

## 延迟约束的分布式演化网络监测模型\*

蔡志平<sup>+</sup>, 殷建平, 刘芳, 刘湘辉

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

### Distributed Monitoring Model with Bounded Delay for Evolving Networks

CAI Zhi-Ping<sup>+</sup>, YIN Jian-Ping, LIU Fang, LIU Xiang-Hui

(School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4573659, Fax: +86-731-4575802, E-mail: caizhiping\_nudt@163.com

Cai ZP, Yin JP, Liu F, Liu XH. Distributed monitoring model with bounded delay for evolving networks. *Journal of Software*, 2006,17(1):117-123. <http://www.jos.org.cn/1000-9825/17/117.htm>

**Abstract:** Monitoring infrastructure should be reconfigured at a minimum cost to obtain up-to-date status information as the network evolves. This paper addresses the problem of optimally upgrading the existing monitoring infrastructure. It tries to minimize the total cost of upgrading the monitoring infrastructure, including adding new pollers and reconfiguring the existing pollers. It is shown that this problem is NP-hard. An approximation algorithm is proposed, and its time complexity and approximation ratio are analyzed.

**Key words:** distributed monitoring; evolving network; delay constraint; NP-hard; approximation algorithm

**摘要:** 在扩展网络或网络拓扑发生变化时,需要用最小的代价重新布置网络监测体系,以保证能收集到所有必需的网络信息.更新网络监测体系包括新增和重新配置收集节点两方面的代价,求解总代价最小的更新方案的问题是 NP 难的.提出了一种基于贪婪策略的近似算法,并分析了算法的时间复杂性和近似比.

**关键词:** 分布式监测;演化网络;延迟约束;NP 难;近似算法

中图法分类号: TP393 文献标识码: A

网络监测系统能够帮助网络管理员及时获取网络的状态信息,发现网络的变化情况.许多重要的网络管理任务都需要借助于网络监测系统实时获取网络信息的能力,例如检测链路故障与瓶颈、服务质量保障、流量工程与资源管理等<sup>[1-3]</sup>.

现有的网络监测系统大都基于 SNMP(simple network management protocol)<sup>[1]</sup>.SNMP 由管理站(management station)和管理代理(management Agent)两类实体组成.管理代理分布在路由器、交换机等关键节点上,管理本地的 MIB(management information base),并响应管理站的命令.管理站通过查询各个管理代理的 MIB 来获取整个网络的状态信息.因为管理站集中收集各个管理代理的网络信息,在大型网络中容易导致管理站的性能和带宽过载<sup>[2]</sup>.在收集过程中,网络信息会形成额外的网络流量.当网络监测进程需要更多的网络信息和采用更高的收集频率时,这部分额外的流量对网络的冲击会更加明显,可能会严重影响路由器的吞吐量和带宽利用率.

\* Supported by the National Natural Science Foundation of China under Grant No.60373023 (国家自然科学基金)

Received 2004-10-23; Accepted 2005-03-11

大型网络中的监测系统可以设计成三层结构<sup>[4]</sup>.中心管理站(central manager)负责统计和分析整个网络的状态信息,分散在网络中的各个监控节点(monitring node)完成本地网络信息的监控和管理.在中心管理站与监控节点之间设置一些收集节点(aggregating node)<sup>[4]</sup>,每个收集节点负责查询和收集一部分监控节点的信息,并把这些信息发送到中心管理站.收集节点只负责收集邻近节点的信息,从而可望减少流量和带宽消耗;中心管理站只需要处理收集节点发送过来的信息,从而也可减轻性能负担.当网络发生变化时,只需增加和变更部分收集站就能保持对全网的监测能力,因此三层监控结构具有很好的可扩展性.

在布置收集节点和中心管理站时要考虑带宽、延迟和抖动率的约束,这些重要的网络性能指标也是设计三层结构的主要依据.为了解决收集过程中的带宽限制问题,Li 研究了带宽约束的分布式监测模型,提出了目标为使收集节点尽可能少的一些启发式算法<sup>[5]</sup>.Marina Thottan 研究了演化网络中带宽约束的分布式监测模型,并通过启发式策略得到了一些较好的优化结果<sup>[6]</sup>.除了带宽以外,链路延迟也是影响收集节点分布的一个重要因素.为了收集实时的网络状态信息,收集过程需要一个稳定、可靠的低延迟路由<sup>[7,8]</sup>.为此,我们研究了延迟约束的分布式监测模型<sup>[9]</sup>,并探讨了延迟抖动对分布式监测模型的影响.

在扩展网络或者网络拓扑发生变化时,可能新增一些网络节点和链路,而一部分原有的网络节点和链路可能被拆除,这时就需要调整监测系统的结构,重新布置监控节点和收集节点,以保证监测系统仍然可以覆盖到所有的节点,收集到所有必需的网络信息.对于网络管理者而言,更新网络监测系统的费用和代价是非常值得关注的指标,他们希望用尽量小的代价更新网络监测系统.

为此,本文研究了延迟约束的分布式演化网络监测模型的优化问题,文中第 1 节给出了总代价函数,用整数规划的形式描述了优化问题,并说明了求解延迟约束的分布式演化网络监测模型的优化问题是 NP 难的.第 2 节提出了一种基于贪婪策略的优化算法,证明了该算法的时间复杂性是  $O(n^3)$  而其近似比不超过  $1 + \ln n$ ,其中  $n$  是演化网络中节点的数目.第 3 节是总结并对未来的研究作了展望.

## 1 系统模型

当网络演化之后,一些数据源对应的收集站可能已不存在,同时可能新增了一些节点,对于这部分节点需要指派收集站来负责收集它们的数据.原有收集站负责的数据源可能发生了变动,或者原收集站与数据源之间可能不再满足延迟约束条件的要求,这就需要变更原有收集站的配置.延迟约束的分布式演化网络监测模型的优化问题就是要用尽量小的代价来完成这些系统更新.

下面用两个无向图  $G_1 = (V_1, E_1)$  和  $G_2 = (V_2, E_2)$  分别表示演化前后的网络拓扑.点集  $V_1, V_2$  和边集  $E_1, E_2$  分别表示两个网络中的节点和链路.

### 1.1 代价函数

重新布置演化网络的监测系统主要包括两种代价:新增收集站和变更原有收集站.新增收集站的代价包括购买附加的软硬件和对收集站进行配置,而变更原有收集站显然也需要耗费一定的人力物力.我们研究的目标是得到折衷这两种代价的最优解,让网络管理员耗费尽量小的代价重新布置演化网络的监测系统,使更新后的监测系统能够收集到所有必需的网络信息.

如果变更原有收集站和新增收集站的代价分别为  $C_r$  和  $C_a$ ,可用参数  $\gamma = C_r/C_a$  来代表两者之间的比例, $\gamma$  的取值范围为  $[0,1]$  区间.因为可以根据实际的网络状况、软硬件价格和维护成本来确定和调整  $\gamma$ ,所以在模型中无须考虑  $C_r$  和  $C_a$  的具体数值,优化解的结果也只与  $\gamma$  相关.

网络中的每个数据源(polllee)的网络信息都会由一个指定的收集站(poller)来收集,用  $o_{ij}$  和  $x_{ij}$  分别表示  $G_1$  和  $G_2$  中的这种指派关系. $o_{ij} = 1$  表示在网络  $G_1$  中节点  $v_i$  作为收集站负责收集数据源节点  $v_j$  的网络信息,否则  $o_{ij} = 0$ .同样, $x_{ij} = 1$  表示在网络  $G_2$  中节点  $v_i$  作为收集站负责收集数据源节点  $v_j$  的网络信息,否则  $x_{ij} = 0$ .

定义指标函数  $I(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \end{cases}$  和指示变量  $p_i$  和  $y_i$ .如果节点  $v_i$  在  $G_1$  中是收集站,则  $p_i = 1$ ,否则  $p_i = 0$ .同

样,如果节点  $v_i$  在  $G_2$  中是收集站,则  $y_i = 1$ , 否则  $y_i = 0$ . 重新布置演化网络的监测系统的总代价函数定义如下:

$$C(\gamma, G_1, G_2) = \sum_{\forall v_i \in V_2} y_i(1 - p_i) + \gamma \sum_{v_i \in V_1 \cap V_2} p_i I \left( \sum_{v_j \in V_1 \cup V_2} |o_{ij} - x_{ij}| \right) \quad (1)$$

这里假定变更原有收集站的代价是一定的,与其负责收集的数据源的数量无关.

## 1.2 整数规划

我们假定网络中的链路延迟是对称的,可叠加的.用函数  $D(e)$  代表边  $e$  的延迟,其中包括排队延迟、传输延迟和传播延迟.若边集  $E(\text{Path}(u, v))$  代表从节点  $u$  到节点  $v$  的最短路径上所有边的集合,则节点  $u$  与  $v$  之间的延迟定义为  $\text{Delay}(\text{Path}(u, v)) = \sum_{e \in E(\text{Path}(u, v))} D(e)$ .

延迟约束的分布式演化网络监测模型的优化问题可以表述为:已知原网络  $G_1$  中的收集站和数据源的指派关系  $O$ ,要求确定演化网络  $G_2$  中的指派关系  $X$ ,并使其满足以下条件:

- 从  $O$  更新到  $X$  的总代价最小;
- 指派关系  $X$  能使  $G_2$  中所有的数据源都被收集;
- 任意数据源到所属收集站之间的延迟不能超过延迟约束  $\delta$ .

可用整数规划形式来表述以上优化问题.优化的目标是

$$\text{Minimize } C(\gamma, G_1, G_2) \quad (2)$$

并满足如下限制条件:

$$\sum_{i=1}^{|V_2|} x_{ij} = 1 (\forall v_j \in V_2) \quad (3)$$

$$x_{ij} \leq y_i (\forall v_i, v_j \in V_2) \quad (4)$$

$$\text{Delay}(\text{Path}(v_i, v_j)) x_{ij} \leq \delta (\forall v_i, v_j \in V_2) \quad (5)$$

$$x_{ij} \in \{0, 1\} (\forall v_i, v_j \in V_2) \quad (6)$$

$$y_i \in \{0, 1\} (\forall v_i \in V_2) \quad (7)$$

第 1 个约束条件保证了每个数据源有且仅有一个收集站负责收集它的信息;第 2 个约束条件保证了数据源的信息一定是由收集站负责收集;第 3 个条件保证了收集站与数据源之间满足延迟约束的限定条件.

## 1.3 复杂性分析

当  $G_1$  为空时,系统更新的代价就等于新增收集站的代价.选取的收集站的数量越小,系统更新的代价也就越小.因此在  $G_1$  为空的情况下,延迟约束的分布式演化网络监测模型的优化问题就转化为求延迟约束的分布式网络监测模型的优化解的问题,我们在以前的研究<sup>[9]</sup>中已经证明了延迟约束的分布式网络监测模型的优化问题是 NP 难的,所以延迟约束的分布式演化网络监测模型的优化问题也是 NP 难的.

## 2 近似算法

既然延迟约束的分布式演化网络监测模型的优化问题是 NP 难的,这就意味着至今尚未找到能在多项式时间内求得一个最优解的算法.下面我们设计了一个基于贪婪策略的启发式算法来求其近似最优解,并且分析了算法的时间复杂性和近似比.

### 2.1 贪婪算法

采用 Floyd 和 Warshall 提出的多源最短路径算法<sup>[10]</sup>可以在  $O(|V_2|^3)$  的时间内求得  $G_2$  中任意两点  $u$  与  $v$  之间的延迟  $\text{Delay}(\text{Path}(u, v))$ . 给定延迟阈值  $\delta$ , 定义节点  $u$  的延迟约束覆盖集  $C(u) = \{v | \text{Delay}(\text{Path}(u, v)) \leq \delta\}$ , 即  $C(u)$  表示与节点  $u$  之间满足延迟约束条件的节点的集合.显然,在已知  $G_2$  中任意两点间延迟的情形下,在  $O(|V_2|^2)$  的时间内可以求得  $G_2$  中每个节点  $u$  的  $C(u)$ , 记为  $C_2(u)$ .

因为变更原有收集站的代价要小于新增收集站的代价,所以我们首先分析原有收集站的情况:(1) 收集站负责的数据源都保留在  $G_2$  中,并且它们与收集站之间仍满足延迟约束条件,我们将这一类数据源未发生变化的

收集站放入集合  $F$  中,  $F = \{v_i | v_i \in V_1 \cap V_2 \wedge p_i = 1 \wedge (\forall v_j \in V_1, o_{ij} = 1 \Rightarrow v_j \in V_2 \wedge v_j \in C_2(v_i))\}$ ; 而将  $v_i \in F$  原来负责的数据源放入集合  $M_i$  中,  $M_i = \{v_j | v_j \in V_1 \cap V_2 \wedge v_i \in F \wedge o_{ij} = 1\}$ ; (2) 收集站负责的数据源不在  $G_2$  中或者它们与收集站之间不满足延迟约束条件. 我们将这类数据源发生了变化的收集站放入集合  $R$  中,  $R = \{v_i | v_i \in V_1 \cap V_2 \wedge p_i = 1 \wedge (\exists v_j \in V_1 - V_2, o_{ij} = 1) \vee (\exists v_j \in V_1 \cap V_2, o_{ij} = 1 \wedge v_j \notin C_2(v_i))\}$ .

算法首先保留  $F$  中的节点作为收集站, 并维持其原有的指派关系. 因为  $R$  中的节点肯定需要重新配置, 所以贪婪算法再优先选取  $R$  中的节点作为收集站并使其覆盖尽可能多的未覆盖节点. 为了覆盖剩下的节点, 算法以尽可能小的代价来覆盖尽可能多的未覆盖节点作为贪婪策略, 循环选取收集站, 直到所有的节点都能被覆盖.

整个算法分为两步:

第 1 步进行预处理, 分析  $V_1 \cap V_2$  中节点的状况, 生成集合  $F, R$  和  $M_i$ ;

第 2 步采用贪婪策略选取收集站.

预处理算法用伪代码形式表示如下:

输入: 原网络  $G_1$ , 演化网络  $G_2$  和原网络  $G_1$  中的指派关系  $O$ .

输出: 节点集合  $F, R$  和  $M_i$ .

Pre-processing ()

- (1)  $F = \emptyset, R = \emptyset;$  /\* 各集合初始化为空 \*/
- (2) for  $\forall v_i \in \{v_i | v_i \in V_1 \cap V_2 \wedge p_i = 1\}$
- (3) if  $\forall v_j \in \{v_j | v_j \in V_1 \wedge o_{ij} = 1\} \Rightarrow v_j \in V_2 \wedge v_j \in C_2(v_i)$  /\*  $v_i$  的数据源都在  $V_2$  中且满足延迟约束条件 \*/
- (4) then  $F = F \cup \{v_i\}, M_i = \{v_j | v_j \in V_1 \cap V_2 \wedge o_{ij} = 1\};$  /\* 数据源未发生变化的收集站  $v_i$  放入  $F$ , 对应的数据源放入  $M_i$  \*/
- (5) else  $R = R \cup \{v_i\};$  /\* 数据源发生了变化的收集站  $v_i$  放入  $R$  \*/

贪婪算法首先保留原有的收集站, 对  $F$  中的收集站保留原有的配置, 对  $R$  中的收集站则使之覆盖尽可能多的节点. 算法把  $F$  中的节点赋权值  $\gamma$  以示若再次选中则要更新其配置, 而把其他节点赋权值 1 以示若选中则要新增收集站. 值得注意的是, 虽然  $R$  中的节点所赋权值也为 1, 但是由算法可知,  $R$  中的节点不会再被选中. 算法中优先选取新覆盖节点数与权值之比最大的节点作为收集站. 贪婪算法用伪代码形式表示如下:

输入: 原网络  $G_1$ , 演化网络  $G_2$ , 原网络  $G_1$  中的指派关系  $O$ , 子集  $F, R, M_i$  和延迟覆盖集  $C_2(v_i)$ .

输出: 演化网络  $G_2$  中的指派关系  $X$  和选取的收集站  $P$ .

PickPollerSet ()

- (1) for  $\forall v_i, v_j \in V_2$
- (2)  $x_{ij} = 0;$  /\* 指派关系初始化 \*/
- (3)  $\omega_i = 1;$  /\* 节点权值初始化 \*/
- (4)  $P = \emptyset;$  /\* 用  $P$  记录已选取的收集站 \*/
- (5)  $S = V_2;$  /\* 用  $S$  记录暂未覆盖的节点 \*/
- (6) for  $\forall v_i \in F$
- (7)  $P = P \cup \{v_i\};$  /\* 选取  $F$  中的节点作为收集站 \*/
- (8)  $\omega_i = \gamma;$  /\* 若再次选中  $v_i$  则要更新其配置, 这时的代价为  $\gamma$  \*/
- (9) for  $\forall v_j \in M_i$
- (10)  $x_{ij} = 1;$  /\* 保留原有的指派关系 \*/
- (11)  $S = S - M_i;$  /\* 删去已覆盖的节点 \*/
- (12) for  $\forall v_i \in R$
- (13)  $P = P \cup \{v_i\};$  /\* 选取  $R$  中的节点作为收集站 \*/
- (14) for  $\forall v_j \in C_2(v_i) \cap S$
- (15)  $x_{ij} = 1;$  /\* 使  $v_i$  覆盖尽可能多的暂未覆盖的节点 \*/

- (16)  $S = S - C_2(v_i) \cap S$ ; /\* 删去已覆盖的节点 \*/  
 (17) while ( $|S| \neq 0$ )  
 (18) Pick  $v_i \in V_2$  使  $v_i = \arg \left( \max_{v_i \in V_2} \left( |C_2(v_i) \cap S| / \omega_i \right) \right)$ ; /\* 采用贪婪策略选取收集站 \*/  
 (19)  $P = P \cup \{v_i\}$ ; /\* 记录新选中的收集站 \*/  
 (20) for  $\forall v_j \in C_2(v_i) \cap S$   
 (21)  $x_{ij} = 1$ ; /\* 使  $v_i$  覆盖尽可能多的暂未覆盖的节点 \*/  
 (22)  $S = S - C_2(v_i) \cap S$ ; /\* 删去已覆盖的节点 \*/

### 2.2 贪婪算法的时间复杂性

关于上述贪婪算法的时间复杂性有以下结果.

**定理 1.** 在输入为  $G_1 = (V_1, E_1)$  和  $G_2 = (V_2, E_2)$  时,算法 Pre-Processing 存在一种时间复杂性为  $O(|V_1 \cap V_2| |V_1 \cup V_2|)$  的实现.算法 PickPollerSet 存在一种时间复杂性为  $O(|V_2|^3)$  的实现.

**证明:**假设  $V_2 - V_1$  中的节点为  $v_1, v_2, \dots, v_D$ ,  $V_2 \cap V_1$  中的节点为  $v_{D+1}, v_{D+2}, \dots, v_{D+E}$ ,  $V_1 - V_2$  中的节点为  $v_{D+E+1}, v_{D+E+2}, \dots, v_{D+E+F}$ . 那么我们采用数组  $p[1:D+E+F]$  中的元素  $p[i]$  来记录  $p_i$  的值,采用数组  $w[1:D+E]$  中的元素  $w[i]$  来记录  $w_i$  的值,采用矩阵  $a[D+1:D+E+F; D+1:D+E+F]$  中的元素  $a[i, j]$  来记录  $a_{ij}$  的值,采用矩阵  $x[1:D+E; 1:D+E]$  中的元素  $x[i, j]$  来记录  $x_{ij}$  的值.对  $D+1 \leq i \leq D+E$ ,我们引进布尔数组  $M_i[1:D+E]$  使得  $M_i[j]$  为真当且仅当  $v_j \in M_i$ .对  $1 \leq i \leq D+E$ ,我们引进布尔数组  $C_2^i[1:D+E]$  使得  $C_2^i[j]$  为真当且仅当  $v_j \in C_2(v_i)$ .最后引进布尔数组  $S[1:D+E]$  使得  $S[i]$  为真当且仅当  $v_i \in S$ ,并用链表来实现集合  $F, R$  和  $P$ .

这时,算法 Pre-Processing 的外层循环次数为  $O(|V_1 \cap V_2|)$ ,每次循环中计算  $M_i$  需要  $O(|V_2|)$  的时间,内层循环次数为  $O(|V_1|)$ ,其他语句每次执行均只需常数时间,所以算法 Pre-Processing 的时间复杂性为  $O(|V_1 \cap V_2| |V_1 \cup V_2|)$ .

对算法 PickPollerSet,语句(1)的循环次数为  $O(|V_2|^2)$ ,语句(2)~(4)执行一次需常数时间,语句(5)执行一次需  $O(|V_2|)$  时间,语句(6)的循环次数不超过  $O(|V_2|)$ ,语句(7)和(8)执行一次需常数时间,语句(9)的循环次数不超过  $O(|V_2|)$ ,语句(10)执行一次需常数时间,语句(11)执行一次需  $O(|V_2|)$  时间,语句(12)的循环次数不超过  $O(|V_2|)$ ,语句(13)执行一次需常数时间,  $C_2(v_i) \cap S$  的计算需  $O(|V_2|)$  时间,语句(14)的循环次数不超过  $O(|V_2|)$ ,语句(15)执行一次需常数时间,语句(16)执行一次需  $O(|V_2|)$  时间,  $|S|$  的计算需  $O(|V_2|)$  时间,语句(17)的循环次数不超过  $O(|V_2|)$ ,语句(18)执行一次需  $O(|V_2|^2)$  时间,语句(19)执行一次需常数时间,语句(20)的循环次数不超过  $O(|V_2|)$ ,语句(21)执行一次需常数时间,语句(22)执行一次需  $O(|V_2|)$  时间,所以算法 PickPollerSet 的时间复杂性为  $O(|V_2|^3)$ .

### 2.3 贪婪算法的近似比

不妨假设算法 PickPollerSet 执行完语句(16)时未覆盖节点的集合为  $U$ ,显然  $U = V_2 - \bigcup_{v_i \in F} M_i - \bigcup_{v_i \in R} C_2(v_i)$ .定义贪婪算法覆盖集合  $U$  的代价为  $Cost_g(U)$ ,因为覆盖  $\bigcup_{v_i \in F} M_i$  和  $\bigcup_{v_i \in R} C_2(v_i)$  的代价分别为 0 和  $\gamma |R|$ ,所以采用贪婪算法的总代价为  $Cost_g(U) + \gamma |R|$ .假设原问题最优解的总代价为  $Cost_{opt}(V_2)$ ,因为其中同样包含了变更  $R$  中收集站的代价  $\gamma |R|$ ,所以我们定义  $C_{opt} = Cost_{opt}(V_2) - \gamma |R|$ .

下面分析  $Cost_g(U)$  与  $C_{opt}$  的关系.不妨假设算法 PickPollerSet 中语句(17)~(22)在循环  $l$  次后结束,第  $i$  次循环开始前暂未覆盖的节点集合为  $S_i$ .假设  $U$  中的节点数量为  $m$ ,在算法中被收集站覆盖的次序为  $v_1, v_2, \dots, v_m$ .如果节点  $v_k$  在算法执行第  $i$  次循环时被覆盖,那么可以将选中的收集站的权值与新覆盖的节点数之比定义为该节点的代价  $p(v_k)$ ,即  $p(v_k) = \omega_{N(i)} / (|S_i| - |S_{i+1}|)$ ,其中  $N(i)$  是第  $i$  次循环中语句(18)选取的节点的编号.这时  $U$  中所有节点的代价之和就等于算法覆盖  $U$  所需的代价,即  $\sum_{v_k \in U} p(v_k) = Cost_g(U)$ .根据贪婪算法选取收集站的

策略,  $S_{i+1} = S_i - C_2(v_{N(i)}) \cap S_i$ ,  $|S_{i+1}| < |S_i|$  以及  $p(v_k)$  的定义, 可以知道序列  $p(v_1), p(v_2), \dots, p(v_m)$  是不减的.

引理 1. 对于  $U$  中每个被覆盖的节点  $v_k$ , 都有  $p(v_k) \leq \frac{C_{opt}}{m-k+1}$  成立.

证明: 假设节点  $v_k$  在算法执行第  $i$  次循环时被覆盖, 那么  $|S_i| \geq m-k+1$ . 假设从原问题最优解中去掉集合  $R$  后剩余的集合为  $opt_R$ , 并且从第  $i$  次循环开始将算法 PickPollerSet 中的贪婪策略改为从  $opt_R$  中选取收集站, 即将语句(18)中的  $v_i \in V_2$  改为  $v_i \in opt_R$ , 同样定义每个被覆盖的节点的代价为收集站的权值与新覆盖的节点数之比, 记为  $p'(v_k)$ . 若这样选出的收集站依次为  $v_{opt_1}, v_{opt_2}, \dots, v_{opt_r}$ , 则  $\sum_{v_k \in S_i} p'(v_k) = \sum_{j=1}^r \omega_{opt_j} \leq C_{opt}$ . 因为序列  $p'(v_{opt_1}), p'(v_{opt_2}), \dots, p'(v_{opt_r})$  同样是不减的, 所以该序列的第 1 个元素满足  $p'(v_{opt_1}) \leq C_{opt} / |S_i| \leq C_{opt} / (m-k+1)$ . 而  $p'(v_{opt_1}) = \omega_{opt_1} / |C_2(v_{opt_1}) \cap S_i|$ . 对原算法 PickPollerSet, 假设第  $i$  次循环时是因为选取节点  $v_{k'}$  才导致覆盖节点  $v_k$ , 那么由  $p(v_k)$  的定义知道  $opt_R \subseteq V_2$ , 根据原算法的贪婪策略有:  $p(v_k) = p(v_{k'}) = \frac{\omega_{k'}}{|C_2(v_{k'}) \cap S_i|} \leq$

$$\frac{\omega_{opt_1}}{|C_2(v_{opt_1}) \cap S_i|} = p'(v_{opt_1}) \leq \frac{C_{opt}}{m-k+1}.$$

引理 2.  $Cost_g(U) \leq H(n)C_{opt}$ , 其中  $H(x) = \sum_{i=1}^x \frac{1}{i}$ ,  $n$  为演化网络  $G_2$  中节点的个数.

证明: 因为  $m \leq n$ , 由引理 1 可知:

$$Cost_g(U) = \sum_{i=1}^m p(v_i) \leq \sum_{i=1}^m \frac{C_{opt}}{m-i+1} = H(m)C_{opt} \leq H(n)C_{opt}.$$

因为  $H(n) \leq \int_1^n \frac{1}{x} dx + 1 = 1 + \ln n$ , 而且  $Cost_g(V_2) = Cost_g(U) + \gamma |R|$  和  $Cost_{opt}(V_2) = C_{opt} + \gamma |R|$ , 所以由引理 2 可以得到如下定理.

定理 2. 如果  $Cost_g(V_2)$  是采用贪婪算法得到的解的总代价,  $Cost_{opt}(V_2)$  是最优解的总代价, 那么有  $Cost_g(V_2) \leq (1 + \ln n)Cost_{opt}(V_2)$ , 即贪婪算法的近似比为  $1 + \ln n$ .

### 3 结论

我们用整数规划的形式描述了延迟约束的分布式演化网络监测模型的优化问题, 并说明了这个问题是 NP 难的, 然后基于贪婪策略给出了一种近似算法, 并分析了算法的时间复杂性和近似比.

因为收集站存在维护和管理的代价, 显然, 设立的收集站的数量越少越好. 本文研究的模型以系统更新的总代价最小作为唯一的优化目标, 在下一步的研究中会考虑多种优化目标的情况<sup>[11-13]</sup>.

致谢 审稿人对原稿提出了中肯的意见, 在此向他们表示感谢.

### References:

- [1] Stallings W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. 3rd ed., Boston: Addison-Wesley Professional, 1998.
- [2] Asgari A, Trimintzios P, Irons M, Pavlou G, Egan R, den Berghe SV. A scalable real-time monitoring system for supporting traffic engineering. In: Proc. of the IEEE Workshop on IP Operations and Management. Dallas: IEEE, 2002. 202-297. <http://www.ist-tequila.org/publications/ipom02-monitoring.pdf>
- [3] Awduche DO. MPLS and traffic engineering in IP networks. IEEE Communications Magazine, 1999, 37(12):42-47.
- [4] Lin YJ, Chan MC. A scalable monitoring approach based on aggregation and refinement. IEEE Journal on Selected Areas in Communications, 2002, 20(4):677-690.
- [5] Li L, Thottan M, Yao B, Paul S. Distributed network monitoring with bounded link utilization in IP networks. In: Proc. of the IEEE INFOCOM. San Francisco: IEEE, 2003. 1189-1198. [http://www.ieee-infocom.org/2003/papers/29\\_03.PDF](http://www.ieee-infocom.org/2003/papers/29_03.PDF)
- [6] Thottan M, Li L, Yao B, Mirrokni VS, Paul S. Distributed network monitoring for evolving IP networks. In: Proc. of the 24th Int'l Conf. on Distributed Computing Systems. Hachioji: IEEE, 2004. 712-719. [http://www.ieee-infocom.org/2003/papers/29\\_03.PDF](http://www.ieee-infocom.org/2003/papers/29_03.PDF)

- [7] Breitbart Y, Chan CY, Garofalakis M, Rastogi R, Silberschatz A. Efficiently monitoring bandwidth and latency in IP networks. In: Proc. of the IEEE INFOCOM. Anchorage: IEEE, 2001. 933–942. <http://http://csdl.computer.org/comp/proceedings/icdcs/2004/2086/00/20860712abs.htm>
- [8] Breitgand D, Raz D, Shavitt Y. SNMP GetPrev: An efficient way to browse large MIB tables. IEEE Journal of Selected Areas in Communication, 2002,20(4):656–667.
- [9] Liu XH, Yin JP, Lu XC, Cai ZP, Zhao JM. Distributed network monitoring model with bounded delay constraints. Wuhan University Journal of Natural Sciences, 2004,9(4):429–434.
- [10] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. 2nd ed., Cambridge: MIT Press, 2001.
- [11] Liu XH, Yin JP, Tang LL, Zhao JM. Analysis of efficient monitoring method for the network flow. Journal of Software, 2003,14(2):300–304 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/300.htm>
- [12] Liu XH, Yin JP, Lu XC, Zhao JM. A monitoring model for link bandwidth usage of network based on weak vertex cover. Journal of Software, 2004,15(4):545–549 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/545.htm>
- [13] Cai ZP, Zhao WT, Yin JP, Liu XH. Using passive measuring to calibrate active measuring latency. In: Kim C, ed. ICOIN 2005. LNCS 3391, Berlin: Springer-Verlag, 2005. 198–206.

#### 附中文参考文献:

- [11] 刘湘辉,殷建平,唐乐乐,赵建民.网络流量的有效测量方法分析.软件学报,2003,14(2):300–304. <http://www.jos.org.cn/1000-9825/14/300.htm>
- [12] 刘湘辉,殷建平,卢锡城,赵建民.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. <http://www.jos.org.cn/1000-9825/15/545.htm>



蔡志平(1975 - ),男,湖南益阳人,博士生,主要研究领域为计算机网络测量,近似算法.



刘芳(1976 - ),女,博士,主要研究领域为信息安全,计算机体系结构.



殷建平(1963 - ),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为算法设计与分析,人工智能,模式识别,信息安全.



刘湘辉(1973 - ),男,博士,主要研究领域为计算机网络 QoS,网络安全.