文章编号:1003－207(2014)09－0040－09

一类带有限制的网络瓶颈容量扩张问题

刘 慧[1],杨 超[2],杨 珺[2]

(1. 湖北经济学院物流与工程管理学院,湖北 武汉 430205;
2. 华中科技大学管理学院,湖北 武汉 430074)

摘 要:在一些物理网络中,当设施(边的容量等)建立后,由于需求增加,需要调整网络的容量来提高服务水平。调整优化的过程中既要考虑扩张成本,同时也要考虑需要调整的总边数,以尽可能小的影响人们的正常生活。本文研究对于一个给定的网络 G,已知边 e_i 的初始容量和单位容量扩张成本,在预算成本和扩张总边数的约束下,如何有效地扩张边的容量至 x_i,使得系统的容量最大,即 $\max\{\min_{e_i\in T} x_i, T$ 是网络 G 中的生成树$\}$。首先求解两个与之相关的模型,然后通过分析两个相关模型与原问题之间的联系与区别,提出了原问题的多项式时间算法。最后,通过算例说明算法的步骤,并分析了不同参数值对系统容量的影响。

关键词:瓶颈容量扩张;最小生成树;多项式时间算法

中图分类号:C931 **文献标识码**:A

1 引言

网络容量扩张问题是网络改进问题的一种重要类型,其目标是改变给定网络中的某些部分的参数,使得整个网络的性能提高,例如在一定的预算约束下,增加某些结点或者边的容量,使系统的容量最大。通常,一个系统的容量是由它的瓶颈容量来决定的,也就是最弱部分的容量。数学上,这是一个 max-min 问题。网络瓶颈容量扩张问题是网络扩张问题的一种,其在现实生活中有着广泛的应用。随着社会经济的发展和人口的增加,原有的网络已经满足不了人们日益增长的需求,这就需要对网络进行改进,使得改进后的网络能更加有效。比如,一个交通网络要增加流量,一个生产系统要提高生产能力,一个通信网络要提高信息处理能力等等。很多学者已经对网络容量扩张问题做了广泛的研究。Zhang Jianzhong 和 Yang Chao[1] 研究在预算约束下的瓶颈容量扩张问题,并提出多项式时间算法来求解这个问题。Zhang Jianzhong 和 Lin Zhenhong[2] 针对上述模型提出了强多项式时间算法。Burkard 等[3] 用一般的约束扩展了 Zhang Jianzhong 和 Yang Chao[1] 的模型。在 Zhang Jianzhong 和 Yang Chao[1] 的研究基础上,Yang Chao 和 Chen Xueqi[4] 考虑了在预算和瓶颈双重约束下的网络容量扩张问题,同样地给出了多项式时间算法。Burkard 等[5] 研究了权重降低问题(WRPs)。Yang Chao 等[6] 研究了纯效益网络容量扩张问题。王洪国等[7－8] 将 Zhang Jianzhong 和 Yang Chao[1] 的网络瓶颈容量扩张模型加以扩充,解决了无向网络容量扩张和有向网络容量扩张的问题,他们引入了固定扩张费用的概念。吴云等[9] 考虑了扩张费用不确定时的瓶颈容量扩张问题,研究了随机网络瓶颈容量扩张的相关机会规划模型。也有一些学者研究了一般的网络改进问题。Zhang Jianzhong 等[10] 研究了逆向选址问题。杨晓光等[11] 考虑了两个逆网络选址的改进问题,即修改网络上各个边的长度,分别使得网络上某个给定的顶点到网络上所有点的最大距离以及该点到其它顶点的距离之和不大于预先给定的上界,并且所做的修改总量最小。Wang Qin 等[12] 研究了反问题中的最一致性问题。Zhang Jianzhong 等[13] 研究了怎样有效改变网络中每条边的权重,使得网络中各点到根节点的最短路径小于给定的参数,证明了该问题是 NP-hard,并在一般的树形结构网络上给出了多项式算法。Zhang Jianzhong 等[14] 研究了用不同范式来衡量改进成本时,顶点到各点(vertex-to-vertices)的距离降低问

收稿日期:2012－03－07;**修订日期**:2013－05－06

基金项目:国家自然科学基金资助项目(71402048);湖北物流发展研究中心资助项目(2014A16)

作者简介:刘慧(1982－),女(汉族),湖北谷城人,湖北经济学院物流与工程管理学院,讲师,博士,研究方向:网络优化与决策、物流管理.

题。杨珺等[15]研究了在物流配送网络中怎样有效地缩短每条边的响应时间,使得物流网络的覆盖范围更大。

上述对网络改进问题的研究主要考虑的是改进网络所需要的费用,然而容量扩张是一个长期复杂的战略性问题,需要考虑的因素有很多,这就使得以上的研究有一定的局限性。网络容量扩张决策必须经过深思熟虑,因为基础设施的建设非常昂贵,而且由于扩张工程持续的时间较长,会给人们的日常生活带来极大的不便。如果网络中需要扩张的边数太多,将会严重影响网络的正常运行,因此需要扩张的边数也是决策中应考虑的重要问题。例如,随着城市化的发展和人口的增加,现有的一些交通网络已经无法满足人们的日常出行,甚至引起严重的交通堵塞,对现有的交通网络进行扩容势在必行。然而,交通网络的扩容工程浩大,持续时间较长,因此在进行决策时,一方面要考虑财政预算问题,另一方面也要考虑到对人们生活的影响。如果扩建的路的边数太多,势必引起交通问题,给人们的日常生活带来极大的不便,所以一般希望扩建的路段数越少越好。再比如,随着信息化社会的发展,现有的通信网络负载越来越重,不断涌现的网络堵塞、瘫痪等故障使得对通信网络的扩容迫在眉睫。通信网络的扩容问题同样需要权衡新增基站数量与财政预算问题,在保证网络畅通的同时,尽可能减少对人们日常生活的影响。

正是基于这些实际问题,本文将在网络容量扩张问题中不仅考虑预算约束,同时也限制扩张的总边数。利用网络中生成树的瓶颈容量来衡量整个网络系统的性能,考虑扩张费用和总的扩张边数两个约束,对网络中边的容量进行扩张,使得扩张后的系统更加有效。首先提出了该问题的数学模型(P_0),由于模型(P_0)是 max-min 问题,且可行域是非凸的,不易求解。本文不是直接求解模型(P_0),而是先考虑两个与之相关的模型:(P_1)带有边数限制的扩张费用最小问题;(P_2)带有边数限制的最大容量问题。在求解模型(P_1)时,先分析最优解的性质,然后再根据生成树算法的特殊性,将求解模型(P_1)转化为当网络中各边的权重是单个参数 r 的线性函数时,寻找网络中的最小生成树问题,显然该问题有多项式时间算法。在求解模型(P_2)时,同样的也是先分析最优解的性质,根据最优解的特殊性,提出模型(P_2)的多项式时间算法。由模型(P_1)得出预算约束与边数约束之间的关系;由模型(P_2)得出如何在只有边数约束的情况下,求解问题(P_0)。最后,通过分析两个模型(P_1)(P_2)与原问题(P_0)之间的联系与区别,最终提出了原问题(P_0)的多项式时间算法,并通过一个算例,进一步阐述了算法的步骤与实现。

2　带有边数约束的网络瓶颈容量扩张模型

给定一个无向连通网络 $G(V,E)$,顶点集 $V=\{v_1,v_2,\cdots,v_n\}$,边集 $E=\{e_1,e_2,\cdots,e_m\}\in V\times V$,每条边对应一个初始容量 c_i 和单位容量扩张费用 w_i,$C=(c_1,c_2,\cdots,c_m)\in R^m$,$W=(w_1,w_2,\cdots,w_m)\in R^m$。图 $G(V,E)$ 的生成树 $T(N,S)$ 是图 G 的子图,它满足以下三个条件:

(1) T 包含 G 中所有的顶点;

(2) T 中边的个数是 $n-1$,即 $|S|=n-1$;

(3) T 是连通图。

最小生成树连接网络中所有的结点并且总的费用最小。最小生成树在规划有效地分配系统中有着广泛的应用,如管道、输电线路或 leased-line 电话网络的设计和其他通信问题。

假设 F 是图 $G(V,E)$ 中所有生成树的族,即 $F=\{T,T是G中的生成树\}$。对于每个 $T\in F$,生成树 T 的容量是 T 的瓶颈容量,定义为 T 中所有边的容量的最小值,用 $cap(T,C)$ 表示,则 $cap(T,C)=\min\{c_i\mid e_i\in T\}$。族 F 的容量定义为 T 中所有生成树容量的最大值,即 $cap(T,C)=\max\{cap(T,C)\mid T\in F\}=\max\{\min\limits_{e_i\in T}c_i\mid T\in F\}$,网络系统的容量即为族 F 的容量。

本文将考虑:对于一个给定初始容量向量 C 和单位容量扩张费用 W 的网络,在一定的预算约束下,怎样有效地扩张每条边的容量,使得当需要扩张的总边数不超过某一限制 k 时,系统的容量最大。此问题可以表述为以下数学模型:

$$(P_0)\quad \max\{cap(T,X)\mid T\in F\} \tag{1}$$

$$s.t.\quad |support(X-C)|\le k \tag{2}$$

$$W^T(X-C)\le B \tag{3}$$

$$x_i\geqslant c_i\quad i=1,\cdots,m \tag{4}$$

其中集合函数 $support(X-C)=\{j:x_j-c_j\neq 0\}$,$|support(X-C)|$ 表示集合 $support(X-C)=\{j:x_j-c_j\neq 0\}$ 的势,表示扩张后边的容量 X 和初始容量 C 两者间不同值的个数,即需要扩张的总边数。参数 k 表示允许扩张或改进的最大边数,B 表示预先给定的预算值。约束(2)称为边数约束,约束(3)称为预算约束。

这个问题在实际生活中有着广泛的应用。假设电信系统要提高其通信网络的信息流量。假设通信网络为 $G(V,E)$，C 是每条边的初始信息流量，W 为提高每条边信息量的单位费用。定义生成树的容量是树上所有边的瓶颈容量，整个网络的容量是最大的生成树容量，则网络中每对节点间的信息流量的值至少是整个网络的流量。假设预算是 B，允许扩张的最大边数为 k。允许扩张的最大边数一般是电信系统根据对整个系统的影响而定的，显然 k 的值越小，影响越小。我们的问题是怎样有效地改进网络，使得在预算约束 B 和允许扩张的最大边数 k 的限制下，整个通信网络系统的流量最大。

令 $H(r)=\{X \mid X \geqslant C$，且 $cap(F, X) \geqslant r\}$，则该问题可以表示为：

$$(P_0) \quad \max r \tag{5}$$

$$s.t. \ | support(X-C) | \leqslant k \tag{2}$$

$$W^T(X-C) \leqslant B \tag{3}$$

$$X \in H(r) \tag{6}$$

问题 (P_0) 是一个非凸规划，因为它的可行域可能是非凸的。首先我们考虑对参数 k 的限制。如果 k 足够大，则约束 $| support(X-C) | \leqslant k$ 恒成立。当 $k \geqslant n-1$ 时，问题(P_0)等价于

$$\max r$$

$$s.t. \ W^T(X-C) \leqslant B$$

$$X \in H(r)$$

此问题已被 Zhang Jianzhong 和 Yang Chao (2001)[1] 研究，他们将问题转化为当网络中边 E 的权重是关于单个参数 r 的线性函数时，在网络 G 中寻找最小生成树的问题，该问题有强多项式时间算法。

不失一般性，令 $0<k<n-1$，假设 $C=(c_1, c_2, \cdots, c_m)$ 包含 k_0 个不同的值并且按递增的顺序排列：

$$c_{i_1} < c_{i_2} < \cdots < c_{i_{k_0}} \tag{7}$$

容易看出问题(P_0)有一个可行解 $r=c_{i_1}$。$X=C$.

3 模型的求解与算法分析

由于模型 (P_0) 是 max-min 问题，且可行域是非凸的，不易求解。本节不是直接求解模型 (P_0)，而是先考虑两个与之相关的模型。

3.1 带有边数限制的扩张费用最小问题

本节我们考虑在限制扩张边数的情况下，怎样有效地扩张每条边的容量，使得当系统容量不低于某一水平 r 时，所使用的扩张费用最小。此问题可表示为：

$$(P_1) \quad \min W^T(X-C) \tag{8}$$

$$s.t. \ | support(X-C) | \leqslant k \tag{9}$$

$$X \in H(r) \tag{10}$$

模型 (P_1) 的可行域可能为空集。假设 $k<n-1$ 且 $r>c_{k_0}$，由于 $r>c_{k_0}$，每条边 $e_i \in T(\forall\ T \in F)$ 都需要扩张，此时 $k=n-1$（$|T|=n-1$），与假设中 $k<n-1$ 矛盾，显然任意 $X \in R^m$ 都不是模型 (P_1) 的可行解，即模型(P_1) 的可行域为空集。另一方面，如果 k 较大，约束(9)失去约束力，则模型 (P_2) 转化为：

$$\min W^T(X-C)$$

$$s.t. \ X \in H(r)$$

Zhang Jianzhong 和 Yang Chao(2001)[1] 已对上述问题进行了研究，并给出强多项式时间算法。本节中，我们假设模型 (P_1) 的可行域为非空集。下面来分析模型(P_1) 的最优解的性质。

定理 1 假设(T^*, Y)是模型(P_1)的最优解。$X^* \in R^m$，定义如下：

$$x(e_i)=\begin{cases} r, & \text{如果 } e_i \in T^*,\ c_i<r \\ c_i, & \text{其他} \end{cases}$$

则(T^*, X^*)也是模型 (P_1) 的最优解。

证明：$| support(X-C) | \leqslant | support(Y-C) | \leqslant k$，并且 $cap(F, X^*) \geqslant cap(T^*, X^*)=r$，$X^*$ 是模型 (P_1) 的可行解。因为 $cap(F)=cap(T^*)=\min\{y_i, e_i \in T\} \geqslant r$，所以，对于任意 $e_i \in T^*$，$y_i \geqslant r$ 且 $y_i \geqslant c_i$，由 X^* 的定义，显然对于任意的 $e_i \in T^*$，$y_i \geqslant x_i$。

$$W^T(X^*-C)=\sum_{e_i \in E} w_i(x_i-c_i)=\sum_{e_i \in T^*} w_i(x_i-c_i) \leq \sum_{e_i \in T^*} w_i(y_i-c_i) \leqslant \sum_{e_i \in E} w_i(y_i-c_i) \leqslant W^T(Y-C)$$

所以(T^*, X^*)也是模型 (P_1) 的最优解。

为了求解模型 (P_1)，我们定义如下两个权重向量 $\alpha(e)$ 和 $\beta(e)$，它们对应于 e_i 的分量 $\alpha(e_i)$ 和 $\beta(e_i)$ 分别定义如下：

$$\alpha(e_i)=\begin{cases} 0, & \text{如果 } c_i \geq r \\ w_i(r-c_i), & \text{如果 } c_i<r \end{cases}$$

$$\beta(e_i)=\begin{cases} 0, & \text{如果 } c_i \geq r \\ 1, & \text{如果 } c_i<r \end{cases}$$

由定理 1 可知，如果 T^* 是问题 (P_1) 的最优

解,我们只需要扩张生成树 T^* 中初始容量小于 r 的边,并且扩张边的总数不超过 k。模型(P_1)可以转化为如下的等价问题:

$$(P_1') \quad \min_{T\in F}\sum_{e_i\in T}\alpha(e_i) \tag{11}$$

$$s.t. \sum_{e_i\in T}\beta(e_i)\leqslant k \tag{12}$$

$$T\in F$$

由于 F 是生成树的族,所以模型(P_1')是带有限制的最小生成树问题。一般的,带有限制的最小生成树问题是 NP-hard。但是在模型(P_1')中,两个权重有相关性:如果 $\alpha(e_i)=0$,则 $\beta(e_i)=0$;如果 $\alpha(e_i)>0$,则 $\beta(e_i)=1$。

对于给定的网络 $G=(V,E)$,我们首先找出 G 中的两个生成树:

(1) 在权重 $\alpha(e)$ 下找出最小生成树 T^α;

(2) 在权重 $\beta(e)$ 下找出最小生成树 T^β;

记 $l=\sum_{e\in T^\beta}\beta(e_i)$。显然 $l\leqslant k$(模型(P_1)的可行域为非空集)。容易看出,如果 $\sum_{e\in T^\alpha}\beta(e_i)\leqslant k$,则 $T^*=T^\alpha$ 就是最优解。下面,我们来证明 $\sum_{e\in T^\alpha}\beta(e_i)=\sum_{e\in T^\beta}\beta(e_i)$,即 $\sum_{e\in T^\alpha}\beta(e_i)=l$。

用 Kruskal's 算法来找最小生成树。假设 $\alpha(e)=\{\underbrace{0,\cdots,0}_{j个},\underbrace{\alpha(e_{j+1}),\alpha(e_{j+2}),\cdots,\alpha(e_m)}_{(m-j)个}\}$,$\beta(e)=\{\underbrace{0,\cdots,0}_{j个},\underbrace{1,1,\cdots,1}_{(m-j)个}\}$。两个向量中的前 j 个分量都是 0。当我们在权重 $\alpha(e)$ 下找最小生成树时,首先搜索 $\alpha(e)$ 的前 j 个分量,假设找到 v 条边作为最小生成树 T^α 的一部分,然后在向量 $\alpha(e)$ 中剩下的分量 $\{\underbrace{\alpha(e_{j+1}),\alpha(e_{j+2}),\cdots,\alpha(e_m)}_{(m-j)个}\}$ 中搜索剩下的 $n-1-v$ 个分量。显然 $\sum_{e\in T^\alpha}\beta(e_i)=n-1-v$。同样,当我们在权重 $\beta(e)$ 下找最小生成树时,先搜索 $\beta(e)$ 的前 j 个分量,容易得出,同样找出 v 条边作为最小生成树 T^β 的一部分,然后在向量 $\beta(e)$ 中剩余的分量 $\{\underbrace{1,1,\cdots,1}_{(m-j)个}\}$ 中搜索剩下的 $n-1-v$ 条边,则 $\sum_{e\in T^\beta}\beta(e_i)=n-1-v$。所以 $\sum_{e\in T^\alpha}\beta(e_i)=\sum_{e\in T^\beta}\beta(e_i)=l$。

引理 1　如果 F 是网络 $G(V,E)$ 中所有生成树的族,如果 T^α 是在权重 $\alpha(e)$ 下的最小生成树,则 T^α 也是在权重 $\beta(e)$ 下的最小生成树。

从以上的讨论和引理 1 可以看出,要解模型(P_1),我们只需要在权重 $\alpha(e)$ 下找出最小生成树 T^α,则 $\sum_{e\in T^\alpha}\beta(e_i)\leqslant k$,$T^*=T^\alpha$ 是最优解,最优目标函数值是 $\alpha=\sum_{e\in T^\alpha}\alpha(e_i)$。

定理 2　如果 F 是网络 $G(V,E)$ 中所有生成树的族,则模型(P_1)可以用在权重 $\alpha(e)$ 下寻找最小生成树的方法来求解,模型(P_1)有多项式时间算法。

由模型(P_1)的求解方法和引理 1 可以看出,模型(P_0)中的边数约束(2)和预算约束(3)之间存在如下关系:对于给定的 r,设 T^α 是在权重 $\alpha(e)$ 下的最小生成树,令 $\varphi(r)=\sum_{e\in T^\alpha}\alpha(e_i)$,$\varphi(r)=\sum_{e\in T^\alpha}\beta(e_i)$,则 $\varphi(r)=\min\{|support(X-C)||X\in H(r)\}$。若 $\varphi(r)=\sum_{e\in T^\alpha}\alpha(e_i)=B$,$\varphi(r)=\sum_{e\in T^\alpha}\beta(e_i)\leqslant k$,则 r 是模型(P_0)的最优解。若 $\varphi(r)=\sum_{e\in T^\alpha}\alpha(e_i)>B$ 或 $\varphi(r)=\sum_{e\in T^\alpha}\beta(e_i)>k$,则不满足约束条件。由于 $\varphi(r)$ 和 $\varphi(r)$ 是关于 r 的单调非减函数,所以需适当地降低 r,使得约束条件(2)和(3)均满足。

3.2　带有边数限制的最大容量问题

本节我们考虑只在限制扩张边数的情况下,怎样有效地扩张每条边的容量,使得系统容量最大。此问题可表示为:

$$(P_2) \quad \max r$$

$$s.t. \ |support(X-C)|\leqslant k$$

$$X\in H(r)$$

与第二节中一样,假设 C 包含 k_0 个不同的值,且按递增的顺序排列。容易看出,模型(P_2)有一个可行解 $r=c_{i_1}$,$X=C$。不失一般性,假设 $0\leqslant k<n-1$。

定理 3　如果 F 是网络 $G(V,E)$ 中所有生成树的族,$0\leqslant k<n-1$,如果 (T^*,r^*) 是模型(P_2)的最优解,则 $r^*\in\{c_i,e_i\in E\}$,即 r^* 必是 $c_{i_1},c_{i_2},\cdots,c_{i_{k_0}}$ 中的某个值。

证明:若 $r^*>c_{i_{k_0}}$,则对任意 $X\in R^m$,$|support(X-C)|>k$,所以 $c_{i_1}\leqslant r^*\leqslant c_{i_{k_0}}$。假设最优解 $r^*\in(c_{i_m},c_{i_{m+1}})$,$T^*$ 是对应的生成树。$G=(e_{i_1},\cdots,e_{i_m},e_{i_{m+1}},\cdots,e_{i_{k_0}})$,定义权重向量 $\sigma^{r^*}(e)$ 如下:$\sigma^{r^*}(e_i)=\begin{cases}1, 若\ c_i<r^*\\0, 若\ c_i\geqslant r^*\end{cases}$,则 T^* 为在权重

$\sigma^{r^*}(e)$ 下的最小生成树，且 $\sigma(T^*)=\sum_{e_i\in T^*}\sigma^{r^*}(e_i)=k$。令 $r'=c_{m+1}$，定义权重向量 $\sigma^{r'}(e)$ 如下：$\sigma^{r'}(e_i)=\begin{cases}1, & \text{若 } c_i<r'\\ 0, & \text{若 } c_i\geqslant r'\end{cases}$，则 $\sigma^{r'}(e)=\sigma^{r^*}(e)$，$T^*$ 为在权重 $\sigma^{r'}(e)$ 下的最小生成树，所以 r' 也是最优解，且 $r'>r^*$，这与 r^* 是最优解矛盾，所以 r^* 必是 $c_{i_1},c_{i_2},\cdots,c_{i_{k_0}}$ 中的某个值。

由定理 3，我们可以设计出一个算法来解模型 (P_2)。算法的具体步骤如下：

算法 1：

令 $s=i_{k_0}$，定义一个权重向量 $\sigma^r(e)$，对应于 e_i 的分量 $\sigma^r(e_i)$ 定义如下：

$$\sigma^r(e_i)=\begin{cases}0, & \text{如果 } c_i\geqslant r\\ 1, & \text{如果 } c_i<r\end{cases}$$

步骤 1　若 $s=1$，转到步骤 3；否则，令 $r=c_s$，在权重 $\sigma^r(e_i)$ 下找出最小生成树 T^r，计算 $\sigma=\sigma(T^r)=\sum_{e_i\in T^r}\sigma^r(e_i)$。

步骤 2　若 $\sigma\leqslant k$，则 $r^*=r$，$T^{r^*}=T^r$，转到步骤 4；否则 $s\Leftarrow s-1$，转到步骤 1。

步骤 3　$r^*=c_{i_1}$，$T^{r^*}=T^{c_{i_1}}$。

步骤 4　若 $e\in T^{r^*}$ 且 $c_i<r^*$，$x_i=r^*$；否则 $x_i=c_i$，结束。

定理 4　如果 F 是网络 $G(V,E)$ 中所有生成树的族，$0\leqslant k<n-1$，则模型 (P_2) 有多项式时间算法。

注意模型 (P_2) 与模型 (P_0) 的区别与联系：模型 (P_2) 比模型 (P_0) 少了预算约束(3)，若预算 B 较大，即对某个值 r，约束(3) 总是满足时，可以通过求解模型 (P_2) 的方法来寻找模型 (P_0) 的最优解。

3.3　带有边数限制的瓶颈容量扩张问题

基于以上两个相关模型 (P_1) 和 (P_2) 的算法及分析，现在来求解原模型 (P_0)。与 3.1 节中相同，令 $\varphi(r)=\min_{T\in F}\sum_{e\in T}\alpha(e_i)$，$\varphi(r)=\min_{T\in F}\sum_{e\in T}\beta(e_i)$。$\alpha(e)$，$\beta(e)$ 的定义也同 3.1 节。显然，$\varphi(r)$ 和 $\varphi(r)$ 是关于 r 的非减函数。不失一般性，假设 $0\leqslant k<n-1$，关于 C 的值假设同上。容易知道，$c_{i_1}\leqslant r\leqslant c_{i_{k_0}}$，模型 (P_0) 有可行解 $r=c_{i_1}$，$X=C$。

由以上的定理和模型 $(P_1)(P_2)$ 的算法，我们来设计求解模型 (P_0) 的算法。算法的主要思想是：通过逐渐降低 r 的值，来检验 r 是否满足约束条件(2) 和(3)。在检验时，首先利用引理 1 以及约束(2) 和(3) 之间的关系，找出满足预算约束(3) 的 r 值的范围。当预算约束满足时，根据定理 3 来寻找最优解，同时注意在权重 $\alpha(e)$ 下寻找，因为由引理 1，若 $T^\#$ 使得 $\varphi(r)$ 最小，则 $T^\#$ 也使得 $\varphi(r)$ 最小。

算法的具体步骤如下：

算法 2：

令 $h=i_{k_0}$，$S=\{e\mid e\in T^r,c_i\leqslant r\}$。

步骤 1 若 $h=i_1$，转到步骤 9。否则令 $r=c_h$。在权重 $\alpha(e)$ 下找出最小生成树 T^r，并计算 $\varphi(r)=\sum_{e\in T^r}\alpha(e_i)$ 和 $\varphi(r)=\sum_{e\in T^r}\beta(e_i)$。

步骤 2　若 $\varphi(r)\leqslant B$，转到步骤 3。若 $\varphi(r)>B$，$h\Leftarrow h-1$，转到步骤 1。

步骤 3　令 $j=h$，

(1)若 $\varphi(c_j)=B$，$\varphi(c_j)\leqslant k$，令 $r^*=c_j$，，$T^{r^*}=T^{c_j}$，转到步骤 10；

(2)若 $\varphi(c_j)=B$，$\varphi(c_j)>k$，转到步骤 7；

(3) 若 $\varphi(c_j)<B$，$\varphi(c_j)\leqslant k<\varphi(c_{j+1})$，令 $r^*=c_j$，$T^{r^*}=T^{c_j}$，转到步骤 10；

(4)若 $\varphi(c_j)<B$，$\varphi(c_{j+1})\leqslant k$，转到步骤 4；

(5)若 $\varphi(c_j)<B$，$\varphi(c_j)>k$，转到步骤 7。

步骤 4　令 $t=0$，$(l_t,u_t)=(c_j,c_{j+1})$.

步骤 5　$r^\#=\dfrac{l_t+u_t}{2}$，在权重 $\alpha(e)$ 下找出最小生成树 $T^{r^\#}$，并计算 $\varphi(r^\#)$。

(1)若 $\varphi(r^\#)=B$，令 $r^*=r^\#$，$T^{r^*}=T^{r^\#}$，转到步骤 10；

(2)若 $\varphi(r^\#)>B$，令 $u_t=r^\#$；

(3)若 $\varphi(r^\#)<B$，令 $l_t=r^\#$。

步骤 6　令 $r'=\dfrac{B+\sum_{e\in S\cap T^{l_t}}w_ic_i}{\sum_{e\in S\cap T^{l_t}}w_i}$。若 $\varphi(r')=B$，$r^*=r'$，$T^{r^*}=T^{r'}$，转到步骤 10。否则，$t\Leftarrow t+1$ 转到步骤 5。

步骤 7　若 $j=i_1$，转到步骤 9；否则，令 $g=j$，$r=c_g$，在权重 $\alpha(e)$ 下找出最小生成树 T^r，并计算 $\varphi(r)=\sum_{e\in T^r}\alpha(e_i)$，$\varphi(r)=\sum_{e\in T^r}\beta(e_i)$。

步骤 8　若 $\varphi(r)\leqslant k$，则 $r^*=r$，$T^{r^*}=T^r$，转到步骤 10；否则 $j\Leftarrow j-1$，转到步骤 7。

步骤 9　$r^*=c_{i_1}$，$T^{r^*}=T^{c_{i_1}}$。

步骤 10　若 $e_i\in T^{r^*}$ 且 $c_i<r^*$，$x_i=r^*$；否则 $x_i=c_i$，结束。

关于算法的说明如下：

(1)步骤 1 和步骤 2 找出在只满足预算约束(2.3)时,r 的值所在的区间。

(2)在步骤 3 中,情形 1 和情形 3 表示已经找到最优解;情形 4 表明最优解在区间 $[c_j, c_{j+1}]$ 中,因为 $\varphi(c_j) < B$, $\varphi(c_{j+1}) > B$, $k \geq \varphi(c_{j+1})$;情形 2 和情形 5 表明最优解在区间 $[c_{i_1}, c_j]$ 中。

(3)若最优解在区间 $[c_j, c_{j+1}]$ 中,步骤 4、5、6 表示采用二分法来寻找最优解,且能够在多项式时间内找到最优解。

(4)若最优解在区间 $[c_{i_1}, c_j]$ 中,由引理 1 我们只需要逐渐降低 r 的值,在权重 $\alpha(e)$ 下找出最小生成树 T^r,直至 T^r 在权重 $\beta(e)$ 下的总权重不超过 k。

定理 5　如果 F 是网络 $G(V,E)$ 中所有生成树的族,$0 \leqslant k < n-1$,则问题 (P_0) 有多项式时间算法。

证明:由算法 2,若最优解在区间 $[c_j, c_{j+1}]$ 中,Zhang Jianzhong 和 Yang Chao[1] 已证明可以在多项式时间内找出最优解。若最优解在区间 $[c_{i_1}, c_j]$ 中,根据定理 3,最优解取 $c_{i_1}, c_{i_2}, \cdots, c_j$ 中的某个值。依次令 r 取 $c_j, c_{j-1}, \cdots, c_{i_2}, c_{i_1}$,在权重 $\alpha(e)$ 下找出最小生成树 T^α,直至 T^α 在权重 $\beta(e)$ 下的总权重不超过 k 时结束。而求网络中的最小生成树具有多项式时间算法。由此,算法 2 是多项式时间的。

4　算例

为了更清楚地阐述模型 (P_0) 及算法 2 的求解步骤,本节针对一个具体的网络来求解模型 (P_0)。本节使用 Zhang Jianzhong 和 Yang Chao[1] 中研究的网络,网络的结构如图 1 所示,图 1 包含 6 个顶点,9 条边。网络中边的初始容量和单位容量扩张费用如表 1 所示。网络中的边包含有 5 个不同的初始容量,分别令 $r = 5, 4, 3, 2, 1$ 时,计算两个权重向量 $\alpha(e)$ 和 $\beta(e)$,它们对应于 e_i 的分量 $\alpha(e_i)$ 和 $\beta(e_i)$ 为:

$$\alpha(e_i) = \begin{cases} 0, & \text{如果 } c_i \geqslant r \\ w_i(r - c_i), & \text{如果 } c_i < r \end{cases}$$

$$\beta(e_i) = \begin{cases} 0, & \text{如果 } c_i \geqslant r \\ 1, & \text{如果 } c_i < r \end{cases}$$

根据权重 $\alpha(e)$,求出 G 中的最小生成树 T^r,计算 $\varphi(r) = \sum_{e \in T^r} \alpha(e_i)$, $\varphi(r) = \sum_{e \in T^r} \beta(e_i)$,得出不同 r 值下的最小生成树以及生成树分别在 $\alpha(e)$ 和 $\beta(e)$ 下的总权重如表 2 所示。显然 $0 \leqslant k < 5$ 。若 $k \geqslant 5$,则模型 (P_0) 中的边数约束(2)失效;当 $0 \leqslant k < 5$ 时,若 $B \geqslant 39$,则模型 (P_0) 中的预算约束(2.3)失效。不失一般性,本文的算例中取 $0 < k < 5$, $0 < B < 39$ 。

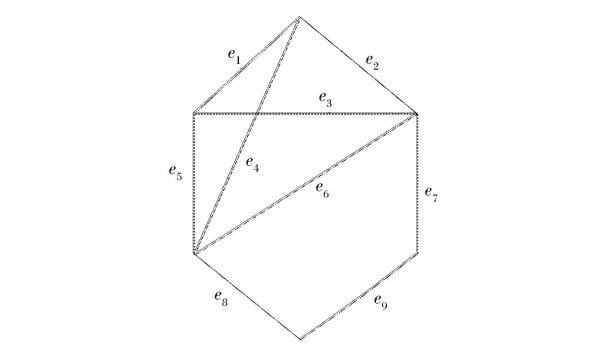

图 1　算例的网络结构图

表 1 网络中边的初始容量和单位容量扩张费用

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
初始容量 c_i	3	1	4	3	3	5	2	2	4
单位容量扩张费用 w_i	10	2	6	6	8	12	8	5	10

表 2　不同 r 值下的最优生成树和生成树的权重

r	对应的最小生成树 T^r	预算权重 $\varphi(r)$	边数权重 $\varphi(r)$
5	$\{e_2, e_3, e_6, e_8, e_9\}$	39	4
4	$\{e_2, e_3, e_6, e_8, e_9\}$	16	2
3	$\{e_1, e_5, e_6, e_8, e_9\}$	5	1
2	$\{e_1, e_5, e_6, e_8, e_9\}$	0	0
1	$\{e_1, e_5, e_6, e_8, e_9\}$	0	0

下面根据预算 B 和边数限制 k 的不同取值,来详细说明算法 2:

(*a*) $B = 16, k = 3$

根据算法 2 的步骤 1 和步骤 2,先令 $r = 5$,计算 $\varphi(5) = 39$,由于 $\varphi(5) > B$,降低 r 的值,令 $r = 4$,计算 $\varphi(4) = 16$,由于 $\varphi(4) \leqslant B$,则 $r^* \in [1,5)$。而 $\varphi(4) = B$, $\varphi(r) = 2 \leqslant k$,此时对应于步骤 3 中的情况(1),直接得到最优解 $r^* = 4$,对应的最优生成树 $T^* = \{e_2, e_3, e_6, e_8, e_9\}$。

(*b*) $B = 16, k = 1$

由于 $\varphi(5) > B$, $\varphi(4) \leq B$,则 $r^* \in [1,5)$。而 $\varphi(4) = B$, $\varphi(4) = 2 > k$,此时对应于步骤 3 中的情况(2),$r = 4$ 时,不满足边数约束(2),逐渐降低 r 的值,使之满足边数约束。令 $r = 3$,计算 $\varphi(3) = 5 < B$, $\varphi(r) = 1 = k$,得到最优解 $r^* = 3$,对应的最优生成树 $T^* = \{e_1, e_5, e_6, e_8, e_9\}$。

再说明任意的 $r \in (3,4)$ 都不是最优解。根据定理 3,当预算约束(3)满足时,r^* 必是 $c_{i_1}, c_{i_2}, \cdots, c_{i_{k_0}}$ 中的某个值。例如令 $r = 3.5$,计算 $\varphi(3.5) = 10.7 \leqslant B$,而 $\varphi(3.5) = 2 > k$, $r = 3.5$ 不满足边数约束(2)。

(*c*) $B = 20, k = 2$

表 3 不同参数值下的最优解和约束的情况

参数的值		最优解			约束	
预算 B	边数 k	容量 r^*	扩张决策 X^*	最优生成树 T^r	预算约束是否为紧约束	边数约束是否为紧约束
16	3	4	(3,3,4,3,3,5,2,4,4)	$\{e_2, e_3, e_6, e_8, e_9\}$	是	否
16	1	3	(3,1,4,3,3,5,2,4,3)	$\{e_1, e_5, e_6, e_8, e_9\}$	否	是
20	2	4	(3,1,4,4,3,5,2,4,4)	$\{e_2, e_3, e_6, e_8, e_9\}$	否	是
20	4	4.148	(3,4.148,4.148,3,3,5,2,4.148,4.148)	$\{e_2, e_3, e_6, e_8, e_9\}$	是	是
20	1	3	(3,1,4,3,3,5,2,3,4)	$\{e_1, e_5, e_6, e_8, e_9\}$	否	是

由于 $\varphi(5) > B$, $\varphi(4) < B$, $\varphi(4) = 2 = k$, 则 $r^* \in [4,5)$。此时对应于步骤 3 中的情况(3),直接得到最优解 $r^* = 4$, 对应的最优生成树 $T^* = \{e_2, e_3, e_6, e_8, e_9\}$。

(*d*) $B = 20, k = 4$

由于 $\varphi(5) > B$, $\varphi(4) < B$, $\varphi(4) = 2 < k$, $\varphi(5) = 4 \geq k$, 则 $r^* \in (4,5)$。此时对应于步骤 3 中的情况(4),当 $r \in (4,5)$ 时,由于边数约束(2)总能满足,只需要找到使得 $\varphi(r) = B$ 的 r 值。根据步骤 4、5、6,利用二分法,在区间 (4,5) 搜索。令 $t = 0$, $(l_0, u_0) = (4,5)$, $r^{\#} = \dfrac{l_0 + u_0}{2} = 4.5$, 在权重 $\alpha(e)$ 下找出最小生成树 $T^{r^{\#}}$, 并计算 $\varphi(r^{\#}) = 27.5$, 由于 $\varphi(r^{\#}) > B$, 则 $r^* \in (4,4.5)$。$T^{l_0} = T^4 = \{e_2, e_3, e_6, e_8, e_9\}$, $r' = \dfrac{B + \sum\limits_{e \in S \cap T^{l_0}} w_i c_i}{\sum\limits_{e \in S \cap T^{l_0}} w_i} = 4.148$。而 $\varphi(r') = 20 = B$, 则 $r^* = r' = 4.148$, 对应的最优生成树 $T^{r^*} = T^{r'} = \{e_2, e_3, e_6, e_8, e_9\}$。

(*e*) $D = 20, k = 1$

由于 $\varphi(5) > B$, $\varphi(4) < B$, $\varphi(4) = 2 > k$, 则 $r^* \in [1,4)$。此时对应于步骤 3 中的情况(5),逐渐降低 r 的值,令 $r = 3$,计算 $\varphi(r) = 1 = k$, 得到最优解 $r^* = 3$, 对应的最优生成树 $T^* = \{e_1, e_5, e_6, e_8, e_9\}$。

在预算 B 和允许扩张的边数 k 的不同取值时,利用算法 2 求解同时带有预算约束和边数约束的网络瓶颈容量扩张模型(P_0),得到系统的最优容量 r^*、扩张策略 X^* 以及对应的最优生成树如表 3 所示,表 3 的最后两列表示约束的松紧情况,如当 $B = 16, k = 3$ 时,最优解是 $r^* = 4$, 边数约束(2)的左端 $|support(X^* - C)| = 2 < k$, 预算约束(3)的左端 $W^T(X^* - C) = 16 = B$, 可得预算约束是紧约束,而边数约束是非紧约束。再如当 $B = 20, k = 4$ 时,最优解是 $r^* = 4.148$, 边数约束(2)的左端 $|support(X^* - C)| = 4 = k$, 预算约束(3)的左端 $W^T(X^* - C) = 20 = B$, 可得预算约束和边数约束都是紧约束。决策者可以根据表 3 中约束是否为紧约束,掌握资源的利用情况。

由表 3 中的结果,决策者可做如下决策:

(1)由于系统容量的增加是以预算的增加或允许扩张边数的增加为代价,决策者要决定这种增加是否值得。如当 $B = 16$ 时,允许扩张的最大边数 k 从 3 条增加到 4 条,边数增加了 1 条,系统的容量也从 3 增加到 4,系统容量的增加是以扩张边数的增加为代价的。决策者可以根据网络的具体情况来决定这种增加是否值得。

(2)给定系统要达到的容量 r 时,决策者应该如何选择预算和允许扩张的最大边数。例如,当 $B = 16, k = 3$ 时,系统的容量是 4,而当 $B = 20, k = 2$ 时,系统的容量也是 4,两组参数的区别是一个预算大,而另一个需要扩张的边数多,而两组参数下的系统容量相同,决策者可以在两组参数中任选其一,取决于预算和扩张边数的权重。

从本节中的算例可以看出,算法 2 能够有效地求解问题(P_0)。本节的算例规模较小,而对于较大规模的网络,决策者可以编程实现算法 2,以便其用于解决现实中大规模的网络瓶颈容量扩张问题。

5 结语

网络容量扩张问题是现实网络改进问题的一种常见类型。在一些物理网络中,如交通网络、通信网络和物流网络等,当设施(边上的运输能力、最大流量等)建立后,由于人们对各种服务的需求日益增长,现有的网络已经不能满足当前的需求,而重新布置设施又花费巨大,因此需要对现有的网络进行改进,以此来提高网络的运行效率。如果对网络做大量的改进,则会严重影响网络的运行。因此,在网络的调整优化过程中既要考虑系统最终的效率和扩张成本,同时也要考虑需要调整的总边数,这样尽可能小地影响人们的正常生活。本文考虑了在一定的预算约束下,如何有效地扩张网络中各条边的容量,使得扩张后的系统更有效,并且需要扩张的总边数不

超过一定的限制。而系统的容量定义为网络中生成树的瓶颈容量。通过求解与之相关的两个模型,最终提出了原问题的多项式时间算法。最后,通过一个算例,说明算法的有效性。决策者可以选取不同的预算值和允许扩张的最大边数,得到不同的扩张策略以及系统的最优容量,并根据系统容量与参数之间的关系作相关决策。但是,本文提出的算法成立的条件是当网络系统的容量是用网络中生成树的瓶颈容量的最大值来衡量时,如果不采用这种衡量方法,算法 2 并不适用。例如,采用网络中两点间所有路径瓶颈容量的最大值来衡量网络的容量,两点间的路径有多条,在做扩张决策时,有的路径扩张成本低,有的路径初始容量大,即需要扩张的边数少,此时,扩张的费用和需要扩张的边数之间并不存在像引理 1 一样的关系,就不能用算法 2。本文研究的问题对于有效提高网络设施的服务水平、提升网络的运行效率具有一定的指导意义,如有效改进现实生活中的交通网络、通信网络和物流网络等。本模型仅对特殊的树形结构的网络提出了有效的算法,今后可以研究更加一般的模型,如 F 是网络中所有从源节点到终节点的路径所组成的集合,或者 F 是二部图中的所有匹配所组成集合等等。这些问题都是以后的研究方向。

参考文献:

[1] Zhang, Jianzhong, Yang Chao, Lin Yixun. A class of bottleneck expansion problems [J]. Computer & Operations Research, 2001, 28(6): 505－519.

[2] Zhang Jianzhong, Lin Zhenhong. An oracle strongly polynomial algorithm for bottleneck expansion problems [J]. Optimization Methods and Software, 2001, 17(1): 61－75.

[3] Burkard R E, Klinz B, Zhang Jianzhong. Bottleneck capacity expansion problem with general budget constraints [J]. RAIRO-Operations Research, 2001, 35: 1－20.

[4] Yang Chao, Chen Xueqi. An inverse maximum capacity path problem with lower bound constraints [J]. Acta Mathematica Scientia, 2002, 22(2): 207－212.

[5] Burkard R E, Lin Yixun, Zhang Jianzhong. Weight reduction problems with certain bottleneck objectives [J]. European Journal of Operational Research, 2004, 153(1): 191－199.

[6] Yang Chao, Hao Chunyan, Zhang Jianzhong. On the optimum capacity of capacity expansion problems [J]. Mathematical Methods of Operations Research, 2007, 66(2): 225－233.

[7] 王洪国, 马诏汉. 关于无向网络容量扩张的问题 [J]. 山东大学学报, 2000, 35(4): 418－424.

[8] 王洪国, 马诏汉. 关于有向网络容量扩张的问题 [J]. 高校应用数学学报, 2001, 16(4): 471－476.

[9] 吴云, 周建, 杨珺. 随机网络瓶颈容量扩张相关机会规划模型 [J]. 中国管理科学, 2004, 12(6): 113－117.

[10] Zhang Jianzhong, Lin Zhenhong, Ma Zhongfan. Some reverse location problems [J]. European Journal of Operational Research, 2000, 124(1): 77－88.

[11] 杨晓光, 张建中, 蔡茂诚. 两个逆网络选址问题的计算复杂性 [J]. 系统科学与数学, 2002, 22: 321－327.

[12] Wang Qin, Yuan Jinjiang, Zhang Jianzhong. An inverse model for the most uniform problem [J]. Operations Research Letters, 2008, 36(1): 26－30.

[13] Zhang Jianzhong, Yang Xiaoguang, Cai M C. Inapproximability and a polynomially solvable special case of a network improvement problem [J]. European Journal of Operational Research, 2004, 155(1): 251－257.

[14] Zhang Jianzhong, Yang Xiaoguang, Cai M C. A network improvement problem under different norms [J]. Computational Optimization and Applications, 2004, 27(3): 305－319.

[15] 杨珺, 王玲, 杨超. 优化设施服务的网络调整费用均衡问题研究 [J]. 中国管理科学, 2009, 17(5): 75－80.

A Class of Network Bottleneck Capacity Expansion Problem with Constraints

LIU Hui[1], YANG Chao[2], YANG Jun[2]

(1. School of Logistics and Engineering Management, Hubei University of Economics, Wuhan 430205, China;
2. School of Management, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract: Established infrastructure makes a great impact on demand changes in particular physical network. It is necessary to adjust the capacities of arcs to improve the service of the network when demand increases. The process of adjusting and optimizing should possible to minimize influences on people's daily life, by considering not only the expansion cost but also the total edges adjustment. In this paper, a network G, the initial capacity of edge e_i and the cost for increasing per unit capacity of e_i are initially set.

Two factors: the improvement cost and the total edges adjustment are considered in this problem. The task is to determine new capacities x_i so that the capacity of the network can be increased to the maximum extent, i.e. $\max\{\min_{e_i \in T} x_i, T \text{ is the spanning tree of network } G\}$. Firstly two related models are solved instead of the original model. The relations and differences between the two related models and the original problem are analyzed. Then an algorithm is present to solve the original problem in polynomial time. Finally, an example is computed to illustrate the steps of the algorithm and analyze the impacts of parameters to the capacity of the system.

Key words: bottleneck capacity expansion; minimum spanning tree; polynomial time algorithm