·体系结构与软件技术· 文章编号：1000-3428(2020)06-0221-09 文献标志码：A 中图分类号：TP393

面向航空信息网络的控制器可靠性部署方法

高航航，王 翔，赵尚弘，彭 聪

（空军工程大学 信息与导航学院，西安 710077）

摘 要： 结合软件定义航空信息网络架构中的控制平面可扩展性，提出一种多控制器部署方案。该方案包括集群域划分和域内控制器部署两个阶段。在集群域划分阶段，针对 k-means 算法初始聚心不稳定的缺点设计基于离散因子的改进 k-means 算法，将航空信息网络划分为多个航空集群域。在域内部署阶段，以网络控制路径故障率最小为目标，采用离散粒子群优化算法对控制器进行部署，以此实现对网络的有效管控。仿真结果表明，该方案在对航空信息网络合理划分的同时能够保证各控制器间的负载均衡，其中离散粒子群优化算法能够有效降低控制路径的故障率，解决动态及大规模网络下的多控制器部署问题。

关键词： 航空信息网络；软件定义网络；控制器部署；网络可靠性；粒子群优化

开放科学（资源服务）标志码（OSID）：

中文引用格式： 高航航，王翔，赵尚弘，等. 面向航空信息网络的控制器可靠性部署方法[J]. 计算机工程，2020，46(6)：221-229.

英文引用格式： GAO Hanghang，WANG Xiang，ZHAO Shanghong，et al. Controller reliability deployment method for aeronautical information network[J]. Computer Engineering，2020，46(6)：221-229.

Controller Reliability Deployment Method for Aeronautical Information Network

GAO Hanghang，WANG Xiang，ZHAO Shanghong，PENG Cong

（School of Information and Navigation，Air Force Engineering University，Xi'an 710077，China）

【Abstract】 To address the scalability problem of control plane in software defined aeronautical information network，this paper proposes a multi-controller deployment scheme. The scheme contains two phases，cluster domain partitioning and intra-domain controller deployment. In the cluster domain partitioning phase，to deal with the unstable initial convergence of the k-means algorithm，an improved k-means algorithm based on Discrete Factor（DF）is designed to partition the aeronautical information network into multiple aeronautical cluster domains. In the intra-domain deployment phase，a discrete particle swarm optimization algorithm is adopted for controller deployment with the goal of minimizing the failure rate of network control paths，so as to implement effective management and control of network. Simulation results show that the proposed scheme can achieve reasonable partitioning of aeronautical information network with load balancing of controllers ensured. The adopted discrete particle swarm optimization algorithm can effectively reduce the failure rate of control paths，and solve the multi-controller deployment problem in dynamic and large-scale networks.

【Key words】 aeronautical information network；Software Defined Network（SDN）；controller deployment；network reliability；Particle Swarm Optimization（PSO）

DOI： 10.19678/j.issn.1000-3428.0054565

0 概述

航空信息网络又称为机载网络、空基网络，主要通过高空航空平台进行信息发送、节点接收或转发，是以节点间的无线通信为传输链路所组成的网络，其具有高动态、覆盖范围广、作战任务多样化、接入数据量大等特点[1-2]。随着网络规模的扩大和作战环境的复杂化，当前的航空信息网络也逐渐暴露出一系列问题[3]，例如：如何对航空平台所获得的战场信息进行高效共享以及灵活调度控制；不同的作战

基金项目： 国家自然科学基金（91638101，61571461）。

作者简介： 高航航（1994—），男，硕士研究生，主研方向为航空信息网络、软件定义网络；王 翔，讲师、博士；赵尚弘，教授、博士生导师；彭 聪，博士研究生。

收稿日期： 2019-04-10 **修回日期：** 2019-05-30 **E-mail：** gaohh321@126.com

计算机工程 www.ecice06.com

需求在网络时延、可靠性方面具有不同 QoS 需求，要求网络应具备较强的差异化服务能力；软件控制和硬件转发紧密耦合的传统网络设备难以满足网络快速发展的需要。面向未来的航空信息网络应当具备提供灵活耦合任务的差异化网络服务、支持灵活高效的网络配置、新网络技术能够简单快速部署等能力，而现有的无线信息网络仅能满足有限任务背景下模式固定的信息交互需求，难以支撑未来作战中航空集群成员间的灵活协同。

软件定义网络[4]（Software Defined Network，SDN）的出现为解决上述问题提供了新思路，其通过将传统网络设备中的控制平面和数据平面相解耦，利用逻辑集中控制的 SDN 控制器对底层转发设备进行统一管控，控制平面负责策略制定与资源调配，底层转发设备进行数据业务的转发，有效提高了网络的信息处理和管理控制能力。然而集中式的控制平面存在单点失效、可靠性低、处理能力受限等缺点，实际中需要重点考虑控制平面的可扩展性，采用物理上分布、逻辑上集中的多控制器部署架构已成为目前解决控制平面可扩展性的重要方法。

当前对于 SDN 多控制器的部署研究主要以地面网络为主，评估指标包括网络时延、可靠性、流量开销等[5]。文献[6]针对 SDN 中控制器部署问题进行研究，首先定义平均传输时延和最大传输时延，在此基础上利用实际网络拓扑进行分析，得到控制器数量对网络时延的影响以及最优时延和平均时延的比较结果，但文中仅考虑了网络传输时延。文献[7]在传输时延基础上增加发送时延以完善现有的时延模型，并对该模型是否存在最优解进行证明，针对是否发送时延分别提出传输算法和输送算法。文献[8]针对传统静态部署方案难以应对高动态网络拓扑和流量变化这一问题，采用一种基于双门限的交换机动态迁移策略以解决控制器失效和资源浪费问题，结果表明该算法在提升系统吞吐量的同时可保证控制器间的负载均衡。文献[9]以航空网络的全网中断概率最小为目标，提出一种融合人工免疫策略、小生境思想以及改进遗传算法的混合优化算法，仿真表明该算法在获得更优值的同时其收敛时间也得到一定减少，但该文未考虑控制器的负载均衡问题。文献[10]针对控制路径平均故障率提出有效控制路径预期百分比这一指标，通过最大化预期百分比实现网络控制路径的强健壮性，但平均故障率仅能反映网络整体故障，无法反映网络的最坏状态。

航空信息网络具有高动态网络拓扑、远距离传输范围、大尺度节点分布以及不稳定的链路质量，因此，传统基于地面网络的多控制器架构、部署算法等不再适用。本文在总结航空信息网络特点的基础上，对传统平面式和层次式 SDN 多控制器架构进行改进，设计一种混合式多控制器部署架构。针对多控制器负载不均衡问题，提出改进的集群域划分算法，并以网络节点和链路中断概率为变量，将网络控制路径故障率最小作为优化目标，设计改进的离散粒子群优化算法实现多控制器部署。

1 模型分析

1.1 软件定义航空信息网络架构特点

航空信息网络位于天基网络和地面网络之间，如图 1 所示，其对上可与天基平台建立信息链路，实现天基信息及时注入航空信息网络，为航空平台提供信息支持，对下可与地面信息系统建立信息链路以保证地面指控信息的输入，实现综合信息处理分发、空中指挥控制和协同传输。目前航空信息网络正朝着网络异构化、业务多样化、功能复杂化的方向发展，加之网络节点的移动性、链路的不稳定性，其对当前的网络性能提出更高的要求。现有的无线信息网络虽然在传输可靠性、端到端时延、传输速率等网络传输性能指标上能够支撑现有的航空网络进行一定程度的作战协同，但其本质上并非契合未来航空集群作战的应用背景，网络仅能满足有限任务背景下模式固定的信息交互需求，与作战任务缺乏灵活的耦合关系，而构建一个基于 SDN 的航空信息网络将更符合未来作战对网络各方面性能的要求。

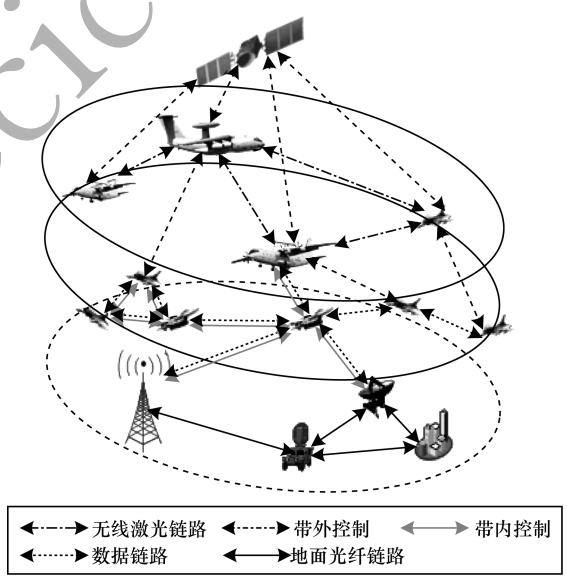


图 1 航空信息网络示意图

Fig. 1 Schematic diagram of aviatial information network

软件定义航空信息网络[3,11]利用 SDN 逻辑集中控制的网络管控策略，能够实时掌握航空信息网络全局视图，实现对航空信息网络中业务流量的优化调度，满足网络中多用户需求，改善整体性能。控

www.ecicee.com

制器部署作为构建控制平面的前提，对提升网络性能具有重要意义，而控制平面中单控制器通常存在单点失效、处理能力受限等问题，因此，多控制器部署架构已经成为目前有效的解决方案。多控制器部署架构包括平面式架构[12]、垂直式架构[13]和混合式架构[14]。由于高空节点的移动性和链路的不可靠性易造成网络通信中断，在瞬息万变的战场中控制器节点也存在一定的故障风险，因此地面网络中的平面式多控制器架构将不再适用。考虑到上述因素，航空信息网络应采用混合式的控制器架构对网络进行集中管控，本文借鉴文献[15]所构建的混合式控制器架构模型，结合具体的应用场景设计一种航空信息网络下的混合式多控制器架构，如图 2 所示。

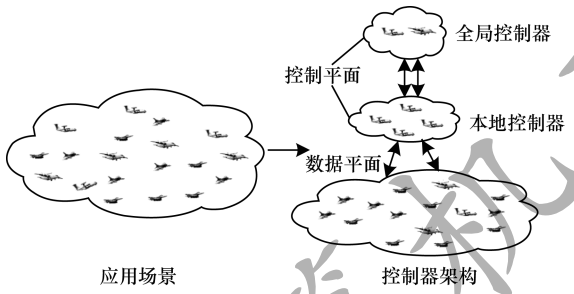


图 2　混合式多控制器架构示意图

Fig. 2　Schematic diagram of hybrid multi-controller architecture

混合式多控制器架构中的控制平面由全局控制器(Global Controller，GC)和本地控制器(Local Controller，LC)组成，其中：GC 可从全域战场视角对航空信息网络实施集中管控，应部署在信息综合处理能力和生存能力较强的飞机节点上，如指通机、预警机等，其控制优先级最高；LC 负责管控其自身控制区域内的网络节点，考虑到航空信息网络的实际需求及特点，可在每个航空平台上布置 LC，实际中网络根据自身状态及 GC 的部署策略开启或关闭相应的 LC，实现对其动态部署。

1.2　数学建模

软件定义航空信息网络中的多控制器部署问题描述如下：

1)$G(V,E,V_c,E_c)$表示航空信息网络拓扑，其中，V 代表网络中飞机节点集合，E 代表飞机节点间的通信链路集合，V_c 代表网络中部署的控制节点集合，E_c 代表控制路径集合，且 $V_c \subseteq V, E_c \subseteq E$。

2)本文假定已知混合式多控制器架构下 GC 的部署个数和位置，仅对控制平面中的 LC 进行部署，下文所提的控制节点与 LC 节点均为同一概念。

3)考虑到实际情况，网络中所有飞机节点均应布置控制器，控制器按照具体部署策略相应打开或关闭。当飞机节点 i 上的控制器打开时，节点 i 为控制节点；当控制器关闭时，节点 i 为交换节点，也称作普通传输节点，网络中任意节点均有机会成为控制节点或交换节点。

4)控制路径包括 LC 与 GC 节点之间的路径以及交换节点与 LC 节点间相连的路径。由于 GC 的优先级最高，因此 GC 之间单独配置控制路径，实现对网络视图信息的共享。其余控制路径为带内方式，不单独配置控制路径，即控制信息和数据信息通过相同的路径进行传输。

5)假定网络中节点和链路发生故障的概率均是独立的，p_o 和 p_v 分别表示链路中断概率和节点失效概率，有 $0 \leqslant p_o < 1, 0 \leqslant p_v < 1$。

6)网络中每个 LC 在同一时间内只能由唯一的 GC 控制，每个交换节点也只能由唯一的 LC 控制。

为方便描述多控制器部署问题，现对具体数学模型中的符号进行说明，如表 1 所示。

表 1　符号说明

Table 1　Symbol description

数学符号	实际含义
v	网络节点
e	网络链路
p_v	节点失效概率
p_o	链路中断概率
g	某 GC 节点
l	某 LC 节点
k	部署的控制器个数
h_{ij}^{v}	节点 v 位于节点 i,j 间路径上
h_{ij}^{e}	链路 e 位于节点 i,j 间路径上
c_l	控制器部署在节点 l
b_{tj}	节点 t 受节点 j 管控
h_{tl}	节点 t 受控制节点 l 管控

根据上述分析，可将软件定义航空信息网络中的多控制器部署问题建立如下目标函数：

$$\min f = f_1 + f_2 \tag{1}$$

$$f_1 = 1 - \prod_{\substack{v,t \in V, l \in V_c \\ e \in E}} h_{tl}^{v} h_{tl}^{e} (1 - p_v)(1 - p_o) \tag{2}$$

$$f_2 = 1 - \prod_{\substack{v \in V, e \in E \\ g \in \{GC\}, l \in V_c}} h_{gl}^{v} h_{gl}^{e} (1 - p_v)(1 - p_o) \tag{3}$$

$$\text{s.t. } p_v, p_o \in [0,1); \forall v \in V, e \in E \tag{4}$$

$$b_{tj} c_l = h_{tl}; l \in V_c, t, j \in V \tag{5}$$

$$\sum_{l \in V_c} c_l = k \tag{6}$$

$$\sum_{j \in V_c} b_{tj} = 1; \ \forall t \mid c_t = 0, t \in V \tag{7}$$

$$h_{tl}^{v}, h_{tl}^{e}, h_{gl}^{v}, h_{gl}^{e}, b_{tl}, c_l \in \{0,1\}; v, t \in V$$
$$l \in V_c, e \in E, g \in \{GC\} \tag{8}$$

www.ecice06.com

式(1)表示最小化控制路径故障率;式(2)表示LC节点和交换节点间的控制路径故障率;式(3)表示LC节点与GC节点间的控制路径故障率;式(4)表示节点、链路失效概率取值范围;式(5)表示节点 t 受控制节点 l 所管控;式(6)表示网络中部署的控制器个数;式(7)表示一个交换节点只能由一个LC节点管控;式(8)表示各式中变量的取值范围。

2 算法设计

2.1 航空集群域划分算法

在航空信息网络中存在着功能各异的飞机节点,本文参考文献[15],假定在执行某作战任务过程中其网络拓扑保持相对稳定,随着作战任务变化,网络拓扑也相应改变,本文在此基础上对航空信息网络进行进一步集群域划分。

2.1.1 k-means 聚类算法

k-means是一种简单高效、快速实现的基于划分原理的聚类算法,其将数据集 $U=\{x_1,x_2,\cdots,x_n\}$ 按照某种准则划分为若干个子集,其中 k 为聚类数目,且聚类满足以下约束:

$$U_j\neq\varnothing,j=1,2,\cdots,k \tag{9}$$

$$U_i\cap U_j=\varnothing,i,j=1,2,\cdots,k \text{ and } i\neq j \tag{10}$$

图3所示为传统k-means算法流程,其中输入为数据集 U 和聚类个数 k(见式(11)),输出为聚类划分结果,算法结束的条件为聚类中心不再发生变化或变化范围在规定的阈值内。

$$\bigcup_{j=1}^{k}U_j=U \tag{11}$$

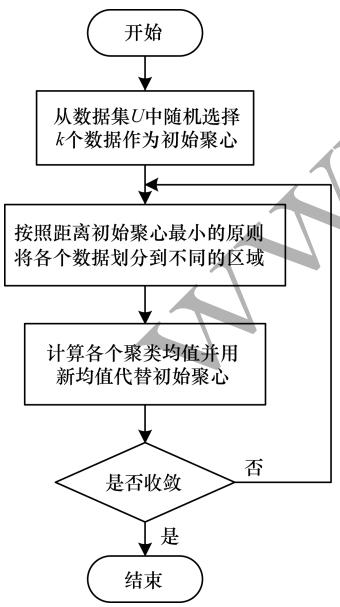


图3 传统k-means算法流程

Fig. 3 Procedure of traditional k-means algorithm

由于k-means算法是随机选择初始聚类中心的,聚类结果会随初始聚心的改变而改变,因此本文引入聚类质量评估函数作为聚类效果的评价准则,其中对聚类质量评估函数定义如下:

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n_j}d(x_i,z_j),x_i\in U_j \tag{12}$$

$$z_j=\frac{1}{n_j}\sum_{i=1}^{n_j}x_i \tag{13}$$

式(12)表示各聚类的内距离和,J 值越小则代表聚类的效果越优;式(13)表示集群域 j 的聚心。

2.1.2 改进 k-means 聚类算法

传统k-means聚类算法是一种贪婪算法,容易陷入局部最优,并且该算法选择的初始聚心极有可能偏离数据密集区,若初始聚心位于孤点或者偏远点,则会导致划分的集群域性能变差。针对该问题,本文参考文献[16-17]中提出的离群因子对k-means算法初始聚心位置随机这一不足加以改进,设计一种基于离散因子(Discrete Factor,DF)的改进k-means算法,通过计算各节点的离散因子值,并选择合适的离散因子值所对应的节点作为集群域初始中心,避免网络中的孤点或离散点成为初始聚心,最终得到的集群域也将更加合理。算法描述具体如下:

定义1 数据 x 的第 k 距离

x 的第 k 距离 $d_k(x)$ 指在数据集合 U 中存在数据 o,数据 o 与数据 x 之间的距离记为 $d(o,x)$,当满足下述条件时,x 的第 k 距离 $d_k(x)$ 等于 $d(o,x)$:

1)数据集 U 中至少存在 k 个数据 $o'\in D\backslash\{x\}$ 满足 $d(x,o')\leqslant d(x,o)$ 。

2)数据集 U 中至多存在 $k-1$ 个数据 $o'\in D\backslash\{x\}$ 满足 $d(x,o')<d(x,o)$。

定义2 数据 x 的第 k 距离邻域

数据 x 的第 k 距离邻域指不超过数据 x 的第 k 距离内所有数据的集合,其表达式如式(14)所示,可以看出,若该集合较大,则说明数据 x 的第 k 距离邻域较大,表明数据偏离程度较大,反之偏离程度较小。

$$N_k(x)=\{o\in D\backslash\{x\}\mid d(x,o)\leqslant d_k(x)\} \tag{14}$$

定义3 数据 x 的离散因子

数据 x 的离散因子 $D_k(x)$ 指在 (x) 的第 k 距离范围内,集合 U 中数据的平均分布密度大小,$D_k(x)$ 值越大,则 x 越可能成为离群点,其计算公式如下:

$$D_k(x)=\frac{\sum_{o\in N_k(x)}\frac{l_k(o)}{l_k(x)}}{|N_k(x)|} \tag{15}$$

计算机工程 www.ecice06.com

$$l_k(x)=1\Big/\left(\frac{\sum_{o\in N_k(x)}\max\{d_k(o),d(p,o)\}}{|N_k(x)|}\right) \tag{16}$$

本文利用欧式距离计算出数据 x 的第 k 距离大小，并根据式(14)得到数据 x 的第 k 距离邻域，最后在 x 的邻域内计算出其余节点相对 x 的平均分布状况，依次对集合 U 内数据进行计算，得到集合 U 内所有节点的平均分布密度大小。本文基于离散因子的改进 k-means 算法描述如下，其通过计算集合 U 中各数据的离散因子值，找出离散因子值较小的节点作为候选初始聚心集合，在该集合中通过计算距离进而得出最终的聚心位置。

算法 1　改进的 k-means 算法

输入　数据集 U，聚类个数 k，取样系数 α

输出　聚类结果 $U_j(\ j=1,2,\cdots,k\ )$

1. for each $x_i \in U$

2. 计算 x_i 的 DF 值，并将计算结果按照升序排列到 ListDF 中

3. 选择 ListDF 中前 $n\times\alpha$ 个数据，生成候选初始聚心集合 CandidateClusterSet

4. 令 $C=\varnothing$，在 CandidateClusterSet 中选择 DF 值最小的数据 x_i 作为初始聚心 C_1，$C=\{C_1\}$，并从 CandidateClusterSet 中删除该数据，Update CandidateClusterSet

5. for each $x_i \in$ CandidateClusterSet

6. 计算数据 x_i 与 C_1 之间的欧氏距离，选择距离最大的数据点作为第 2 个初始聚心 C_2，$C=\{C_1,C_2\}$，并从 CandidateClusterSet 中删除该数据，Update CandidateClusterSet

7. for each $x_i \in$ CandidateClusterSet

8. 分别计算 x_i 与 $\{C_1,C_2\}$ 之间的距离，将计算结果相加，选择最大值所对应的数据作为下一个初始聚心 C_3，$C=\{C_1,C_2,C_3\}$，并从 CandidateClusterSet 中删除该数据，Update CandidateClusterSet

9. Repeat and Update C

10. until　$i>k$

11. 输出 k 个初始聚心结果 $u_j(\ j=1,2,\cdots,k)$

12. 从得到 k 个初始聚心出发，利用最短路径算法求出聚类结果

根据算法 1 可得到最终的聚心分布结果，相比于传统 k-means 算法，该算法有效地避免了聚心位于孤点或偏远点，使得聚心分布更加合理。算法在构造集群域时主要考虑节点的分布特点，而在实际部署过程中由于 LC 存在处理能力受限等问题，易造成某 LC 负载过载或欠载，从而引起数据拥塞或不能充分利用资源等问题，本文在此基础上加入 LC 负载受限这一约束条件对集群域划分结果进行优化调整。通过引入负载均衡指数 B 对各集群域间的负载是否均衡进行计算判断，其表达式如下：

$$B=\max_{p}\sum_{x_i\in U_p}a(x_i)-\min_{q}\sum_{x_j\in U_q}a(x_j) \tag{17}$$

$$\sum_{x_i\in U_p}a(x_i)\leqslant A(p) \tag{18}$$

其中，$a(x_i)$ 表示节点 x_i 向其集群域内的 LC 提交的数据请求信息，且集群域内所包含节点提交的总数据请求信息不能超过该集群域内 LC 的负载能力。LC 负载约束的集群域划分算法描述如下：

算法 2　LC 负载约束的集群域划分算法

输入　初始聚心结果 u_j，负载均衡指数 B

输出　LC 负载约束的聚类划分结果 U_j

1. 利用改进 k-means 算法求出初始聚心。

2. 按照最短距离算法将 x_i 分配给各 u_j，形成集群域 U_j。

3. 计算各集群域内 LC 负载以及负载均衡指数 B，并判断 B 是否符合预先设置值，若符合，则转到步骤 5，否则执行步骤 4。

4. 将 LC 过载对应的集群域内节点分配至距离次短的集群域内，并转到步骤 3。

5. 得到 LC 负载约束的集群域划分结果。

2.2　集群域内的控制器部署算法

将航空信息网络划分为多个集群域后，在集群域中以控制路径总故障率最小为目标进行 LC 部署，其目标函数如式(1) ~ 式(8)所示，本文采用一种改进的离散粒子群算法 BPSO 对目标函数进行求解。

2.2.1　离散粒子群算法

离散粒子群优化算法是一种基于群体智能理论的优化算法，其用空间中的粒子表示问题的解，并根据适应度函数判断粒子的好坏，粒子依据个体最优和全局最优进行位置更新，具有收敛速度快、易实现等特点[18-20]。BPSO 算法中的粒子 X_i 由 d 维二进制编码组成，将粒子 $\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{id})$ 的每一维限制为 0 或 1，其速度 $V_i=(v_{i1},v_{i2},\cdots,v_{id})$ 不加限制，每个位根据式(19)改变速度，改变后的速度转换为位变量取 1 的概率，通常利用式(20)中的 sigmoid 函数计算该概率值，在算法搜索过程中，粒子 X_i 通过自身和种群状态对其位置动态调整，其每一维的速度和位置计算公式如下：

$$v_d=w\times v_d+c_1\times r_1\times(p_d^{\mathrm{Best}}-x_d)+c_2\times r_2\times(g_d^{\mathrm{Best}}-x_d) \tag{19}$$

$$S(v_d)=\frac{1}{1+\exp(-v_d)} \tag{20}$$

$$x_d=\begin{cases}1, & r\leqslant S(v_d)\\ 0, & \text{其他}\end{cases} \tag{21}$$

式(19)表示粒子的速度改变式，其中，w 表示粒子自身的惯性权重，c_1 和 c_2 表示加速因子，r_1 和 r_2 均表示区间[0,1]内的随机数，且两者独立，p_d^{Best} 和 g_d^{Best} 分别为粒子迭代过程中的个体历史最优值和全局历史最优值，d 表示维数。式(20)和式(21)表示用粒子速度更新该粒子的位置，其中用式(20)计算 x_d 位置取 1 的概率，并通过式(21)改变自身位置，而利用 sigmoid 函数仅能体现出位取 1 的概率，不能体现出位改变的概率[21]。此外，BPSO 算法计算 x_d 位

计算机工程 www.ecice06.com

值时会受前一次迭代结果的影响,并且随着迭代次数的增多,算法的随机性将增大,因此,该算法不能够达到理想的结果。

2.2.2 改进的离散粒子群优化算法

由上述分析可知,BPSO 算法中的 sigmoid 函数仅能求解出粒子位取 1 或者取 0 的概率,无法求出粒子位的变化值大小。在实际生活中,人们在解决问题时通常会依赖以往的个人历史经验和社会历史经验[22]。结合上述分析,本文提出一种改进的离散粒子群算法进行求解。

与传统粒子群算法不同,由于 p_i^{Best} 和 g_i^{Best} 中已包含粒子的历史最优信息,具有较高的可信度,因此每次粒子位的变化均取决于 p_i^{Best} 和 g_i^{Best} 的值,并可以通过设置合理的 p_i^{Best} 和 g_i^{Best} 值以避免对其过度参考。针对集群域中的 LC 部署问题,考虑用每个粒子表示一种 LC 部署方案,假设网络中共有 n 个节点,对于粒子 $\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{in})$,$x_{im}$ 表示在第 i 种部署方案中节点 m 是否为控制节点。粒子根据 p_i^{Best} 和 g_i^{Best} 对其位置进行更新,并定义节点 i 在第 $T+1$ 次位置更新时成为控制节点的概率为 $P(x_i^{T+1}=1)$,否则为 $P(x_i^{T+1}=0)$,假设 p_i^{Best} 和 g_i^{Best} 在寻找最优值的过程中是独立的,其中 p_i^{Best} 的信任度为 p_p,g_i^{Best} 的信任度为 p_g,则可求出其具体的概率值为:

$$P(x_i^{T+1}=1\mid p_i^{\text{Best T}}=1,g_i^{\text{Best T}}=1)=\frac{p_p\times p_g}{p_p\times p_g+(1-p_p)\times(1-p_g)}=\alpha \tag{22}$$

$$P(x_i^{T+1}=1\mid p_i^{\text{Best T}}=0,g_i^{\text{Best T}}=0)=\frac{(1-p_p)\times(1-p_g)}{p_p\times p_g+(1-p_p)\times(1-p_g)}=1-\alpha \tag{23}$$

$$P(x_i^{T+1}=1\mid p_i^{\text{Best T}}=1,g_i^{\text{Best T}}=0)=\frac{p_p\times(1-p_g)}{p_p\times p_g+(1-p_p)\times(1-p_g)}=\beta \tag{24}$$

$$P(x_i^{T+1}=1\mid p_i^{\text{Best T}}=0,g_i^{\text{Best T}}=1)=\frac{(1-p_p)\times p_g}{p_p\times p_g+(1-p_p)\times(1-p_g)}=1-\beta \tag{25}$$

由于 p_i^{Best} 和 g_i^{Best} 均为粒子迭代过程中的历史最优值,因此其信任度高于平均值,即有 $p_p>0.5$,$p_g>0.5$,同时为避免算法陷入局部最优,应保证 $p_p>p_g$,本文参考文献[20]中设置 $p_p=0.8$,$p_g=0.7$,此时算法可获得最优性能,式中 α 表示当 p_i^{Best} 和 g_i^{Best} 相同时两者对 x_i 取值的影响,β 表示当 p_i^{Best} 和 g_i^{Best} 不同时两者对 x_i 取值的影响。与改进的 BPSO 算法相比,BPSO 需要先计算粒子的速度,在此基础上利用 sigmoid 函数获得位为 1 的概率,而改进的 BPSO 算法则直接利用式(22)~式(25)计算位为 1 的概率,其粒子搜索效率更高,并且改进的 BPSO 算法直接利用 p_i^{Best} 和 g_i^{Best} 对位的取值概率进行确定,有效地避免了前一代位值对后期取值的影响。在集群域内部署控制器时,该算法的优势在于仅需要求出最优粒子即可得到最优的域内部署方案,粒子的每一维仅包括 0 和 1,代表网络中的控制器部署与否,其与实际问题结合更加合理。

在寻找最优解的过程中,粒子的好坏程度用适应度函数来评价,本文用式(1)作为改进 BPSO 算法的适应度函数,即 $F=f$,若某粒子的适应度函数值小,则代表该粒子所对应的解更优,否则较差,集群域内的 LC 部署算法描述如下:

算法 3 改进 BPSO 算法的控制器部署算法

输入 各集群域 U_j

输出 控制器部署方案

1. 设置粒子种群规模 N,迭代次数 T_{max},控制器数目 k。

2. 初始化粒子位置 X_i,速度 V_i,个体最优 p_{id} 并计算全局最优 p_{gd}。

3. 计算当前时刻粒子的适应度值 F。

4. 比较当前时刻粒子的适应度值 F 与个体最优下的适应度值和全局最优下的适应度值大小,判断是否更新个体最优和全局最优。

5. 更新粒子速度 V_i 并根据表 2 更新粒子位置 X_i。

6. 是否达到迭代次数上限 T,若未达到,则执行步骤 3。

7. 判断控制器数目 $i<k$ 是否成立,若成立则执行步骤 2。

8. 得到不同的控制器个数 k 以及不同的 LC 部署方案。

上述 LC 部署算法根据式(19)对粒子速度进行更新,在对粒子位置更新时采用如下策略:

1. 产生随机数 r = rand()

2. if $p_i^{\text{Best}}=1$ and $g_i^{\text{Best}}=1$

if $r<\alpha$ then $x_i=1$ else $x_i=0$

3. else if $p_i^{\text{Best}}=0$ and $g_i^{\text{Best}}=0$

if $r<1-\alpha$ then $x_i=1$ else $x_i=0$

4. else if $p_i^{\text{Best}}=1$ and $g_i^{\text{Best}}=0$

if $r<\beta$ then $x_i=1$ else $x_i=0$

5. else

if $r<1-\beta$ then $x_i=1$ else $x_i=0$

3 仿真结果分析

3.1 仿真设置

本文设置航空信息网络的范围为 600 km × 600 km,为体现出网络的动态性,在该范围内分别随机生成 2 种大小规模不同的网络拓扑,小规模网络包括 36 个节点和 84 条链路,大规模网络包括 72 个节点和 176 条链路。假定只有一个 GC,且 GC 的位置随机指定,并忽略 GC 的数量和位置对网络性能和部署结果造成的影响。在部署 LC 过程中,假定所有 LC 的处理能力和负载容量均相同,所有传输节点

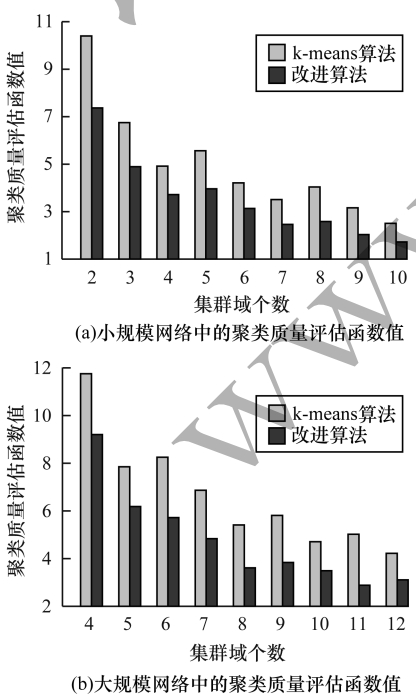
www.ecice06.com 计算机工程

向 LC 传输的请求信息量也相同，节点和链路发生故障的概率分别为[0，0.03]和[0，0.05]内的随机数。此外，为体现 2 种不同规模网络中部署 LC 的差异性，在小规模中的每个集群域内仅部署一个控制器，而在大规模网络中每个集群域内部署若干个控制器。

在仿真中同时设置 Random 算法、ANIGA 算法[9]和 Survivor[23]算法进行性能对比。Random 算法在网络中随机选择节点部署 LC；ANIGA 是一种基于启发式的随机搜索算法，其通过循环迭代找出满足中断概率条件时的 LC 数量和位置。Survivor 算法通过选择不相交路径最多的节点部署 LC，并按最短距离为其分配交换机节点。此外，为排除干扰因素，仿真中将实验重复 20 次，并计算结果的平均值作为最终结果输出。

3.2　结果对比与分析

图 4(a)和图 4(b)分别为传统 k-means 算法和改进算法在 2 种不同网络中的聚类质量评估函数值，可以看出，改进算法的 J 值波动范围与 k-means 算法相比较小，且整体 J 值均低于 k-means 算法，这是由于在改进算法中加入了负载均衡模块，划分的每个集群域中节点个数差异较小，进而计算出的 J 值波动范围较小。此外，由于改进算法在选择初始聚心时充分考虑到网络中各节点的离散系数，有效地避免了孤点或离散点成为初始聚心，进而计算出的 J 值更小，说明了采用改进算法划分的集群域更合理。



(a)小规模网络中的聚类质量评估函数值

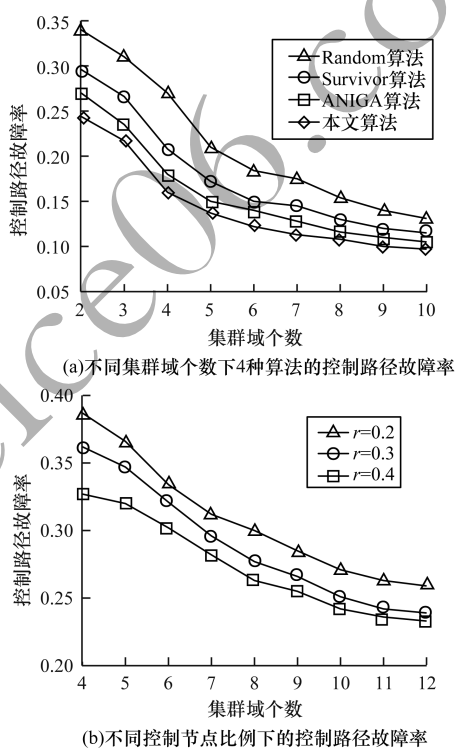
(b)大规模网络中的聚类质量评估函数值

图 4　聚类质量评估函数值

Fig. 4　Value of clustering quality evaluation function

图 5(a)显示了在小规模网络中控制路径故障率随集群域个数变化的情况，可以看出，随着集群域个数的增加，4 种算法下的控制路径故障率均逐渐降低，如集群域个数为 2、6、10 时对应的控制路径故障率分别为 0.246、0.124、0.093，这是由于随着集群域个数的增加，小规模网络中的控制器数量随之增加，控制节点与交换节点间的控制路径增多，进而控制路径的可靠性增大。

图 5(b)显示了在大规模网络中控制节点所占比例对网络控制路径故障率的影响，可以看出，在各集群域中控制节点比例逐渐增加的同时，控制路径故障率相应地减小，如集群域个数为 4 时，当控制节点比例为 0.2、0.3 和 0.4 时所对应的控制路径故障率分别为 0.386、0.363、0.327，可见在集群域中通过增加控制器的个数能够有效地降低网络控制路径故障率。



(a)不同集群域个数下4种算法的控制路径故障率

(b)不同控制节点比例下的控制路径故障率

图 5　控制路径故障率

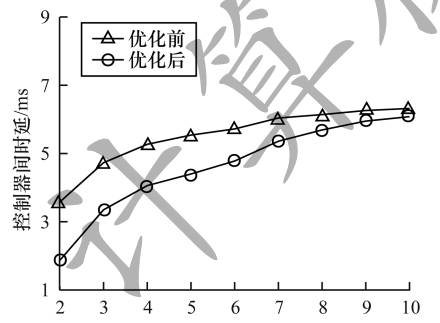
Fig. 5　Failure rate of control path

图 6(a)显示了小规模网络下控制器间的数据同步时延，可以看出，随着集群域个数的增多，控制器间同步时延呈现上升的趋势，如优化前和优化后同步时延分别增加了 2.741 ms 和 4.212 ms，这是由于随着集群域数量的增加，网络中的控制器个数随之增加，控制器节点间的控制路径增加，进而造成网络中的同步时延增大。此外，随着集群域个数的增多，集群域中节点数量减少，控制器部署位置的差异

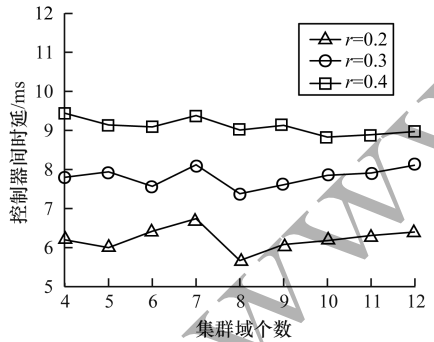
www.ecice06.com

度也随之减小,从而优化前后控制器间同步时延的差距也逐渐缩小。

图 6(b)显示了大规模网络下不同控制器数量对控制器间时延的影响,可以看出,随着集群域个数的增加,图中曲线的变化趋势较为缓慢,这是由于当集群域个数较少时,各集群域内的节点数量较多,其集群域内的控制节点也相对较多,而当集群域个数增多时,集群域内的节点数量减少,在同等控制节点数量比例下其集群域内的控制节点也相应减少。因此,尽管集群域个数在逐渐增多,但其控制器总数量与集群域个数较少时网络中的控制器总数量相差不大,意味着在网络中不能一味地追求集群域的数量,如图中集群域个数 $k=8$ 时网络中的控制器间同步时延最小,在该集群域数量下可获得较小的时延开销。此外,在相同的集群域个数下,随着网络中控制节点比例的增加,控制路径数量也随之增加,进而使得控制器间同步时延开销增大,如当控制节点比例从 0.2 增加到 0.4 时,其控制器间平均时延值分别增加了 1.608 ms 和 2.896 ms。



(a)小规模网络中的控制器间时延



(b)不同控制节点比例下的控制器间时延

图 6 控制器间时延

Fig. 6 Delay between controllers

4 结束语

本文针对软件定义航空信息网络架构中的控制平面可扩展性问题展开研究,将航空信息网络划分为多个集群域。在划分集群域时,为避免集群域中心位于孤点或偏远点,提出一种基于离散因子的改进 k-means 算法,结果表明该算法得到的聚心分布更合理。在集群域内 LC 部署方面,以控制路径故障率最小为优化目标设计改进的 BPSO 算法。仿真结果验证了本文在解决控制器部署问题方面的有效性。本文重点考虑控制器无故障下的部署方案,下一步将针对控制器的可生存性进行研究,以确保控制器工作的可连续性。

参考文献

[1] SCHNELL M, EPPLE U, SHUTIN D, et al. LDACS: future aeronautical communications for air-traffic management[J]. IEEE Communications Magazine, 2014, 52(5): 104-110.

[2] KWAK K, SAGDUYU Y, YACKOSKI J, et al. Airborne network evaluation: challenges and high fidelity emulation solution[J]. IEEE Communications Magazine, 2014, 52(10): 30-36.

[3] ZHAO Shanghong, CHEN Kefan, LÜ Na, et al. Software defined airborne tactical network for aeronautic swarm[J]. Journal on Communications, 2017, 38(8): 140-155. (in Chinese)
赵尚弘,陈柯帆,吕娜,等. 软件定义航空集群机载战术网络[J]. 通信学报, 2017, 38(8): 140-155.

[4] KREUTZ D, RAMOS F, VERISSIMO P, et al. Software-defined networking: a comprehensive survey[J]. Proceedings of the IEEE, 2015, 103(1): 14-76.

[5] GAO Xianming, WANG Baosheng, DENG Wenping, et al. Survey of controller placement problem in software defined network[J]. Journal of Communications, 2017, 38(7): 155-164. (in Chinese)
高先明,王宝生,邓文平,等. SDN 网络中控制器放置问题综述[J]. 通信学报, 2017, 38(7): 155-164.

[6] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 473-478.

[7] YAO Linyuan, CHEN Ying, SONG Fei, et al. Delay-aware controller placement for fast response in software-defined network[J]. Journal of Electronics and Information Technology, 2014, 36(12): 2802-2808. (in Chinese)
姚琳元,陈颖,宋飞,等. 基于时延的软件定义网络快速响应控制器部署[J]. 电子与信息学报, 2014, 36(12): 2802-2808.

[8] YANG Li, KONG Zhixiang, SHI Huaifeng. Multi-controller dynamic deployment strategy of software defined spatial information network[J]. Computer Engineering, 2018, 44(10): 58-63. (in Chinese)
杨力,孔志翔,石怀峰. 软件定义空间信息网络多控制器动态部署策略[J]. 计算机工程, 2018, 44(10): 58-63.

www.ecice06.com 计算机工程

[9] ZHAO Jing, ZHAO Shanghong, ZHAO Weihu, et al. Strategy of controller placement in high reliability software defined airborne optical information network[J]. Systems Engineering and Electronics, 2018, 40(4): 885-890. (in Chinese)
赵静,赵尚弘,赵卫虎,等. 高可靠性软件定义航空光信息网络的控制器分布优化策略[J]. 系统工程与电子技术,2018,40(4):885-890.

[10] HU Yannan, WANG Wendong, GONG Xiangyang, et al. On the placement of controllers in software defined networks[J]. The Journal of China Universities of Posts and Telecommunications, 2012, 19(S2): 92-97.

[11] THANGAMURUGAN K A. Software Defined Networking (SDN) for aeronautical communications[C]//Proceedings of Digital Avionics Systems Conference. Washington D. C., USA: IEEE Press, 2013: 1-20.

[12] TOOTOONCHIAN A, GANJALI Y. HyperFlow: a distributed control plane for OpenFlow[C]//Proceedings of Internet Network Management Conference on Research on Enterprise Networking. [S.l.]: USENIX Association, 2011: 3-8.

[13] YEGANEH S H, GANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]// Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks. New York, USA: ACM Press, 2012: 19-24.

[14] FU Yonghong, BI Jun, CHEN Ze, et al. A hybrid hierarchical control plane for flow-based large-scale software-defined networks[J]. IEEE Transactions on Network and Service Management, 2015, 12(2): 117-131.

[15] LÜ Na, LIU Chuang, CHEN Kefan, et al. A method for centralized control network deployment of aeronautic swarm[J]. Acta Aeronautica et Astronautica Sinica, 2018, 39(7): 172-184. (in Chinese)
吕娜,刘创,陈柯帆,等. 一种面向航空集群的集中控制式网络部署方法[J]. 航空学报,2018,39(7):172-184.

[16] BREUNIG M M, KRIEGEL H P, NG R T. LOF: identifying density-based local outliers[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2000: 93-104.

[17] TANG Dongkai, WANG Hongmei, HU Ming, et al. Optimizing initial cluster center of improved K-means algorithm[J]. Journal of Chinese Computer Systems, 2018, 39(8): 1819-1823. (in Chinese)
唐东凯,王红梅,胡明,等. 优化初始聚类中心的改进 K-means 算法[J]. 小型微型计算机系统,2018,39(8):1819-1823.

[18] CAI Qing, GONG Maoguo, MA Lijia, et al. Greedy discrete particle swarm optimization for large-scale social network clustering[J]. Information Sciences, 2015, 316(C): 503-516.

[19] QIN Jin, LI Xin, YIN Yixin. An algorithmic framework of discrete particle swarm optimization[J]. Applied Soft Computing, 2012, 12(3): 1125-1130.

[20] XU Yichun, XIAO Renbin. An improved binary particle swarm optimizer[J]. Pattern Recognition and Artificial Intelligence, 2007, 20(6): 788-793. (in Chinese)
徐义春,肖人彬. 一种改进的二进制粒子群算法[J]. 模式识别与人工智能,2007,20(6):788-793.

[21] LIU Jianhua, YANG Ronghua, SUN Shuihua. The analysis of binary particle swarm optimization[J]. Journal of Nanjing University (Nature Science Edition), 2011, 47(5): 504-514. (in Chinese)
刘建华,杨荣华,孙水华. 离散二进制粒子群算法分析[J]. 南京大学学报(自然科学版),2011,47(5):504-514.

[22] WANG Xiao, YANG Jinmin. Optimization scheme for SDN controllers' placement based on IBPSO[J]. Application Research of Computers, 2019, 36(10): 3069-3071. (in Chinese)
王潇,杨金民. 一种基于 IBPSO 的 SDN 控制器放置优化方案[J]. 计算机应用研究,2019,36(10):3069-3071.

[23] MÜLLER L F, OLIVEIRA R R, LUIZELLI M C, et al. Survivor: an enhanced controller placement strategy for improving SDN survivability[C]//Proceedings of IEEE Global Communications Conference. Washington D. C., USA: IEEE Press, 2014: 1909-1915.

编辑　金胡考

计算机工程 www.ecice06.com