QoS 路由问题的反向优化算法

张　品[*,**]　　李乐民[*]　　王　晟[*]

[*](电子科技大学宽带光纤传输与通信系统技术重点实验　成都　610054)

[**](杭州电子科技大学通信学院　杭州　310017)

摘　要：寻找满足两个加性 QoS 约束条件的路径是网络 QoS 路由研究的核心问题，线性搜索算法是重要近似算法之一。本文提出一种结合了反向优化策略的线性搜索算法。当线性搜索过程所得到的路径不满足 QoS 需求时，对搜索到的路径选取合适的节点进行反向优化。算法的时间复杂度为 $O(K(m+n\log_2(n)))$ 。仿真显示本文的搜索策略扩大了搜索空间，提高了寻找可行路径的成功率。

关键词：两约束路由问题，线性搜索算法，反向优化

中图分类号：TN915.01　　**文献标识码**：A　　**文章编号**：1009-5896(2005)06-0952-05

Reverse Optimization Algorithm for QoS Routing Problem

Zhang Pin[*,**]　　Li Le-min[*]　　Wang Sheng[*]

[*]*(Key Lab of Optical Fiber and Communication Networks, UEST of China, Chengdu* 610054, *China)*

[**]*(College of Communications, Hangzhou Dianzi University, Hangzhou* 310017, *China)*

Abstract　Finding the path satisfying two additive QoS constraints is the key question of QoS research. The linear search algorithm is one of important approximation algorithms. This paper proposes a new linear search algorithm combined with the reverse optimization scheme. If the path found by the linear search procedure does not satisfy the QoS constraints, the proper nodes of the path are chosen to make the reverse optimization. The time complexity of proposed algorithm is $O(K(m+n\log_2(n)))$. The simulation shows that the new approach extends the search fields and improves the succeeding ratio of finding the feasible paths.

Key words　Bi-constraint path, Linear search algorithm, Reverse optimization

1　引言

目前，大规模的实时数据应用传输诸如多媒体应用，视频及音频会议，以及各种大规模数据发布等等已经在局域网或广域网得到应用，这些应用对于服务质量(Quality of Service，QoS)有严格的要求。各种 QoS 需求体现为一系列参数(metrics)，这些 metrics 对于应用程序的接收者是至关重要的，决策者(路由器)所确立的具体连接(通常是路径或树)必须满足相关的 QoS 需求。典型的 QoS metrics 包括带宽(bandwidth)，延迟(delay)，延迟漂移(delay jitter)，损失率(lost rate)等等[1,2]。

QoS 约束条件分为以下 3 种基本种类：(1) 加性约束，如延迟，延迟漂移，跳数等；(2) 乘性约束，如损失率等；(3)凹性约束，如带宽等。我们称一个 metrics 是加性或乘性的，如果端到端连接的 metrics 为构成该连接的所有链路的 metrics 的和或积。称一个 metrics 是凹性的，如果端到端连接的 metrics 为构成该连接的所有链路的 metrics 的最小值。本质上只有两种 metrics，链路乘性 metrics 可以通过取负对数转化为加性 metrics[3]。

最基本的 QoS 路由问题如下：给定源结点 s 和目标节点 t，一系列 QoS 约束 D，以及可能的代价目标函数，要求发现一条(s, t)间的路径满足所有端到端约束条件或同时优化代价函数。该问题称为多约束路由问题(Multi-Constraint Path problem，MCP)以及多约束优化路由问题(Multi-Constraint Optimization Path problem，MCOP)。

文献[3]证明一般带有两个或两个以上独立加性QoS约束的路由问题是NPC(NP-Complete)的。特殊情况下多项式可解的多约束路由问题包括[4]：(1) 其中一个约束条件是跳数的由两个加性约束条件组成的端到端路径问题；(2) 由一个凹性约束和一个加性约束组成的端到端路径问题；(3) 由多个凹性约束组成的端到端路径问题。最基本的NPC难度的QoS路由问题是两约束路径问题(Bi-Constraint Path problem，BCP)以及受约束最短路径问题(Restricted Shortest Path problem，RSP)，BCP要求寻找(s, t)间满足两个互相独立

2004-01-09 收到，2004-06-25 改回

国家自然科学基金(60002004) 资助课题

的加性约束条件的路径。RSP要求寻找(s, t)间满足加性约束条件且代价(cost)最小的路径，也就是要求满足一个加性约束条件，同时优化另一个加性约束条件。本质上BCP是RSP对应的判定问题，RSP属于FPTAS [5](Fully Polynomial Time Approximation Scheme)类，即RSP可以由网络节点数和$1/\varepsilon$的多项式求解，ε为算法的相对误差。

BCP或RSP有密切联系，任何解决RSP的算法都可以用来解决BCP，而任何BCP的算法则可以通过对优化目标运用两分法得到RSP的ε近似解。这两个问题都是NPC的[5]，关于BCP或RSP已经提出了很多算法，文献[6]对此做了综述。有关BCP或RSP的算法可以分成3类：(1) 理论近似算法，文献[7,8]提出了RSP的ε-多项式算法，文献[8]的算法的时间复杂度为$O(m(n^2/\varepsilon)\log_2(n/\varepsilon))$，该结果表明BCP或RSP属于“最简单”的FPTAS类问题，该问题可以用多项式算法无限逼近，然而无论是文献[7]还是文献[8]的多项式近似算法由于其过高的时间复杂度在实际情况中无法应用。(2) 动态规划算法，以文献[9,10]为代表，该算法解决链路参数以及约束参数为整数时的RSP以及BCP，对于一般情况通过使链路参数离散化来求解，其中文献[10]的算法的时间复杂度为$O(D_1D_2nm)$，D_1为D_2为离散化的路径约束条件值。虽然BCP或RSP能够用动态规划的方法解决，但这些算法由于时间开销太大仍然仅仅具备理论价值。(3) 启发式算法，启发式算法追求效率与时间复杂度的平衡，缺点是算法的实际效果在理论上难以预测。在众多启发式算法中，线性搜索算法以其较低的时间复杂度和易实施性成为重要的一类算法。文献[11]首先提出了线性搜索算法，该算法的核心是用约束条件的加权线性和$h(e)=\alpha w_1(e)+\beta w_2(e)$来表示代价函数，再用标准最短路径算法求最短途径。文献[12]进一步发展了这一思想。线性搜索算法的核心是联系两个约束条件的参数的调节方法，文献[13]提出的双重搜索(Binary Search，BS)算法是目前最有效的线性搜索算法。BS的核心思想是每次迭代用$h(e)=w_1(e)+kw_2(e)$或$kw_1(e)+w_2(e)$作为链路权重，运用标准最短路径算法进行双重搜索，其时间复杂度为$O(K(m+n\log_2(n)))$，K为迭代次数，该算法的特点是取得了算法时间复杂度和搜索效率的平衡。

本文提出一种启发式算法，该算法结合了线性搜索过程和反向优化过程。文献[11－13]中的线性搜索算法，集中在$h(e)=\alpha w_1(e)+\beta w_2(e)$的搜索参数$\alpha$或$\beta$的调节上。但是，线性搜索算法本身的特点决定了它所能确定搜索到的可行解大约只有所有可行解的一半[11]。因此，本文提出了反向优化算法，当用$h(e)=w_1(e)+kw_2(e)$搜索失败后，不是用$kw_1(e)+w_2(e)$从源节点s开始搜索[13]，而是从用$w_1(e)+kw_2(e)$作为链路权重优化所得到的路径p中选出一个节点u根据不同情况用$w_1(e)$或$w_2(e)$作为权重计算该节点到目标节点t的最短路径$u\to t$，检测由原来p的路径段$s\to u$和$u\to t$连接组成的路径是否满足约束条件，然后进行下一步搜索，这一过程称为反向优化过程。本文提出了两种从路径p中选出反向优化起始节点u的方法并进行了仿真比较。反向优化算法的核心思想是：在某次线性搜索失败后，并不是抛弃所求得的路径重新搜索，而是认为该路径仍然是有价值的，因此选择对该路径根据不同情况用$w_1(e)$或$w_2(e)$作为权重进行局部优化。本文的仿真证明对于设计合理的网络如格形网络或ANSNET [13]，反向优化算法扩大了搜索空间，提高了搜索效率，同时时间复杂度仍为$O(K(m+n\log_2(n)))$。

2 BCP的反向优化算法

本文讨论BCP，其数学模型如下：

定义1 BCP，考虑网络由图$G=(V, E)$表示，V表示节点集，E表示链路集，E中的每一条链路$e\in E$都与两个QoS参数$w_k(e)$相联系，$k=1$，2。两个参数都非负且满足可加性。令D_k表示QoS约束条件，$k=1$，2。对于联结源节点s和目标节点t的路径p，若p满足$w_k(p)=\sum_{e\in p}w_k(e)\leqslant D_k$，$k=1, 2$，则称路径$p$满足QoS约束。

如果把其中一个约束条件看成代价(cost)，另一个看成延迟(delay)，联结源节点s和目标节点t的延迟约束为D，要求寻找(s, t)间的路径p，满足$\mathrm{delay}(p)=\sum_{e\in p}\mathrm{delay}(e)\leqslant D$，且$\mathrm{cost}(p)=\sum_{e\in p}\mathrm{cost}(e)$最小，则称为RSP。

说明 通过令所有链路的$w_k(e)$乘上$1/D_k$，$k=1$，2。可以使链路两个参数处于同一数量级，同时端到端约束$D_1=D_2=1$，对约束条件归一一般能提高算法的执行效率。此外，在表示算法的时间复杂度时，本文用n表示节点数，m表示链路数。

几乎所有关于BCP的近似算法都有一个缺点——优化过程存在倾向性，当对其中一个约束条件进行优化时，另一个约束条件的指标就变得恶劣。对于线性搜索算法具体有以下结论。

定理1 p与p^*分别为用$h(e)=w_1(e)+kw_2(e)$与$h^*(e)=w_1(e)+k^*w_2(e)$作为链路权重运用标准最短路径算法求得的路径，若$k<k^*$，则$w_1(p)<w_1(p^*)$且$w_2(p)>w_2(p^*)$。

证明 由已知条件知

$$w_1(p)+kw_2(p)<w_1(p^*)+kw_2(p^*) \tag{1}$$

$$w_1(p)+k^*w_2(p)>w_1(p^*)+k^*w_2(p^*) \tag{2}$$

式(1)×k^*－式(2)×k：$(k^*-k)w_1(p)<(k^*-k)w_1(p^*)$，由 $k<k^*$，得 $w_1(p)<w_1(p^*)$。

式(2)－式(1)：$(k^*-k)w_2(p)>(k^*-k)w_2(p^*)$，得

$w_2(p)>w_2(p^*)$。 证毕

该定理证明用 $w_1(e)+kw_2(e)$作为链路权重计算最短路径时，k 越大，该过程越倾向于优化第 2 个约束条件，所得路径的第 1 个参数值越差。反之 k 越小，用 $w_1(e)+kw_2(e)$作为链路权重计算最短路径时越倾向于优化第 1 个约束条件，所得路径的第 2 个参数值越差。事实上 $k<1$ 时倾向优化第 1 个约束条件，$k>1$ 时倾向优化第 2 个约束条件。当 $k=0$ 时或足够小时相当于只对第 1 个约束条件优化，当 $k=\infty$或足够大时相当于只对第 2 个约束条件优化。

关于BCP的BS算法的核心思想如下[13]：先用$h(e)= w_1(e)+kw_2(e)$作为链路权重运用标准最短路径算法搜索，若失败就用 $kw_1(e)+w_2(e)$进行第二次搜索，若仍然找不到可行路径就调整 k 值，算法设定 k 的范围为 $1\leqslant k\leqslant B$，$B=\max\{n\cdot w_k(e), k=1,2\}$。其时间复杂度为 $O(\log_2(B)(m+ n\log_2(n)))$，在实际运算过程中一般指定迭代次数为 K，因此其时间复杂度为 $O(K(m+n\log_2(n)))$，该算法兼顾了时间复杂度与搜索效率的平衡。该算法的局限性是当用 $h(e)=w_1(e)+kw_2(e)$搜索不成功时，则完全抛弃原来路径用 $kw_1(e)+w_2(e)$搜索，搜索得到的路径要么倾向于优化约束条件 1，要么倾向于约束条件 2。例如当 k 比较大时，从用 $h(e)=w_1(e)+kw_2(e)$作为链路权重转向 $kw_1(e)+w_2(e)$相当于从完全倾向于优化约束条件 2 转向完全优化约束条件 1，必然会影响算法的效率。

本文对此进行了改进，首先利用定理 1，采用了两分法搜索。与此同时，认为用 $h(e)=w_1(e)+kw_2(e)$作为权重优化所得的路径即使不满足约束条件也是有价值的。因此选择该路径的某个节点作为起始优化节点，保留前一段路径，对后一段根据不同情况针对某个参数进行优化再同前一段连接即可能得到可行路径，该过程称为反向优化过程。线性搜索算法本身的特点决定了它所能确定搜索到的可行解大约只有所有可行解的一半，某些情况下无论怎样调整搜索参数也不可能找到可行路径，这也是 BS 算法的局限性。通过反向优化过程，扩大了搜索空间，提高了搜索到可行路径的成功率。

在叙述算法之前，需要建立以下两个定义。

定义 2 路径的中间节点：对路径 $p=s\to t$ 从源节点 s（标号为 0）起进行标号，分别为 $0,1,2,\cdots,|p|$，$|p|$为路径 p 的跳数，即 t 的标号，p 的中间节点是标号为$[|p|/2]$的节点（$[\cdots]$是高斯取整符号）。

定义 3 路径 p 标记为$\{s, a_1, a_2,\cdots, u\}$，$s=a_0$，$u=a_{|p|}$，路径 q 标记为$\{u, b_1, b_2,\cdots, t\}$，$u=b_0$，$t=b_{|q|}$；$|p|$与$|q|$为路径 p 与 q 的跳数，(s, t)间的路径 $r=p\oplus q$ 的定义如下，令 $i=\max\{i, b_i\in p\}$，即路径 p 与 q 的公共节点中在路径 q 中标号最大的节点。$r=p\oplus q$ 定义为路径 p 的 $s\to b_i$ 段以及路径 q 的 $b_i\to t$ 段连接而成的路径，称 r 为 p 与 q 的连接。

该定义保证了由 p 与 q 连接而成的路径 $r=p\oplus q$ 不包含圈。

例 1 如图 1 所示，$p_1=\{s=a_0, a_1, a_2, a_3, a_4=u\}$，$q_1=\{u=b_0, b_7, b_8, b_9, t=b_{10}\}$，$q_2=\{u=b_0, b_9, t=b_{10}\}$，$q_3=\{u=b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, t=b_{10}\}$。

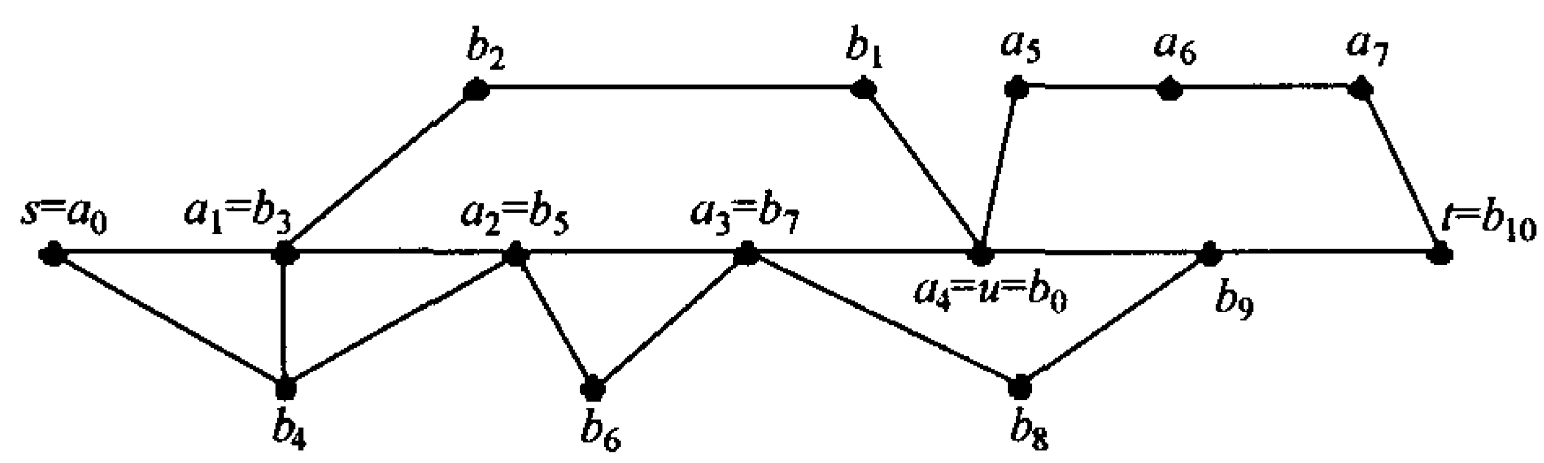

图 1 $r=p\oplus q$ 示例

p_1 与 q_1 的公共节点有 1 个，为 $a_3=b_7$，$r_1=p_1\oplus q_1=\{s=a_0, a_1, a_2, a_3=b_7, b_8, b_9, t=b_{10}\}$。

p_1 与 q_2 没有公共节点，$r_2=p_1\oplus q_2=\{s=a_0, a_1, a_2, a_3, u, b_9, t=b_{10}\}$。

p_1 与 q_3 的公共节点有 3 个，分别为 $a_1=b_3$，$a_2=b_5$，$a_3=b_7$，其中在 q_3 中标号最大的为 b_7，即 $\max\{i, b_i\in p\}=7$，则有 $r_3=p_1\oplus q_3=\{s=a_0, a_1, a_2, a_3=b_7, b_8, b_9, t=b_{10}\}$。

反向优化算法 I

(1) 令 $B=\max\{n\times w_k(e), k=1, 2\}$，$k0=1/B$，$k1=B$，$k=1$，以 $w_1(e)+kw_2(e)$作为链路权重运用最短路径算法求出路径 p，若 p 满足 $w_1(p)\leqslant D_1$ 且 $w_2(p)\leqslant D_2$ 则输出可行解 p，若 p 满足 $w_1(p)>D_1$ 且 $w_2(p)>D_2$ 则宣布无解。若 $w_1(p)>D_1$ 且 $w_2(p)\leqslant D_2$ 转(4)，若 $w_1(p)\leqslant D_1$ 且 $w_2(p)>D_2$ 转（5）。

(2) $k1=k$，$k=(k0+k)/2$，以 $w_1(e)+kw_2(e)$作为链路权重运用最短路径算法求出路径 p，若 p 满足 $w_1(p)\leqslant D_1$ 且 $w_2(p)\leqslant D_2$ 则输出可行解 p，若 $w_1(p)>D_1$ 且 $w_2(p)\leqslant D_2$ 转(4)，若 $w_1(p)\leqslant D_1$ 且 $w_2(p)>D_2$ 转(5)。(注：此时不可能出现两个约束条件都不满足的情况)。

(3) $k0=k$，$k=(k1+k)/2$，以 $w_1(e)+kw_2(e)$作为链路权重运用最短路径算法求出路径 p，若 p 满足 $w_1(p)\leqslant D_1$ 且 $w_2(p)\leqslant D_2$ 则输出可行解 p，若 $w_1(p)>D_1$ 且 $w_2(p)\leqslant D_2$ 转(4)，若 $w_1(p)\leqslant D_1$ 且 $w_2(p)>D_2$ 转(5)。(注：此时不可能出现两个约束条件都不满足的情况)。

(4) (反向优化过程)对 p 作如下处理：取 p 的“中间节点” u，以 $w_1(e)$作链路参数求最短路径 $p_2=u\to t$，令路径 p 的 $s\to u$ 段为 p_1，令 $r=p_1\oplus p_2$，检测 r 是否满足约束条件，若满足则输出所得路径，否则若 $w_1(p)>D_1$ 且 $w_2(p)\leqslant D_2$ 转(2)，若 $w_1(p)\leqslant D_1$ 且 $w_2(p)>D_2$ 转(3)。

(5) (反向优化过程)对 p 作如下处理：取 p 的“中间节点” u，以 $w_2(e)$作链路参数求最短路径 $p_2=u\to t$，令路径 p 的 $s\to u$ 段为 p_1，令 $r=p_1\oplus p_2$，检测 r 是否满足约束条件，若满足则输出所得路径，否则若 $w_1(p)>D_1$ 且 $w_2(p)\leqslant D_2$ 转(2)，若 $w_1(p)\leqslant D_1$ 且 $w_2(p)>D_2$ 转(3)。

(6) 在规定迭代次数达到后输出最后结果或宣布无解。

反向优化算法的主要特点是：(1) 针对搜索机制，在每一次迭代过程中，BS 运用 $h(e)=w_1(e)+kw_2(e)$或 $kw_1(e)+w_2(e)$作为链路权重进行双重搜索，$k\in[1, B=\max\{n\times w_k(e), k=1, 2\}]$，反向优化算法运用 $h(e)=w_1(e)+kw_2(e)$进行两分法搜索，$k\in[1/B, B]$。根据定理 1，假定搜索到的路径为 p，如果 p 不满足约束条件，则进行反向优化过程，若还得不到可行解，此时如果 $w_1(p)>D_1$，则减小 k 值进行搜索（搜索倾向优化第一个约束条件），若 $w_2(p)>D_2$，则增大 k 值进行搜索（搜索倾向优化第二个约束条件）以及继续反向优化。在迭代过程中遇到可行路径就终止。(2) 针对反向优化过程：由于线性搜索的局限性，单纯的线性搜索很可能永远无法找到可行解。但线性搜索失败后，本文认为所得的路径 p 仍然是有价值的，因此不像 BS 算法那样抛弃 p 重新搜索，而是认为对它们进行适当的优化就有可能得到可行解。例如对于线性搜索得到的路径 p，若 p 满足 $w_1(p)\leqslant D_1$，$w_2(p)>D_2$。由于 $w_2(p)>D_2$，路径 p 不满足第 2 个约束条件，因此对它进行适当修正，从 p 的中间节点 u 起对路径 p 的后半段对第 2 个参数进行优化求出路径 $u\to t$ 并与路径 p 的前半段重新连接组合为一条新路径，即由路径 p 的 $s\to u$ 段和 $u\to t$ 连接就可能得到可行路径。

在反向优化过程中，算法选择路径 p 的中间节点 u 作为反向优化起点，本文还考虑了以下两种选点方式，(1) 在路径 p 的除了 t 的所有节点中随机选取反向优化起点；(2) 在标号为（$[|p|/2],\cdots,|p|-1$）的节点中随机选取，即不包括 t 的后半段节点中随机选取。本文的测试发现前一种优化方法无优势可言，后一种对较大的图效果较好。本文把采用后一种反向优化选点方式的算法称为反向优化算法Ⅱ。

在算法执行过程中，对约束条件归一可以提高运算效率，反向优化算法Ⅰ与Ⅱ以及 BS 算法都是如此，具体做法是对通过令所有链路的 $w_k(e)$乘上 $1/D_k$，$k=1$，2。使链路两个参数处于同一数量级，同时 $D_1=D_2=1$，k 的范围变为$[1/n, n]$。显然反向优化算法Ⅰ和 Ⅱ 的时间复杂度仍为 $O(\log_2(B)(m+n\log_2(n)))$，在指定搜索次数 K 的情况下为 $O(K(m+n\log_2(n)))$。

3　实验测试

本文的测试采用用 8×8，20×20 格形图和 ANSNET（图 2），对于所有链路的参数 w_1 和 w_2，我们分配以 1 到 42 之间的随机正整数，对反向优化算法Ⅰ，Ⅱ以及 BS 算法进行了 1000 次测试，迭代次数 $K=3$。测试结果如图 3，图 4 和图 5 所示。

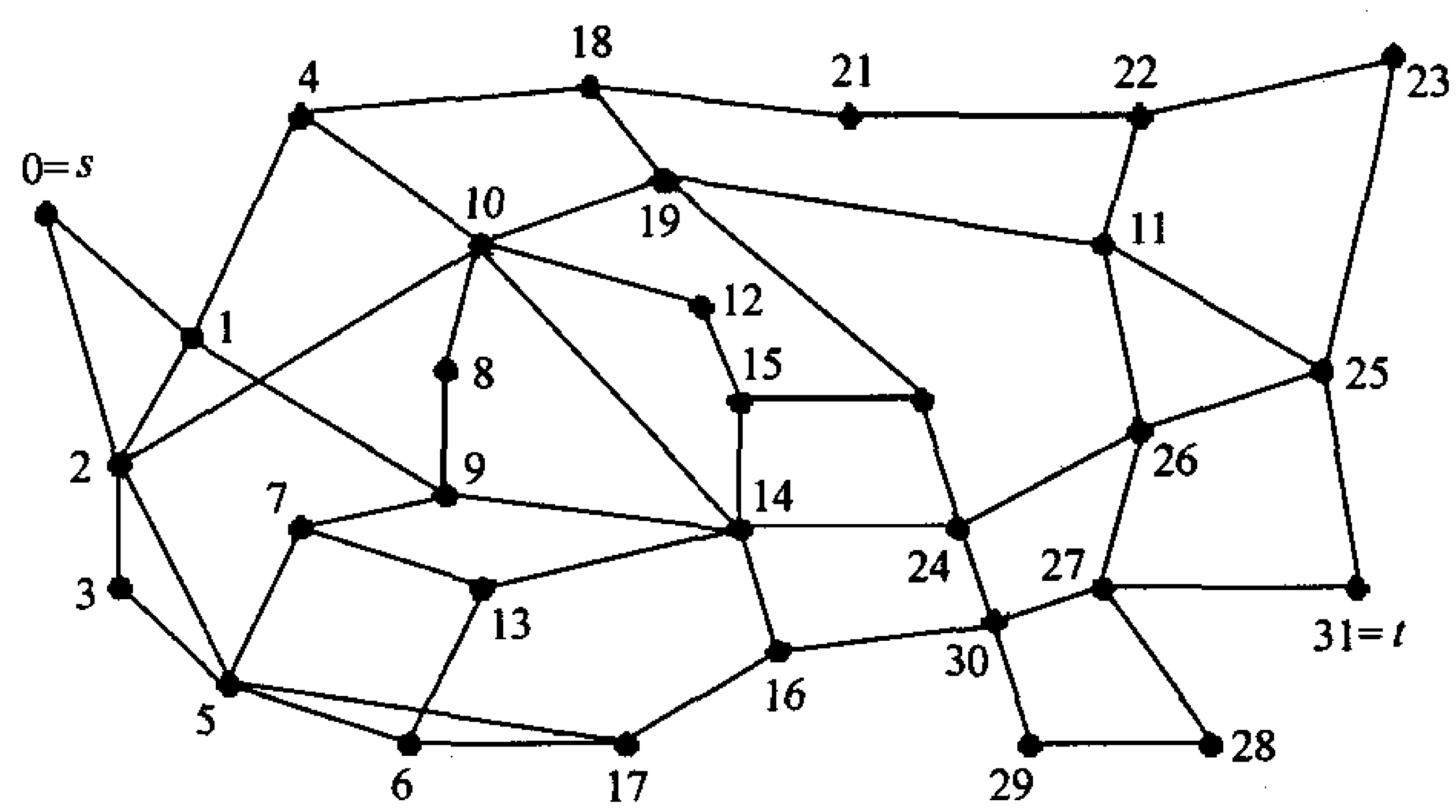

图 2　ANSNET

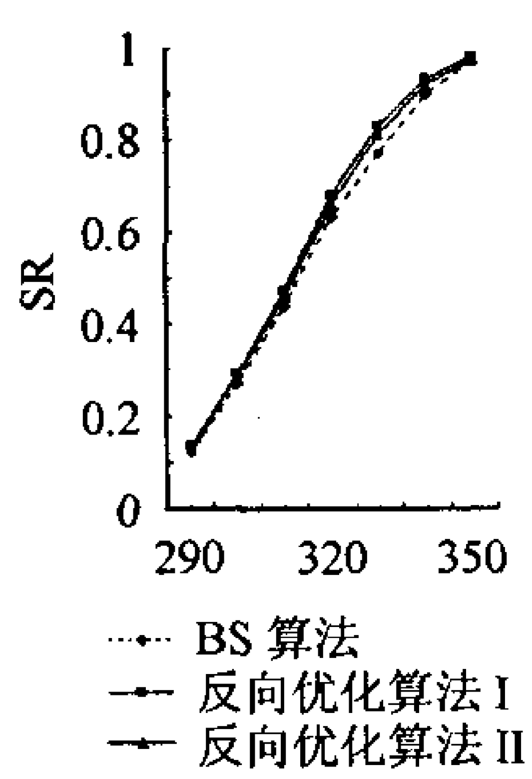

图 3 算法比较
(8×8 格形图)

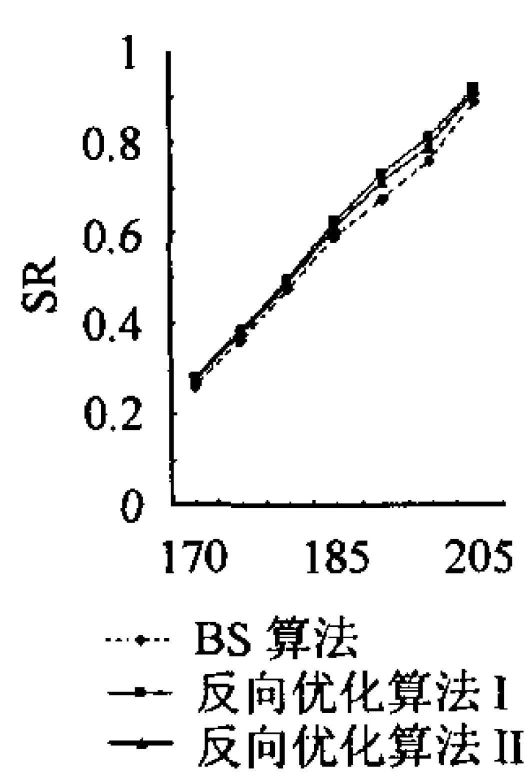

图 4 算法比较
(ANSNET)

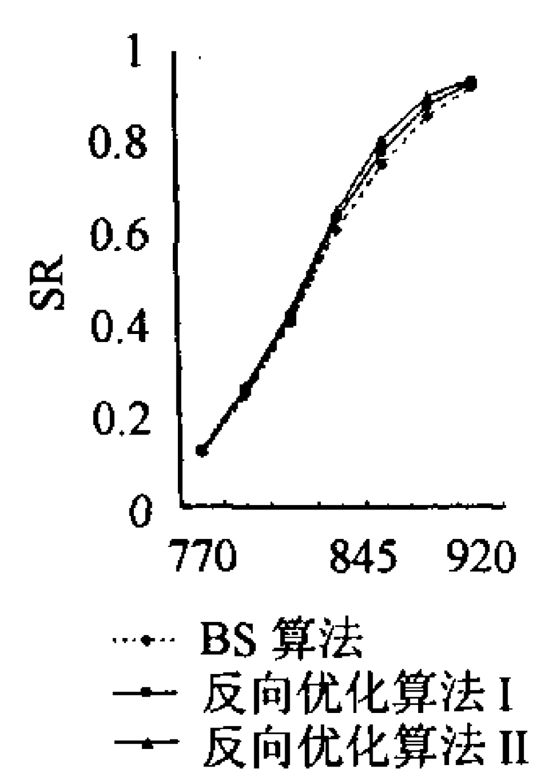

图 5 算法比较
(20×20 格形图)

说明：图 3，图 4 和图 5 中，横轴代表约束条件，假定 $D_1=D_2$。纵轴代表找到可行路径的成功率(Succeeding Ratio, SR)。

结果表明反向优化算法Ⅰ和Ⅱ都好于 BS 算法，对 8×8 格形图和 ANSNET 反向优化算法Ⅰ略好于Ⅱ，而对 20×20 格形图反向优化算法Ⅱ略好于Ⅰ。这些结果是可以预料的，因为对线性搜索求得的路径 p 而言，虽然不满足约束条件，但由于其经过优化，可能已经很接近可行路径了，因此对其进行部分微调即可能得到可行路径。反向优化算法Ⅱ相对于反向优化算法Ⅰ调整得更小，但若图比较小，p 的后半段节点集($[|p|/2],\cdots,|p|-1$)数量过少，则调整达不到目的，但若图比较大如 20×20 格形图，反向优化算法Ⅱ的在路径 p 的后

半段随机选取优化起点的方式相比反向优化算法 I 选取中间节点作为优化起点的方式就有一定优势。此外，综合来看，反向优化算法 I 和 II 都优于 BS 算法，但在约束条件较紧或较松时优势不明显，在一般情况下总体大致比 BS 算法好 5% 左右。

4 结论

本文对 QoS 路由问题的核心问题即满足两个加性约束的路由问题进行了研究，提出了一种结合了反向优化过程的线性搜索方法。该方法的核心思想是在线性搜索无法找到可行路径时，对搜索到的路径选取合适的节点进行反向优化。同双重搜索算法 BS 相比，反向优化算法利用了线性搜索所得路径的价值。实验测试表明反向优化算法增大了搜索空间，提高了运算效率，其时间复杂度仍为 $O(K(m+n\log_2(n)))$。

参 考 文 献

[1] Crawley E, Nair R, Rajagopalan B, Sandick H. A framework for QoS-based Routing in the internet. Internet Draft, RFC2386, 1998.

[2] Guerin R, kamat S, Orda A, Przygienda T, Williams D. QoS routing mechanisms and OSPF extensions. Internet Draft, RFC2676, 1998.

[3] Wang Z, Crowcroft J. QoS routing for supporting resource reservation. *IEEE J. on SAC*, 1996, 14(9): 1228 – 1234.

[4] Sobrinho J. Algebra and algorithms for path computation and hop-by-hop routing in the internet. *IEEE/ACM Trans. on the Networking*, 2002, 10(4): 541 – 551.

[5] Garey M, Johnson D. Computers and Intractability: A Guide to the Theory of NP Completeness.[M]. San. Francisco, New York: W. H. Freeman, 1979, chp. 4.

[6] Chen S, Nahrstedt K. An overview of quality of service routing for next-generation high-speed networks: Problems and Solutions. *IEEE Network*, 1998, 12(6): 64 – 79.

[7] Warburton A. Approximation of Pareto optima in multiple-objective, shortest path problems. *Operations Research*, 1987, 35(1): 70 – 79.

[8] Hassin R. Approximation schemes for the restricted shortest path problems. *Mathematics Operations Research*, 1992, 17(1): 136 – 142.

[9] Yuan.X. Heuristic algorithms for multiconstrained quality-of-service routing. *IEEE/ACM Trans. on the Networking*, 2002, 10(2): 244 – 256.

[10] Cheng S, Nahrstedt K. On finding multi-constrained paths [A]. IEEE ICC'98[C], Atlanta, GA, 1998: 874 – 879.

[11] Jaffe J. Algorithms for finding paths with multiple constraints. *Networks*, 1984, (14): 95 – 116.

[12] Neve H, Mieghem P. A multiple quality of service routing algorithm for PNNI.[A] Proceedings IEEE ATM Workshop[C], May 26-29, Fairfax, VA, 1998: 324 – 328.

[13] Korkmaz T, Krunz M. An efficient algorithm for finding a path subject to two additive constraints [J]. *Computer Communications Journal*, 2002, 25(3): 225 – 238.

张 晶： 男，1971 年生，博士生，主要研究方向为网络 QoS 问题、WDM 光网络等.

李乐民： 男，1932 年生，教授，博士生导师，中国工程院院士，主要研究方向为宽带网络技术、IP 网络、WDM 光网络等.

王 晟： 男，1971 年生，博士，副教授，主要研究方向为 IP 网络技术、WDM 光网络等.