第29卷 第8期 控 制 与 决 策 2014年 8月
Vol. 29 No. 8 Control and Decision Aug. 2014

文章编号: 1001-0920 (2014) 08-1513-04 DOI: 10.13195/j.kzyjc.2013.0083

网络监测节点序列部署问题的占线竞争算法设计

代文强[1]，冯 博[2]

(1. 电子科技大学 经济与管理学院，成都 610054；2. 华南理工大学 工商管理学院，广州 510640)

摘 要: 万维网的高速发展需要在网络内部构建部署相应的网络监测系统, 但由于耗资巨大, 在设计网络监测系统时, 网络节点部署初期往往不能一次性监测完所有的边, 只能选择有限的网络节点以监测少部分的边, 再逐渐增加部署新的网络监测节点. 在占线理论与竞争策略的基础上, 研究网络监测系统网络节点序列占线优化部署问题, 给出一个竞争算法, 证明了该算法具有常数竞争比, 该竞争比结果优于已有的结果.

关键词: 网络监测系统；顶点覆盖问题；占线；竞争算法

中图分类号: TP273 **文献标志码:** A

An online competitive algorithm for network monitor nodes sequence deployment problem

DAI Wen-qiang[1]，*FENG Bo*[2]

(1. School of Management and Economics，University of Electronic Science and Technology of China，Chengdu 610054, China；2. School of Business Administration，South China University of Technology，Guangzhou 510640，China. Correspondent：DAI Wen-qiang，E-mail：wqdai@uestc.edu.cn)

Abstract: With the rapid development of the world wide web, the network monitoring systems need to be built within the network, but due to huge cost, when the network monitoring system is designed, all of the edges can not be monitored at one time. Instead only a limited number of network nodes can be chosen to monitor a small part of the edge, and later the deployment of a new network monitoring node is increased. Based on the online theory, the online vertex covering problem is studied. A competitive algorithm is presented with a constant competitive ratio. The performance of the competitive ratio is better than the existed result.

Key words: network monitoring system；vertex covering problem；online；competitive algorithm

0 引 言

随着万维网的快速发展和网络用户的快速增长, 网络流量逐渐增大. 为了向客户提供更好的服务质量, 网络服务提供商需要不断地建设和升级网络. 与此同时, 提供商也需要在网络内部构建一个面向全局的网络监测系统, 该监测系统可以用来测量如链路流量等性能指标, 实时地获取网络的性能数据, 进而加强全局网络性能. 考虑网络监测系统部署的最基本问题: 如果在某个顶点上部署了监测器, 并且认为它可以监测 (或服务) 到与其相连接的边, 则称该边被该顶点所覆盖 (Covering). 同时, 假设各顶点具有一定的非负权重 (例如可以表示为部署该监测点的初始建设费用等), 如果问题是如何以最小的代价 (即选取一个最小权重和的顶点子集) 来监测服务网络上所有的边, 则这个问题为著名的顶点覆盖问题 (Vertex Covering Problem), 目前已经证明该问题是NP-hard的[1-2].

目前, 测量Internet甚至仅测量ISP主干网络的网络性能数据都需要花费相当大的代价. 在全局各个节点部署测量设备需要消耗相当多的人力和物力, 而且安装在网络设备上的测量设备和工具也会占用物理空间和系统资源. 因此, 在设计全局网络监测系统时, 既需要监测尽可能多的网络信息, 又需要尽量减少系统的部署代价和维护代价, 并且监测设备一旦建好, 最好不被删去或移动. 由于资金限制等因素的存在, 实际监测节点部署决策过程本质上是一个长期的建设部署过程. 在初始阶段不能一次监测完所有的边,

收稿日期: 2013-01-07；**修回日期:** 2013-04-22.

基金项目: 国家自然科学基金项目(70901012)；国家自然科学基金重大项目(71090403, 71090400)；广东省普通高校人文社会科学重点研究基地重大项目(11JDXM63003).

作者简介: 代文强(1978−), 男, 副教授, 博士, 从事网络优化、组合优化与决策等研究；冯博(1981−), 女, 教授, 博士生导师, 从事运作管理与决策分析等研究.

只能选择有限的监测节点监测少部分的边, 随着限制因素的放松再逐渐增加部署新的监测节点. 此外, 还要求已经部署好的网络监测节点不能被删除. 针对上述问题, 可以考虑用占线算法和竞争策略[3-5]的相关理论来建立模型并进行研究. 利用该理论建立的模型输出是一个待部署的网络监测节点序列, 算法需要保证该序列的每一个 k 元前序 (要求至少覆盖 k 条边) 的费用, 都与相应的只需覆盖至少 k 条边的最优解费用非常接近. 对于每一个 k, 这 2 个数值间的比值就衡量了该 k 元前序的优劣, 因此将该序列的竞争比定义为该序列中所有前序所计算的比值中的最大者. 这样竞争比便可度量序列间的优劣, 显然竞争比越小越好.

上述利用占线理论来研究分阶段建设网络监测节点部署序列的模型被称为占线顶点覆盖问题, 该模型是网络监测系统部署问题的一个基本问题. 对于该问题, 已证明除非 $P = \mathrm{NP}$, 否则该竞争比小于 1.360 7 是不可能的, 并且除非唯一博弈假设成立, 否则该问题不存在竞争比小于 2 的多项式时间竞争算法[4]. 文献[4]进一步给出了结构性下界 1.618. 文献[6]给出了竞争比为 $4c$ 的竞争算法, 其中 c 为 k-顶点覆盖问题的近似比. 本文将对这一问题进行研究, 给出一个常数竞争比的竞争算法, 该竞争比结果将优于以往的结果.

1 占线顶点覆盖问题的定义

本文所有的讨论均基于如下基本定义: 给定无向图 $G = (V, E)$ 为拟部署网络监测节点的拓扑网络. 其中: V 为顶点的集合, $n = |V|$; E 为边的集合. 为了去除平凡的情况, 该图不存在孤立顶点, 且在顶点 V 上存在一个正的权重函数 $w : V \to R^+$. 单位权重表示 $w \equiv 1$, 顶点 v 的度表示与该顶点相关联的边数, 记为 $\deg(v)$.

定义 1 给定顶点集 $S \subseteq V$, 如果 $\forall e \in E$, 至少有一个端点在 S 中, 则称 S 是一个顶点覆盖.

定义 2 顶点集 $S \subseteq V$ 的权重定义为该子集中所有元素的权重之和, 即 $w(S) = \sum\limits_{x\in S} w(x)$.

顶点覆盖问题的目标是: 寻求一个顶点集 V 的子集 $S \subseteq V$, 使得它在能覆盖所有边的全部顶点子集合中, $w(S)$ 值最小[1]. 相应地, 定义 k-顶点覆盖问题[2](也称为部分顶点覆盖问题)的目标是: 对于给定的需要至少覆盖的边的个数 k, $1 \leqslant k \leqslant |E|$, 寻求一个顶点集 V 的子集 $S \subseteq V$, 使得它在能至少覆盖 k 条边的所有顶点子集合中, $w(S)$ 值最小. k-顶点覆盖问题是标准顶点覆盖问题的推广, 因此也是 NP-hard 的. 目前, 针对该问题的最优的近似算法所达到的近似比为 2[2].

占线顶点覆盖问题的数学描述如下[4-6]: 寻求子集序列 $F_i(i = 1, 2, \cdots, |E|)$, 使得 F_i 为 i-顶点覆盖问题的一个可行解, 即其至少覆盖了 i 条边. 其中: $F_1 \subseteq F_2 \subseteq \cdots \subseteq F_{|E|} \subseteq V$, $w(F_i) \leqslant \alpha \times \mathrm{opt}_i$, opt_i 为 i-顶点覆盖问题的最优解费用, α 为与问题无关的常数, 称为竞争比, 且 α 越小越好. 该竞争比表明了在最坏的情况下, 所给出的网络监测节点部署序列的每个阶段的费用, 都在相应阶段最优解费用的竞争比倍数以内, 因此可以保证给出的节点部署序列在每个阶段都非常接近最优解.

对于这个问题, 最简单的想法是利用贪婪的思想: 1) 选取当前图中具有最小势(如权重与对应顶点的度的比值) 的顶点; 2) 删去与该顶点关联的边, 在剩下的图中再选取具有最小势的顶点; 3) 重复步骤1)和步骤2). 但即使是针对单位权重的顶点覆盖问题, 也存在反例表明上述贪婪算法不能得到常数近似比[7], 故贪婪算法针对占线顶点覆盖问题不能得到常数竞争比. 同时, 贪婪算法的反例也表明, 在优化决策各阶段部署网络监测节点位置的方案时, 应综合考虑到未来可能出现的各种情况.

2 竞争算法设计与分析

2.1 定义及性质

定义 3 对于任意给定的顶点覆盖问题输入, 定义

$$\varDelta = \frac{\max\limits_{x\in V} w(x)}{\min\limits_{x\in V} w(x)} > 0$$

为输入空间中的最大相对权重, 则 $\varDelta$ 在输入空间给定后为常数.

引理 1 $\mathrm{opt}_{|E|} \leqslant \left\lceil \dfrac{n}{2} \right\rceil \varDelta \mathrm{opt}_1$, 其中: $n = |V|$, 函数 $\lceil x \rceil$ 表示对数值 x 上取整.

证明 设给定顶点集合 $V = \{v_1, v_2, \cdots, v_n\}$, 且 $\{v_1\}$ 可使 1-顶点覆盖问题达到最优. 由于假设给定的图不存在孤立的顶点, v_1 必定是所有顶点中权重最小的顶点, 即 $\mathrm{opt}_1 = w(v_1) = \min\limits_{x\in V} w(x)$. 此外, V 中必存在 $\left\lceil \dfrac{n}{2} \right\rceil$ 个顶点覆盖了所有的边, 记该顶点集合为 S. 故有

$$\mathrm{opt}_{|E|} \leqslant w(S) = \sum_{x\in S} w(x) \leqslant \left\lceil \frac{n}{2} \right\rceil \varDelta \min_{x\in V} w(x) = \left\lceil \frac{n}{2} \right\rceil \varDelta \mathrm{opt}_1. \qquad \square$$

2.2 竞争算法设计及时间复杂度分析

本文设计的竞争算法如下.

输入: 无向图 $G = (V, E)$, 函数 $w : V \to R^+$;

输出: 顶点子集序列 $F_i(i = 1, 2, \cdots, |E|)$, 使 F_i 至少覆盖 i 条边, 且有 $F_1 \subseteq F_2 \subseteq \cdots \subseteq F_{|E|} \subseteq V$.

Step 1: 初始化. 令 $i=1, k=1, T(k)=1$.

Step 2: 输出 1-顶点覆盖问题的最优解 x_a, 即权重最小的顶点 (如果存在多个最优解, 则取度数最大的顶点). 输出 $F_1=F_2=\cdots=F_{\deg(x_a)}=\{x_a\}$. 如果 F_1 能够覆盖所有的边, 则继续输出 $F_{\deg(x_a)+1}=F_{\deg(x_a)+2}=\cdots=F_{|E|}=F_1$, 否则执行 Step 3.

Step 3: $k=k+1$. 寻求最大下标 i, 使得 $w(F_i^*)\leqslant q^{k-1}w(F_1^*)$. 其中: 参数 $q>1$ 在下文证明中确定, F_i^* 是 i-顶点覆盖问题的近似比为 c 的近似解, 记 $T(k)=i$.

Step 4: 如果 $T(k-1)=T(k)$, 则转到 Step 3; 否则有 $T(k-1)<T(k)$, 对 $t=T(k-1)+1,\cdots,T(k)$, 输出 $F_t=F_{T(k-1)}\cup F^*_{T(k)}$.

Step 5: 如果 $T(k)=|E|$, 则算法停止; 否则转到 Step 3.

上述算法的思想类似于文献[5]等所设计的竞争算法. 但与之不同的是, 由于这里涉及的基本问题是顶点覆盖问题, 根据覆盖问题的特点, 可以直接利用已有覆盖问题的解构造出新阶段下的问题的解, 并限制新增加的费用, 同时也有利于估计算法的竞争比的度量. 由上述算法的构造过程可知, 算法的输出满足: 1) $F_1\subseteq F_2\subseteq\cdots\subseteq F_{|E|}$; 2) F_j 至少覆盖了 j 条边, $\forall j=1,2,\cdots,|E|$. 此外, 算法的时间受制于 Step 3, 并进而受制于寻求最大下标 i 所需的时间. 由于可以采用预处理的方式 (即对每一个 i 产生所有的 c-近似解, 进而利用二分搜索得到满足条件的最大的下标 i), 算法的时间复杂度即受制于产生近似解 F_i^* 所需的耗费时间. 故如果此时允许使用非多项式时间, 则可令 $c=1$; 否则根据现有的近似算法, 可以令 $c=2$[2], 算法的总时间耗费为多项式时间. 由上述分析可得如下的定理.

定理 1　算法是一个多项式时间竞争算法.

2.3　算法竞争比度量

引理 2　对于 $\forall j\in\{1,2,\cdots,|E|\}$, 有如下不等式成立:

$$w(F_j)\leqslant\sum_{l=1}^{l=k,T(k-1)<j\leqslant T(k)}w(F^*_{T(l)}).$$

证明　$\forall j\in\{1,2,\cdots,|E|\}$, 由算法 Step 4 可知, 若 $j=T(k)$, 则有

$$w(F_j)=w(F_{T(k)})=w\Big(\bigcup_{l=1}^{k}F^*_{T(l)}\Big)\leqslant\sum_{l=1}^{k}w(F^*_{T(l)});$$

若 $j\neq T(k)$, 则存在 k, 使得

$$T(k-1)<j<T(k),$$

$$w(F_j)=w(F_{T(k)})\leqslant\sum_{l=1}^{l=k}w(F^*_{T(l)}).\qquad\square$$

定理 2　令 $q=1+\sqrt{1-1\Big/\Big(c\Big\lceil\dfrac{n}{2}\Big\rceil\varDelta\Big)}$, 算法得到的集合序列 $\{F_1,F_2,\cdots,F_{|E|}\}$ 满足

$$w(F_j)\leqslant\alpha\cdot\mathrm{opt}_j,$$

其中

$$\alpha=2\Big(1+\sqrt{1-1\Big/\Big(c\Big\lceil\frac{n}{2}\Big\rceil\varDelta\Big)}\Big)c,$$

即算法的竞争比为 α.

证明　首先注意到 $F_1,F_2,\cdots,F_{\deg(x_a)}$ 只输出了 1 个顶点, 因此是最优的, 下面的讨论中可以忽略这种特殊情况.

由引理 2 可知, $\forall j\in\{1,2,\cdots,|E|\}$, 设 $T(k-1)<j\leqslant T(k)$, 则有

$$w(F_j)\leqslant\sum_{l=1}^{l=k}w(F^*_{T(l)}).$$

由 Step 3 可得 $w(F_l^*)\leqslant q^{l-1}w(F_1^*)$, 因此

$$w(F_j)\leqslant\sum_{l=1}^{l=k}q^{l-1}w(F_1^*)=\frac{q^k-1}{q-1}w(F_1^*).$$

由于 $T(k)$ 为满足 $w(F_l^*)\leqslant q^{k-1}w(F_1^*)$ 的最大下标, 而 $T(k-1)<j\leqslant T(k)$, 则有

$$w(F_j^*)\geqslant q^{k-2}w(F_1^*).$$

故有

$$w(F_j)\leqslant\frac{q^k-1}{(q^{k-2})(q-1)}w(F_j^*)\leqslant\frac{q^k-1}{(q^{k-2})(q-1)}c\cdot\mathrm{opt}_j.$$

由此可知, 本文只需考察如下函数的上界:

$$f(q)=\frac{q^k-1}{(q^{k-2})(q-1)}=\frac{q^2}{q-1}-\frac{1}{q^{k-2}(q-1)}.$$

设最大的 k 值为 $k_{\max}$, 即 $T(k_{\max})\leqslant|E|$, 则有

$$w(F^*_{T(k_{\max})})\geqslant q^{k_{\max}-2}w(F_1^*),$$

$$w(F^*_{T(k_{\max})})\leqslant w(F^*_{|E|})\leqslant c\cdot\mathrm{opt}_{|E|},$$

从而由引理 1 可得

$$q^{k_{\max}-2}\leqslant c\Big\lceil\frac{n}{2}\Big\rceil\varDelta.$$

由 $q>1, 1\leqslant k\leqslant k_{\max}$ 可得

$$f(q)\leqslant\frac{q^2}{q-1}-\frac{1}{c\Big\lceil\dfrac{n}{2}\Big\rceil\varDelta(q-1)}.$$

对该上界求解最小值可以得到, 当 $q=q^*=1+\sqrt{1-1\Big/\Big(c\Big\lceil\dfrac{n}{2}\Big\rceil\varDelta\Big)}$ 时, 最小值为 $2q^*$, 即算法的竞争比为 $2\Big(1+\sqrt{1-1\Big/\Big(c\Big\lceil\dfrac{n}{2}\Big\rceil\varDelta\Big)}\Big)c$. $\square$

推论 1　本文设计的算法在多项式时间内具有的常数竞争比至多为 $4\Big(1+\sqrt{1-1\Big/\Big(2\Big\lceil\dfrac{n}{2}\Big\rceil\varDelta\Big)}\Big)$, 在非多项式时间内具有的常数竞争比至多为 $2\Big(1+\sqrt{1-1\Big/\Big(\Big\lceil\dfrac{n}{2}\Big\rceil\varDelta\Big)}\Big)$.

特别地, 对于单位权重的占线顶点覆盖问题, 所有的权重均相同, 即 $\varDelta=1$, 由此可得推论 2.

推论 2 对于单位权重的占线顶点覆盖问题, 算法在多项式时间内具有的常数竞争比为 $4\left(1+\sqrt{1-1\Big/\left(2\left\lceil\frac{n}{2}\right\rceil\right)}\right)$, 非多项式时间内具有的常数竞争比为 $2\left(1+\sqrt{1-1\Big/\left(\left\lceil\frac{n}{2}\right\rceil\right)}\right)$.

3 算例分析

下面通过算例验证算法的有效性. 这里对贪婪算法不能得到单位权重下的顶点覆盖问题常数近似比的反例[7]进行分析, 如图 1 所示 (原始反例为图 1 的推广).

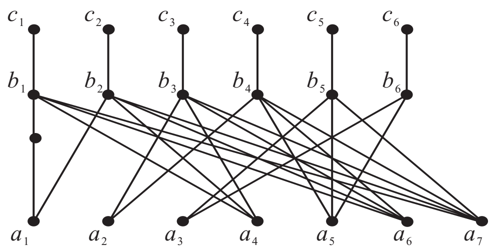

图 1 算例分析

该例子共有 19 个顶点, 27 条边. 容易验证顶点覆盖问题的最优解为 $\{b_1,b_2,b_3,b_4,b_5,b_6\}$, 且有

$$\mathrm{opt}_1=\cdots=\mathrm{opt}_5=1,\ \mathrm{opt}_6=\cdots=\mathrm{opt}_{10}=2,$$
$$\mathrm{opt}_{11}=\cdots=\mathrm{opt}_{15}=3,\ \mathrm{opt}_{16}=\cdots=\mathrm{opt}_{20}=4,$$
$$\mathrm{opt}_{21}=\cdots=\mathrm{opt}_{25}=5,\ \mathrm{opt}_{26}=\mathrm{opt}_{27}=6.$$

为了简单起见, 本文只考虑 $c=1$ 的情形. 在算法的初始阶段, 初始化 $i=1, k=1, T(1)=1$, 输出 $\{a_7, b_1,b_2,b_3,b_4\}$ 中的任意一个顶点. 为了便于分析, 假设输出为最坏情形: 输出不是顶点覆盖问题最优解中的任一顶点, 即输出为 $F_1=F_2=\cdots=F_5=\{a_7\}$. 由于 a_7 不能覆盖所有的边, 需执行 Step 3. 这里令 $q=2$, 寻求最大的下标 i, 使得 $w(F_i^*)\leqslant 2w(F_1^*)=2$, 求得 $i=10$, 则 $T(2)=10$. 在 Step 4 中, 计算 F_{10}^*, 最优解为 $\{a_7,b_1,b_2,b_3,b_4\}$ 中的任意 2 个点, 同样假设此时输出为最坏情形: 假设该顶点为 b_1,b_2(不是 a_7 的任意 2 个点), 此时算法输出 $F_6=F_7=\cdots=F_{10}=\{a_7,b_1, b_2\}$. 在 Step 5 中, 由于 $T(2)\neq 27$, 转到 Step 3, 此时 $k=3$, 寻求使 $w(F_i^*)\leqslant 4$ 最大的下标 i, 输出 $i=20$, $T(3)=20$. 在 Step 4 中, 计算 F_{20}^*, 最优解为 $\{a_7,b_1,b_2, b_3,b_4\}$ 中的任意 4 个点, 假设输出为 $F_{20}^*=\{b_1,b_2,b_3, b_4\}$, 此时算法输出为 $F_{11}=F_{12}=\cdots=F_{20}=\{a_7, b_1,b_2,b_3,b_4\}$. 在 Step 5 中, 检验 $T(3)\neq 27$, 再次转到 Step 3, 此时 $k=4$, 寻求使 $w(F_i^*)\leqslant 8$ 最大的下标 i, 输出 $i=27, T(4)=27$. 在算法的 Step 4 中, 计算 F_{27}^*, 输出为 $\{b_1,b_2,b_3,b_4,b_5,b_6\}$, 此时算法输出为 $F_{21}=F_{22}=\cdots=F_{27}=\{a_7,b_1,b_2,b_3,b_4,b_5,b_6\}$, 算法结束. 因此算法针对此特例的输出为

$$\{F_i|i=1,2,\cdots,27\}=$$
$$\Big\{\underbrace{\{a_7\}}_{i=1,2,\cdots,5},\ \underbrace{\{a_7,b_1,b_2\}}_{i=6,7,\cdots,10},\ \underbrace{\{a_7,b_1,b_2,b_3,b_4\}}_{i=11,12,\cdots,20},$$
$$\underbrace{\{a_7,b_1,b_2,b_3,b_4,b_5,b_6\}}_{i=21,22,\cdots,27}\Big\}.$$

经验证, 竞争比为 $5/3\leqslant 1.667$. 事实上, 本文已经证明 (推论 2), 设计的算法在最坏情况下竞争比最多为 $2(1+\sqrt{1-1/10})\leqslant 3.898$.

4 结 论

本文在占线理论和竞争策略的基础上, 研究了网络监测节点序列部署问题. 基于问题的基本结构特点, 考虑了占线顶点覆盖问题, 设计了常数竞争算法并证明了竞争比. 在未来的研究中, 一方面可以考虑如何结合网络监测系统问题特点设计出更优的竞争策略以改进竞争比; 另一方面, 可以考虑如何将相关网络监测系统的更多实际因素融入到模型中以增加模型的指导性.

参考文献(References)

[1] Hochbaum D. Approximation algorithms for set covering and vertex cover problems[J]. SIAM J of Computing, 1982, 11(3): 555-556.

[2] Gandhi R, Khuller S, Srinivasan. Approximation algorithms for partial covering problems[J]. J of Algorithms, 2004, 53(1): 55-84.

[3] Borodin A, El-Yaniv R. Online computation and competitive analysis[M]. Cambridge: Cambridge University Press, 1998.

[4] 代文强. 占线顶点覆盖问题的结构性下界[J]. 系统工程理论与实践, 2012, 32(1): 134-138.
(Dai W Q. Structural lower bound analysis for online vertex covering problem[J]. Systems Engineering-Theory & Practice, 2012, 32(1): 134-138.)

[5] 代文强. 具有建设成本的占线中心选址问题及其竞争算法设计[J]. 系统工程理论与实践, 2011, 31(12): 2342-2347.
(Dai W Q. Online median problem with constructive cost and its competitive algorithm analysis[J]. Systems Engineering-Theory & Practice, 2011, 31(12): 2342-2347.)

[6] Lin G, Nagarajan C, Rajaraman R, et al. A general approach for incremental approximation and hierarchical clustering[J]. SIAM J of Computing, 2010, 39(8): 3633-3669.

[7] Korte B, Vygen J. Combinatorial optimization: Theory and Algorithms[M]. Berlin Heidelberg: Springer-Verlag, 2012.

(责任编辑: 闫 妍)