;选取权重值最大的簇成员节点为预备簇首,实现预备簇首实时更新和最优化,在确定预备簇首后,通过广播告知簇内成员。

周期性推选预备簇首节点的同时,及时检测簇首的权值信息值的变化,通过设置两者的权值差阈值,作为触发簇首更换的条件。随着网络环境变化,当现任簇首节点权值和预备簇首权值之间的关系满足簇首更换条件时,簇首节点发送 APPOINT 消息给预备簇首,完成簇首地位转移。预备簇首收到 APPOINT 消息后,广播 CH 消息,告知其他成员,自己成为新的簇首。

簇首更换条件为

$$\begin{cases} W_{CH} < W_{PCH} \\ W_{PCH} - W_{CH} < \Phi \end{cases} \quad (41)$$

其中: W_{CH} 为现任簇首的权值; W_{PCH} 为预备簇首的权值; 阈值 Φ 需要根据簇首节点权值信息的变化状态动态调整其值的大小,可以避免采用固定阈值带来的簇首更换策略存在的虚警和操作滞后问题。定义式(42)中的 $\delta(t)$ 为簇首权值的状态变化函数,其中 T 为采样周期。

$$\delta(t) = \left| \frac{W_{CH}(t) - W_{CH}(t-T)}{T} \right| \quad (42)$$

由于面临簇首的更换场景,默认是簇首节点的权值随着时间推移在减小,如果簇首权值在增大也就不会进行簇首的更换,则式(42)可写为

$$\delta(t) = \frac{W_{CH}(t-T) - W_{CH}(t)}{T} \quad (43)$$

通过 $\delta(t)$ 定义阈值调整函数为 $\Phi(t)$, 如式(44)所示,其中 Φ_0 为初始阈值。由式(44)可知,通过跟踪簇首权值下降的速度,使得簇首更换阈值随着簇首权值变化率而改变。当簇首节点的权值下降速度较快时,此时说明簇首的性能下降严重,不稳定性高,则应选择较小的阈值触发簇首的更换;当簇首节点的权值下降较为平缓,簇的整体稳定性较高,通过设置较大的阈值,保证分簇的稳定性。

$$\Phi(t) = \Phi_0 \cdot e^{-\delta(t)} \Rightarrow \Phi(t) = \Phi_0 \cdot \frac{W_{CH}(t-T) - W_{CH}(t)}{T} \quad (44)$$

采样周期 T 可以根据网络节点移动速度调整,网络中节

点移动速度越快,簇成员与簇首之间的链路断开的概率增大,分簇结构的稳定性降低,采样周期 T 应该越小。同时,当节点低速移动时,簇的结构稳定性较高时, T 值应增大,减少不必要的计算开销。

通过设立预备簇首,可以实现当簇首已不再适合担任簇首后,能够从现有簇成员中快速选出最佳的簇首替代节点,有效避免簇首轮换的空白期带来的通信中断,降低簇的初始化频率,提高簇的稳定性。

3.3.3 簇的合并

当两个簇首互相之间可以接收到对方发送的 HELLO 消息时,就可能需要进行簇的合并。当某一簇首 j 及其邻居节点 $S[j]$ 是另一簇首 i 邻居节点集合 $S[i]$ 的子集时,簇首 i 成为两个合并后的簇的簇首,簇 j 消失。如果两簇的大部分已经重合,两簇首计算相互之间的链路保持时间 T_{ij} ,如果 T_{ij} 的值较大,说明两个簇会在相互的范围内停留时间较大,需要进行簇的合并。通过比较两簇首的权值 W ,选取两者中 W 值大的节点作为新的簇首。

3.4 算法复杂性分析

WSCA 算法的计算复杂度可从时间、存储两个方面表征。设网络规模为 N (即网络中节点总数为 N),网络直径为 L ,节点的平均节点度为 D 。在极端情况下,所有簇首形成一张直线图时路由搜索时间最大,在该拓扑图中每次只能搜索到一个分簇,寻找一个节点的时间复杂性为 $O(|clusterheads|)$,其中 $|clusterheads|$ 为支配集的势。节点保存有邻居表和附属成员表,邻居列表的复杂度为 $O(D)$,在极端情况下 (D 个邻节点各自有 D 个邻居节点为附属簇成员,且不重合)附属成员表复杂度为 $O(D^2)$,因此总的存储复杂度为 $O(D(D+1))$ 。

4 仿真结果分析

本文通过 OPNET14.5 网络仿真软件对 WSCA 分簇算法的性能进行仿真验证。比较对象选取 WCA^[13]、IWCA^[16] 和 AHPECA 算法^[17]。仿真区域大小设置为 $100\text{ km} \times 100\text{ km}$,节点在区域内随机分布,各个节点参数设置相同,节点的通信半径为 $15 \sim 20\text{ km}$ 。节点运动模型为 RWP 模型^[21] (暂停时间为 0),最大运动速度取值为 $50 \sim 100\text{ m/s}$,通信模型采用 CBR 数据源,数据包大小为 512 Byte,信道传输速率为 2 Mbps,信号传播损失模型为 Friis 模型,仿真运行时间为 20 min。通过改变网络中的节点个数、节点的运动速度和节点通信半径大小,设置不同的仿真场景。仿真完成后,通过分簇数、平均簇首保持时间、簇间切换次数和簇维护开销四项性能指标的对比分析,反映分簇算法性能,结果取 10 次实验的平均值。

通过改变网络中节点数量的大小,验证算法对分簇网络扩展性能的影响。如图 4 所示,在保证网络中其他条件不变时,持续增加网络节点数量,各算法的分簇数量都在增大,WCA 算法的分簇数量的增大最为明显,IWCA 与 AHPECA 算法的性能相近。而 WSCA 算法通过设立附属成员的身份,使得簇内结构更加灵活,分簇成员数量可实现动态改变,避免了过小规模簇的形成,使得簇内成员个数更加合理。同时,在簇首的选举时,WSCA 算法充分考虑到网络中最佳簇首个数,在几种算法中所提算法的分簇个数最少,且分簇数小于理想值,可能会由于簇成员个数较多,带来簇首负载过大的问题。但是,就簇的变化趋势可以看出,算法的分簇个数的增长趋势更加平稳,能够更好地应对网络中节点数量的增大,提高了网络的扩展性。

在保持节点数量为 200,通信半径为 15 km 的条件下,通过改变节点最大移动速度,比较不同分簇算法的平均簇首保持

时间。随着最大移动速度的增加,簇内成员节点运动更易移出簇首的通信范围内,不同簇首间相遇的概率增大,面临簇的合并的可能性增加。由图 5 可以看出,各分簇算法的簇首的平均保持时间在不同程度地降低,而 WSCA 算法在四种算法中始终为最大值。这是由于 WSCA 算法在进行簇首选举时,充分考虑了节点间的速度相似性和距离可靠性,同时在簇维护阶段,采用了稳定分簇保持策略,通过设置动态阈值,减少簇首的更换频率,所选簇首能更好应对节点高速移动的网络场景。

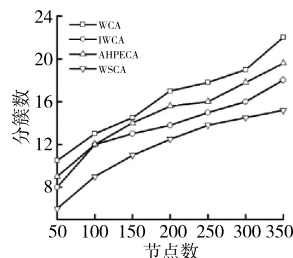


图4 节点数量对分簇数量的影响
Fig.4 Influence of nodes' number on cluster's number

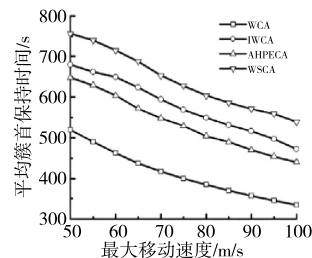


图5 平均簇首保持时间与最大移动速度的关系
Fig.5 Relationship between average cluster head holding time and maximum moving speed

在保持节点最大移动速度为 60 km/h 条件下,不同分簇算法的平均簇首保持时间受节点通信半径变化的比较,如图 6 所示。四种算法的平均簇首保持时间都在增大,主要原因是随着节点通信范围的增大,在相同的相对速度和节点分布情况下,节点间的链路连接时间随之增加,一定程度上降低了节点移动性的影响。相较于其他三种算法,WSCA 算法始终具有最大的平均簇首保持时间,提高了分簇的稳定性。

平均簇首保持时间是分簇算法性能在簇首节点的反映,而簇间切换次数是簇成员节点体现分簇算法稳定性的性能指标。因此,通过改变节点最大移动速度和通信半径,比较各算法簇间切换次数指标变化。

由图 7 可知,随着节点速度的增大,网络拓扑结构变化剧烈,四种算法的簇成员的簇间切换次数增大,WSCA 算法受节点运动速度增大的影响最小。原因在于 WSCA 算法引入了附属簇员,降低了节点簇间切换次数,在簇首选取阶段,充分考虑了节点的分布状况,节点间的链路连接时间等因素,同时根据网络的变化,调整各测度指标的权值,能够保证簇成员节点与所选簇首间关系的稳固性,降低了节点簇间切换次数。

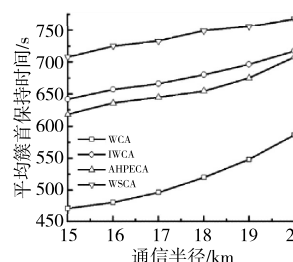


图6 平均簇首保持时间与通信半径的关系
Fig.6 Relationship between average cluster head holding time and communication radius

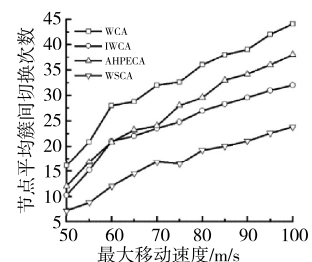


图7 平均簇间切换次数与最大移动速度的关系
Fig.7 Relationship between average number of inter cluster handoffs and maximum moving speed

随着最大通信半径增大,簇内节点能在分簇内保持更长的连接时间,更不容易移出簇首的通信覆盖范围。从图 8 可以看出,四种算法的簇间切换次数都在减少,WSCA 算法始终维持为各算法的最小值。WSCA 算法在进行簇首选举时,充分考虑了节点的具体分布情况,能有效区分出位于通信边界的节点,节点与原簇首间的关系更加稳固,减少了节点的簇间切换次数。同时,结合图 4,WSCA 算法的分簇数最少,所以簇内成

员的簇间切换次数也会相应较少。

由图 9 可知,随着网络中节点个数的增多,四种算法的簇维护开销都在增大。在节点个数相对较少时,WSCA 算法需要周期性推选预备簇首,同时,由于引入了附属簇员,使得簇内控制消息数增加,一定程度上增大了分簇的维护开销。但是,随着节点数量的增大,WSCA 算法由于形成的簇更加稳定,减少了因簇首改变引起的控制开销,相较于其他算法簇维护开销相对降低。

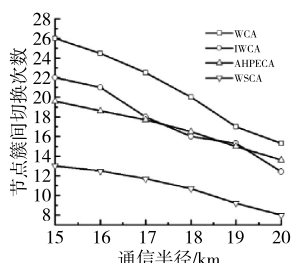


图8 平均簇间切换次数与通信半径的关系

Fig.8 Relationship between average inter cluster handoff times and communication radius

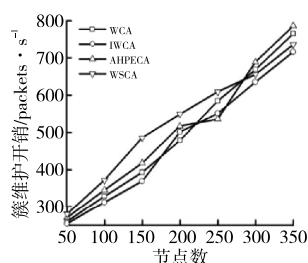


图9 节点个数与簇维护开销的关系

Fig.9 Relationship between the number of nodes and the overhead of cluster maintenance

5 结束语

针对无人机自组网具有节点移动速度更快、分布范围更广、网络规模更大等特点,并结合分簇结构在网络控制管理方面的优势,在 WCA 算法的基础上,提出了一种基于加权的稳定分簇算法。该算法从分簇测度指标选取、确定参数指标权值和分簇稳定保持策略三个关键点出发,结合无人机集群组网通信的需要进行了改进。仿真结果表明,该算法能够明显改善分簇的稳定性,提高了网络性能。下一步,在 WCA 分簇算法基础上,将对与之适应的路由协议进行研究。

参考文献:

- [1] 牛轶峰,肖湘江,柯冠岩. 无人机集群作战概念及关键技术分析[J]. 国防科技,2013,34(5):37-43. (Niu Yifeng, Xiao Xiangjiang, Ke Guanyan. Operation concept and key techniques of unmanned aerial vehicle swarms[J]. National Defense Science & Technology, 2013,34(5):37-43.)
- [2] 王祥科,刘志宏,丛一睿,等. 小型固定翼无人机集群综述和未来发展[J]. 航空学报,2019,41(4):023732. (Wang Xiangke, Liu Zhihong, Cong Yirui, et al. Miniature fixed-wing UAV swarms: review and outlook[J]. Acta Aeronautica et Astronautica Sinica, 2019,41(4):023732.)
- [3] Gupta L, Jain R, Vaszkun G. Survey of important issues in UAV communication networks[J]. IEEE Communications Surveys & Tutorials,2016,18(2):1123-1152.
- [4] Bekmezci I, Sahingoz O K, Temel S. Flying Ad hoc networks (FANETs): a survey[J]. Ad hoc Networks,2013,11(3):1254-1270.
- [5] Chriki A, Touati H, Snoussi H, et al. FANET: communication, mobility models and security issues[J]. Computer Networks,2019,163(9):106877.
- [6] 孟洛明,江彦彦,刘彦君,等. 基于相对移动性预测的 k 跳 Ad hoc 网络分簇算法[J]. 电子与信息学报,2018,40(12):163-170. (Meng Luoming, Jiang Yanfu, Liu Yanjun, et al. Relative mobility prediction based k-hop clustering algorithm in Ad hoc networks[J]. Journal of Electronics & Information Technology,2018,40(12):163-170.)
- [7] 于云龙,茹乐,方望,等. 面向高动态移动自组织网络的生物启发

分簇算法[J]. 电子学报,2018,46(4):918-929. (Yu Yunlong, Ru Le, Fang Kun, et al. Bio-inspired clustering algorithm for highly dynamic mobile ad hoc networks[J]. Acta Electronica Sinica,2018,46(4):918-929.)

- [8] Raza A, Khan M F, Maqsood M, et al. Adaptive K-means clustering for flying Ad hoc networks[J]. KSII Trans on Internet and Information Systems,2020,14(6):2670-2685.
- [9] 张文庆,李旭,黄文俊. 多跳分簇自组网络的最优簇内跳数分析[J]. 兵工学报,2018,39(11):2220-2227. (Zhang Wenqing, Li Xu, Huang Wenjun. Analysis of optimal inner-cluster hop number of multi-hop cluster-based Ad hoc network[J]. Acta Armamentarii, 2018,39(11):2220-2227.)
- [10] Yu J Y, Chong P H J. A survey of clustering schemes for mobile Ad hoc networks[J]. IEEE Communications Surveys and Tutorials, 2005,7(1):32-48.
- [11] Bentaleb A, Boubetra A, Harous S. Survey of clustering schemes in mobile Ad hoc networks[C]//Proc of Spring International Conference on Wireless Communications and Networks. [S. l.]: Scientific Research,2013:8-14.
- [12] Mehta S, Sharma P, Kotecha K. A survey on various cluster head election algorithms for MANET[C]//Proc of Nirma University International Conference on Engineering. Piscataway, NJ: IEEE Press, 2011:1-6.
- [13] Chatterjee M, Das S K, Turgut D. WCA: a weighted clustering algorithm for mobile Ad hoc networks[J]. Cluster Computing,2002,5(2):193-204.
- [14] 马豫青,李晓宇. Ad hoc 网络自适应安全加权分簇算法[J]. 计算机工程与设计,2014,9(10):3346-3350. (Ma Yuqing, Li Xiaoyu. Adaptive security weighted clustering algorithm of Ad hoc network[J]. Computer Engineering and Design, 2014,9(10):3346-3350.)
- [15] 黄金科,樊晓光,万明,等. 基于稳定分簇的移动自组织网络路由协议[J]. 北京航空航天大学学报,2016,42(11):2332-2339. (Huang Jinke, Fan Xiaoguang, Wan Ming, et al. Stable cluster-based routing protocol for mobile Ad hoc networks[J]. Journal of Beijing University of Aeronautics and Astronautics, 2016,42(11):2332-2339.)
- [16] 王沁飞,南建国,黄金科,等. 基于加权的无人机集群组网分簇算法[J]. 计算机应用研究,2019,36(5):1500-1503,1514. (Wang Qinfei, Nan Jianguo, Huang Jinke, et al. Weighting based clustering algorithm for FANET[J]. Application Research of Computers, 2019,36(5):1500-1503,1514.)
- [17] 陈德江,王君,赵崇丞. 基于 AHP-熵权分簇算法的多弹协同作战研究[J]. 火力与指挥控制,2018,43(8):53-57. (Chen Dejiang, Wang Jun, Zhao Chongcheng. Research about multi missile cooperative combat based on AHP and entropy weight clustering algorithm[J]. Fire Control & Command Control,2018,43(8):53-57.)
- [18] Wang Jian, Yang Liusong, Zhao Yafeng. Research of wireless sensor networks cross-layer energy optimization based on link quality[C]//Proc of the 3rd International Conference on Measuring Technology & Mechatronics Automation. Piscataway, NJ: IEEE Press, 2011:1092-1094.
- [19] Correlle L H A, Macedo D F, Santos A L D, et al. Transmission power control techniques for wireless sensor networks[J]. Computer Networks,2007,51(17):4765-4779.
- [20] Gupta P, Kumar P. The capacity of wireless networks[J]. IEEE Trans on Information Theory,2000,46(2):388-404.
- [21] Bujari A, Calafate C T, Cano J C, et al. Flying Ad hoc network application scenarios and mobility models[J]. International Journal of Distributed Sensor Networks,2017,13(10):1-17.