

## 多加性 QoS 约束下的链路分离路由算法

熊轲<sup>1,2</sup>, 裘正定<sup>1</sup>, 张煜<sup>1</sup>, 张宏科<sup>3</sup>

(1. 北京交通大学 计算机与信息技术学院, 北京 100044; 2. 清华大学 电子工程系, 北京 100084;

3. 北京交通大学 电子信息工程学院, 北京 100044)

**摘要:** 对多个加性 QoS 约束下的链路分离路径问题进行了研究, 针对现有算法求解结果依赖于网络结构, 难以保证对任意网络都可求得可行解和最优解的问题, 提出了一种与网络结构无关的多约束链路分离路径路由算法 (MCLPRA, multiple constrained link-disjoint path routing algorithm)。该算法基于 SAMCRA, 采用对解空间先分类, 然后按类进行处理和搜索的方法, 引入了控制搜索深度的参数, 可保证对任意网络都能求得可行解。理论分析表明, MCLPRA 能够在现有算法不能求解的情况下解得可行解和最优解。仿真结果显示, MCLPRA 的可行解平均求解成功率明显高于现有算法且所求路径对长度也比现有算法更短。

**关键词:** 服务质量路由; 链路分离; 可靠路由; 多约束路由

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2010)06-0127-09

## Link-disjoint routing algorithm under multiple additive QoS constraints

XIONG Ke<sup>1,2</sup>, QIU Zheng-ding<sup>1</sup>, ZHANG Yu<sup>1</sup>, ZHANG Hong-ke<sup>3</sup>

(1. School of Computer & Information Technology, Beijing Jiaotong University, Beijing 100044, China;

2. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China;

3. School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** The problem of finding link-disjoint paths under multiple additive QoS constraints was studied. Since the existing algorithms depended on network's structure and could not guarantee to find feasible solutions for arbitrary networks, a novel algorithm called multiple constrained link-disjoint path routing algorithm (MCLPRA) was proposed. MCLPRA was based on SAMCRA and didn't rely on the network's structure. By introducing the parameter to control its search depth, dividing the solution space into different classes and performing searching according to the classes respectively, MCLPRA was able to obtain the feasible solutions for arbitrary networks. Theoretic analysis shows that MCLPRA can get the feasible and optimal solutions when traditional schemes can not. Comprehensive simulations also show that MCLPRA has better performances than existing algorithms in terms of higher average successful rate of getting feasible solutions with shorter average total length of the obtained path pair.

**Key words:** QoS routing; link-disjoint; reliable routing; multi-constrained routing

收稿日期: 2008-04-09; 修回日期: 2010-04-20

基金项目: 国家重点基础研究发展计划 (“973” 计划) 基金资助项目 (2007CB307101); 教育部科技创新工程重大项目培育资金项目 (706005); 高等学校学科创新引智计划 (“111 计划”) 基金资助项目 (B08002)

**Foundation Items:** The National Basic Research Program of China (973 program) (2007CB307101); Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (706005); The Program of Introducing Talents of Discipline to Universities (111 Program) (B08002)

### 1 引言

网络多媒体业务和实时性业务的飞速增长，迫切要求网络为其提供相应的 QoS 保障。与此同时，光交换技术和传输技术的发展使得网络的传输速率成倍增长，在高速网络中即使很短暂的链路失效，也会造成数据包的大量丢失和网络 QoS 的急剧下降。如何提高网络可靠性和 QoS 恢复能力已成为网络研究的重要课题<sup>[1~3]</sup>。

在通信的源和目的节点间寻找满足 QoS 约束的 2 条分离路径，一条作为主用，另一条作为备用。在主用路径失效时，将业务流迅速切换至备用路径传输，被认为是可同时提高网络可靠性和 QoS 保障的重要方法<sup>[1,2,4~8]</sup>。多约束分离路径亦十分有利于网络流量工程、负载均衡和拥塞避免的实施。

通常 QoS 约束可分为链路约束和路径约束 2 类。链路约束是对单条链路的约束，如带宽约束，属于凹性约束；路径约束则指对传输路径上所有链路 QoS 参数端到端总和的约束，如花费(cost)、延时(delay)等约束，属于加性约束。链路约束处理起来相对简单，只需在路径计算前将不满足约束条件的链路删去即可<sup>[8]</sup>。路径约束则需要将路径上端到端的参数叠加起来考虑，因此要复杂的多。当路径约束个数大于 2 时，路由问题就变 NP 问题<sup>[9]</sup>。分离路径包括链路分离和节点分离 2 类。前者要求路径间无共用链路，后者要求路径间无共用节点。节点分离一定为链路分离，反之则不成立。因此讨论链路分离路径是研究分离路径问题的基础。

本文主要研究多个加性 QoS 约束下的链路分离路径算法，旨在在一对通信的源和目的节点间寻找 2 条链路分离且满足多个加性 QoS 约束的路径，并提出了与网络结构无关的多约束链路分离路径路由算法(MCLPRA, multiple constrained link-disjoint path routing algorithm)，该算法基于多约束下的最短路径精确算法(SAMCRA)<sup>[9,10]</sup>，在解的搜索过程中，首先对解空间按照解的不同构成形式进行分类，然后按类进行搜索处理；引入了控制搜索精度和深度的控制参数，能够保证对任意结构网络求得可行解。

### 2 问题描述及研究现状

将网络抽象为加权有向图，记为  $G(V, E)$ 。网

络中的路由设备和通信链路分别用图中的节点和有向边表示。 $V$  代表  $G$  中节点的集合， $E$  代表  $G$  中有向边的集合， $(u, v)$  表示从节点  $u$  到节点  $v$  的一条有向边。 $G$  中每条边都带有  $m$  维的加性权重向量  $W(u, v) = [w_1(u, v), w_2(u, v), \dots, w_m(u, v)]$ 。用  $s$  和  $t$  分别代表通信的源和目的节点， $P$  为从  $s$  到  $t$  的一条路径， $w_i(P)$  表示  $P$  的第  $i$  维权重，路径  $P$  的权向量  $W(P) = [w_1(P), w_2(P), \dots, w_m(P)]$ ，其中

$$w_i(P) = \sum_{(u,v) \in P} w_i(u, v), \quad i = 1, \dots, m \quad (1)$$

用  $E(P) = \{(u, v) | (u, v) \in P\}$  代表路径  $P$  上的边的集合，用  $V(P)$  代表路径  $P$  上的节点集合， $V(P) = \{u | u, v \in P \text{ 或 } v, u \in P\}$ 。

**定义 1** 非线性长度。给定带  $m$  维权重的加权有向图  $G(V, E)$  和  $m$  维约束向量  $C$ ， $C = [c_1, c_2, \dots, c_m]$ ， $G$  中路径  $P$  的非线性长度定义为<sup>[9,10]</sup>

$$l(P) = \max_{1 \leq i \leq m} \left( \frac{w_i(P)}{c_i} \right) \quad (2)$$

非线性长度是归一化的长度。当  $l(P) > 1.0$  时，路径  $P$  至少有一维权重超出了约束  $C$ 。

**定义 2** 链路分离路径(link-disjoint path)假设  $P_1$  和  $P_2$  为  $G(V, E)$  中的 2 条路径，若  $E(P_1) \cap E(P_2) = \emptyset$ ，则  $P_1$  与  $P_2$  为链路分离路径。

**定义 3** 多约束链路分离路径对(MCLPP, multiple constrained link-disjoint path pair)问题。给定一个加权有向图  $G(V, E)$  和一个  $m$  维的约束向量  $C = [c_1, c_2, \dots, c_m]$ ， $m \geq 2$ 。MCLPP 问题的目的是在源  $s$  和目的  $t$  节点之间找到 2 条路径  $P_1$  与  $P_2$ ，要求  $E(P_1) \cap E(P_2) = \emptyset$ ，且  $P_1$  与  $P_2$  均满足约束  $C$ ，即  $w_i(P_1) \leq c_i, w_i(P_2) \leq c_i, 1 \leq i \leq m$ 。

实际上，在  $s$  和  $t$  间可能同时存在多条满足约束  $C$  的链路分离路径对，往往希望找到路径总长最小的一对。当  $l(P_1) + l(P_2)$  为最小时，定义  $\{P_1, P_2\}$  为 MCLPP 的最优解。

近年来，多 QoS 约束链路分离路由问题受到了研究者的广泛关注，然而新的研究基本都是针对 delay 单个约束的，旨在找出一对链路分离路径，在满足 delay 约束的前提下达到 2 条路径总的 cost 最小。当给定的 delay 约束针对的是路径对中的每条路径的延时，问题被称为 DCLDOP-I，当 delay 约束针对的是路径对中 2 条路径延时的总和时，问题被称为 DCLDOP-II。文献[13]对 DCLDOP-I 和

DCLDOP-II 问题进行了建模,证明了这 2 种问题都属于 NP 完全问题。文献[14]提出了针对这 2 种问题的近似求解算法。文献[15]对 DCLDOP-II 问题进行了研究,提出了单条链路失效下的路径恢复算法。文献[16]研究了 Min-Min 问题,旨在求解 2 条满足 QoS 约束的分离路径,且实现较短的路径 cost 最小。文献[17]采用了将 2 个加性权值进行线性组合的方法来求解满足 QoS 约束的分离路径,并给出了求解算法,所提算法虽然简单,却不能保证一定能求得可行解。

MCLPP 问题由文献[11,12]提出并证明了 MCLPP 为 NP 完全问题,分析了多约束链路分离路径问题与单约束链路分离路径问题的不同,且给出了启发式求解算法 DIMCRA。本文在文献[11,12]的基础上,对 MCLPP 进行了进一步的研究,旨在通信的源和目的节点间找到一对链路分离路径,且每条路径都满足 2 个或 2 个以上的 QoS 约束。

目前求解 MCLPP 问题的最直观的算法为 RF 算法。其原理是:先计算一条满足 QoS 约束的最短路径,然后删去该路径上的所有链路的修正图,再在修正图中解得另一条最短路径。RF 算法能保证所求得的 2 条路径为链路分离路径,但由于在求解过程中删去了第一条路径上的链路,破坏了原网络的连通性,导致很多情况下得不到最优解,甚至连可行解也得不到。

DIMCRA<sup>[10,11]</sup>算法由单约束的链路分离路径算法 LBA<sup>[10,11]</sup>演化而来,对 RF 算法有一定的改进。给定  $G(V,E)$  和约束  $C$ ,其步骤如下。第 1 步:执行 SAMCRA,寻找 1 条最短路径  $P_1$ ;第 2 步:将  $P_1$  的所有链路反向并置其权重为 0,得修正图  $G'(V,E)$ ;第 3 步:在  $G'(V,E)$  中执行 SAMCRA,寻找 1 条满足约束  $2C$  的最短路径  $P_2$ ,若  $P_2$  不存在,算法停止;第 4 步:取  $P_1$  和  $P_2$  的并集,删去反向链路出现在  $P_1$  上的  $P_2$  的链路和反向链路出现在  $P_2$  上的  $P_1$  的链路,将余下的链路组成 2 条路径  $\{P'_1, P'_2\}$ ;第 5 步:检查集合  $\{P'_1, P'_2\}$  中的每条路径,若路径  $P'_i$  ( $i=1,2$ ) 不满足约束,则从修正图  $G'(V,E)$  中删去  $P'_i$  与  $P_1$  未发生交叠的链路集合,得到更新的修正图  $G''(V,E)$ ,返回第 3 步;否则,算法停止。

与 RF 一样, DIMCRA 也能保证所求解一定满足链路分离,但由于在解搜索过程中采用了删除不满足约束条件的路径上的部分或全部链路的方法,

见算法第 5 步,亦破坏了网络的连通性。因此 DIMCRA 同样不能保证对任意结构的网络都可求得可行解和最优解,如本文第 5 节的算例 3 和算例 5。

### 3 MCLPRA

针对上述问题,本文提出了 MCLPRA。与 DIMCRA 算法相似, MCLPRA 也以 SAMCRA 为基础,充分利用了 SAMCRA 的几个特点:①基于非线性长度;②  $k$ -shortest path 特性,即在求解最短路径的搜索过程中,不仅可以得到最短路径,也可以得到第 2 短路径,……,第  $k$  短路径;③多约束路由最短路径的无环路精确算法。

MCLPRA 采用先将解空间分类,然后按类进行搜索处理的方法。在算法处理过程中,保持了原网络的结构和连通性,确保了解空间的完整性,引入了搜索深度控制参数,能够保障对任意网络求得可行解。需要补充说明的是, DIMCRA 的第 3 步仅求得了最短的一条  $P_2$ ,而 MCLPRA 利用了 SAMCRA 的  $k$ -shortest path 特性,对 SAMCRA 进行了修改,使其在一次运行过程中可将满足约束的所有  $P_2$  路径记录下来,然后将这些路径按类处理,进行解的搜索。

#### 3.1 MCLPRA 的求解步骤

下面给出 MCLPRA 的求解步骤,给定加权有向图  $G(V,E)$  和约束向量  $C$ 。

第 1 步:执行 SAMCRA,求 1 条最短路径  $P_1$ ;

第 2 步:将  $P_1$  的所有链路反向并置其权重为 0,得修正图  $G'(V,E)$ ;

第 3 步:在修正图  $G'(V,E)$  中运行 SAMCRA,计算出从  $s$  到  $t$  的满足约束向量  $2C$  的  $k$  条最短路径集合  $S = \{P'_1, \dots, P'_k\}$ ,如果  $S$  为空,算法停止;

第 4 步:将  $S$  按如下规则分成 2 个子集:

$$\begin{cases} S_\alpha = \{P \mid E(P) \cap E(P_1) = \emptyset \text{ 且} \\ \quad V(P) \cap \{V(P_1) - s - t\} = \emptyset, P \in S\} \\ S_\beta = S - S_\alpha \end{cases}$$

第 5 步:初始化空路径  $P_i^* = null$ ,  $i=1,2$ ,  $l(null)=1.0$ 。寻找  $S_\alpha$  中的最短路径  $P_\alpha$ ,调用  $Search\_S_\beta(P_1, S_\beta, T)$  搜索  $S_\beta$  与  $P_1$  构成的满足  $C$  的最优解  $\{P_1^*, P_2^*\}$ 。若  $\{P_1, P_\alpha\}$  不满足  $C$ ,则  $\{P_1^*, P_2^*\}$  为最终解;若  $\{P_1, P_\alpha\}$  满足  $C$ ,则选取  $\{P_1, P_\alpha\}$  和  $\{P_1^*, P_2^*\}$  中非线性长度和最小的一组作为

MCLPRA 的最终解。

下面给出  $Search\_S_\beta(P_1, S_\beta, T)$  的处理过程, 其中  $T$  为搜索深度控制参数。

Initialization

$P_a^* = null; P_b^* = null;$

Loop

while ( $S_\beta \neq \emptyset$  and  $T > 0$ )

{

Find the shortest path of  $S_\beta$ ;

if  $E(P_\beta) \cap E(P_1) \neq \emptyset$

{ $P'_a, P'_b$ } = Search\_in\_Linkjoint ( $P_1, P_\beta$ );

else /\*  $P_\beta$  和  $P_1$  为链路分离

{

if  $V(P_\beta) \cap \{V(P_1) - s - t\} = \emptyset$

{/\*  $P_\beta$  和  $P_1$  除源、目的地外无其他共用节点

if  $P_\beta$  和  $P_1$  均满足约束向量  $C$

{ $P'_a = P_1; P'_b = P_\beta$ }

else

{ $P'_a = Null; P'_b = Null;$ }

}

else /\*  $P_\beta$  和  $P_1$  虽链路分离但有共用节点

{ $P'_a, P'_b$ } = Search\_in\_nodeshared( $P_1, P_\beta$ );

}

if  $l(P'_a) + l(P'_b) < l(P_a^*) + l(P_b^*)$

{/\*如果  $S_\beta$  中当前最短路径  $P_\beta$  与  $P_1$  可以构成路径总长更小的解, 就更新  $P_a^*$  和  $P_b^*$

$P_a^* = P'_a; P_b^* = P'_b;$

}

$S_\beta = S_\beta - P_\beta$ ; /\*从  $S_\beta$  去掉当前已处理的路径

$T = T - 1$ ;

}

Return { $P_a^*, P_b^*$ } /\*返回  $S_\beta$  中的最优解

接下来介绍  $Search\_in\_Linkjoint(P_a, P_b)$  的步骤。功能是从有共用链路的 2 条路径  $P_a, P_b$  的链路所构成的集合中搜索满足约束的可行解。

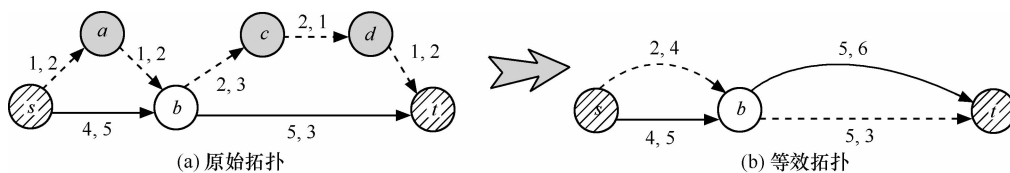


图 1 等效变换

第 1 步: 取  $P_a$  和  $P_b$  的并集, 从中删去反向链路出现在  $P_a$  上的  $P_b$  链路和反向链路出现在  $P_b$  上的  $P_a$  上的链路, 将余下的链路组成 2 条路径  $\{P_x, P_w\}$ ;

第 2 步:

if  $V(P_x) \cap \{V(P_w) - s - t\} = V(E(P_x) \cap E(P_w)) \neq \emptyset$

{/\*  $P_x$  和  $P_w$  除源、目的节点外只共享共用链路上的节点

if  $P_x$  和  $P_w$  均满足约束向量  $C$

Return { $P_x, P_w$ };

else

Return {null, null};

}

else

{/\*  $P_x$  和  $P_w$  除源、目的节点外, 不只是共享共用链路上的节点

Return Search\_in\_nodeshared ( $P_x, P_w$ );

}

最后介绍  $Search\_in\_nodeshared(P_a, P_b)$  的步骤。功能是从有共用节点的 2 条链路分离路径  $P_a, P_b$  所构成的链路集合中, 找到满足约束  $C$  且长度和最小的 2 条链路分离路径。

第 1 步: 取  $P_a$  和  $P_b$  的并集, 由  $P_a$  和  $P_b$  的链路构成一个  $G(V, E)$  的子图  $G_{sub}$ ;

第 2 步: 对  $G_{sub}$  做如图 1 所示的等效变换, 在等效图中找到由  $G_{sub}$  链路构成的 2 条满足约束的链路分离路径  $\{\tilde{P}_a, \tilde{P}_b\}$ , 且  $l(\tilde{P}_a) + l(\tilde{P}_b)$  为最小。

对图 1(a) 的结构, 通过合并出度与入度都为 1 的节点相连接的链路, 可得到等效图 1(b), 在图 1(b) 的结构中易找到  $s$  和  $t$  之间的非线性长度和最小的链路分离路径对, 图 1(b) 中为  $\{sabt, sbcdt\}$ 。

### 3.2 MCLPRA 的求解过程

下面通过算例说明 MCLPRA 的求解过程。

**算例 1** 如图 2(a) 所示网络,  $C = (10, 10)$ 。目的是在源节点  $s$  和目的节点  $t$  间寻找一对满足  $C$  约束条件的链路分离路径。第 1 步在图 2(a) 上运行 SAMCRA 得到最短路径  $P_1$  为  $sabt$ 。第 2 步如图 2(b) 所示, 将  $sabt$  上的链路反向, 并且重置链路权重为

0. 第 3 步在图 2(b)上运行 SAMCRA, 求得满足  $2C$  约束的路径集合  $S = \{sdt, sbat\}$ 。第 4 步对  $S$  进行分类,  $sbat \in S_\beta$ ,  $sdt \in S_\alpha$ 。第 5 步先查找  $S_\alpha$  中的最短路径,  $S_\alpha$  中只有一条路径  $sdt$ , 故  $P_\alpha = sdt$ 。然后调用  $Search\_S_\beta()$  搜索  $S_\beta$  构成的可行解中的最优解。取  $S_\beta$  中的最短路径  $sbat$ , 由于  $sbat$  与  $P_1$  有共用链路, 所以调用  $Search\_in\_Linkjoint()$  处理, 从  $sbat$  和  $sbat$  链路构成的并集中去掉共用链路  $ab$ , 剩余链路构成 2 条链路分离路径  $\{sat, sbt\}$ , 经判断  $sat$  与  $sbt$  均满足约束  $C$ , 故  $\{sat, sbt\}$  为  $S_\beta$  与  $P_1$  构成的最优解。因为  $P_1$  和  $sdt$  均满足约束  $C$  且  $l(sat) + l(sbt) < l(sbat) + l(sdt)$ , 故  $\{sat, sbt\}$  为最终解。显然, 该解为图 2(a)上的最优解。

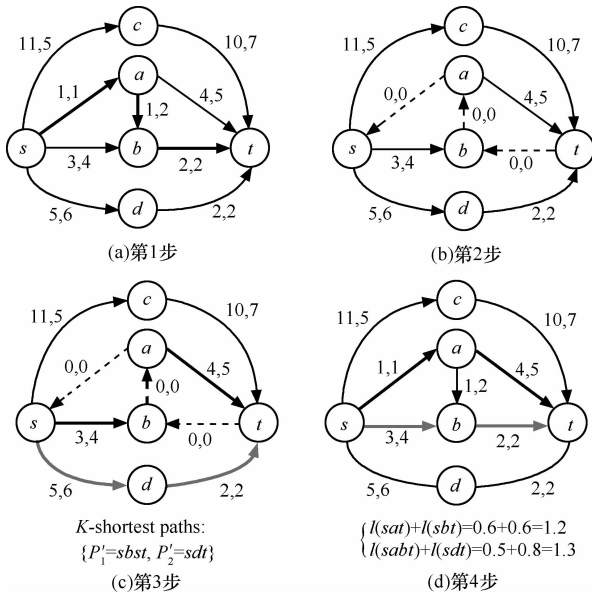


图 2 MCLPRA 算例 1

**算例 2** 如图 3(a)所示网络,  $C = (10, 10)$ 。MCLPRA 第 1 步是在图 3(a)上运行 SAMCRA 得到最短路径  $P_1$  为  $sabct$ 。第 2 步如图 3(b)所示, 将  $sabct$  上的链路反向, 重置链路权重为 0。第 3 步在图 3(b)上运行 SAMCRA, 求得满足  $2C$  约束的路径集合  $S$ , 求解结果只有一条路径  $sbt$  满足要求。第 4 步对  $S$  进行分类,  $sbt$  属于  $S_\beta$ ,  $S_\alpha$  为空。第 5 步搜索  $S_\beta$ , 调用  $Search\_in\_Linkjoint()$  的过程, 因为  $sbt$  与  $P_1$  无共用链路, 所以调用  $Search\_in\_nodeshared()$ 。在由  $P_1$  和  $sbt$  链路构成的并集中, 除了  $\{sabct, sbt\}$  这组链路分离的路径, 还有一组为  $\{sabt, sbct\}$ 。由于路径  $sbt$  的权向量为  $(11, 2)$ , 不满足约束条件  $(10, 10)$ , 故  $\{sabct, sbt\}$  不能作为 MCLPRA 的解。经判断,  $sabt$  和  $sbct$  均满

足约束条件, 故在算例 2 中, MCLPRA 的解为  $\{sabt, sbct\}$ , 该解也是图 3(a)上的最优解。

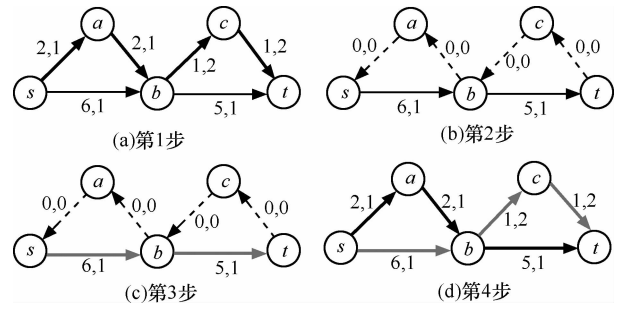


图 3 MCLPRA 算例 2

### 4 MCLPRA 特性分析

本节将从理论上分析 MCLPRA 设计的科学性, 从 MCLPP 解的构成形式入手证明 MCLPRA 求解结果与网络结构无关, 在此基础上给出 MCLPRA 求得可行解和最优解的控制参数  $T$  的取值。

**定理 1** 给定加权有向图  $G(V, E)$ ,  $P_1$  为节点  $s$  和  $t$  间非线性长度最短路径, 若  $s$  和  $t$  间满足约束向量  $C$  ( $\dim(C) \geq 2$ ) 的 MCLPP 问题的最优解  $\{P_1^*, P_2^*\}$  存在, 则  $\{E(P_1^*) \cup E(P_2^*)\} \cap E(P_1) = \emptyset$  成立。

**证明** 假设  $\{E(P_1^*) \cup E(P_2^*)\} \cap E(P_1) = \emptyset$ , 则有  $E(P_1^*) \cap E(P_1) = \emptyset$  且  $E(P_2^*) \cap E(P_1) = \emptyset$ 。因为  $P_1^*$  和  $P_2^*$  链路分离, 故  $E(P_1^*) \cup E(P_2^*) = \emptyset$ 。因此,  $P_1^*$ 、 $P_2^*$  和  $P_1$  为链路彼此分离的 3 条路径。又因为  $P_1$  为  $s$  和  $t$  间的最短路径, 可得

$$\begin{cases} l(P_1) + l(P_1^*) < l(P_2^*) + l(P_1^*) \\ l(P_1) + l(P_2^*) < l(P_1^*) + l(P_2^*) \end{cases} \quad (3)$$

这与定理 1 题设  $\{P_1^*, P_2^*\}$  为最优解矛盾, 故假设不成立。定理 1 得证。

**推论 1** 给定加权有向图  $G(V, E)$ ,  $P_1$  为节点  $s$  和  $t$  间非线性长度最短路径, 若  $s$  和  $t$  间满足约束向量  $C$  ( $\dim(C) \geq 2$ ) 的 MCLPP 问题的最优解  $\{P_1^*, P_2^*\}$  存在, 当且仅当  $E(P_1^*) \cap E(P_1) = \emptyset$  时,  $P_1 = P_2^*$  ( $P_1^*$  和  $P_2^*$  互换推论 1 仍然成立)。

**证明** 由定理 1  $\{E(P_1^*) \cup E(P_2^*)\} \cap E(P_1) \neq \emptyset$ , 所以有  $E(P_1^*) \cap E(P_1) \neq \emptyset$  或  $E(P_2^*) \cap E(P_1) \neq \emptyset$ 。假设  $E(P_1^*) \cap E(P_1) = \emptyset$ , 若  $P_2^* \neq P_1$ , 最优链路分离路径对则为  $\{P_1, P_1^*\}$ , 这与推论 1 题设矛盾, 假设不成立, 必要性得证。若  $P_2^* = P_1$ , 因和  $P_2^*$  为链路  $P_1^*$  分离路径对,  $E(P_1^*) \cap E(P_1) = \emptyset$ , 充分性得证。故推论 1 成立。

定理 1 和推论 1 说明了对于任意结构的网络, 如果 MCLPP 的最优链路分离路径对  $\{P_1^*, P_2^*\}$  存在, 最短路径要么与  $P_1^*$  和  $P_2^*$  都相交, 要么和其中的一条重合。故给定加权有向图  $G(V, E)$ ,  $P_1$  为从源、目的间的最短路径。  $S = \{P'_1, \dots, P'_k\}$  为 MCLPRA 第 3 步求得结果。  $\forall P_2 \in \{P'_1, \dots, P'_k\}$ ,  $P_1$  与  $P_2$  的结构关系只能为下列 4 种情况之一。

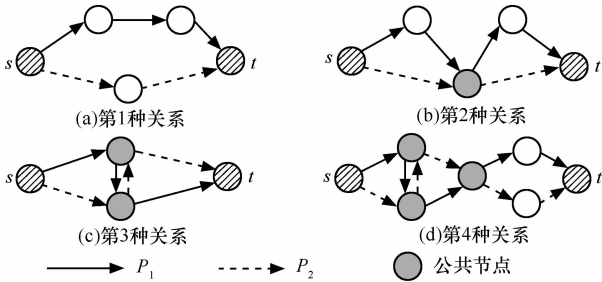


图 4  $P_1$  与  $P_2$  路径的 4 种关系

①  $E(P_1) \cap E(P_2) = \emptyset$  且  $V(P_1) \cap \{V(P_2) - s - t\} = \emptyset$ , 如图 4(a) 所示,  $P_1$  与  $P_2$  既无共用节点, 也无共用链路;

②  $E(P_1) \cap E(P_2) = \emptyset$ ,  $V(P_1) \cap \{V(P_2) - s - t\} \neq \emptyset$ , 如图 4(b) 所示,  $P_1$  与  $P_2$  有共用节点, 但无共用链路;

③  $E(P_1) \cap E(P_2) \neq \emptyset$ ,  $V(P_1) \cap \{V(P_2) - s - t\} \neq \emptyset$  且  $V(P_1) \cap \{V(P_2) - s - t\} = V(E(P_1) \cap E(P_2))$ , 如图 4(c) 所示,  $P_1$  与  $P_2$  既有共用节点, 也有共用链路, 但所有共用节点都和共用链路关联;

④  $E(P_1) \cap E(P_2) \neq \emptyset$ ,  $V(P_1) \cap \{V(P_2) - s - t\} \neq \emptyset$  且  $V(P_1) \cap \{V(P_2) - s - t\} \neq V(E(P_1) \cap E(P_2))$ , 如图 4(d) 所示,  $P_1$  与  $P_2$  既有共用节点也有共用链路, 但并非所有共用节点都与共用链路关联;

在上述表达式中,  $E(P_1) \cap E(P_2)$  表示  $P_1$  与  $P_2$  共用的链路的集合,  $V(E(P_1) \cap E(P_2))$  表示  $P_1$  与  $P_2$  所有共用链路上的节点的集合;

$\forall P_2 \in \{P'_1, \dots, P'_k\}$ , 若  $P_1$  和  $P_2$  属于第①种关系, 则  $\{P'_1, \dots, P'_k\}$  中的最短路径  $\min\{P'_1, \dots, P'_k\}$  和  $P_1$  构成的链路分离路径对即为总长最短的链路分离的路径对, 如果  $P_1$  和  $\min\{P'_1, \dots, P'_k\}$  均满足约束, 则  $\{P_1, \min\{P'_1, \dots, P'_k\}\}$  为最优解; 若  $P_1$  和  $P_2$  属于第②种关系,  $P_1$  和  $P_2$  可构成多组链路分离路径对 (如图 4(b) 的结构中存在 2 组链路分离路径对), 处理方法是满足约束的路径对中选取总长度最小的一组作为最优解, 具体过程见  $Search\_in\_nodeshared()$  的

处理过程; 若  $P_1$  和  $P_2$  属于第③种关系, 可先将  $P_1$  和  $P_2$  共用的链路去除, 将其转换成情况①进行处理; 若  $P_1$  和  $P_2$  属于第④种关系, 可先将  $P_1$  和  $P_2$  共用的链路删除, 将其转换成情况②进行处理; 对情况③和④的详细处理过程见第 2 节的  $Search\_in\_Linkjoint()$ 。

在给定的任意网络中, 由 MCLPRA 第 3 步求得的集合  $S = \{P'_1, \dots, P'_k\}$  上的路径  $P_2$  与  $P_1$  的关系必然属于上述 4 种关系中的一种。MCLPRA 第 4 步先对  $S$  上的路径按照与  $P_1$  的结构关系进行分类, 再针对不同情况分别进行处理, 最后比较选取各种情况下的最优解作为最终求解结果, 以此来保证寻求所得可行解中的最优解。定理 1、推论 1 以及 MCLPP 解构成的 4 种可能情况表明了本文所提算法 MCLPRA 采用先计算最短路径  $P_1$ , 再求  $\{P'_1, \dots, P'_k\}$ , 然后按类处理求解过程的科学性。MCLPRA 在求解过程中未对原网络拓扑进行改变, 保持了网络的连通性和解空间的完整性, 故求解结果与网络的结构无关。

由 MCLPRA 求解步骤可以看出, 在  $Search\_S_\beta()$  操作中引入了参数  $T$  来控制搜索的精度和深度, 本文将  $T$  设置为减计数器来控制搜索循环的次数。

**定理 2** 当  $T$  取最大有效值  $\|S_\beta\| (T \geq \|S_\beta\|)$  时, MCLPRA 对任意结构的网络均可能求得最优解, 其中,  $\|S_\beta\|$  为 MCLPRA 中  $S_\beta$  集合中元素个数。

**证明** MCLPRA 第 3 步采用 SAMCRA 计算出图中所有满足  $2C$  约束的路径集合  $S = \{P'_1, \dots, P'_k\}$ 。根据定理 1、推论 1 和对解构成的 4 种情况的分析, MCLPP 的最优解一定由  $P_2$  与  $P_1$  构成,  $P_2 \in S$ 。根据 MCLPRA,  $S_\alpha \cup S_\beta = S$ ,  $S_\alpha$  为满足情况①的  $P_2$  的集合,  $S_\beta$  为情况②、③和④的集合, MCLPRA 分别求得  $S_\alpha$  上的最优解和  $S_\beta$  上的最优解, 选择其中最优秀的一组为最终解。对于  $S_\alpha$ ,  $S_\alpha$  中最短的路径  $P_\alpha$  与  $P_1$  组成的分离路径对即为  $S_\alpha$  上的最优解。对于  $S_\beta$ , 在  $Search\_S_\beta()$  中参数  $T$  可控制搜索  $S_\beta$  的深度。根据算法过程, 在搜索的每一次循环中,  $S_\beta$  的当前最短路径在处理完后被删除, 而  $T$  的值也被减 1, 当  $S_\beta$  中的路径全被处理完毕时, 即可搜索到  $S_\beta$  上的最优解, 当  $T \geq \|S_\beta\|$  可以保证  $S_\beta$  上的路径被全部处理, 因而  $T \geq \|S_\beta\|$  时可确保 MCLPRA 所得最优解不丢失。

**推论 2** 当  $T$  取判决条件——在  $S_\beta$  中搜索到与

$P_1$  构成满足约束条件的链路分离路径的一组可行解就退出搜索循环时, MCLPRA 能够保障对任意结构的网络求得可行解, 并且该可行解为当前搜索深度下的可行解中的最优解。

**证明** 根据前文所述, MCLPRA 的可行解落在  $S_\alpha$  和  $S_\beta$  与最短路径  $P_1$  构成的路径对上。  $S_\alpha$  中最短的路径  $P_\alpha$  与  $P_1$  组成的分离路径对即为  $S_\alpha$  上的最优解。对于  $S_\beta$ , 如果在  $S_\beta$  中搜索到与  $P_1$  构成满足约束条件的链路分离路径的一组可行解就退出搜索。根据算法过程, MCLPRA 的最终解为该组可行解与  $S_\alpha$  上可行解中最优的一组。推论 2 得证。

### 5 MCLPRA 与 DIMCRA 的性能比较

DIMCRA 是目前求解 MCLPP 问题最有效算法, 故对 MCLPRA 和 DIMCRA 进行对比分析。

#### 5.1 算例比较

**算例 3** 仍取算例 1 中的网络和相同的约束, 如图 5(a)所示。若运行 DIMCRA, 第 1、2 步的运行过程和结果与 MCLPRA 的第 1、2 步一样(见图 5(a)和图 5(b)), 所得最短路径  $P_1$  为  $sabt$ 。第 3 步, 在图 5(b)上运行 SAMCRA, 解得最短的路径  $P_2$  为  $sdt$ 。第 4 步, 因为  $P_1$  和  $P_2$  已经是链路分离路径, 转至第 5 步。经判断,  $P_1$  和  $P_2$  均满足约束  $C$ , 算法结束。所以 DIMCRA 在图 5(a)的网络上的求解结果为  $\{sabt, sdt\}$ 。显然  $\{sabt, sdt\}$  仅为图 5(a)上的一个可行解, 并非最优解, 而 MCLPRA 可以求得最优解(见算例 1)。

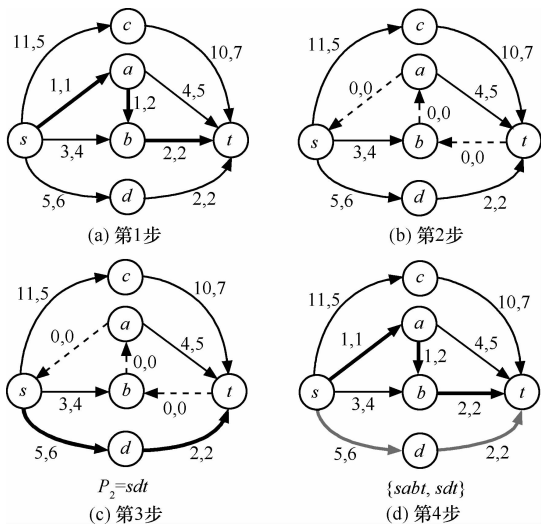


图 5 DIMCRA 算例 3

**算例 4** 仍采用图 5(a)的拓扑, 将约束向量修改为  $C = (7, 7)$ 。运行 MCLPRA, 求解过程与结果

仍与算例 1 相同。若运行 DIMCRA, 前 4 步过程和图 5(a)、图 5(b)和图 5(c)相同, 可求得  $P_1 = sabt$ ,  $P_2 = sdt$ 。在第 5 步中, 因为  $P_2$  的权向量为  $(7, 8)$ , 不满足  $C$  的约束, 故  $P_2$  上与  $P_1$  不相交链路被删除, 结果得到图 6(a)所示的修正图, 并返回第 3 步, 在图 6(a)上重新运行 SAMCRA 求得新的最短路径  $sbat$ 。第 4 步将  $sbat$  与  $P_1$  的链路并集中共用链路  $ab$  删除, 将剩余链路组成 2 条新的链路分离路径  $\{sat, sbt\}$ 。第 5 步判断  $\{sat, sbt\}$  满足约束为 DIMCRA 的最终解。

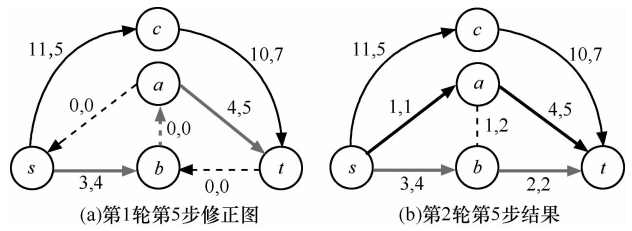


图 6 DIMCRA 算例 4

在算例 4 中, 虽然 DIMCRA 得到了与 MCLPRA 相同的解, 但运行了 3 次 SAMCRA, 而 MCLPRA 仅运行了 2 次 SAMCRA。

**算例 5** 采用图 3(a)所示的拓扑和相同的约束。若运行 DIMCRA, 前 3 步与 MCLPRA 相同, 如图 3(a)、图 3(b)和图 3(c)所示, 可得到  $P_1 = sabct$ ,  $P_2 = sbt$ 。第 4 步, 因为  $P_1$  与  $P_2$  无共用链路, 转而执行第 5 步。由于  $P_2$  的权向量为  $(11, 2)$ , 不满足  $C$  的约束, 故  $P_2$  与  $P_1$  不相交的链路被删除, 结果得图 7 所示修正图, 然后返回第 3 步。显然, 在图 7 的结构上, 已不存在从  $s$  到  $t$  的路径, 算法终止, 故此算例中 DIMCRA 返回为无解。而算例 2 已表明 MCLPRA 可以在图 3(a)的网络上得到最优解。

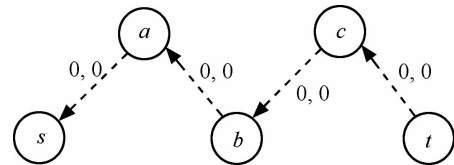


图 7 DIMCRA 算例 5

综上所述, MCLPRA 的求解能力要高于 DIMCRA。原因在于: ① DIMCRA 对  $P_2$  采用了计算出一条就处理一条的方法, 一旦满足约束条件就终止算法。这样的过程实际上求得的只是算法最先搜索到的可行解, 并未进行最优解的搜索处理, 因而 DIMCRA 不能保证对任意网络求得最优解; ② DIMCRA 的第 5 步, 对不能和  $P_1$  构造满足约束条件的分离路径对的  $P_2$ , 采

用删链路的处理，破坏了网络的原本结构和连通性，导致了可行解和最优解的丢失，因而也不能保证对任意网络求得可行解；③MCLPRA 避免了 DIMCRA 上述几个问题，是一种与网络结构无关的 MCLPP 求解算法，这一点第 4 节已论证。

### 5.2 仿真比较

本小节通过仿真实验对这 2 个算法进行比较，仿真采用随机拓扑图(RGU)<sup>[18]</sup>，RGU 图的节点数为  $N$ ，链路密度  $\rho$  取值为 0.2。每条链路都带有  $m$  维的加性权重，每维权重均服从[0,1]上的均匀分布。仿真所用计算机 CPU 主频为 1.9GHz，内存为 1G。RGU 图的节点数取为 100、150、200、250、300、350、400、450 和 500，每个  $N$  上生成 1 000 个 RGU。取  $c_i = 1$  ( $1 \leq i \leq m$ )，在  $m = 2$  和  $m = 3$  的条件下分别进行实验。每个 RGU 上运行 MCLPRA 和 DIMCRA 各 1 次。

图 8 给出了 MCLPRA 和 DIMCRA 分别在 2 约束和 3 约束下每个  $N$  上的 1 000 个 RGU 上成功求解的比率。

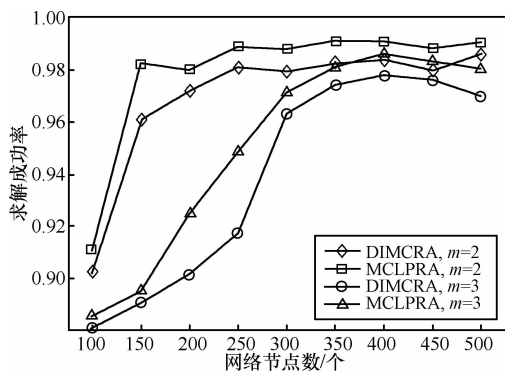


图 8 算法求解成功率比较

由图 8 可以看出：① MCLPRA 的可行解求解概率都高于相同情况下的 DIMCRA。2 个算法的求解成功率都随着网络节点数的增加而提高，这是因为  $\rho$  一定的情况下， $N$  越大网络连接密度越高，网络可行路径数量也在增长，客观存在可行解的概率就越大。②另外，仿真结果显示 MCLPRA 的可行解求解率并不总是 100%。这是因为：MCLPRA 基于 SAMCRA，MCLPRA 对任意拓扑均可保证求得可行解的前提条件是在第 3 步运行 SAMCRA 时求得并存储所有满足约束的可行解，因为计算机的实际存储能力有限，仿真实验中只选取所有满足 2C 路径中前 20 条最短路径进行存储和搜索，因而会有解的损失，要提高求解成功率，可以适当增加  $T$  的值；随机生成的 RGU 很可能客观上并不存在可行解，因而也就

难以求解，这也是网络节点数较少时，2 种算法求解成功率都偏低的原因之一，即便如此，仿真结果仍表明即使在限制了搜索范围的情况下，MCLPRA 的可行解求解平均概率仍要高于 DIMCRA。③在 2 约束条件下，MCLPRA 和 DIMCRA 的平均求解概率都高于 3 约束的条件，这是因为约束个数越少网络的可行路径数目越多。

图 9 给出了 MCLPRA 和 DIMCRA 分别在 2 约束和 3 约束下在每个  $N$  上的 1 000 个 RGU 上所求分离路径对的平均非线性长度和情况。结果表明，MCLPRA 所求解的平均总长度明显小于 DIMCRA 所求解的平均总长度，即 MCLPRA 所求解优于 DIMCRA。

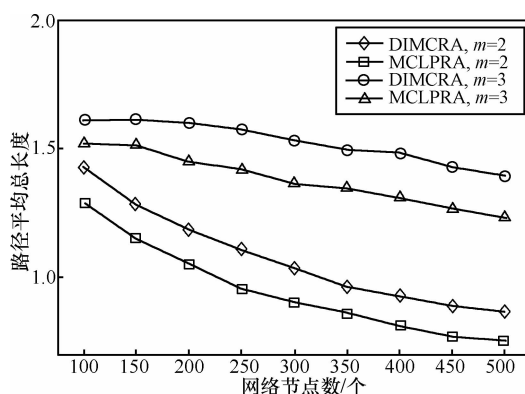


图 9 算法所求解的平均长度比较

图 10 给出了 MCLPRA 和 DIMCRA 分别在 2 约束和 3 约束下在每个  $N$  上的 1 000 个 RGU 平均执行时间开销对比结果。结果显示，MCLPRA 的时间开销略高于 DIMCRA，这是因为要实现算法更高的求解率和更优的求解结果，往往需要以增加一定的复杂性为代价。从图中可以看出，节点数在 500，平均连接度达 200 的情况下(即高密度连接的复杂网络)，算法的执行时间开销仍在数百毫秒，这对网络的路由计算仍处于可行的范围之内。

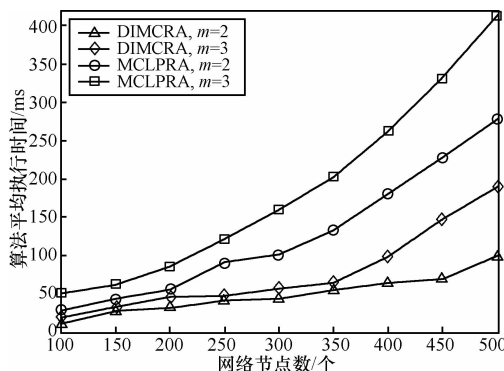


图 10 算法所求解的平均长度比较



上述实验说明 MCLPRA 在可行的执行开销内, 求解成功率和所求解都明显优于现有算法。

## 6 结束语

本文针对多个加性 QoS 约束下的链路分离算法进行了研究, 提出了基于解空间分类搜索的算法 MCLPRA, 给出了 MCLPRA 的求解过程。从理论上分析了 MCLPP 问题解的构成形式, 证明了 MCLPRA 对任意结构的网络均可求得最优解, 并给出了 MCLPRA 求得最优解的控制参数取值条件。通过理论分析和仿真比较, MCLPRA 均可获得比 DIMCRA 更优的求解性能。笔者下一步工作将对 MCLPRA 进行优化和改进, 并进一步降低其算法复杂性。

### 参考文献:

- [1] DAS A, MARTEL C, MUKHERJEE B, *et al.* A better approach to reliable multi-path provisioning[A]. IEEE Global Communications Conferences (GLOBECOM)[C]. 2007. 2724-2728.
- [2] SAWADA N, KANEKO K. Pairwise disjoint paths in pancake graphs[A]. Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies, DPCAT '07[C]. 2007. 376-382.
- [3] CHEN S, NAHRSTEDT K. On finding multi-constrained paths[A]. IEEE International Conference on Communications ICC'98[C]. 1998. 874-879.
- [4] TAFT-PLOTKIN N, BELLUR B, OGIER R. Quality-of-service routing using maximally disjoint paths[A]. The 7th International Workshop on Quality-of-Service[C]. 1999. 119-128.
- [5] GUO L, LI L M, CAO J, *et al.* On finding feasible solutions with shared backup resources for surviving double-link failures in path-protected WDM mesh networks[J]. Journal of Lightwave Technology, 2007, 25(1): 287-296.
- [6] XIONG K, QIU Z D, ZHANG H K. Towards link-disjoint paths under multiple additive QoS constraints[A]. The 2nd IET International Conference on Wireless Mobile and Multimedia Networks (ICWMMN)[C]. 2008. 119-127.
- [7] XU D H, QUAO C M, XIONG Y Z. Ultrafast potential-backup-cost (PBC)-based shared path protection schemes[J]. Journal of Lightwave Technology, 2007, 25(8): 2251-2259.
- [8] VAN MIEGHEM P, DE NEVE H, KUIPERS F. Hop-by-hop quality of service routing[J]. Computer Networks, 2001, 37(3/4): 407-423.
- [9] XUE G L, SEN A, ZHANG W Y, *et al.* Finding a path subject to many additive QoS constraints [J]. IEEE/ACM Transactions on Networking, 2007, 15(1): 201-211.
- [10] VAN MIEGHEM P, KUIPERS F. Concepts of exact QoS routing algorithms[J]. IEEE/ACM Transactions on Networking, 2004, 12(5): 851-864.
- [11] GUO Y C, KUIPERS F, VAN MIEGHEM P. Link disjoint paths algorithm for reliable QoS routing[J]. International Journal of Communication Systems, 2003, 16(9): 779-798.
- [12] 郭宇春, KUIPERS F, MIEGHEM P V 等. 多约束分离路径算法[J]. 铁道学报, 2005, 27(2): 49-57.  
GUO Y C, KUIPERS F, VAN MIEGHEM P, *et al.* Disjoint multiple-constrained paths algorithms[J]. Journal of the China Railway Society, 2005, 27(2): 49-57.
- [13] 张品, 张坚武, 李乐民等. QoS 约束下的链路分离问题的研究[J]. 通信学报, 2006, 27(6): 37-42.  
ZHANG P, ZHANG J W, LI L M, *et al.* Researches on the problem of link disjoint paths with QoS constraints[J]. Journal on Communications, 2006, 27(6): 37-42.
- [14] CHAO P, HONG S. An improved approximation algorithm for computing disjoint QoS paths[A]. IEEE ICN/ICONS/MCL[C]. 2006.
- [15] NASER H, GONG M. Link-disjoint shortest-delay path-pair computation algorithms for shared mesh restoration networks[A]. IEEE Symposium on Computers and Communications[C]. 2007. 269-274
- [16] XU D H, CHEN Y, XIONG Y Z, QIAO C M, *et al.* On the complexity of and algorithms for finding the shortest path with a disjoint counterpart[J]. IEEE/ACM Transactions on Networking, 2006, 14(1): 147-158.
- [17] XIAO Y, THULASIRAMAN K, XUE G L. Constrained shortest link-disjoint paths selection: a network programming based approach[J]. IEEE Transactions on Circuits and Systems-I: Regular Papers, 2006, 53(5): 1174-1187.
- [18] BOLLOB S B. Random Graphs[M]. MA: Cambridge Univ, Press, 2001.

### 作者简介:



熊轲 (1981-), 男, 陕西汉中, 博士, 清华大学电子工程系博士后, 主要研究方向为下一代互联网络、多媒体通信和网络信息论。

裘正定 (1944-), 男, 浙江嵊县人, 北京交通大学教授、博士生导师, 主要研究方向为现代信息处理与通信。

张煜 (1983-), 男, 广西桂林人, 北京交通大学博士生, 主要研究方向为下一代互联网络和多媒体通信。

张宏科 (1957-), 男, 山西大同人, 北京交通大学教授、博士生导师, 主要研究方向为下一代互联网络。