

# SDN中基于蚁群优化的网络测量节点选择算法

叶和元<sup>1</sup>, 韩 俐<sup>1</sup>, 孙士民<sup>2</sup>

(1.天津理工大学 计算机科学与工程学院, 天津 300384; 2.天津工业大学 计算机科学与技术学院, 天津 300387)

**摘 要:** 在软件定义网络(SDN)中,当流传输路径信息获取受限时,现有的测量节点选择算法只能基于网络拓扑的中心性指标进行测量节点选择,存在测量精度较低、测量负载不均衡、运行时间长等问题。将SDN网络中测量节点选择问题抽象为最小顶点覆盖模型,提出一种基于蚁群优化的测量节点选择算法ACO-NS。利用复杂网络的度分布理论缩减状态转移过程中的候选集规模,同时设计一种信息素局部增强-全局挥发机制,增大可行解的信息素浓度,提高算法的准确度和收敛度,并且缩短搜索时间。通过OpenFlow消息在线计算测量节点的负载,采用邻域搜索策略对过载节点进行筛选和替换,以降低过载处理的时间。实验结果表明,与ACO算法相比,该算法的准确度和收敛度分别提高56.7和28.2个百分点,且单位时间内的过载处理开销降低79.8个百分点,具有较高的测量精度。

**关键词:** 网络测量;测量节点选择;蚁群优化;邻域搜索;软件定义网络

开放科学(资源服务)标志码(OSID):



中文引用格式:叶和元,韩俐,孙士民.SDN中基于蚁群优化的网络测量节点选择算法[J].计算机工程,2022,48(5):191-199.

英文引用格式:YE H Y, HAN L, SUN S M. Network measurement node selection algorithm based on ant colony optimization in SDN[J]. Computer Engineering, 2022, 48(5): 191-199.

## Network Measurement Node Selection Algorithm Based on Ant Colony Optimization in SDN

YE Heyuan<sup>1</sup>, HAN Li<sup>1</sup>, SUN Shimin<sup>2</sup>

(1.School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China;

2.School of Computer Science and Technology, Tiangong University, Tianjin 300387, China)

**[Abstract]** In Software Defined Network (SDN), when access to stream transmission path information is limited, existing measurement node selection algorithms can only select nodes based on the central index of the network topology, resulting in low measurement accuracy, an unbalanced measurement load, and a long run time. The measurement node selection problem in SDN network is abstracted as the minimum vertex coverage model, and a measurement node selection algorithm ACO-NS based on Ant Colony Optimization (ACO) is proposed. The degree distribution theory of complex networks is used to reduce the size of candidate sets in the state transition process. Simultaneously, a pheromone local enhancement global volatilization mechanism is designed to increase the pheromone concentration of feasible solutions, improve the accuracy and convergence of the algorithm, and shorten the search time. The load of the measurement node is calculated online through the OpenFlow message, and the Neighborhood Search (NS) strategy is used to filter and replace the overloaded nodes to further reduce the time overhead of overload processing. The experimental results show that compared to the ACO algorithm, the proposed algorithm improves the accuracy and convergence by 56.7 and 28.2 percentage points, respectively, and has high measurement accuracy, and the unit time overhead is reduced by 79.8 percentage points.

**[Key words]** network measurement; measurement node selection; Ant Colony Optimization (ACO); Neighborhood Search (NS); Software Defined Network (SDN)

DOI: 10.19678/j.issn.1000-3428.0062248

### 0 概述

互联网应用呈现多样化、精细化和快速化发展,

同时网络管理的难度也越来越大。随着流量监测技术的不断提升,网络测量从流量数据中分析用户的行为特征<sup>[1]</sup>,为网络管理提供依据。高清视频、虚拟

**基金项目:**国家自然科学基金(61802281, 61702366);天津市自然科学基金(19JCYBJC15800);天津市高等教育委员会科技发展基金(2017KJ090)。

**作者简介:**叶和元(1994—),男,硕士研究生,主研方向为软件定义网络测量、网络空间安全;韩 俐(通信作者),副教授、博士;孙士民,讲师、博士。

收稿日期:2021-08-02

修回日期:2021-10-16

E-mail: hanli@tjut.edu.com

现实等应用使网络带宽快速提升,直接导致网络流量难以复现,为攻击者隐藏恶意流量提供了便利。如果防护措施不规范,频发的网络安全事件严重威胁用户隐私和社会经济安全<sup>[2]</sup>。网络测量是网络管理的基础,细粒度的流量统计特征可以准确反映潜在的网络问题,能够有效地指导异常检测工作。测量节点作为网络测量的环境要素<sup>[3]</sup>,其主要采集流量的网络节点,如软件定义网络(Software Defined Network, SDN)中的 OpenFlow 交换机。测量节点选择方法用于决策整个网络中的测量节点,SDN 控制器可以在线查询交换机上的流统计信息,然而查询哪些 OpenFlow 交换机获取了细粒度流信息成为研究热点<sup>[4]</sup>。

SDN 网络的控制平面和数据平面是解耦合的,控制器能够收集并管理网络的全局状态信息,包括测量节点选择需要的流传输路径。在一条流的传输路径上,任意一个 OpenFlow 交换机均记录了该流的统计信息<sup>[5]</sup>。控制器每次选择流信息最多的 OpenFlow 交换机作为测量节点,直到获取当前网络中全部流信息<sup>[6]</sup>。为避免攻击者通过流传输路径推断敏感信息,在注重隐私保护的 SDN 场景中,通常采取多路径技术<sup>[7]</sup>防止机密信息泄露。如果获取的流传输路径不精确,那么 SDN 网络只能基于网络拓扑的中心性指标<sup>[8]</sup>进行测量节点选择,此时精度越高的流量测量耗费的测量资源越多。单个中心性最高的测量节点对网络行为特征进行估计,大幅降低了测量资源的开销,但是部分流统计信息监测不准确,使得估计误差较大;选择全部网络节点对网络行为特征进行估计,不仅增加控制器轮询 OpenFlow 交换机的频率,而且相邻测量节点间的流统计存在较大重复的问题。因此,采用最少测量节点覆盖全部流信息是测量节点选择的趋势<sup>[9-10]</sup>,以实现测量资源和准确性的平衡。

SDN 中基于网络拓扑的测量节点选择问题,可以抽象成最小顶点覆盖模型,即寻找无向图顶点集的最小子集,使其包含图中任一条边的至少一个端点。文献[11]指出,最小顶点覆盖模型是 NP-Hard 的,不可能在小于 1.360 6 的复杂度因子内近似求解,研究人员分别采用分支界限法<sup>[12]</sup>、贪心算法<sup>[13]</sup>、元启发式算法<sup>[14]</sup>和局部搜索算法<sup>[15]</sup>对该抽象模型进行求解。然而这些算法均存在一定不足。分支界限法能够获取模型的最优解,但是其复杂度高使得算法在大规模无向图中的运行时间无法预估。贪心算法显著降低问题复杂度,但是局部的最优决策不一定能够达到全局最优,并且误差随图的规模逐渐增大。蚁群优化(Ant Colony Optimization, ACO)作为元启发式算法之一,具有正反馈、贪心启发两种机制和较优的鲁棒性,但存在收敛过早、搜索时间长等问题。属于局部搜索的邻域搜索算法(Neighborhood Search, NS)目标在于优化某个特定顶点而不是整个

无向图,能够有效缩短搜索时间,但前提是问题的初始解较准确。

现有 SDN 测量节点选择方案对测量节点状态的考虑有所欠缺<sup>[16-17]</sup>,对不同位置的 OpenFlow 交换机施加了不均匀的测量负载,最佳的测量结果可能导致不均匀的网络负载分布。在真实环境中的网络设备,同一时间运行着多个测量任务,处理性能和内存容量消耗较大,如果继续向其下发测量任务,该设备的测量负载维持在较高水平,可能会出现宕机现象<sup>[18]</sup>。控制器利用 OpenFlow 消息查询 OpenFlow 交换机的计数器,需要综合考虑查询频率和测量精度。文献[19]指出,CPU 吞吐量和三态内容寻址存储器(Ternary Content Addressable Memory, TCAM)容量是制约 OpenFlow 交换机性能的两大因素,在当前高带宽环境下尤为明显。因此,测量节点的负载因素应纳入 SDN 测量节点选择问题的范畴。

针对流路径信息获取受限的 SDN 场景,本文提出一种基于蚁群优化的网络测量节点选择算法。通过对 ACO 算法的概率状态转移规则和信息素更新规则进行优化,利用 OpenFlow 消息在线监控测量节点的负载,使用邻域搜索策略替换负载超过阈值的测量点,从而在保证准确性的前提下提高 ACO 算法的收敛度,缩短搜索时间。

## 1 相关工作

网络测量节点选择是网络管理工作的基础。随着网络技术的发展,研究人员提出各种智能选择算法,例如 ACO 算法,但都局限于特定场景。

对于抽象的最小顶点覆盖模型,ACO 算法能够提高测量节点集合的准确性,但也存在局部最优、搜索时间长等问题,可以从概率状态转移和信息素更新 2 个规则进行优化。此外,较优的鲁棒性使得 ACO 算法可以结合其他算法的优势,弥补其自身不足<sup>[20]</sup>。

针对概率状态转移规则,文献[21]使用轮盘赌的选择方式构造可行解,摒弃了传统 ACO 算法的  $\alpha$ 、 $\beta$  双参数方案,使得计算复杂度降至  $O(1)$ ,避免了局部最优现象的发生,然而降低信息素的重要程度可能导致算法收敛速度过慢。文献[22]在概率状态转移规则中引入故障排除因子,尽可能避免选择网络中的故障节点,以增加可行解搜索的多样性,从而提高算法的准确度。文献[23]提出现有概率状态转移规则容易导致局部最优解,需要设置平衡参数进行动态调整:初始时平衡参数值较大,蚂蚁倾向于选择信息素浓度高的节点;迭代数次后降低平衡参数的值,蚂蚁选择信息素浓度低的节点,从而避免局部最优;最后再次提高平衡参数的值,从而加快算法收敛过程。

针对信息素更新规则,文献[24]表明,部分异常节点可能影响全局最优解的质量,提出一种排除异常节点的信息素更新规则,以筛选迭代最优解中信息素异常的节点,后续迭代优先选择正常节点,从而将蚂蚁引



导到新的搜索区域。文献[21]表明,当信息素挥发系数为常值时,算法对于信息素较低节点的搜索能力较低,因此提出一种信息素动态挥发机制,每次迭代后,挥发系数根据信息素浓度动态变化。文献[25]构建一种改进最大最小蚂蚁系统(Max-Min Ant System, MMAS),以避免结果陷入局部最优解,将顶点信息素初始为最大值,在蚁群搜索过程中,信息素被严格限制在最值范围内,每次迭代后仅更新迭代最优解或全局最优解中顶点的信息素。然而这些信息素更新规则无法兼顾算法的准确性和收敛性,在SDN测量节点选择问题的应用效果不佳。

现有SDN测量方案中均考虑了测量节点选择问题,利用有限测量资源实现较优的测量精度,但是尚未考虑测量节点的过载现象。文献[26]将控制器作为测量节点,考虑部署多个测量节点来测量网络参数的问题,分析变量特征后转换为0-1线性规划问题,并提出一种基于拉格朗日松弛的动态规划算法。该算法能够有效降低不同网络拓扑下的测量成本和运行时间。大多数SDN测量方案从OpenFlow交换机中采集流量信息,如果流路由信息获取不受限制,文献[5]提出5种查询策略,分别为流路径上最后、均匀随机、轮询、非均匀随机和负载最少的单个交换机,它们对网络设备施加不均匀的负载且难以扩展到大规模网络。文献[6]和文献[27]均采用贪心策略选择多个测量节点,其原理是第1轮迭代选择流数最多的一个节点,将第1个测量节点关联的流从流矩阵中删除,后续迭代重复选择点-删流过程,直到流矩阵不包含任何流时停止,以得到1个测量节点集合。然而在实际网络环境中流的数量通常较大,如 $10^6$ 条,使得持续统计并更新流矩阵的时间开销不可忽略。

在流路径信息获取受限的情况下,SDN网络测量节点选择问题只能依赖于网络的物理拓扑。文献[27]借助社交网络的中心性指标度量图节点的重要性,提出3种基于中心性的单个测量节点选择方案:介数中心性,接近中心性和度中心性。然而单点测量能够节约测量资源,但是测量准确度低。因此,该文献又提出一种基于扩展中心性的多测量节点选择算法。该算法基于介数中心性选择第1个测量节点,后续测量节点在其邻接点中按照最短路径重叠最少规则进行选择。

根据以上研究和分析,SDN网络下测量节点选择算法存在以下问题:1)隐私保护场景下流路径信息获取受限,基于流的测量节点选择方案失效;2)单个测量点采集的流统计信息不全面,导致测量精度较低;3)基于扩展中心性的多测量节点选择算法未考虑测量点的负载因素,实际测量效果较差。因此,本文在SDN网络中基于网络拓扑进行测量节点选择,改进ACO算法的概率状态转移规则和信息素更新规则,分别提高模型结果的准确性和收敛性,增加一种基于NS策略的过载处理机制局部调整满足条

件的测量节点,在不破坏链路覆盖率的情况下,缩短过载处理时间。

2 SDN中基于蚁群优化的测量节点选择算法

基于蚁群优化的测量节点选择算法的参数信息如表1所示。

表1 基于蚁群优化的测量节点选择算法的参数信息

Table 1 Parameters information of measurement node selection algorithm based on ant colony optimization

参数	含义	符号
网络拓扑图	与网络拓扑等价的无向图	graph
初始测量集	所有测量节点构成的集合	selects
过载测量集	存放负载超过阈值的测量节点	loads
优化测量集	过载处理后的测量节点集	uploads
负载阈值	用于评估过载测量点的临界值	$\delta$
信息素指数	概率状态转移中信息素重要性	$\alpha$
信息素增强常量	模拟蚂蚁释放适量信息素	$Q$
信息素挥发系数	信息素浓度随迭代降低的比例	$\rho$

2.1 SDN网络中测量节点选择方案

网络测量节点选择的SDN架构如图1所示。基于蚁群优化的测量节点选择方案部署在SDN应用层,包括拓扑发现、节点搜索、负载监测和过载处理4个模块,同一模块流向其他模块的数据相同。

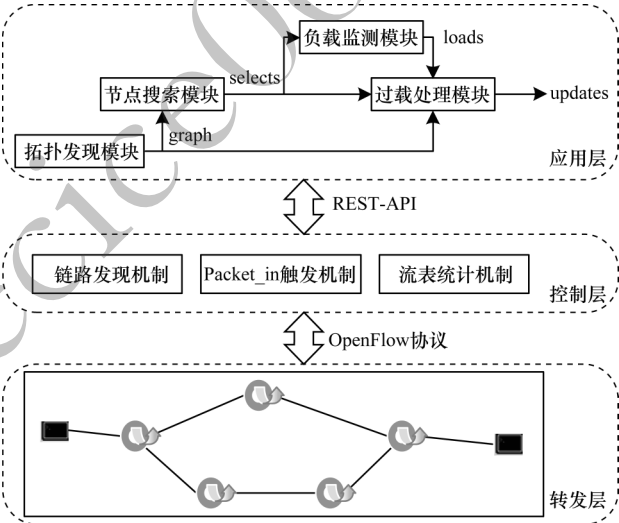


图1 网络测量节点选择的SDN架构

Fig.1 SDN framework for network measurement node selection

基于蚁群优化的测量节点选择方案的具体流程为:1)拓扑发现模块通过链路发现机制的REST-API获取模拟网络的所有交换机和链路,汇总成网络拓扑图;2)节点搜索模块利用改进ACO算法搜索graph的初始测量集;3)负载监测模块通过Packet-in消息触发机制和流表统计接口在线计算selects中节点的负载,若监测值超过阈值 $\delta$ ,该测量点被添加到过载测量集;4)过载处理模块利用NS策略将loads中满足覆盖条件的过载测量点从selects中删除,最终得到给定无向图的优化测量集updates。

## 2.2 ACO算法的规则改进

在无向图  $G=(V,E)$  中,  $V$  和  $E$  分别表示顶点集和边集, 覆盖  $E$  中每条边的顶点集合是  $V$  的子集, 最小顶点覆盖  $M$  要求该子集包含的顶点最少。假设其中过载顶点的邻接矩阵为  $S$ , 附加负载的最小顶点覆盖要求使得  $M$  包含最少数量的过载顶点。

ACO算法的规则改进是采用ACO-NS算法求解附加负载因素的最小顶点覆盖模型, 首先ACO算法从图  $G$  中搜索出最小顶点覆盖  $M$ , 然后NS算法结合邻接矩阵  $S$  得到最小状态顶点覆盖  $M'$ , 基于蚁群优化的网络测量节点选择算法流程如图2所示。

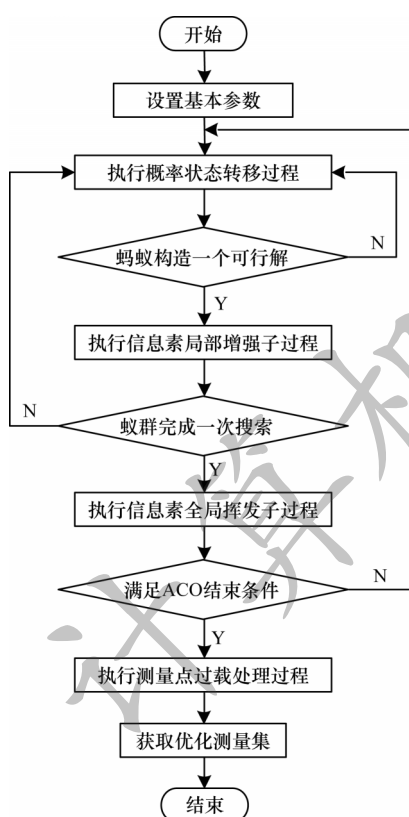


图2 基于蚁群优化的网络测量节点选择算法流程

Fig.2 Procedure of network measurement node selection algorithm based on ant colony optimization

对于ACO算法, 本文主要改进概率状态转移过程中的候选集定义, 对信息素更新过程的优化体现在局部增强和全局挥发2个子过程; 而对NS算法优化主要分为构造筛选、替换和冗余检验3个操作, 处理测量节点的过载现象。

### 2.2.1 概率状态转移优化

在概率状态转移过程中, 候选集动态减小为空, 每次根据概率选择一个顶点后, 从候选集中删除点和度均为0的顶点。网络拓扑顶点的度通常服从以下分布<sup>[28]</sup>: 仅部分节点的度明显较大, 多数节点的度在3附近。但是原始候选集更新策略还存在一定的不足: 蚂蚁不断构造初始测量集, 所选顶点的邻接点的度逐渐减小为1, 在信息素和启发信息的作用下, 它们得不到及时处理。仿真结果表明, 假设无向图规模为  $n$ , 度为1顶点的数量随着蚂蚁搜索次数增加

而增加。区间  $\left(0, \frac{n}{8}\right)$  内按正态曲线增大到  $\frac{n}{4}$ , 而区间  $\left(\frac{n}{8}, \frac{n}{2}\right)$  内以线性关系降低到0。因此, 未处理的度为

1的顶点近似为  $\frac{3n^2 + 8n}{64}$ , 概率状态计算的时间复杂度和空间复杂度均为  $O(n^2)$ 。

蚂蚁每次选择一个顶点后, 新候选集更新策略在原始候选集更新策略的基础上继续删除度为1的顶点。假设当前候选集用  $C$  表示, 则新候选集更新策略的数学描述如式(1)所示:

$$C = C_p - s - V_0 - V_1 \quad (1)$$

其中:  $C_p$  为上次更新后的候选集;  $s$  为所选顶点;  $V_0$  为度为0的顶点构成集合; 集合  $V_1$  包含所有度为1的顶点。

根据新候选集更新策略, ACO算法的概率状态转移规则进行了相应调整。候选集划分为优选集和计算集。优选集由度为1顶点的邻接点构成的集合, 计算集是当前候选集中优选集的补集, 必须保证优选集的优先级。在某次迭代中, 如果蚂蚁先选择计算集的顶点, 由于会删除关联的边, 可能使优选集中顶点的度为0, 导致优选集顶点被候选集更新策略删除。度为1的顶点与邻接点的边没有被 selects 覆盖, 进而影响模型结果的准确性。优选集顶点的选择概率恒定为1, 而计算集顶点的选择概率由信息素浓度和启发信息值计算得到。假设优选集、计算集分别为  $F$  和  $R$ , 蚂蚁  $k$  选择顶点  $v_i$  的概率为  $p_i^k$ , 则调整后的概率状态转移规则如式(2)所示:

$$p_i^k = \begin{cases} 1, & v_i \in F \\ \frac{\tau_i^\alpha \eta_i}{\sum_{j \in R} \tau_j^\alpha \eta_j + 1}, & v_i \in R \end{cases} \quad (2)$$

其中:  $\tau_i$  为顶点的信息素浓度, 初始值设为1, 不构成顶点选择的元素, 其值随蚁群的搜索逐渐增大;  $\alpha$  为信息素指数, 用于衡量信息素相对于启发信息的重要程度, 指数较传统  $\alpha, \beta$  双因子方案平均降低了3.63;  $O(n^2)$  数量级的度为1顶点执行概率状态转移规则的计算开销较大;  $\eta_i$  为顶点的启发信息值。邻接点被选择使得顶点的度减少一个单位, 实际上是顶点动态变化的度, 参数值的计算如式(3)所示:

$$\eta_i = |v_i| \quad (3)$$

当计算集顶点和优选集顶点的选择概率相同时, 蚂蚁极有可能选择计算集中的顶点, 破坏规定的优先级。通过将概率状态转移规则的分母增加1, 使得计算集的概率状态始终小于优选集, 以确保模型结果的准确性。

### 2.2.2 信息素更新优化

信息素是自然界蚂蚁间交互的介质, 蚂蚁经过节点或路径会释放一定浓度的信息素, 但是信息素浓度随时间的增加不断降低。因此, 信息素更新因素包括信息素增强和信息素挥发。由于信息素更新因素的触发时机不同, 因此构成了不同的ACO算法, 正反馈机制强弱也存在一定的差异。全局更新和局部更新是信息素策略的典型执行时机, 局部更新指蚂蚁构造一个可行解结束后, 全局更新则发生在蚁群搜索到迭代最优解。结合信息素更新因素和



触发时机, 本文提出一种信息素局部增强-全局挥发规则, 增大可行解的信息素浓度, 使得蚂蚁趋向于局部最优轨迹, 以减少算法的迭代次数, 从而有效解决蚁群搜索时间较长的问题。通过动态降低迭代解的信息素, 并适当增加蚁群的搜索面, 使得全局解与模型最优解的偏差逐渐减小。

信息素局部增强发生在蚂蚁构造出一个可行解, 无论可行解是否最优, 其相关节点的信息素均适当增加。节点  $v_i$  在第  $t+1$  次迭代中信息素浓度如式(4)所示:

$$\tau_i(t+1) = \tau_i(t) + Q \quad (4)$$

信息素局部增强为充分建立 ACO 算法的正反馈机制, 通过当前蚂蚁增强潜在最优节点的信息素, 指导蚁群中的其他蚂蚁, 以构造较优解。

每当蚁群迭代出一个局部解时, 执行信息素全局挥发子策略。根据迭代解降低相关节点  $v_j$  的信息素浓度, 如式(5)所示:

$$\tau_j(t+1) = \left(1 - \frac{|V_c^{vc}|}{|V_g^{vc}|}\right) \rho \tau_j(t) \quad (5)$$

其中:  $|V_c^{vc}|$  和  $|V_g^{vc}|$  分别为迭代最优解和全局最优解包含的测量节点数。在迭代最优解中, 节点  $v_j$  的信息素按  $|V_c^{vc}|$  与  $|V_g^{vc}|$  的比值进行挥发, 比值越大说明迭代最优解偏离全局最优解的程度越高, 挥发的信息素也就越多。信息素局部增强可能使算法陷入局部最优解, 然而信息素全局挥发可以弥补这种不足, 增大非局部最优解节点被搜索的可能性, 以保证所提规则在提高模型收敛性时不影响其准确性。

### 2.3 测量点过载处理机制

#### 2.3.1 OpenFlow 交换机负载监测

测量点过载处理机制主要监测 selects 中测量点的负载, 在实际 SDN 环境中, 在线获取 OpenFlow 交换机的负载。在不同场景下, 交换机负载的定义和监测方式有所不同, 由于 CPU 吞吐量和 TCAM 容量是 SDN 交换机的两大性能瓶颈, OpenFlow 交换机的负载  $I_{load}$  如式(6)所示:

$$I_{load} = \frac{T_{CPU} + U_{TCAM}}{2} \quad (6)$$

其中:  $T_{CPU}$  和  $U_{TCAM}$  分别为测量点的 CPU 吞吐率和 TCAM 使用率。因此, 测量点负载监测分为统计 CPU 吞吐率和 TCAM 使用率。

在 SDN 交换机中, CPU 主要用于处理各种 OpenFlow 消息, 其中 Packet-in 消息参与多种报文处理机制。假设单个交换机单位时间内发送的 Packet-in 消息数量为  $in\_count$ , 根据 SDN 特性, 该指标是有限的。CPU 吞吐率计算的关键在于控制器统计单位时间内指定交换机发送的 Packet-in 消息数。控制器解析 Packet-in 消息, 获取指定交换机对应的 dpid 标识, 进而根据 dpid 对 Packet-in 消息进行累加。截止统计时间, 控制器输出指定交换机的 Packet-in 消息数  $in\_count$ 。因此, CPU 吞吐率由  $in\_count$  与  $in\_max$  比值计算得到, 如式(7)所示:

$$T_{CPU} = \frac{in\_count}{in\_max} \quad (7)$$

在 OpenFlow 交换机中, TCAM 用于存储流表项, 而流表项是交换节点处理数据包的依据。从成本和功

耗两个方面考虑, TCAM 存储器支持的流表项数量有限, 假设最大流表项数为  $entry\_volumn$ 。交换机当前使用的流表项数在一定程度上反映了其负载状态。

OpenFlow 协议提供流表统计机制, 包括 TableStatsRequest 和 TableStatsReply 消息。控制器向交换机发送不包含数据的 TableStatsRequest 消息, 交换机回复 TableStatsReply 消息给控制器, TableStatsReply 消息体结构如下:

```
/* Body of reply to OFPMP_TABLE request.*/
struct ofp_table_stats {
    uint8_t table_id; /* Identifier of table.
    Lower numbered tables are consulted first.*/
    uint8_t pad[3]; /* Align to 32-bits.*/
    uint32_t active_count; /* Number of active entries.*/
    uint64_t lookup_count;
    /* Number of packets looked up in table.*/
    uint64_t lookup_count;
    /* Number of packets looked up in table.*/
    uint64_t matched_count;
    /* Number of packets that hit table.*/
};
```

OFP\_ASSERT (sizeof(struct ofp\_table\_stats) == 24);  
active\_count 字段指示单个流表中活跃的流表项数, 累加所有流表的 active\_count 字段值, 可以得到整个交换机当前占用流表项数  $entry\_current$ 。

因此, TCAM 使用率是当前占用流表项数与最大流表项数的比值, 如式(8)所示:

$$U_{TCAM} = \frac{entry\_current}{entry\_volumn} \quad (8)$$

OpenFlow 交换机在线获取测量节点的负载后, 需要进一步判断该节点的状态是否异常。假设负载阈值  $\delta=0.8$ , 若式(6)计算的负载大于  $\delta$ , 该测量节点不适合下发测量任务, 添加到 loads 后, 等待过载处理模块筛选并替换。

#### 2.3.2 基于 NS 的过载处理机制

测量点过载处理规则的相关假设是在短时间内网络拓扑不会发生大规模变化, 并且负载主要发生在测量节点。随着网络测量持续运行, 测量节点可能出现过载现象, 需要重新执行 ACO 算法等。然而全局搜索算法将耗费大量资源, 同时时间开销较大。NS 策略在初始解的基础上执行添加、删除和交换等操作, 以获取最优解。NS 策略存在平衡测量资源和时间的可能, 应用在最小顶点覆盖模型时, 还可以兼顾链路覆盖率。

如果 2 个过载测量点由一条边连接, 则定义这种边为过载边。过载节点可以集中替换的节点取决于过载边的统计结果, 如果没有过载边, 则替换整个过载节点集; 否则不断筛选覆盖最多过载边的负载异常的测量节点, 删除已覆盖过载边, 直到删除完全, 过载节点集中的剩余节点作为替换节点。

链路覆盖率维持的关键是 NS 策略的替换过程。基于网络拓扑图构造每个替换测量点的邻域, 即所有邻接点构成的集合, 若某个邻接点不在 selects 中, 则将其添加到初始测量集。整个邻域遍历结束后, 从初始测量集删除对应替换测量点。替换过程结束, selects 还需要进一步处理。

冗余覆盖检验的目的在于降低模型结果的链路的冗余度。如果 selects 中每个测量点邻域被包含在更新

的初始测量集中,说明该测量点与所有邻接点间的边被重复覆盖,直接将该测量点从初始测量集中删除,在不影响链路覆盖率的情况下,能够降低模型结果的冗余度,从而优化测量集。

测量点过载处理后的 updates 包含最少数量的过载测量点,算法 1 是邻域搜索策略的伪代码。

#### 算法 1 邻域搜索策略

输入 graph, selects, loads

输出 uploads

1. 假设基于过载边信息筛选出  $m$  个替换测量点。
2. 初始化  $i = 1$ 。
3. 基于 graph 构造替换测量点  $i$  的邻域。
4. 遍历替换测量点  $i$  的邻域,若邻接点不属于初始测量集,则将其添加到 selects。
5. 将替换测量点  $i$  从 selects 中删除。
6. 令  $i = i + 1$ 。
7. 如果条件  $i > m$  不成立,返回步骤 3;否则停止循环。
8. 更新的 selects 进行冗余覆盖校验,原理优化测量集。

算法 1 的原理是 1 个双层循环结构(步骤 2~7),外层循环控制当前访问的替换测量点,而内层循环遍历该替换测量点的邻域,除 selects 以外的邻接点被添加到初始测量集,遍历结束,再删除该替换测量点。假设  $n$  表示替换测量点的数目,根据分析可得,算法的时间复杂度是  $O(n^2)$ ,由于需要记录每个替换测量点的邻接点信息,算法的空间复杂度也是  $O(n^2)$ 。

### 3 实验与分析

本节从实验设置、规则优化效果和仿真拓扑验证 3 个方面对 SDN 中基于蚁群优化的网络测量节点选择算法进行评估。

#### 3.1 实验环境与基本参数

3.60 GHz、16 GB RAM 的 4 核 Windows 10 主机作为硬件环境,SDN 平台基于 Ubuntu16.04 系统搭建,使用开源控制器 Ryu-4.32,拓扑仿真器采用 Mininet-2.3.0d6,软件交换机为 Open vSwitch-2.11.0,算法相关模块由 Python 语言实现。

算法输入的 3 种无向图参数信息如表 2 所示,最小测量数是无向图可以部署测量节点的最少数量。

表 2 不同规模无向图的参数信息

Table 2 Parameters information of undirected graphs with different scales

无向图	顶点数	边数	最小测量数
karate	34	78	14
jazz	198	2 742	158
email	1 133	5 451	594

ACO 算法的参数空间比较复杂,参数取值与算法的验证结果相关,通过参考文献或简单实验确定算法的参数,如表 3 所示。其中:iterate 表示算法最大迭代次数;verge 表示算法的收敛条件,即模型解长度不再减少的连续迭代次数;符号  $|V|$  表示无向图的顶点数,算法中蚂蚁数量随无向图规模发生变化。

表 3 算法的参数值

Table 3 Parameter values of algorithm

参数	值
iterate	100
verge	6
ant	$\sqrt{ V }$
$\alpha$	2
$Q$	0.5
$\rho$	0.05
$\delta$	0.8

#### 3.2 算法规则的优化效果

##### 3.2.1 实验与数据对比

在算法执行过程中,平均测量数是指多个测量节点数的均值,命中频数是命中对应最小测量数的次数,两者决定算法的准确性。平均迭代次数是多次执行迭代次数的均值,与算法收敛性相关。

第 1 组对比实验用于验证新候选集定义对于 ACO 算法准确性的改进效果,在基本参数保持一致的条件下,本文采用新候选集定义和原始候选集定义的 ACO 算法在无向图 karate、jazz 和 email 中进行平均测量数、命中频数和平均迭代数对比,各执行 10 次,结果如表 4 所示。

表 4 两组对比实验的统计数据

Table 4 Statistical datas of two groups of comparative experiments

无向图	新候选集			原始候选集			局部增强-全局挥发规则			MMAS 更新规则		
	平均测量数	命中频数	平均迭代数	平均测量数	命中频数	平均迭代数	平均测量数	命中频数	平均迭代数	平均测量数	命中频数	平均迭代数
karate	14	10	7.0	14.0	10	9.3	14.0	10	7.1	14	10	7.5
jazz	158	10	7.0	158.8	3	44.2	158.6	4	10.1	158	10	19.5
email	594	10	7.1	604.0	0	7.9	603.6	0	8.9	603	0	35.4

第 2 组对比实验用于验证信息素局部增强-全局挥发规则对 ACO 算法收敛性的优化效果,同理保持基本参数一致,本文采用信息素局部增强-全局挥发规则和 MMAS 更新规则的 ACO 算法分别在 karate、jazz 和 email 中执行 10 次,同样将第 2 组对比实验数据的结果统计到表 4。

从表 4 可以看出,对比 ACO 算法中的典型策略,新候选集定义一定程度上提高了抽象模型的准确性,而信息素局部增强-全局挥发规则主要优化抽象模型的收敛性。过载处理机制执行的前提是无向图的初始测量集,为了保证实验环境相同,第 3 组对比实验采用改进 ACO 算法获取初始测量集。基本负载处理机制将

负载作为概率状态转移规则的决策因素之一。第3组对比实验中过载节点随机产生,基本负载处理机制与测量点过载处理机制在3种无向图中各执行10次,主要用于验证过载处理机制对时间的优化效果。

图3分别记录2种负载处理机制在无向图 karate、jazz 和 email 运行的测量节点数和运行时间。num\_aco、num\_ns 分别表示基本负载处理和测量点过载处理机制的测量节点数;time\_aco、time\_ns 分别为采用2种机制的运行时间,由于jazz 和 email 图下两种负载处理机制的运行时间差别很大,因此图3(b)和图3(c)中time\_ns 指标非常接近于横坐标。

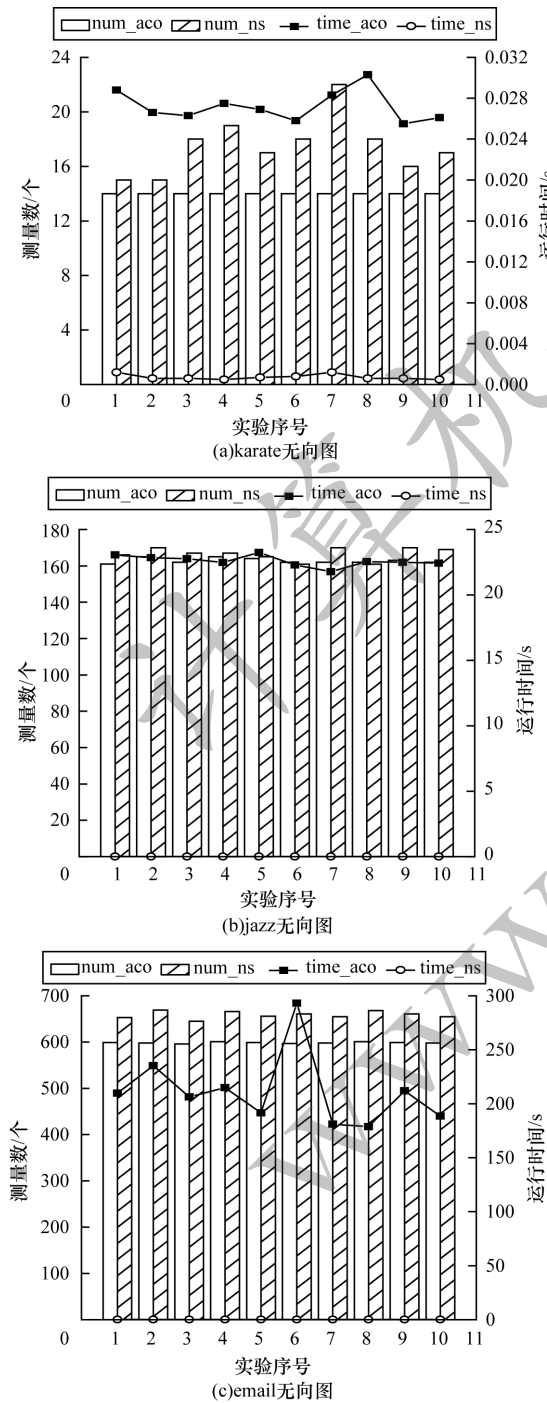


图3 不同负载处理机制的测量节点数和运行时间对比  
Fig.3 Measurement nodes and runtime comparison among different load handling mechanisms

3.2.2 数据分析与结果

本文定义3种衡量算法的指标:准确性为命中频数的平均数;收敛性是收敛条件与平均迭代数的比值;单位时间开销是平均运行时间与无向图顶点数的比值。

对于表4中的统计数据,尽管无向图规模不同,新候选集定义的平均测量数、平均迭代次数均小于或等于原始候选集定义,且命中频数较大,结果更接近无向图的最小测量数。相比MMAS更新策略,虽然信息素局部增强-全局挥发规则增加了平均测量数,但是大幅降低了平均迭代数,在无向图email中更为明显。从图4可以看出,测量点过载处理机制与基本负载处理机制的测量节点数比较接近,但是测量点过载处理机制的运行时间具有2~3个数量级的优势,并且随着无向图顶点数增多而增大,其原因为将负载因素引入概率转移状态规则中,破坏信息素和启发信息的指导作用,增加了算法的运行时间。

因此,新候选集定义显著提高了ACO算法的准确性,信息素局部增强-全局挥发规则集中改进了收敛性,而测量点过载处理机制主要降低了算法的单位时间开销。若考虑3种规则在准确性、收敛性及单位时间开销的精确效果,ACO算法与ACO-NS算法的性能指标对比如表5所示。

表5 ACO与ACO-NS算法的性能指标对比  
Table 5 Performance indexes comparison of ACO and ACO-NS algorithms

算法	准确度	收敛度	单位时间开销
ACO算法	0.433	0.426	0.899
ACO-NS算法	1.000	0.708	0.101

从表5可以看出,相比ACO算法,ACO-NS算法的准确度和收敛度分别提高了56.7和28.2个百分点,且在单位时间开销方面降低了79.8个百分点。

3.3 SDN网络中不同测量节点选择方案

本节在SDN环境中使用Mininet工具模拟欧洲光纤网核心拓扑,如图4所示。

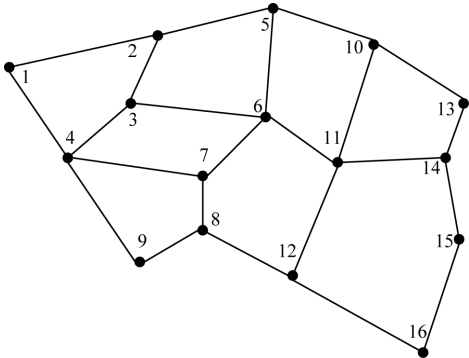


图4 仿真的欧洲光纤网核心拓扑  
Fig.4 Simulated core topology of European optical fiber network



为评估单个中心测量点、基于扩展中心性的多个测量点、ACO-NS算法和全部节点4种方案的覆盖率和冗余度,本文定义链路的覆盖率和冗余度。覆盖率是指所选测量点覆盖的链路占全部链路的比率,冗余度是指测量节点重复覆盖的链路数量。

仿真的欧洲光纤网核心拓扑包括16个节点、23条链路。假设单个中心测量点选择度最大的节点,即  $\text{single}=\{6\}$ ,扩展中心性方案的测量点集合  $\text{extend}=\{11,6,7,4,3,1\}$ ,ACO-NS算法的测量点集合  $\text{aco\_ns}=\{6,4,2,8,10,14,16,12\}$ ,全部节点方案  $\text{all}$  包含该网络拓扑中所有节点。

不同方案的覆盖率和冗余度如图5所示。 $\text{single}$ 方案的覆盖率和冗余度都是最低的,但是网络测量更关注覆盖率,因此 $\text{single}$ 方案应用效果最差。虽然 $\text{aco\_ns}$ 和 $\text{all}$ 这2种方案的覆盖率均达到100%,但是 $\text{all}$ 方案的冗余度过高,流量重复采集的概率较大。 $\text{extend}$ 方案能够以较低的冗余度实现较高的覆盖率,但仍存在优化的空间, $\text{aco\_ns}$ 方案的冗余度和覆盖率均优于 $\text{extend}$ 方案。因此,ACO-NS算法在基于网络拓扑的SDN场景下选择测量节点的性能最佳。

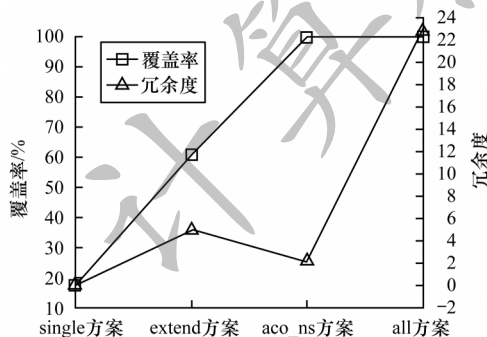


图5 不同方案的覆盖率和冗余度对比

Fig.5 Coverage and redundancy comparison among different schemes

#### 4 结束语

本文提出一种基于蚁群优化的网络测量节点选择算法ACO-NS,利用CPU吞吐率和三态内容寻址存储器在线监测测量交换机的负载,为SDN环境中网络测量节点选择提供理论依据,通过优化概率状态转移规则和信息素更新规则,提高测量集的收敛度,同时仅对过载的测量交换机进行邻域搜索,有效缩短调整负载的时间。实验结果表明,相比ACO算法,ACO-NS算法的准确度提高了56.7个百分点,单位时间开销降低79.8个百分点,能够有效提高SDN控制器的工作效率,从而降低负载对网络测量结果的影响。后续将基于不同网络拓扑结构分析测量点选择与流长分布估计的关系,进一步优化算法设计。

#### 参考文献

- [1] WU X B, LI P L, RAN Y Y, et al. Network measurement for 100 GbE network links using multicore processors[J]. Future Generation Computer Systems, 2018, 79(1): 180-189.
- [2] 戴冕,程光,周余阳. 软件定义网络的测量方法研究[J]. 软件学报, 2019, 30(6): 1853-1874.  
DAI M, CHENG G, ZHOU Y Y. Survey on measurement methods in software-defined networking[J]. Journal of Software, 2019, 30(6): 1853-1874. (in Chinese)
- [3] 张恒,蔡志平,李阳. SDN网络测量技术综述[J]. 中国科学:信息科学, 2018, 48(3): 293-314.  
ZHANG H, CAI Z P, LI Y. An overview of software-defined network measurement technologies[J]. Chinese Science: Information Science, 2018, 48(3): 293-314. (in Chinese)
- [4] TOOTOONCHIAN A, GHOBADI M, GANJALI Y. OpenTM: traffic matrix estimator for OpenFlow networks[C]// Proceedings of International Conference on Passive and Active Network Measurement. Berlin, Germany: Springer, 2010: 201-210.
- [5] CHOWDHURY S R, BARI M F, AHMED R, et al. PayLess: a low-cost network monitoring framework for software defined networks[C]// Proceedings of IEEE Network Operations and Management Symposium. Washington D. C., USA: IEEE Press, 2014: 1-9.
- [6] DENG J, CAI H, CHEN S, et al. Improved flow awareness among edge nodes by learning-based sampling in software defined networks[J]. Mobile Networks and Applications, 2019, 29: 1-10.
- [7] ZHANG Y, TAN Y, JIE T, et al. TOHIP: a topology-hiding multipath routing protocol in mobile Ad Hoc network[J]. Ad Hoc Networks, 2014, 21(5): 109-122.
- [8] YOON S, HA T, KIM S, et al. Scalable traffic sampling using centrality measure on software-defined networks[J]. IEEE Communications Magazine, 2017, 55(7): 43-49.
- [9] PERDICES D, MUELAS D, PEDRO L D, et al. Network performance monitoring with flexible models of multi-point passive measurements[C]// Proceedings of the 14th International Conference on Network and Service Management. Washington D. C., USA: IEEE Press, 2018: 1-9.
- [10] COCIGLIO M, FIOCCOLA G, MARCHETTO G, et al. Multipoint passive monitoring in packet networks[J]. IEEE Transactions on Networking, 2019, 27(6): 2377-2390.
- [11] LI R Z, HU S L, ZHANG H C, et al. An efficient local search framework for the minimum weighted vertex cover problem[J]. Information Sciences, 2016, 372(1): 428-445.
- [12] WANG L Z, HU S L, LI M Y, et al. An exact algorithm for minimum vertex cover problem[J]. Mathematics, 2019, 7(7): 1-8.
- [13] GAO W R, FRIEDRICH T, NEUMANN F, et al. Randomized greedy algorithms for covering problems[C]// Proceedings of the Genetic and Evolutionary Computation Conference. New York, USA: ACM Press, 2018: 309-315.
- [14] GUAN B X, ZHAO Y H, LI Y. An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems[J]. Expert Systems with Applications, 2021, 164: 1-7.
- [15] HU S L, WU X L, LIU H, et al. Multi-objective neighborhood search algorithm based on decomposition for multi-objective minimum weighted vertex cover problem[J]. Sustainability, 2019, 11(13): 1-21.



- [16] YU C, LUMEZANU C, SHARMA A, et al. Software-defined latency monitoring in data center networks[C]// Proceedings of International Conference on Passive and Active Network Measurement. Berlin, Germany: Springer, 2015: 360-372.
- [17] VAN ADRICHEM N L M, DOERR C, KUIPERS F A. OpenNetMon: network monitoring in OpenFlow software-defined networks [C]// Proceedings of IEEE Network Operations and Management Symposium. Washington D. C. , USA: IEEE Press, 2014: 1-8.
- [18] 张荣, 杨谈, 金跃辉, 等. 分布式网络测量中测量节点的智能选择算法[J]. 计算机科学, 2015, 42(9): 70-77.  
ZHANG R, YANG T, JIN Y H, et al. Intelligent selection algorithm of measurement nodes in distributed network measurement[J]. Computer Science, 2015, 42(9): 70-77. (in Chinese)
- [19] DIEGO K, PAULO E V, SIAMAK A. Software-defined networking: a comprehensive survey[J]. Proceedings of the IEEE, 2015, 103(1): 14-76.
- [20] 覃远年, 梁仲华. 蚁群算法研究与应用的新进展[J]. 计算机工程与科学, 2019, 41(1): 173-184.  
QIN Y N, LIANG Z H. New progress of the ant colony algorithm in research and applications [J]. Computer Engineering and Science, 2019, 41(1): 173-184. (in Chinese)
- [21] EBADINEZHAD S. DEACO: adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem[J]. Engineering Applications of Artificial Intelligence, 2020, 92: 1-10.
- [22] MIAO C W, CHEN G Z, YAN C L, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm[J]. Computers and Industrial Engineering, 2021, 156: 1-10.
- [23] QU W, WEN D L, FENG X S. A dynamic energy-saving routing strategy based on ant colony optimization [C]// Proceedings of the 29th Chinese Control and Decision Conference. Chongqing, China: [s. n. ], 2018: 2079-2084.
- [24] JOVANOVIĆ R, TUBA M. An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem[J]. Applied Soft Computing, 2011, 11(8): 5360-5366.
- [25] LI X L, WANG Y. Scheduling batch processing machine using max-min ant system algorithm improved by a local search method[EB/OL]. [2021-07-01]. <https://downloads.hindawi.com/journals/mpe/2018/3124182.pdf>.
- [26] SUN M, WANG L, HE Y. An optimization algorithm for multiple measurement points placement in SDN [C]// Proceedings of IEEE International Conference on Computer and Communication Systems. Washington D. C. , USA: IEEE Press, 2019: 203-207.
- [27] HARK R, GHANMI M, KAR S, et al. Representative measurement point selection to monitor software-defined networks[C]// Proceedings of the 43rd Conference on Local Computer Networks. Washington D. C. , USA: IEEE Press, 2018: 511-518.
- [28] MA F, LUO X D, WANG P. Random growth networks with exponential degree distribution[J]. Chaos, 2020, 30: 1-11.

编辑 薛晋栋