

分布式网络测量的测量节点自动部署优化算法^①

荣自瞻^{②*} 金跃辉^{*} 崔毅东^{**} 杨 谈^{**}

(^{*} 北京邮电大学网络与交换技术国家重点实验室 北京 100876)

(^{**} 北京邮电大学软件学院 北京 100876)

摘 要 研究了分布式网络测量中的测量节点自动部署,以实现测量节点自动部署的优化。将一种模拟进化算法——蚁群算法应用到分布式网络测量中测量节点的自动部署问题中,形成了解决该问题的基本算法,并通过对该基本算法进行改进和创新,提出了一种基于蚁群算法的测量节点自动部署的优化算法。该算法能够在待测网络拓扑中选择合理的测量节点,通过在所选取的一部分网络节点上部署测量探针进行监测,以获取整个网络的性能状况,并能有效减少网络测量对待测网络带来的带宽占用和软硬件资源消耗。实验验证了该算法的实用性和有效性。

关键词 网络测量, 测量节点, 自动部署, 蚁群算法, 优化算法

0 引 言

网络测量是指对网络相关性能指标进行测量,主要包括对链路流量、端到端带宽、时延和丢包等指标的测量^[1]。分布式网络测量系统 TANC^[2]是一个能够测量某个网络区域内链路性能指标的分布式系统,它分为云监测平台子系统和测量探针子系统。云监测平台子系统采用基于 Hadoop 实现的云计算架构。测量探针子系统则部署到待测网络中的网络节点上进行链路性能指标测量,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。

随着待测网络规模的扩大,分布式网络测量系统必须改进其测量节点的部署方法,以实现较大规模网络的整体性能状况的测量,如果在该待测网络中所有的网络节点部署测量探针进行监测,不但会在该待测网络中产生较高的测量流量,而且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。所以,如果能够通过一种方法,在待测网络中选择出部分节点部署测量探针,并以此测量结果反映出整个待测网络性能状况,则既能减少测量所产生的网络流量,又能有效降低因为执行测

量任务对网络节点带来的资源消耗。因此,设计一种科学的测量节点自动部署算法,通过在所选出的节点上部署测量探针来监测整个网络的性能状况,就变得尤为有价值。近年来,关于测量节点的部署问题已经进行了很多研究。文献[3]提出可以将测量节点的部署问题映射为集合覆盖问题。Chaudet^[4]等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^[5]等提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,并且该问题为 NP-Hard 问题。文献[6]针对无向图中的弱顶点覆盖问题提出了一种基于贪婪策略的近似算法。文献[7]提出了一种利用蚁群算法求解最小弱顶点覆盖问题的算法,表现出了较好的性能,但若将该算法直接应用于分布式网络测量中仍然会存在很多不足。本文将蚁群算法应用于分布式网络测量中的测量节点自动部署中,形成了解决该问题的基本算法,并综合考虑测量节点负载、分布等多种因素,对此基本算法进行了改进和优化,提出了测量节点自动部署的优化算法。实验证明,此优化算法既能满足分布式网络测量中的网络链路的覆盖需求,又能够有效保证所选择节点的质量,有利于获取网络性能状况。

① 863 计划(2011AA01A102)资助项目。
② 男,1990 年生,硕士;研究方向:计算机网络和计算机应用;联系人,E-mail: rongzizhan@gmail.com
(收稿日期:2013-12-04)

1 基于蚁群算法的测量节点自动部署问题

蚁群算法是由 Dorigo 等人于 1991 年提出的一种新型的模拟进化算法,它已被应用到通信网络优化等多个领域,并表现出了良好的性能^[8]。本文将蚁群算法引入到分布式网络测量的测量节点的自动部署中,将其解决最小顶点覆盖问题的算法作为测量节点自动部署的基本算法,并对其进行了优化和改进,提出了一种基于蚁群算法的测量节点自动部署优化算法。

分布式网络测量中测量节点的自动部署需要考虑两个重要因素:一是所选择测量节点的代表性,二是所选择测量节点的合理性。

(1) 所选择节点的代表性

因为我们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^[9],主动测量和被动测量中的测量节点选择问题均可以映射为集合覆盖问题^[3]。考虑到分布式网络测量中主动测量和被动测量的结合,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部署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,本文将在 2.1 节中简要介绍该算法,该算法是目前求解这一问题性能较好的一种算法。本文在分布式网络测量测量节点的自动部署中引入了这一算法。

(2) 所选择节点的合理性

分布式网络测量所选择测量节点的合理性,是指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,因为我们在选择节点时首先需要考虑所选节点的代表性,即需要满足上文所述的覆盖问题。所以我们在选点过程中,并不能简单地直接将整个待测网络中负载最低的点选出,而是要将本因素和前文所述的覆盖问题结合起来。为了实现这样的结合,我们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。

此外,针对网络测量中的实际需求,我们还提出了节点权值控制、局部最优解避免和终端节点清理等步骤,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 2.2 节中进行详细介绍。

2 基于蚁群算法的测量节点自动部署算法

2.1 基于蚁群算法的测量节点自动部署基本算法

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,简称蚁群优化(ant colony optimization, ACO)算法^[7,10],该算法是目前求解该问题的一种表现较为出色的算法,因为本文已将分布式网络测量中测量节点的自动部署问题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部署的基本算法。

ACO 算法的基本思路是将待测网络拓扑图抽象为一个图,并为其建立关联矩阵,从关联矩阵中删除边数最多的顶点 V_i 对应的行以及该行中值为 1 的元素所在列,然后在剩下的关联矩阵中再次删除所有行元素之和为 0 的其他行以及这些行中值为 1 的元素所对应的列,直到不能再删除新的行和列为止。然后重复上述过程。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^[10]。

ACO 算法引入了信息素的概念,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子集(部分可行解)的概率。随机比例状态转移概率的计算按下式进行:

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

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, 但其值的选取需要人为经验的指定。

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 $\tau_{\max}$ 。

顶点的期望启发信息采用动态方式获得, 其表达式为

$$\eta_i = |v_i| \quad (2)$$

当顶点 v_i 被蚂蚁 k 选择之后, 将该顶点加入蚂蚁 k 的部分解集合 S 中, 同时将该顶点和与其相连的所有边都删除, 然后按照上述近似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 $A = 0$, 即所有边都已被部分解集合中的顶点所覆盖, 那么这个集合 S 就是蚂蚁 k 构造的顶点覆盖集。

每次迭代过程完成后, 采用 MMAS 的信息素更新规则更新信息素, 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 ρ ($\rho \in [0, 1]$) 模拟信息素挥发, 定义如下式:

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

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式:

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

其中, WVC_c 是循环最优解; 而 WVC_g 则是全局最优解。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 中的顶点将得到大量的信息素增强。 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 WVC_c 和 WVC_g 中的顶点个数。

虽然上述基本算法具有诸多优点, 比如能够选出最小顶点覆盖集, 可以选出能够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) 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, 不适合选为监测点的情况, 若选点过程中未考虑这一因素, 可能造成选点后执行测量任务时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。

(2) 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。而若正反馈强度太强, 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。

(3) 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优解。然而, 在分布式网络测量中, 终端节点可能占很大数量。对某个待测网络区域的性能衡量, 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。

(4) 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, 会存在根据人为经验, 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。

2.2 基于蚁群算法的测量节点自动部署优化算法

针对上节所提到的基于蚁群算法的测量节点自动部署基本算法的不足, 本节对基本算法进行几点优化和改进, 并将改进后的算法作为自动部署的优化算法。在基本算法的流程上, 本文所做的改进和优化主要包括算法开始前节点权值的初始化、蚁群算法随机比例转移规则的改进、局部最优解避免策略的提出和搜索结果终端节点的清理等。下面依次进行介绍。

2.2.1 节点权值初始化

基本算法初次迭代前, 需要先将各顶点的信息素轨迹强度初始化为 $\tau_{\max}$ 。在分布式网络测量中, 会存在不同测量节点基本条件不同, 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, 为了体现不同节点的重要度差异, 本节所提的改进中加入了节点权值初始化环节, 并将这一初始化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

$$a\tau_{\max} + b\tau_{\max} \cdot p \tag{5}$$

其中, $\tau_{\max}$ 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,均介于 0,1 之间,且 a, b 满足

$$a + b = 1 \tag{6}$$

其中, a, 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的最大影响比例。

2.2.2 改进的随机比例状态转移规则

蚁群算法的搜索过程中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,我们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。为此,我们对随机比例状态转移规则进行了改进,首先定义了负载的计算方法,如下式所示:

$$g = 0.5v_{\text{cpu}} + 0.5v_{\text{memory}} \tag{7}$$

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

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

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。 m 为负载影响因子,由经验指定。 $f(g)$ 为节点负载转换函数,我们将其定义为

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

其中 g 为式(7)的计算结果。当节点负载大于 0.8 时,则 $f(g)$ 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 的值小于 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,我们加入了节点负载这一节点选择过程中的重要考量因素,使得节点的选择更趋于科学合理。

2.2.3 局部最优解避免策略

基本算法在优化的过程中容易陷入停滞。针对基本算法的这一缺点,本文提出了一种有效的局部最优解避免策略,这是保证解质量的重要措施。本文在当前对此问题的常用解决方案的基础上,提出了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,一是在优化过程中加入随机扰动,二是引入收敛调整策略。

(1) 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:

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

其中 λ 为我们定义的扰动因子, λ 的定义如下:

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

其中, k 为我们定义的扰动概率, 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,有效的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。

(2) 收敛调整

为了能进一步进行最优解探索,本文提出的局部最优解避免策略还引入了收敛调整策略。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。我们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。这一调整策略可描述如下:

首先对所有节点的信息素按照从大到小的顺序排序:

$$\tau_{i_1} > \tau_{i_2} > \tau_{i_3} > \cdots > \tau_{i_n} \tag{12}$$

对上式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 $\tau_{i_{\lceil n/2 \rceil + 1}}$ 到 τ_{i_n} 两部分。然后对于按照对应关系进行信息素的调整,调整公式如下:

$$\tau_{i_k} = (0.5 + t)(\tau_{i_k} + \tau_{i_{(\lceil n/2 \rceil + k)}}) \tag{13}$$

$$\tau_{i_{\lceil n/2 \rceil + k}} = (0.5 - t)(\tau_{i_k} + \tau_{i_{(\lceil n/2 \rceil + k)}}) \quad (14)$$

其中 t 为信息素调整系数 ($t \in (0,0.5)$), $k \in [1, \lfloor n/2 \rfloor]$ 。通过这一调整,在解集趋于收敛时,缩小较高信息素和较低信息素之间的差距。完成信息素调整之后,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。

2.2.4 终端节点清理

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,基本算法将无法进行测量的测量节点从选点结果中删除后,为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,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。对于具备这一特征的终端节点,我们将进行终端节点的清理,终端节点清理策略描述如下:

对于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,则在其中通过随机指定的方式,保留一个节点,并删除其余与该节点类型相同的终端节点。

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。

3 实验

我们选择具有 1000 个节点规模的拓扑网络来进行测量节点的选择实验,该网络拓扑由 Waxman 网络模型产生^[11,12],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。

我们将基于蚁群算法的测量节点自动部署优化算法与基本算法进行比较。每种算法运行 10 次。算法中的参数是根据实验和相关文献的研究进行选取^[13,14],由于篇幅所限,本节不再详述。本次实验我们选择的参数为: $\alpha = 1, \beta = 2, \rho = 0.7, k = 0.1, m = 0.3$ 。

从表 1 可以看出,我们所提出的基于蚁群算法的测量节点自动部署优化算法在最优解和平均解的解集规模上均要明显小于基于蚁群算法的测量节点自动部署基本算法,这表明优化算法在解的质量上有明显提高。而优化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,优化算法虽然在解的质量上有所提高,但在性能上有一定下降。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,通过对优化算法和基本算法在解集规模和算法循环次数上的差距进行比较,我们认为优化算法在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公式,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,优化算法在全部实验平均负载上要显著低于基本算法,而优化算法的最小平均负载也要小于基本算法。由此可以看出,在考虑测量节点负载状况的前提下,优化算法选出的节点在一般情况下的合理性要显著优于基本算法。

表 1 基本算法与优化算法的实验结果

	基本算法	优化算法
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

基于蚁群算法的测量节点自动部署优化算法已经应用到 TANC^[2] 系统中,目前在小规模网络中已经顺利使用。虽然 TANC 系统目前尚没有进入大规模部署阶段,但通过上文的实验结果可以得出,未来将本算法应用到 TANC 系统的大规模部署中也是可行和有效的。

4 结 论

分布式网络测量中测量节点的自动部署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自动部署问题中,并将其作为测量节点自动部署的基本算法。在此基础上根据实际测量中的需求对基本算法进行了优化和改进,包括网络节点权值初始化、随机比例状态转移概率中负载因子的加入,避免局部最优解策略的提出和终端节点清理策略等,进而提出了一种基于蚁群算法的测量节点自动部署优化算法。

为验证本文所提的测量节点自动部署优化算法的实用性和有效性,本文进行了实验,将优化算法同基本算法进行了比较。实验证明,本文提出的基于蚁群算法的测量节点自动部署优化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。但是,本算法尚还存在一定的不足,比如因为对基本算法的优化使得算法在性能上有一定程度的下降,如何最大限度的减少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. 西安电子科技大学学报(自然科学版),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[软件著作权]. 北京邮电大学,2013
- [3] 蔡志平,刘芳,赵文涛. 网络测量部署模型及其优化算法. 软件学报,2008,19(2):419-431
- [4] Chaudet C, Fleury E, Guérin Lassous I, et al. Optimal positioning of active and passive monitoring devices. In: Proceedings of CoNEXT, New York, USA, 2005. 71-82
- [5] Suh K, Guo Y, Kurose J, et al. Locating network monitors: Complexity, heuristics and coverage. In: Proceedings of IEEE INFOCOM, Miami, USA, 2005. 351-361
- [6] 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. 软件学报,2004,15(4):545-549
- [7] 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. 计算机应用研究,2009,26(4):1480-1486
- [8] 胡小兵,黄席樾. 蚁群优化算法及其应用. 计算机仿真,2004,21(5):81-85
- [9] 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. 电子学报,2003,31(8):1227-1233
- [10] 彭震宇. 最大独立集和最小弱顶点覆盖问题求解及其应用研究. 无锡:江南大学,2008
- [11] Breitbart Y, Chan C Y, Garofalakis M, et al. Efficiently monitoring bandwidth and latency in IP networks. In: Proceedings of the IEEE INFOCOM, Anchorage, USA, 2001. 933-942
- [12] Waxman B M. Routing of multipoint connections. *IEEE Journal on Selected Areas in communication*, 1988, 6(9):1617-1622
- [13] 吴庆洪,张纪会,徐心和. 具有变异特征的蚁群算法. 计算机研究与发展,1999,36(10):1240-1245
- [14] 叶志伟,郑肇葆. 蚁群算法中参数 α , β , ρ 设置的研究. 武汉大学学报(信息科学版),2004,29(7):597-601

An optimization algorithm for measurement nodes' automatic deployment in distributed network measurement

Rong Zizhan*, Jin Yuehui*, Cui Yidong**, Yang Tan**

(* State Key Laboratory of Networking and Switching Technology,

Beijing University of Posts and Telecommunications, Beijing 100876)

(** School of Software, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract

The automatic deployment of measurement nodes in distributed network measurement was studied to achieve the optimization of measure nodes' automatic deployment. The ant colony algorithm, a simulated evolutionary algorithm, was applied to automatic deployment of measurement nodes to form a basic algorithm for the solution of the deployment, and then, based on some improvements and innovations of the basic algorithm, an optimization algorithm for measurement nodes' automatic deployment in distributed network measurement was put forward. The algorithm selects reasonable measurement nodes in a network to be measured, and monitors the performance indicators of the entire network by deploying measurement probes to a part of nodes in network. It can effectively reduce the bandwidth consumption and resource consumption of hardware or software brought in by network measurement. Its practicability and effectiveness were verified by experiment.

Key words: network measurement, measurement nodes, automatic deployment, ant colony algorithm, optimization algorithm