

分布式网络测量中测量节点的智能选择算法

张荣¹ 金跃辉¹ 杨谈^{1,2} 荣自瞻¹

(北京邮电大学网络与交换技术国家重点实验室 北京 100876)¹ (北京邮电大学软件学院 北京 100876)²

摘要 大规模网络结构复杂,需要有针对性的网络监测方法。测量节点的自动选择必须在测量代价和覆盖范围之间进行权衡。合理地测量节点选择,能在获取全网性能状况的同时,有效减少测量给待测网络带来的带宽占用和软硬件资源消耗的影响。以最小化测量节点数量为目标,选择蚁群算法作为测量节点自动选择的基本算法,并通过基本算法进行改进和创新,可形成一种针对分布式网络测量的测量节点智能选择算法。

关键词 网络测量,测量节点,智能选择,蚁群算法

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.015

Intelligent Selection Algorithm of Measurement Nodes in Distributed Network Measurement

ZHANG Rong¹ JIN Yue-hui¹ YANG Tan^{1,2} RONG Zi-zhan¹

(State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)¹

(School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)²

Abstract The complexity of large-scale networks calls for monitoring techniques of special consideration. The automatic selection of measurement nodes must make a balance between costs and coverage. With appropriate selection of measurement nodes, not only the performance status of the overall network can be obtained, but also the impact of monitoring on the monitored network in terms of bandwidth and consumption of software/hardware resources can effectively be reduced. By targeting minimum number of measurement nodes, applying ant colony optimization as the basic algorithm, and making improvements and innovations on the foundation of the basic algorithm, an intelligent selection algorithm of measurement nodes was formed and proposed.

Keywords Network measurement, Measurement nodes, Intelligent selection, Ant colony algorithm

1 引言

随着网络规模的不断扩大和业务种类的多样化, Internet 的发展给人们带来了巨大的方便;同时也使网络中出现各种异常的机会大大增加,给网络测量带来了更大的挑战。网络测量是对网络相关性能指标进行测量,主要包括链路流量、端到端带宽、时延和丢包等^[1]。分布式网络测量系统 TANC^[2] 是一个能够测量指定网络区域内网络性能指标的分布式系统,分为云监测平台子系统和测量探针子系统。云监测平台子系统基于 Hadoop 和 HBase 实现;测量探针子系统则部署在待测网络中的网络节点上,进行网络性能指标测量,并将测量结果回传至云监测平台进行处理分析,从而得到所测网络区域的网络性能状况。在 TANC 实现的分布式网络中进行网络性能监测,网络管理员的一项主要任务就是监测网络设备的运行情况和流量负载。当待测网络的规模不大、网络负载较低时,网络管理员可以在所测网络拓扑中的所有节点上部署探针,实现对网络性能的全网监测。而在监测大规模网

络时,如果选择在待测网络的所有节点上部署测量探针进行网络性能监测,不但会在该待测网络中产生较高的测量流量,也给测量系统自身以及待测网络中的节点带来巨大的软硬件资源消耗。为解决这个问题,可以从以下两个方向努力:一种是改进现有测量方法,降低测量带来的流量和软硬件资源消耗^[3];另一种是在待测网络中选择合理有效的测量节点^[4]。

本文旨在提出一种合理的网络测量节点的智能选择方法,在待测网络中以部分节点的测量结果反映整个待测网络性能,达到减少测量所产生的网络流量并降低因为执行测量任务而给网络节点带来的资源消耗的目的。

此外,网络的复杂性要求测量节点的智能选择算法能适应处理网络的各种异常情况,如网络节点数量的变化,网络节点的软硬件资源不足、节点故障、节点损坏问题,网络链路拥塞、故障等情况。测量节点的智能选择算法需要综合考虑各种网络状况,从而合理地选择节点。

近年来,已经有很多关于测量节点的自动选择问题的研究。关于算法模型的抽象,文献[5]提出可以将测量节点的自

到稿日期:2014-05-17 返修日期:2014-07-21 本文受国家 863 项目(2011AA01A102),国家 973 项目(2009CB320505),中央高校基本科研业务费专项资金(2014RC0501)资助。

张荣(1988-),女,硕士,主要研究方向为网络测量与评估、移动互联网软件, E-mail: kugirl@bupt.edu.cn; 金跃辉(1965-),女,硕士,教授,主要研究方向为网络测量与评估、移动互联网应用; 杨谈(1982-),女,博士,讲师,主要研究方向为网络测量与性能分析、社会网络分析、移动互联网; 荣自瞻(1990-),男,硕士,主要研究方向为网络测量与性能分析。

动选择问题映射为集合覆盖问题;Chaudet 等^[6]提出集合覆盖问题可以使用集合覆盖问题的优化算法来解决;Suh 等^[7]不仅指出基于流守恒的网络链路带宽测量问题可以映射为最小弱顶点覆盖问题,还提出该问题为 NP-Hard 问题。针对 NP-Hard 问题,初期的理论研究均为近似算法,文献^[8]提出了求解弱顶点覆盖问题的一种基于贪婪策略的近似算法,文献^[9]提出求解弱顶点覆盖问题的贪婪秩算法,文献^[10]提出了弱化的贪心算法。贪心算法能够在较短的时间内给出一个较为满意的结果,但却不能保证准确性,因为根据贪心的思想,在每一步都做了最优的决策,但从最终结果来看,却可能并非最优,可能存在更符合人们需求的节点选择策略。

目前来看,网络流量有效测量节点的自动选择方案,基本上是根据网络的拓扑结构建立部署模型来进行优化的,主要采用以下 3 种优化方法:映射到经典优化问题,利用经典优化问题的难解性结论和近似算法^[9];采用整数规划描述和解决优化问题^[11];设计启发式算法求近似解^[12]。网络流量有效测量节点的自动选择问题本质上是有待解决的全局优化问题。鉴于全局优化问题的数学理论体系还不完善,一方面人们需要进一步深化完善全局优化问题的数学理论,另一方面人们希望找到一种不需要预先在数学理论上承认,却能够直接达到优化效果的方法。这种情况下,启发式算法应运而生。启发式算法以其灵活性和易操作性等特征受到人们的青睐,其中元启发式算法^[13]发展迅速,取得了令人鼓舞的成果,例如蚁群优化算法(Ant Colony Optimization, ACO)、模拟退火算法(Simulated Annealing, SA)、遗传算法(Genetic Algorithms, GA)、禁忌搜索算法(Tabu Search, TS)、神经网络算法(Neural Networks, NN)等。这些元启发式算法从各种不同的角度利用不同的搜索机制和策略来实现全局优化性能。其中 ACO 算法模型和原理都非常简单,具有较强的鲁棒性,易于和其他启发式算法相结合,得到了学者们的研究与应用,并且已经取得了较快的发展。ACO 算法虽然拥有很多优点,但也存在一定的缺点,比如,搜索时间较长、容易陷入局部最优解,以及容易出现停滞现象。为弥补这些不足,进一步提高算法的正确性和搜索效率,可将 ACO 与其他启发式算法混合,结合各种算法的优势,形成混合优化算法,最终达到改进 ACO 算法性能的目的。文献^[4]基于此思想,在 ACO 算法基础上实现改进的混合优化算法,求解最小弱顶点覆盖问题,以实现分布式网络测量中有效测量节点的自动选择。该混合优化算法表现出了较好的性能,但直接应用于分布式网络测量时仍然存在诸多不足。

本文将蚁群算法应用于分布式网络测量中测量节点的自动选择,形成解决该问题的基本选择算法,并综合考虑节点负载、资源、权限信息等多种因素,对基本选择算法进行了改进和优化,提出了针对分布式网络测量的测量节点智能选择算法。

2 基于蚁群算法的测量节点智能选择问题描述

蚁群算法是由 M. Dorigo 等人于 1991 年提出的一种新型的模拟进化算法,已经被应用到旅行商问题、通信网络优化问题等许多类型的问题中,并表现出了良好的性能^[14]。蚁群优化算法作为一种新的元启发式算法,具有分布式计算、正反馈、自组织以及贪心启发等特点,可成功解决许多 NP-

Hard 组合优化问题。蚁群算法利用蚂蚁觅食行为与旅行商问题的相似性,模拟产生了一种仿生算法。蚂蚁通过信息素进行信息交流,根据信息素的多少进行路径选择和更新,迭代产生全局最优解。利用信息素进行的信息交流是一种正反馈机制。对比以往的其它模拟进化算法,蚁群算法具有如下优点^[15]:

(1) 蚁群算法结合了分布式计算、正反馈机制和贪婪搜索的思想,搜索较优解的能力较强。其中,分布式计算避免了早熟收敛问题的出现,正反馈能快速地发现较优解,而贪婪搜索则有助于在搜索过程早期找出可接受的解决方案,缩短了搜索时间。

(2) 蚁群算法并行性很强,个体之间不需要直接通信,可通过信息素进行合作,以这样的算法构建的系统具有更好的可扩充性。随系统中个体增加而增加的系统通信开销在这里将非常小,这样的特征增强了网络测量中有效测量节点的智能选择算法在规模日益扩大的互联网中的适应性。

(3) 蚁群算法具有较强的鲁棒性,且易于与其它启发式方法相结合,结合不同算法优势,形成混合优化算法,以改进和优化基本蚁群算法的性能,解决自身搜索时间较长、容易陷入局部最优,以及容易出现停滞现象的问题。

本文将蚁群算法引入到分布式网络测量中测量节点的自动选择问题中,将其解决最小顶点覆盖问题的算法作为测量节点自动选择的基本算法,并对其进行了优化和改进,提出了一种基于蚁群算法的适合分布式网络的测量节点智能选择算法。

分布式网络测量中测量节点的智能选择问题需要考虑两个重要因素:1)所选测量节点的代表性;2)所选测量节点的合理性。

所选测量节点需要有代表性,因为没有必要对待测网络中所有节点和链路均进行监测,而是希望通过在选出来的一部分节点上部署测量探针来达到监测整个待测网络性能状况的目的,所以,所选择的节点需要尽可能地覆盖整个待测网络的链路。分布式网络测量包括主动测量和被动测量^[16],主动测量和被动测量中的测量节点自动选择问题均可以映射为集合覆盖问题^[5]。考虑到分布式网络测量中采用了主动测量技术和被动测量技术的结合,以及可能存在部分骨干节点没有操作权限部署测量探针的情况,本文同样引入了图论中的覆盖这一概念,并将分布式网络测量中的测量节点自动选择问题抽象为顶点覆盖问题。容易看出,顶点覆盖既能够满足网络流量测量等被动测量的覆盖需求,同样也可以比较充分地满足主动测量中的节点覆盖需求,使得可测链路基本覆盖整个待测网络中的链路。最小顶点覆盖问题是一个 NP-Hard 问题,到目前为止,还没有多项式时间的求解算法。基于蚁群算法的最小顶点覆盖问题求解算法是一种启发式算法,是目前求解这一问题性能较好的一种算法,本文将在 3.1 节对其进行简要介绍。

所选测量节点需要有合理性,即所选节点的状态是否适合下发测量任务。比如从节点负载因素这一角度考虑,有些节点可能在某些时刻已经在运行多个测量任务或者其他应用,所以负载较高,在这样的情况下,如果再向此测量节点下发测量任务,可能因为节点负载过高而影响节点已在进行的测量任务或其他应用的正常执行,甚至可能造成节点宕机等

问题。一般来说,在其他因素都相同的前提下,负载越低的节点越适合选为测量节点。然而,在选择节点时首先需要考虑的是所选节点的代表性,即需要满足上文所述的最小顶点覆盖问题。所以在选点过程中,不是简单地直接将整个待测网络中负载最低的点选出,而是要将负载因素和前文所述的最小顶点覆盖问题结合起来,选出既具有代表性又有合理性的测量节点。为了实现这样的结合,对基本算法进行了改进,在求解最小顶点覆盖问题的过程中加入了负载这一指标,使得选出的点集在满足链路覆盖的基础上又能够满足整体负载状态最优,降低了测量给网络带来的负面影响。

此外,针对网络测量中的实际需求,还提出了测量节点的预选择、节点权值的预设、随机比例状态转移规则的改进、局部最优解的避免和终端节点的清理等,使得所提出的方法尽可能地满足分布式网络测量的需要。这些改进均在3.2节中详细地介绍。

3 基于蚁群算法的测量节点智能选择算法实现

3.1 基于蚁群算法的测量节点选择算法

本节将介绍一种基于蚁群算法的求解最小顶点覆盖问题的启发式算法(Ant Colony Optimization meta-heuristic, ACO算法)^[17-19]。分布式网络测量中测量节点的自动选择问题可抽象为最小顶点覆盖问题,所以本文将此算法作为分布式网络测量中测量节点自动选择的基本算法。

ACO算法的基本思路是一种贪心算法,该贪心算法的基本步骤如下。

算法1 贪心算法

1. 将网络拓扑图抽象为关联矩阵,无向图 $G=(V, E)$ 的关联矩阵 $A=(a_{ij})$ 为如下定义的 $n \times m$ 矩阵:
$$a_{ij} = \begin{cases} 1, & \text{如果顶点 } v_i \text{ 与 } e_j \text{ 相关联} \\ 0, & \text{其他} \end{cases}$$
2. 选取网络拓扑中一个拥有边数最多的顶点(关联矩阵中行元素和最大的点) v_i 加入最小弱顶点覆盖集合 S_k ;
3. 删除关联矩阵中顶点 v_i 所对应的行元素及该行中值为1的元素所在的列;
4. 在修改后的关联矩阵中删除节点度数(行元素之和)不超过1的其他行的所有行元素,以及这些行中值为1的元素所对应的列元素,直到不能再删除新的行和列为止;
5. 重复步骤2—步骤4,直到包含所有的边,即 $A=0$,从而得到该近似算法的最小弱顶点覆盖集 S_k 。

随着问题规模的扩大,通过此近似算法求解得出的最小顶点覆盖集将是一个近似解,而不是最优解。所以为了求解最小顶点覆盖问题的最优解,必须在此近似解的基础上,继续使用ACO算法对其进行全局寻优^[15]。

ACO算法引入信息素的概念,由每个顶点上的信息素轨迹和期望启发信息共同确定选择顶点加入最小弱顶点覆盖集 S_k 的概率。随机比例状态转移概率的计算按式(1)进行:

$$p_i^k = \frac{\tau_i^{\alpha} \eta_i^{\beta}}{\sum_{j \in C_k} \tau_j^{\alpha} \eta_j^{\beta}} \quad (1)$$

其中, τ_i 表示顶点 v_i 的信息素轨迹强度, η_i 表示顶点 v_i 的期望启发信息, C_k 表示蚂蚁 k 的候选顶点集; α 为信息启发式因子, β 为期望启发式因子,这两个参数都是常量,但其值的选取需要依据经验人为指定。

在选择顶点时,尤其是节点度数相同的顶点时,我们既希

望该顶点排除尽量少的顶点,同时也希望它能排除尽可能多的边。因此,采用动态方式获得顶点的期望启发信息,其表达式为:

$$\eta_i = \frac{\sum_{v_j \in N(v_i)} |v_j|}{|v_i|} \quad (2)$$

其中, $N(v_i)$ 表示与顶点 v_i 相邻的顶点集合。

当顶点 v_i 被蚂蚁 k 选择之后,将该顶点加入蚂蚁 k 的最小弱顶点覆盖集 S_k 中,同时将该顶点和与其相连的所有边都删除,然后按照上述贪心算法思想更新关联矩阵,继续删除顶点和边,直到没有任何可以删除的顶点和边。重复以上步骤,直到关联矩阵 $A=0$,即所有边都已被最小弱顶点覆盖集中的顶点所覆盖,那么这个集合 S_k 就是蚂蚁 k 构造的顶点覆盖集。

每次迭代过程完成后,采用最大最小蚂蚁系统(Max-Min Ant System, MMAS)的信息素更新规则^[20]更新信息素,信息素的更新包括信息素的挥发和信息素的增加。定义一个信息素挥发系数 $\rho \in [0, 1]$ 模拟信息素挥发:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) \quad (3)$$

每次蚁群循环后,循环最优蚂蚁进行信息素释放,相应顶点的信息素增量表达式如下:

$$\Delta\tau_i(t, t+1) = \frac{Q}{1 + |WVC_c| - |WVC_g|} \quad (4)$$

其中, WVC_c 是循环最优解,即本次循环中蚂蚁 k 找到的包含顶点数最少的独立集;而 WVC_g 则是全局最优解,即算法运行以来蚁群找到的包含顶点数最少的独立集。在蚁群完成每次循环之后,循环最优蚂蚁把一定量的信息素释放在 WVC_c 中的各个顶点上。通过循环最优蚂蚁信息素的释放,循环最优解 WVC_c 中顶点的信息素大量增强。 Q 为信息素增量系数,其值由经验所定, $|WVC_c|$ 和 $|WVC_g|$ 分别表示独立集 WVC_c 和 WVC_g 中的顶点个数。则信息素最终的表达式为式(3)和式(4)的和,即

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \frac{Q}{1 + |WVC_c| - |WVC_g|}$$

为防止算法运行中各顶点上的信息素差异过大,各顶点的信息素轨迹强度被限制在 $[\tau_{\min}, \tau_{\max}]$ 范围内,在初次迭代前,各顶点的信息素轨迹强度被初始化为 τ_{\max} 。

基于蚁群算法的测量节点的基本选择算法,简称基本选择算法,其具体实现步骤如下,详细步骤描述见文献^[15]。

算法2 基本选择算法

1. 参数初始化。令时间 $t=0$, 循环次数 $N_c=0$, 设置最大循环次数 N_{\max} , 将 m 只蚂蚁随机置于 n 个网络节点上。令有向图上每条边 (i, j) 的初始信息素量为设定的最大信息素值 τ_{\max} 。
2. repeat, 循环次数 $N_c \leftarrow N_c + 1$
3. for 蚁群中的每一只蚂蚁 k
4. 随机选择点 v_i 加入最小弱顶点覆盖集 S_k , 并以 v_i 作为搜索的起始点
5. 根据问题约束条件更新候选解集合 C_k
6. while 候选解集合 C_k 非空 do
7. 根据状态转移概率函数 p_i^k 的值选择下一网络节点 v_j 加入最小弱顶点覆盖集 S_k
8. 根据问题约束条件更新候选解集合 C_k
9. end of while
10. end of for

11. 根据蚁群算法中每只蚂蚁找到的解的质量进行信息素的全局更新或局部更新
12. until 满足循环结束条件或者循环次数 $N_c \geq N_{max}$
13. return 最终解

虽然上述基本算法具有诸多优点,比如该方法能够选出能最大限度地覆盖待测网络区域的网络链路的最少节点集;同时,该算法利用正反馈的原理,在一定程度上加快了进化过程,而且是一种本质并行的算法,不同个体之间不断进行信息交流和传递,从而能够相互协作,有利于发现较优解。然而,该方法也存在着很多不足之处,具体如下:

(1)基本算法在进行信息素的初始化时,将所有节点的信息素都初始化为最大值。而在分布式网络测量中,由于网络的拓扑特征以及网络上不同业务的部署特点,会出现某些节点的测量价值高于其他节点的情况。在这种情况下,统一初始化信息素初始值的做法无法体现节点之间的价值区别。

(2)该方法仅以顶点覆盖作为选点标准,未考虑节点的状态。而实际测量过程中,可能存在着某些节点负载较高而不适合选为监测点的情况,若选点过程中未考虑这一因素,则选点后执行测量任务时可能会给个别节点造成过大压力,甚至影响测量任务和节点上运行的其他应用的正常进行。

(3)由于蚁群算法利用了正反馈原理,使得解的性能直接受到正反馈强度的影响。如果正反馈强度太小,会造成优化过程较慢,需要较长的搜索时间。而若正反馈强度太大,优化过程又容易陷入局部最优,出现停滞现象,即搜索进行到一定程度后,所有个体所发现的解完全一致,不能对解进行进一步搜索,不利于搜索全局最优解。

(4)在分布式网络测量中,可能存在对某些测量节点不具备操作权限的情况。这样的节点无法进行测量软件部署。然而,在某些非终端节点因为没有操作权限而被从最优解中排除时,为了实现顶点覆盖,可能大量与该节点相连接的终端节点会被选入最小覆盖点集。然而,在分布式网络测量中,终端节点可能占据很大数量。对于某个待测网络区域的性能衡量来说,将大量终端节点与其上一级节点之间的链路都纳入测量范围是没有必要的,也会造成一定的资源浪费。

此外,用户在特殊需求下,希望自动选择算法选择的测量节点一定包含某些特殊节点,即用户需要在算法自动选择测量节点之前做节点的预选择,或在算法自动选择测量节点后进行节点的微调。基于这种需求,在 TANC 系统中将其定义为用户制定模式,用户可以在系统建议选点方案之前预选择节点,指定必须覆盖的中心节点或者主干网络设备等。在系统根据全网拓扑结构和节点属性得出选点方案后,用户可以根据实际情况和需求对选点结果进行微调。在系统给出方案后进行节点调节并不影响网络测量节点的自动选择,而在系统给出自动选点方案前对网络节点预选择,实现人工选点和智能选点相结合,还需要进一步的研究。

3.2 基于蚁群算法的测量节点智能选择算法

针对上节所提到的基于蚁群算法的测量节点的基本选择算法的不足,本节对基本选择算法进行了优化和改进,提出了测量节点的智能选择算法。在基本算法的基础上,本文所做改进和优化主要包括:算法开始前对节点的预选择,算法开始对节点权值的预设、改进算法随机比例转移规则、提出局

部最优解避免策略和清理最小顶点覆盖集中的终端节点等。下面就这些改进依次进行介绍。

3.2.1 测量节点的预选择

在算法自动选择测量节点之前,用户可以根据自身需求,对网络拓扑中的节点进行预选择,即给出用户的期望,期望在所给网络监测节点中哪些测量节点是必须覆盖的中心节点或者主干网络设备。我们称与同一个骨干节点直接相邻的网络节点为兄弟节点。当兄弟节点均处于网络边缘即均为终端节点时,它们在反映链路性能状态上起到的作用也相近。我们认为,这样的兄弟节点在监测系统中只选其一即可。所以,当用户给出预选节点后,如果该节点属于网络边缘节点,首先清理兄弟节点(即预清理),将所给预选节点的兄弟节点全部清除。此举减小了参与蚁群算法选点的节点规模,有效降低了算法计算的复杂度,同时不会对整个网络性能的监测和评估产生影响。

3.2.2 节点权值的预设

在算法自动选择测量节点初次迭代前,需要先将各顶点的信息素轨迹强度初始化为 τ_{max} 。在分布式网络测量中,会存在不同测量节点的基本条件不同或用户对不同节点位置有不同测量需求的情况。在这种情况下,本节在改进算法中加入了节点权值的预设环节,并将这一预设加入信息素的初始化环节。根据用户设定的权值,来进行信息素的初始化。在这里,将初始化信息素计算公式定义为:

$$a\tau_{max} + b\tau_{max} \cdot p \quad (5)$$

其中, τ_{max} 为信息素最大值, p 为人工设定的节点权值, a 和 b 为信息素初始化系数,均在 $[0,1]$ 之内,且 a 和 b 满足关系:

$$a + b = 1 \quad (6)$$

其中, a 和 b 的具体值由经验指定,在这里, b 是初始化信息素受人为设置权值影响的最大比例。观察式(5)可得,节点权值的预设将影响到节点的初始化信息素大小,节点权值越大,初始化信息素越大,越有利于该节点被系统选为最小顶点覆盖集的候选节点。

3.2.3 随机比例状态转移规则的改进

蚁群算法的搜索过程将基于随机比例状态转移概率确定搜索路径。如上文所述,基本算法在选点时并没有考虑到节点的状态信息,然而节点状态信息,特别是节点负载状况,会对网络测量产生重要的影响。一般地,会优先选择负载较低的节点,这有利于减轻测量节点的负载压力,有效地均衡节点间的负载水平,避免因执行测量任务使得负载过度集中到部分测量节点而对这部分节点上测量任务以及其它应用的正常运行造成影响。文献[21]在基于云环境的虚拟机批量部署问题中,提出在随机比例规则中加入性能感知的策略。受此启发,对随机比例状态转移规则进行了改进,将节点的状态信息加入到随机比例状态转移规则中,以寻找更符合测量条件的测量节点。

首先,定义了负载的计算方法,如下式所示:

$$g = 0.5v_{cpu} + 0.5v_{memory} \quad (7)$$

其中, v_{cpu} 表示 CPU 利用率, v_{memory} 表示内存利用率。经过改进的随机比例状态转移规则定义为:

$$p_i^k = \frac{\tau_i^g \eta_i^g (1 - f(g))^m}{\sum_{j \in C_k} \tau_j^g \eta_j^g} \quad (8)$$

其中, τ_i 为顶点 v_i 的信息素轨迹强度, η_i 为顶点 v_i 的期望启发信息, C_k 为蚂蚁 k 的候选顶点集; α 是信息启发式因子, β 是期望启发式因子, m 是负载影响因子, 这 3 个参数都是常量, 但值的选取需要依据经验人为指定。 $f(g)$ 为节点负载转换函数, 将其定义为:

$$f(g) = \begin{cases} g, & g \leq 0.8 \\ 1, & g > 0.8 \end{cases} \quad (9)$$

其中, g 为式(7)的计算结果。当节点负载大于 0.8 时, $f(g)$ 的值为 1。此时将 1 代入式(8)可知, 随机比例状态转移概率将为 0, 即该点不会被选为下一个路径点。当 $g \leq 0.8$ 时, 随机比例状态转移概率将随着节点负载的增大而减小, 其减小趋势与负载影响因子 m 有关。

通过此改进的随机比例状态转移规则表达式, 本文加入了节点负载这一节点选择过程中的重要考量因素, 使得节点的选择更科学合理。

3.2.4 局部最优解的避免

在基本蚁群算法中, 根据随机比例状态转移规则, 蚂蚁总是倾向于选择当前解中较好的解, 导致了较好解路径上的信息素增大, 继而在下次迭代时被选择的概率也增大。这种情况容易降低其他路径被选择的可能性, 致使算法易于陷入局部最优解。为了避免这种算法早熟现象, 需提出一种有效的局部最优解避免策略, 扩大选择概率的范围, 即不仅选取当前最好解, 还有机会选择其他解^[22]。本文在当前对此问题的常用解决方案的基础上, 提出一种随机扰动和收敛调整相结合的局部最优解避免策略。

本策略包括两项具体策略, 第一, 在优化过程中加入随机扰动。这是一种比较常用的避免局部最优解问题的处理方案。本文采用的随机扰动方案为设定一个扰动因子, 并以此扰动因子影响随机比例状态转移概率, 带有随机扰动的随机比例状态转移概率如下式所示:

$$p_i^k = \frac{\tau_i^{\alpha} \eta_i^{\beta} (1-f(g))^m}{\sum_{j \in C_k} \tau_j^{\alpha} \eta_j^{\beta}} \cdot \lambda \quad (10)$$

其中, λ 为所定义的扰动因子:

$$\lambda = \begin{cases} k^{\frac{1}{3}}, & r_i^k < k \\ 1, & \text{else} \end{cases} \quad (11)$$

其中, k 为定义的扰动概率, r_i^k 为 0 到 1 之间的一个随机数。在计算每个节点的随机比例状态转移概率时, 将有 k 概率的可能性产生扰动, 而在其余情况下 λ 为 1, 也即转移概率保持原值, 并未发生扰动。随机扰动的加入, 有效地增加了随机选择过程中解的多样性, 有利于解空间的深度探索。

为了能进一步进行最优解探索, 在此基础上, 本文提出的局部最优解避免策略还引入了另一个策略, 即收敛调整策略。收敛调整是指在搜索到一定程度时, 对节点的信息素分布进行调整。设定的调整条件为当解集第一次收敛或循环次数达到最大循环次数的 1/2 时, 缩小信息素较高的节点和信息素较低的节点之间的信息素差距, 并继续进行搜索, 在再次达到收敛时, 得到最优解。这一调整策略具体如下: 首先对所有节点的信息素按照从大到小的顺序排序:

$$\tau_{i_1} > \tau_{i_2} > \tau_{i_3} > \dots > \tau_{i_n} \quad (12)$$

对式(12)中的信息素进行调整, 将所有信息素值按顺序

平均分为 τ_{i_1} 到 $\tau_{i_{\lfloor n/2 \rfloor}}$, 和 $\tau_{i_{\lfloor n/2 \rfloor+1}}$ 到 τ_{i_n} 两部分。然后按照对应关系进行信息素的调整, 调整公式如下:

$$\tau_k = (0.5+t)\tau_k + (0.5-t)\tau_{n-k} \quad (13)$$

$$\tau_{n-k} = (0.5-t)\tau_k + (0.5+t)\tau_{n-k} \quad (14)$$

其中, t 为信息素调整系数 ($t \in (0, 0.5)$, $k \in [1, \lfloor n/2 \rfloor]$)。对于这个调整, 给出如下证明:

证明: 对于 $1 \leq k \leq \lfloor n/2 \rfloor$, 有

$$\tau_k = (0.5+t)\tau_k + (0.5-t)\tau_{n-k}$$

$$\leq (0.5+t)\tau_k + (0.5-t)\tau_k$$

$$\leq \tau_k$$

$$\tau_{n-k} = (0.5-t)\tau_k + (0.5+t)\tau_{n-k}$$

$$\geq (0.5-t)\tau_{n-k} + (0.5+t)\tau_{n-k}$$

$$\geq \tau_{n-k}$$

所以, 调整后的 $\tau_k - \tau_{n-k}$ 的值一定比原来小, 故可以减小差距。通过这一调整, 在解集趋于收敛时, 缩小了较高信息素和较低信息素之间的差距。完成信息素调整以后, 继续进行搜索, 当解再次达到收敛时, 即认为得到了最终的搜索结果。

3.2.5 终端节点的清理

在实际的待测网络中, 可能存在部分网络节点没有权限部署测量探针, 或者部分测量节点出现故障等一些情况的发生。在这种情况下, 基本选择算法将无法进行测量的测量节点从选点结果中删除以后, 为了实现顶点覆盖, 会相应地选出其他节点来进行补充。那么, 存在一种情况, 即在终端节点较多的待测网络拓扑图中, 若终端节点的上一级节点不具备操作权限或探针出现故障, 则与该节点相连接的所有未被覆盖的度为 1 的终端节点也会被加入选点结果集。而在网络测量中, 我们认为处于网络边缘区域的链路的性能状况对于整个待测网络的性能状况并没有明显影响, 所以对于同一类型的终端节点的重复性测量是没有必要的。因此, 本节加入了终端节点清理策略。

所谓终端节点清理, 是指在蚁群算法搜索完成后, 对于搜索结果中同一类型的终端节点, 只保留一个, 对于其余节点则从结果集中删除。这里所说的“同一类型的终端节点”, 是指有且只有一个父节点并且父节点相同的叶子节点。同一类型的终端节点互称为兄弟节点。对于具备这一特征的终端节点, 将进行终端节点的清理。终端节点清理策略描述如下: 对选出的测量节点集合进行判断, 若存在同一类型的终端节点, 则在其中通过人工设定的节点权值 p 选择节点权值最大的一个节点。如果 p 相同, 则随机保留一个节点, 并删除节点的兄弟节点。

终端节点清理将有效地缩减测量节点结果集的规模, 而对待测网络整体性能测量的影响微乎其微, 有效减轻了测量给待测网络和各测量节点带来的压力。

根据上述 5 点改进方法, 提出基于蚁群算法的测量节点的智能选择算法, 简称智能选择算法, 其具体实现步骤如下。

算法 3 智能选择算法

1. 测量节点预选择, 清理预选节点的兄弟节点, 简化网络拓扑。
2. 参数初始化, 令时间 $t=0$, 循环次数 $N_c=0$, 设置最大循环次数 N_{\max} , 将 m 只蚂蚁随机置于网络节点上。
3. 按照节点的重要度差异, 进行节点权值的预设定。
4. 根据各节点预设定的权值 p , 初始化各节点的信息素大小为 $\alpha\tau_{\max} +$

$b_{r_{max}} \cdot p_0$

5. repeat, 循环次数 $N_c \leftarrow N_c + 1$
6. for 蚁群中的每只蚂蚁 k
7. 随机选择顶点 v_i 加入最小弱顶点覆盖集 S_k , 并以 v_i 作为搜索的起始点
8. 根据问题约束条件更新候选解集合 C_k
9. while 候选解集合 $C_k \neq \emptyset$ do
10. 根据加入随机扰动因子的概率选择函数 $p^k(10)$ 选择一个顶点 v_j 加入最小弱顶点覆盖集 S_k
11. 根据问题约束条件更新候选解集合 C_k
12. end of while
13. end of for
14. 根据蚁群中每只蚂蚁找到的解的质量进行信息素的全局更新或局部更新
15. until 循环次数达到最大循环次数的 1/2 或者达到第一次收敛
16. 按照式(13)和式(14)调整信息素, 缩小信息素较高节点和信息素较低节点之间的信息素差距, 并继续进行搜索, 重复步骤 5—步骤 15, 直到再次达到收敛。
17. return 最优解

4 算法验证

4.1 在 Planetlab 测量中的智能选择算法验证

在我们的 TANC 系统中, 可以实现对 Planetlab 网络的管理和测量。Planetlab 上所有节点都连接到 Internet。这些节点绝大多数都与地区和长途骨干路由存在连接。通过网络可以实现真实世界条件下分布式网络测量中测量节点的智能选择。图 1 为 TANC 系统所管理的一个 Planetlab 网络中网络节点的分布情况。该网络由 73 个 Planetlab 网络节点组成。首先抽象出 Planetlab 上的层级网络拓扑图, 将其所有节点按 Internet(中心云)、国家(骨干路由 $R_i, i \in (1, 12)$)、Planetlab 节点(带序号的小圆圈, 序号从 1—73)分为三级, 得到如图 2 所示的 Planetlab 网络拓扑层级图。

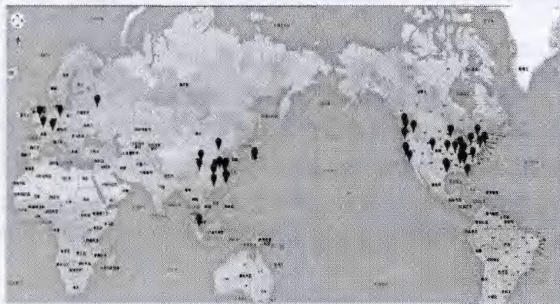


图 1 Planetlab 网络节点分布图

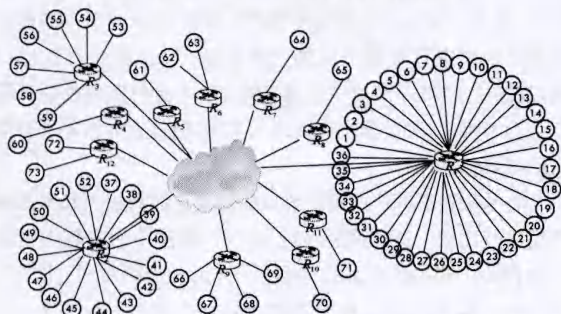


图 2 三级 Planetlab 网络拓扑层级图

在三级 Planetlab 网络拓扑层级图的基础上, 对智能选择算法进行验证。

4.1.1 测量节点的基本选择算法选点实验

按照算法 1 提到的测量节点选择基本算法, 可求出的最小覆盖点均为骨干节点, 如图 3 所示, 将被选择的节点用深色标识, 如 R_1, R_2 等。

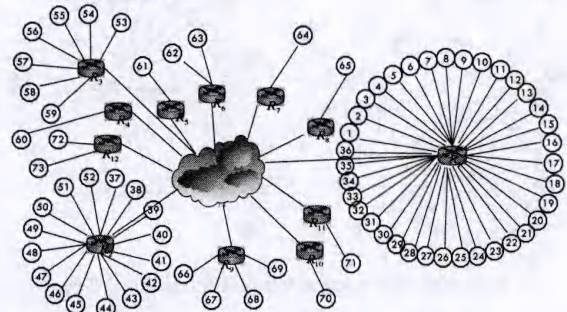


图 3 蚁群基本算法算出的测量节点分布图

通过拓扑观察, 很容易发现所选节点是能够实现网络所有边覆盖的最少顶点。这个结果在一定程度上说明了基本选择算法可以实现集合的最小顶点覆盖需求。

4.1.2 智能选择算法的测量节点预选择问题

测量节点的预选择方式, 有助于减小参与蚁群搜索的网络节点规模, 从而提高运算效率。首先, 基于图 2 所示的 Planetlab 网络拓扑层级图, 选择一个节点作为预选点, 运行测量节点的智能选择算法, 并观察智能选择算法的程序运行总时间和蚁群搜索时间。表 1 的节点 ID 是预选节点 ID, 我们选择 6 个有代表性的节点(与不同的骨干路由相连, 且兄弟节点个数互不相同)分别作为预选节点, 用于计算蚁群搜索所需时间和程序的总运行时间。计算结果及不同预选节点条件下的算法运行时间如表 1 所列。

表 1 不同预选节点条件下算法运行时间的比较

预选节点 ID	兄弟节点个数	预清理的兄弟节点 ID	程序运行总时间(ms)	蚁群搜索时间(ms)
1	35	2—36	7090	5888
37	15	38—52	7922	6643
53	6	54—59	8434	7244
66	3	67—69	8663	7650
62	1	63	8942	7820
61	0	0	9329	8068

不进行节点预选择时, 算法所需要的总运行时间为 9273ms, 蚁群搜索的时间为 8275ms。从表 1 中可以看到, 进行节点预选择可以有效减少蚁群搜索时间, 而且预清理掉的兄弟节点个数越多, 蚁群所需要的搜索时间越少。因为进行预选节点的兄弟节点的清理时, 程序需要一部分计算时间, 所以程序所需要的总时间不一定会减少, 如预选节点 ID 为 61 时, 该节点没有兄弟节点, 所以并没有给程序带来搜索时间的减少, 反而增加了节点预选择开销。但是绝大部分情况下, 如果预选节点的兄弟节点个数不为 0, 进行节点预选择可以提高程序的运算效率。

4.1.3 智能选择算法的测量节点权值预设定问题

在分布式网络测量中, 可能存在某些测量节点不具备操作权限, 这样的节点无法进行测量软件部署。如假设骨干节点 R_0 不具备操作权限, 为了实现顶点覆盖, 节点 66, 67, 68,

69 可能被选择,而选择的依据是这些节点的节点权值。节点权值越大,说明该节点在网络中的位置越重要,其越可能被选择为最终的测量节点。图 4 为所有的骨干节点都不具备操作权限的一种极端情况,将最终选择的节点用深色标识。

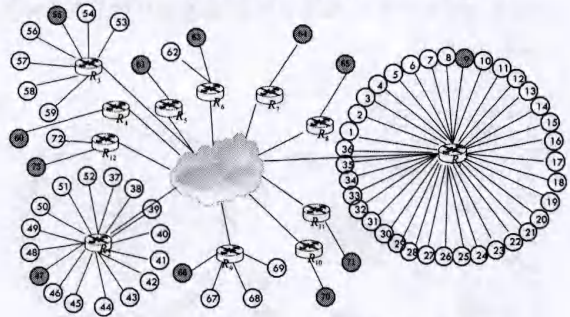


图 4 骨干路由由节点均无操作权限的测量节点分布图

4.1.4 智能选择算法的随机比例状态规则的改进

上节提到了当骨干节点不具备操作权限时,需要选出不少于 1 个的子节点替代该骨干节点的位置,并指出,子节点权值越大,该节点越重要,越可能被选为最终的测量节点。这里说的是可能,而非一定,是因为节点权值越大,该节点的初始信息素越大,越容易被选为候选节点。而根据随机比例状态转移规则(见式(8)),节点被选为候选节点的概率是由信息素和该节点的负载状况共同决定的。权衡节点负载和信息素的大小,通过 P_i^k 的值选择下一个候选节点。加入负载因素考量所选择的节点,其软硬件资源充分,是适合作为探针测量的节点。

4.1.5 智能选择算法的终端节点清理问题

图 4 为在骨干节点均无操作权限的情况下采用智能选择算法计算出的测量节点分布图。如果从最小顶点覆盖问题来看,很显然,我们并没有完成所有边覆盖的目标,甚至有很多边都没有被覆盖。但是,我们认为这种选点是合理的。因为,我们并不是严格要求解决最小顶点覆盖这样一个数学问题,而是在待测网络中选择部分网络的节点部署测量探针,并以此测量结果反映出整个待测网络的性能状况。

我们所选择的节点既减少了测量所产生的网络流量,又有效降低了因为执行网络性能测量任务而给网络测量节点带来的软件硬件资源消耗。同时,通过实际网络测量数据证明,在具有同一父节点的兄弟节点上进行网络性能指标测量(如链路流量、端到端带宽、双向时延和丢包率等)得到的结果非常相近,而在不同父节点上的子节点进行网络性能测量得到的指标数值相差较大,包含了较多的信息。图 5—图 8 表示的是 Planetlab 上 4 个终端节点的双向时延测试结果。

图 5 和图 6 所示的双向时延数据来自中国北邮校区的两个 Planetlab 节点,在相同的时间内,两者的时延变化趋势几乎一致。图 7 为中国上海交大节点的时延数据,时延变化趋势和北邮的两个节点也比较接近。而在同一时间,图 8 显示的来自哈佛的节点数据时延变化趋势与北邮、上海交大节点明显不同。以上数据证明,在分布式网络中,对于网络测量来说,所处位置越相近的节点起到的作用越相似;所处位置差别越大的节点起到的作用越不同。很明显,我们需要的是来自不同位置的网络测量数据来反映全网的网络性能。因此,将

测量网络进行合理的层次划分,并实现终端节点清理的做法是合理的、可靠的,不会因为实现所有的边覆盖,而给网络测量带来局部信息的缺失。

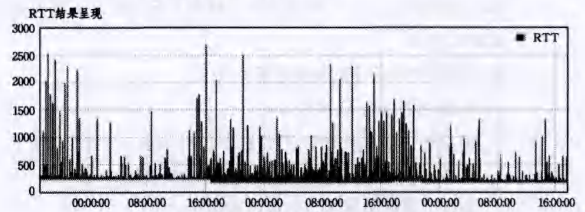


图 5 Planetlab 北邮节点 1 所得到的双向时延测试结果

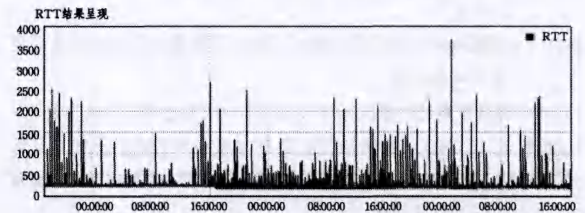


图 6 Planetlab 北邮节点 2 所得到的双向时延测试结果

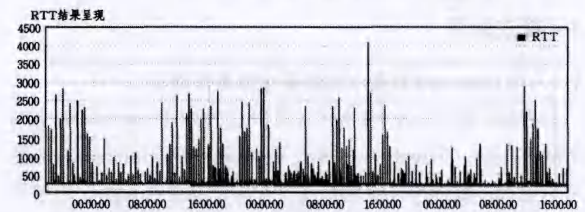


图 7 Planetlab 上海交大节点所得到的双向时延测试结果

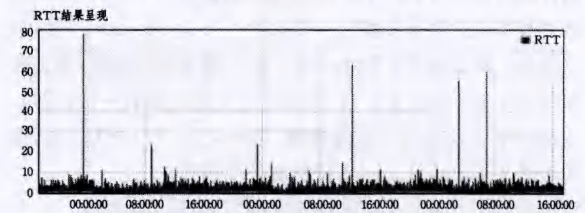


图 8 Planetlab 哈佛大学节点所得到的双向时延测试结果

4.2 基于 Waxman 网络模型的算法仿真实验

基于 Planetlab 测量的智能选择算法验证实验中并没有提到局部最优避免策略的实验效果,是因为所构建的网络拓扑层次比较简单,采用蚁群算法并没有陷入到局部最优解中。局部最优避免策略没有明显反映出其优化效果,反而增加了计算时间。

鉴于基于 Planetlab 测量的算法验证实验中节点的规模较小,需要更为复杂的网络去验证本文智能选择算法的性能。因此,采用了 Waxman 网络模型产生具有 1000 个节点规模的拓扑网络来进行测量节点的选择实验^[9,23]。算法每次运行前,均在这 1000 个节点中随机选择 950 个具有监测设备部署权限的节点,同时根据实验网络的实际状态,人为指定所有节点的负载信息。

将基于蚁群算法的测量节点的智能选择算法与基本选择算法进行比较。每种算法运行 10 次。算法中的参数根据实验和相关文献的研究进行选取^[24,25],由于篇幅所限,本节不再详述。本次实验选择的参数为: $\alpha=1, \beta=2, \rho=0.7, k=0.1, m=0.3$ 。

基本选择算法与智能选择算法的比较结果如表 2 所列。

表 2 基本算法与智能选择算法的实验结果比较

结果评价参数	基本选择算法	智能选择算法
最优解(节点数量)	472	399
平均解(节点数量)	573	452
平均循环次数	47	53
最小循环次数	39	43
全部平均负载	0.73	0.51
最小平均负载	0.51	0.36

从表 2 可以看出,所提出的基于蚁群算法的测量节点智能选择算法在最优解和平均解的解集规模上均明显小于基于蚁群算法的测量节点的基本选择算法,这表明智能选择算法在解的质量上有明显提高。而智能选择算法的收敛所需的平均循环次数和最小循环次数虽然均大于基本算法,但差距并不明显。由此可以看出,智能选择算法虽然在解的质量上有所提高,但在计算性能上有一定下降。但是,因为对一个节点进行测量所产生的时间和资源开销要远远大于算法执行过程中的一次循环所产生的时间和性能开销,所以,通过对智能选择算法和基本选择算法在解集规模和算法循环次数上的差距进行比较,认为智能选择算法在计算性能上的下降在一个可以接受的范围内,而分布式测量系统的测量性能仍然会因解集质量的提高而得到明显提升。式(7)已经给出了负载的计算方法,在实验中,本文计算了 10 次实验中节点平均负载最小的一次实验的平均负载值和 10 次实验中全部节点的平均负载值,可以看到,智能选择算法在全部实验平均负载上要显著于基本选择算法,而智能选择算法的最小平均负载也要小于基本选择算法。由此可以看出,在考虑测量节点负载状况的前提下,智能选择算法选出的节点在一般情况下的合理性要显著优于基本选择算法。

结束语 分布式网络测量中测量节点的自动选择问题依然是当前研究的一个热点。本文将蚁群算法解决最小顶点覆盖问题的思路引入到分布式网络测量中测量节点的自动选择问题中,并将其作为测量节点自动选择的基本选择算法。在此基础上根据实际测量中的需求对基本选择算法进行了优化和改进,包括测量节点的预选择、节点权值的预设、随机比例状态转移概率中负载因子的加入、局部最优解避免策略和终端节点清理策略的提出等,进而提出了一种基于蚁群算法的测量节点的智能选择算法。

为验证本文所提的测量节点智能选择算法的实用性和有效性,本文进行了 Planetlab 测量实验和 Waxman 仿真实验,将智能选择算法同基本选择算法进行了比较。实验证明,本文提出的基于蚁群算法的测量节点智能选择算法既能满足分布式网络测量中网络链路的覆盖需求,又能够有效保证所选节点的质量,使得选出的节点在数量和平均负载水平上均明显优于基本选择算法,较好地满足测量需要,在分布式网络测量中是可行和有效的。本算法尚还存在一定的不足,比如因为对基本选择算法的优化使得算法在性能上有一定程度的下降,如何最大限度地减小优化改进对算法性能的影响还需要在今后进一步研究。

参 考 文 献

[1] 朱畅华,裴昌幸,李建东,等. 网络测量及其关键技术[J]. 西安电

子科技大学学报,2002,29(6):813-818

Zhu Chang-hua, Pei Chang-xing, Li Jian-dong, et al. Network measurement and its key technologies [J]. Journal of Xidian University, 2002, 29(6): 813-818

- [2] 金跃辉. 大规模网络及业务性能动态监测系统 V1.0 [Z]. 北京邮电大学, 2013
- Jin Yue-hui. Large-scale network and business performance monitoring system V1.0 [Z]. Beijing University of Posts and Telecommunications, 2013
- [3] Wei Xiao, Wu Wei, Liu Yun-xiang. A network monitor system model with performance feedback function [C]// International Conference on E-Business and Information System Security, 2009 (EBISS'09). 2009; 1-5
- [4] 葛洪伟,彭震宇,岳海兵. 基于混合优化算法的网络流量有效测量点选择[J]. 计算机应用研究, 2009, 26(4): 1480-1486
- Ge Hong-wei, Peng Zhen-yu, Yue Hai-bing. Hybrid optimization algorithm for efficient monitor-nodes selection in network traffic [J]. Application Research of Computers, 2009, 26(4): 1480-1486
- [5] 蔡志平,刘芳,赵文涛,等. 网络测量部署模型及其优化算法[J]. 软件学报, 2008, 19(2): 419-431
- Cai Zhi-ping, Liu Fang, Zhao Wen-tao, et al. Deploying Models and Optimization Algorithms of network measurement [J]. Journal of Software, 2008, 19(2): 419-431
- [6] Chaudet C, Fleury E, Guérin Lassous I, et al. Optimal positioning of active and passive monitoring devices [C]// CoNEXT. 2005; 71-82
- [7] Suh K, Guo Y, Kurose J, et al. Locating network monitors: complexity, heuristics and coverage [C]// IEEE INFOCOM. 2005; 351-361
- [8] 刘湘辉,殷建平,卢锡城,等. 基于弱顶点覆盖的网络链路使用带宽监测模型[J]. 软件学报, 2004, 15(4): 545-549
- Liu Xiang-hui, Yin Jian-ping, Lu Xi-cheng, et al. A monitoring model for link bandwidth usage of network based on weak vertex cover [J]. Journal of Software, 2004, 15(4): 545-549
- [9] Breitbart Y, Chan Chee-yong, Garofalakis M, et al. Efficiently monitoring bandwidth and latency in IP networks [C]// Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001). 2001; 933-942
- [10] 刘湘辉,殷建平,唐乐乐,等. 网络流量的有效测量方法分析[J]. 软件学报, 2003, 14(2): 300-304
- Liu Xiang-hui, Yin Jian-ping, Tang Le-le, et al. Analysis of efficient monitoring method for the network flow [J]. Journal of Software, 2003, 14(2): 300-304
- [11] Raza S, Huang Guan-yao, Chuah C N, et al. Measurouting: A framework for routing assisted traffic monitoring [J]. IEEE/ACM Transactions on Networking, 2012, 20(1): 45-56
- [12] Dorigo M, Birattari M, Stutzle T. Ant colony optimization [J]. IEEE Computational Intelligence Magazines, 2006, 1(4): 36-39
- [13] Gallo M, D'Acerno L, Montella B. A meta-heuristic approach for solving the urban network design problem [J]. European Journal of Operational Research, 2010, 201(1): 144-157
- [14] 胡小兵,黄席隼. 蚁群优化算法及其应用[J]. 计算机仿真, 2004, 21(5): 81-85
- Hu Xiao-bing, Huang Xi-yue. Ant colony optimization algorithm and its application [J]. Computer Simulation, 2004, 21(5): 81-85

(下转第 93 页)

- ceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, Hong Kong, China, ACM, 2013, 36(4): 339-350
- [8] Chan A J, Pande A, Baik E, et al. Temporal quality assessment for mobile videos[C]//Proceedings of the 18th Annual International Conference on Mobile Computing and Networking. ACM, 2012: 221-232
- [9] Knoche H, De Meer H G, Kirsh D. Utility curves: mean opinion scores considered biased[C]//1999 Seventh International Workshop on Quality of Service, 1999 (IWQoS'99). IEEE, 1999: 12-14
- [10] Chen Kuan-ta, Tu Chen-chu, Xiao Wei-chen. OneClick: A framework for measuring network quality of experience[C]//INFOCOM 2009. IEEE, 2009: 702-710
- [11] Choi S G, Jong Kim H. A study on a QoS / QoE correlation model for QoE evaluation on IPTV service[C]//The 12th International Conference on Advanced Communication Technology, 2010 (ICACT). IEEE, 2010: 2, 1377-1382
- [12] Zoran Bojković, Zoran Miličević. Subjective Video Quality Assessment in H. 264/AVC Video Coding Standard [J]. Telfor Journal, 2012, 4(2): 111-115
- [13] Tan H L, Li Z, Tan Y H, et al. A Perceptually Relevant MSE-Based Image Quality Metric [J]. IEEE Transactions on Image Processing, 2013, 22(11): 4447-4459
- [14] 臧伟. 基于无线网络环境的手机视频码率自适应技术研究 [J]. 软件, 2012, 33(4): 81-83, 88
Zang Wei. Research of Mobile Video Rate Adaptive Method Based on Wireless Network Environment [J]. Computer Engineering & Software, 2012, 33(4): 81-83, 88
- [15] 邵阳, 马越, 郭捷. 移动互联网在线视频播放优化研究与实现 [J]. 微型电脑应用, 2009, 25(11): 10-12
Shao Yang, Ma Yue, Guo Jie. The Research and Implementation of the Optimization of Online Video Player Based on Mobile Network [J]. Microcomputer Applications, 2009, 25(11): 10-12
- [16] Jumisko-Pyykk S, H J, Kkinen. Evaluation of subjective video quality of mobile devices [C]// Proceedings of the 13th annual ACM international conference on Multimedia. Hilton, Singapore; ACM, 2005: 535-538
- [17] Garcia M, Canovas A, Edo M, et al. A QoE Management System for Ubiquitous IPTV Devices [C]// Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009 (UBICOMM'09). Siema, 2009: 147-152
- [18] Jumisko S H, Ilvonen V P, Vaananen-Vainio-Mattila K A. Effect of TV content in subjective assessment of video quality on mobile devices [C]// Electronics Imaging 2005. International Society for Optics and Photonics, 2005: 243-254
- [19] 余忆晗. 手机视频用户满意度调查研究 [J]. 城市建设理论研究 (电子版), 2013(21)
Yu Yi-han. The Research of Satisfaction of Mobile Video Users [J]. ChengShi Jianshe LiLun Yan Jiu, 2013(21)
- [20] Dobrian F, Sekar V, Awan A, et al. Understanding the impact of video quality on user engagement [J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 362-373
- [21] Watson A, Angela S M. Measuring perceived quality of speech and video in multimedia conferencing applications [C]// Proceedings of the Sixth ACM International Conference on Multimedia. ACM, 1998: 55-60

(上接第 77 页)

- [15] 彭震宇. 最大独立集和最小弱定点覆盖问题求解及其应用研究 [D]. 无锡: 江南大学, 2008
Peng Zhen-yu. Maximum independent set and minimum weak vertex cover problem solving and its application research [D]. Jiangnan University, 2008
- [16] 林宇, 程时端, 邬海涛, 等. IP 网端到端性能测量技术研究的进展 [J]. 电子学报, 2003, 31(8): 1227-1233
Lin Yu, Cheng Shi-duan, Wu Hai-tao, et al. The achievement of end-to-end performance measurement technologies in ip networks [J]. Wuxi: Acta Electronica Sinica, 2003, 31(8): 1227-1233
- [17] Dorigo M, Maniezzo V, Colomni A. Ant system: optimization by a colony of cooperating agents [J]. IEEE transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41
- [18] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66
- [19] Dorigo M, Caro G D. Ant colony optimization: a new meta-heuristic [C]// Proceedings of the 1999 Congress on Evolutionary Computation, 1999 (CEC 99). 1999
- [20] 王芳, 李美安, 段卫军. 基于动态自适应蚁群算法的云计算任务调度 [J]. 计算机应用, 2013, 33(11): 3160-3162
Wang Fang, Li Mei-an, Duan Wei-jun. Cloud computing task scheduling based on dynamically adaptive ant colony algorithm [J]. Journal of Computer Applications, 2013, 33(11): 3160-3162
- [21] 杨星, 马自堂, 孙磊. 云环境下基于改进蚁群算法的虚拟机批量部署研究 [J]. 计算机科学, 2012, 39(9): 33-37
Yang Xing, Ma Zi-tang, Sun Lei. Research on Extended Ant Colony Optimization Based Virtual Machine Deployment in Infrastructure Clouds [J]. Computer Science, 2012, 39(9): 33-37
- [22] 李成兵, 郭瑞雪, 李敏. 改进蚁群算法在旅行商问题中的应用 [J]. 计算机应用, 2014, 34(S1): 131-132, 165
Li Cheng-bing, Guo Rui-xue, Li Min. Application of improved ant colony algorithm in travelling salesman problem [J]. Journal of Computer Applications, 2014, 34(S1): 131-132, 165
- [23] Waxman B M. Routing of multipoint connections [J]. IEEE Journal on Selected Areas in Communications, 1988, 6(9): 1617-1622
- [24] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法 [J]. 计算机研究与发展, 1999, 36(10): 1240-1245
Wu Qing-hong, Zhang Ji-hui, Xu Xin-he. An ant colony algorithm with mutation features [J]. Journal of Computer Research and Development, 1999, 36(10): 1240-1245
- [25] 叶志伟, 郑肇葆. 蚁群算法中参数 α, β, ρ 设置的研究 [J]. 武汉大学学报 (信息科学版), 2004, 29(7): 597-601
Ye Zhi-wei, Zheng Zhao-bao. Configuration of parameters α, β, ρ in ant algorithm [J]. Geomatics and Information Science of Wuhan University, 2004, 29(7): 597-601