

多约束分离路径算法

郭宇春^{*†}, Fernando Kuipers^{**}, Piet Van Mighem^{**}

[†]北京交通大学电子信息工程学院

^{**}荷兰代尔夫特大学电子工程数学与计算机科学学院

摘要: 多媒体流和视讯会议等新应用的出现要求网络提供可靠的服务质量 (QoS) 保证。为网络中的每对节点建立两条多约束链路/节点分离路由是提供可靠 QoS 的一条重要途径。如何建立两条链路/节点分离多约束路径是一个关键问题, 这一问题与经典的一维测度下的分离路径问题有着很多概念上的不同。本文分析了这一问题的困难性, 并提出一组解决此问题的启发式算法。
关键词: 可靠 QoS 路由、分离路径、多约束路径

Disjoint Multiple-Constrained Paths Algorithms

Yuchun Guo^{*†}, Fernando Kuipers^{**}, Piet Van Mighem^{**}

^{*}School of Electronic and Information Engineering, Beijing Jiaotong University, 100044 Beijing, P.R. China

^{**}Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands

Abstract: Networks are expected to provide reliable quality of service (QoS) imposed by new applications such as multimedia streaming and video conferencing. Using two link/node-disjoint paths between a pair of nodes in a network is a common approach to provide reliable QoS. The problem of finding link/node disjoint paths satisfying multiple constraints is a major issue in this context. This problem faces many different concepts compared with the classical one-dimension disjoint paths problem. This paper addresses these conceptual problems and presents some heuristics for solving this problem.

Key Words: reliable QoS routing, disjoint paths, multiple-constrained paths

1. 引言

随着多媒体流和视讯会议等新应用的出现要求网络提供可靠的服务质量 (QoS) 保证, 不仅要满足应用的 QoS 要求, 还要在网络故障时能够持续保证业务不间断地进行。要达到这些要求, 通常为一个连接提供两条链路/节点分离的路径, 其中一条主用, 一条备用。当主用路径故障时, 将其承载的业务流切换到备用路径上, 从而实现快速的业务恢复。此外, 负载均衡也需要分离路径实现网络中业务流的均匀分布, 避免网络拥塞, 优化网络吞吐量。健壮性 (Robustness) 和负载均衡是可靠 QoS 路由的两个重要方面。光网络和 MPLS/GMPLS 技术的发展提供资源预约和显式路由能力, 使得在网络中提供可靠的 QoS 保证成为可能。如何在节点之间建立链路/节点分离路径成为提供可靠 QoS 的主要问题, 因此重新得到研究者的关注 [2,7,12,11,14]。在使用一维测度度量链路的网络中建立一对节点间的链路/节点分离路径的问题是图论中的一个经典问题, 已有很多文献论述其理论意义以及在集成电路布线设计、电信网络可生存性设计等诸多领域的实际应用。但是对于 QoS 路由而言, 需要采用多维测度衡量网络链路, 提供满足应用多方面的质量要求的多约束路径。如何在多维测度下寻找两条链路/节点分离多约束路径是一个值得研究的新问题。

在一个网络中, 给定一对信源节点和信宿节点, 如果这对节点之间的两条或两条以上的路径之间不存在相同 (共用) 链路, 则称这些路径是链路分离路径; 若这对节点之间的两条或两条以上的路径之间不存在相同 (共用) 节点 (信源、信宿节点除外), 则称这些路径是节点分离路径。在一对源-宿节点间建立两条最短链路分离路径的一种简单方法有两个步骤: 1) 在这对节点之间寻找一条最短路径; 2) 从图中删除这条路径上所有的链路得到修正图, 在修正图中寻找一条最短路径。在本文中我们称此算法为删除-搜索 (Remove-Find) 算法, 简称 RF 算法。RF 算法虽然简单直接但至少存在两个缺陷: 1) 即使存在两条链路分离路径, RF 算法不一定能找到, 详见 3.1 节的解释; 2) 第二条最短路径可能太长。

为了克服 RF 算法的缺陷, 人们提出了很多寻找总长最小的最短分离路径对的算法 [1,12,14,15,16,18,19]。[18]中提出 Suurballe 算法利用路径增广 (*path augmentation*) 方法寻找总长最小的 K 条节点分离路径。增广路径方法在图论中被用来扩大 matching 的规模 [4], 也用来寻找网络中的最大流或最小代价流 [5,17]。寻找链路/节点分离路径的问题可以视为最小代价流问题的一个特例 [1,18,19]。Suurballe 算法的基本思想是在最短路径

[†]Email: gyc@telecom.njtu.edu.cn

本工作得到国家留学基金委 Nuffic 奖学金和国家自然科学基金 60132030 及 60202001 项目支持。

和最短增广路径的基础上构造一个分离路径对的解集。\$k\$ (\$1 < k \le K\$) 条分离路径可以通过增广 \$k-1\$ 条最优分离路径递归地得到。1984 年 Suurballe 和 Tarjan[19] 提出增强算法 S-T 算法, 进一步降低了 Suurballe 算法的时间复杂度。此后的很多工作[9,10,13] 都建立在 [18,19] 的基础之上。Suurballe 算法 [18] 中采用 Dijkstra 算法寻找最短路径, 为此引入一个特殊的代价变换以保证 Dijkstra 算法所要求的非负链路权重。Bhandari[1] 通过修改原始的 Dijkstra 算法使之允许最短路上的负权重, 避免了代价变换, 允许直接将第一最短路的所有链路设为负权重的操作。

上述工作都是针对一维测度下的分离路径。就我们所知, 除了我们在 [8] 中所作的初步探索外, 尚无文献述及多个加性测度下的分离路径问题。最近出现了一些关于 QoS 路由中的分离路径问题的工作, 但都是以带宽或时延作为 QoS 测度 [2,7,12,11,14]。唯一涉及多测度分离路径的是 Taft-Plotkin 等人 [20] 提出的 MADSWIP 算法, 该算法对 S-T 算法 [19] 进行改进用来寻找一个源节点到其它每个节点之间的一对最宽或最短的最大程度链路分离路径。不过该算法采用的两个测度——时延和带宽中只有时延是加性的, 因此说该算法解决的不是多个加性测度的分离路由问题。

本文在 [8] 的基础上进一步研究多维加性测度下的链路/节点分离路径问题, 提出了解决多约束链路分离路径对 (MCLPP) 问题的启发式算法 DIMCRA-L1 和 DIMCRA-L2, 以及解决多约束链路分离路径对 (MCNPP) 的启发式算法 DIMCRA-N。本文第 2 节首先描述所要分析的 MCLPP 问题和 MCNPP 问题并给出本文中将要用到的基本定义。第 3 节分析论证采用路径增广方法构造一维链路分离路径的基本思想。第 4 节指出构造精确的多维分离路径算法的困难性。第 5 节提出了几个解决 MCLPP 问题和 MCNPP 问题的启发式算法。第 6 节给出结论。

2. 问题描述

在本文中, 我们将网络表示为一个有向图 \$G(V,E)\$, 其中 \$V\$ 是节点集, \$E\$ 是链路集。一条从节点 \$u\$ 到节点 \$v\$ 的有向链路记为 \$u \to v, u, v \in V\$。每条链路的权重矢量 \$\vec{w}\$ 含 \$M\$ 个分量 \$w_m(u \to v), m = 1, \dots, M\$。设仅使用非负链路权重 (但在分离路径的计算过程中可能引入负权重)。QoS 测度可以是加性的也可以是凸性或凹性的。对于加性测度, 如时延、抖动和分组丢失的对数等, 路径的权重矢量为构成该路径的所有链路的链路权重矢量之和。对于凹性或凸性的测度, 如带宽、策略标志等, 路径的权重矢量为构成该路径的所有链路的链路权重矢量之最大或最小值。加性测度在路由问题中会带来更大的困难 [21], 不失一般性, 我们假设所有测度为加性的。在 QoS 路由或多约束路由中, 若一条路径的所有权重分量均符合约束矢量 \$\vec{L}\$ 的各个分量, 称之为一条可用路径。我们将一对源-宿节点 \$(s, t)\$ 之间的路径 \$P(s, t)\$, 简称为 \$P\$, 将路径 \$P\$ 的链路集合也记为 \$P\$。若 \$P_1\$ 与 \$P_2\$ 链路分离, 即这两条路径的链路集合无相同的元素, 记为 \$P_1 \cap P_2 = \emptyset\$, 否则记为 \$P_1 \cap P_2 \neq \emptyset\$。若 \$P_1\$ 与 \$P_2\$ 节点分离, 不仅 \$P_1 \cap P_2 = \emptyset\$, 而且两条路径的节点集合 \$P_{1V}\$ 与 \$P_{2V}\$ 也没有相同的元素, \$P_{1V} \cap P_{2V} = \emptyset\$。
 路径长度: 给定图 \$G(V,E)\$, 其中每条链路有 \$M\$ 维测度。定义由源节点 \$s\$ 到目的节点 \$t\$ 的路径 \$P\$ 的非线性长度 [3] 为

$$l(P) = \max_{m=1, \dots, M} \left(\frac{w_m(P)}{L_m} \right) \quad (1)$$

其中 \$w_m(P) = \sum_{u \to v \in P} w_m(u \to v)\$。

对于一维路由即 \$M = 1, l(P) = w_1(P) / L_1\$, (1) 式中定义的非线性长度简化为线性长度, 链路矢量 \$\vec{w}\$ 简化为标量 \$w(u \to v)\$。无约束情况下 (如下面将论述的 SLPP 问题), 路径的线性长度为 \$\sum_{(u \to v) \in P} w(u \to v)\$, 为了表述的简单性, 仍使用上面定义的 \$l(P)\$ 并且令 \$L_1 = 1\$。若路径 \$P_1\$ 与 \$P_2\$ 链路分离, 则对于 \$M = 1\$ 有 \$l(P_1 \cup P_2) = l(P_1) + l(P_2)\$; 但当 \$M \ge 2\$ 时, 则有 \$l(P_1 \cup P_2) \le l(P_1) + l(P_2)\$。本文的目标是寻找两条满足多重约束且链路/节点分离的路径, 两条路径的总长定义为

$$l(P_1) + l(P_2) \quad (2)$$

最短链路分离路径对 (Shortest Link-Disjoint Path Pair : SLPP) 问题: 给定图 \$G(V,E)\$, 其中每条链路有 1 维权重。在一对源-宿节点 \$(s, t)\$ 间寻找两条链路分离路径 \$P_1\$ 和 \$P_2\$, 使总长 \$l(P_1) + l(P_2)\$ 最小。

多约束路径 (Multiple Constrained Path : MCP) 问题: 给定图 \$G(V,E)\$, 其中每条链路有 \$M\$ (\$M \ge 2\$) 维权重, 约束矢量为 \$\vec{L}\$, 在一对源-宿节点 \$(s, t)\$ 间寻找一条路径 \$P\$ 满足约束矢量 \$\vec{L}\$

$$w_m(P) \le L_m, m = 1, \dots, M,$$

多约束分离路径算法

其中 $w_m(P) = \sum_{u \rightarrow v \in E} w_m(u \rightarrow v)$, $m = 1, \dots, M$ 。即 $l(P) \leq 1$ 。

多约束链路分离路径对 (Multiple Constrained Link-Disjoint Path Pair: MCLPP) 问题: 给定图 $G(V, E)$, 其中每条链路有 $M (M \geq 2)$ 维权重, 约束矢量为 \vec{L} , 在一对源-宿节点 (s, t) 间寻找一对链路分离路径 P_1 和 P_2 , 且 P_1 和 P_2 均满足约束矢量 \vec{L} 。

多约束节点分离路径对 (Multiple Constrained Node-Disjoint Path Pair: MCNPP) 问题: 给定图 $G(V, E)$, 其中每条链路有 $M (M \geq 2)$ 维权重, 约束矢量为 \vec{L} , 在一对源-宿节点 (s, t) 间寻找一对节点分离路径 P_1 和 P_2 , 且 P_1 和 P_2 均满足约束矢量 \vec{L} 。

SLPP 问题可以在多项式时间内求解 [1,18,19]。MCP 问题则是 NP 完备的[6,22]。

定理 1 MCLPP、MCNPP 是 NP 完备的。

证明: MCP 问题是 MCLPP 问题的特例, 因此 MCLPP 也必然是 NP 完备的 (严格的数学证明见[8])。同理, MCNPP 也是 NP 完备的。

3. 增广路径方法

Bhandari 算法[1]是基于增广路径的最短跨距 (span) 分离路径算法, 为了说明增广路径这一概念, 我们将该算法简化为一个解决 SLPP 问题的最短链路分离路由算法, 记为 **LBA**(Link-disjoint version of Bhandari's Algorithm)。

3.1 LBA 的基本步骤

在说明 LBA 的步骤之前, 我们先引入一些表示法。若将路径 $P_1(s, t)$ 上的每条链路的权重和方向符号取反, 即 $w(v \rightarrow u) = -w(u \rightarrow v)$, $\forall (u \rightarrow v) \in P_1$, 我们可以得到一条由反向 P_1 链路组成的从 t 到 s 的路径, 记为 $-P_1$, 且其长度为¹ $l(-P_1) = -l(P_1)$ 。出现在路径 P_2 上的 $-P_1$ 链路及其对应的 P_1 链路所组成的集合记为 $P_1 \tilde{\cap} P_2 = \{(u \rightarrow v), (v \rightarrow u) \mid (u \rightarrow v) \in P_1 \text{ 且 } (v \rightarrow u) \in P_2\}$ 。在本文以下部分的所有例图中, 以粗线表示选中 (最短) 路径上的链路, 以虚线表示原始图中不存在的反向链路。

LBA (G, s, t): 给定有向图 $G(V, E)$, 对于源-宿节点对 (s, t) ,

第 1 步. 寻找一条从 s 到 t 的最短²路径 P_1 ;

第 2 步. 将 P_1 替换成 $-P_1$, 得到修正图 $G(V, E')$, $E' = E \cup (-P_1) \setminus P_1$;

第 3 步. 在修正图 $G(V, E')$ 中寻找一条从 s 到 t 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, 从中删去 $P_1 \tilde{\cap} P_2$ 链路, 将余下的链路组成两条路径 P'_1 和 P'_2 , 即 $P'_1 \cup P'_2 = (P_1 \cup P_2) \setminus (P_1 \tilde{\cap} P_2)$ 。

我们以图 1 所示的拓扑图为例说明 LBA 算法的步骤。设需要在源-宿节点 (a, b) 之间寻找两条分离路径, 第 1 步, $P_1 = acdb$, 长度为 4, 如图 1(a)所示; 第 2 步, 将路径 P_1 替换为 $-P_1$, 得到修正图如图 1(b)所示; 第 3 步, 找到修正图中由 a 至 b 的最短路 $P_2 = adcb$, 长度为 6, 如图 1(c)所示; 第 4 步, 如图 1(d)所示, 从链路集合 $P_1 \cup P_2$ 中删除链路集合 $P_1 \tilde{\cap} P_2 = \{c \rightarrow d, d \rightarrow c\}$, 余下的链路组合成两条分离路径 $\{P'_1, P'_2\} = \{acb, adb\}$ 。该路径集合的总长为 $5 + 5 = 10$, 为最优解。

作为对比, 我们采用 RF 算法在相同的拓扑上按同样的要求进行路由。在第一步, 我们得到最短路 $P_1 = acdb$; 在第二步, 在删除 $acdb$ 上的所有链路后得到修正图中搜索到最短路 $P_2 = aeb$, 长度为 11。集合 $\{acdb, aeb\}$ 总长为 $4 + 11 = 15$, 非最优解集。这个例子表明 RF 算法不能保证发现最优解, RF 算法所得到的第二最短路比 LBA 的长很多。更重要的是, 在如图 2(a)所示的拓扑中, 尽管 (a, b) 节点对之间存在两条链路分离路径, 在采用 RF 算法删除最短路 $acdb$ 上的所有链路后, 如图 2(b)所示, 修正图不连通, 算法无解。相反, 在该拓扑上运行 LBA 算法则仍可提供最优解。

¹ 根据定义(1), 仅对 $M=1$ 有 $l(-P_1) = -l(P_1)$ 。

² 若原始图或修正图中存在多条最短路, 可以选择其中任何一条。选择不同的最短路可能导致不同的解集, 但这些解集有相同的最小总长。

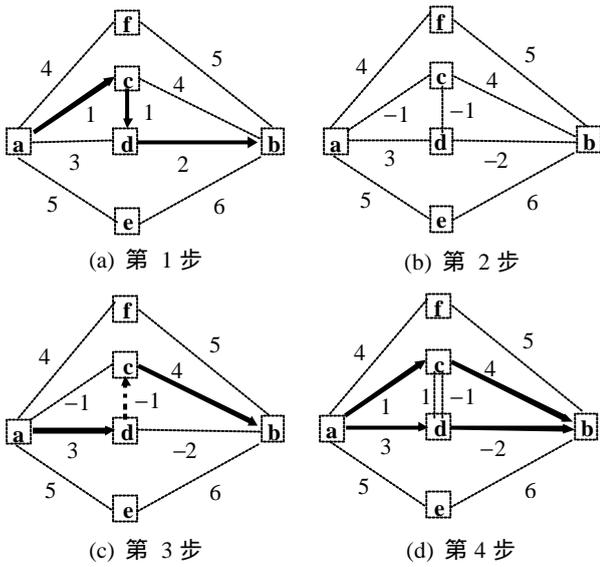


图 1. LBA 算法示例

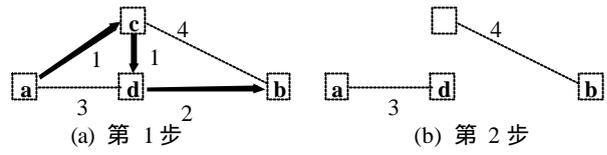


图 2. RF 算法示例 2

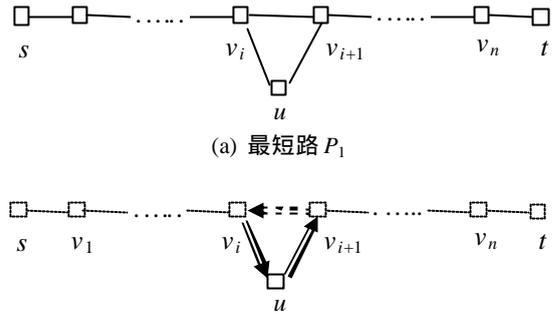


图 3. 含有负权重链路的环路

3.2 基于最短路建立解集

在这一节里我们提出的理论基于最小代价流理论[5,17]，这些阐述对进一步分析解决 MCLPP/MCNPP 问题具有指导意义。

给定一个有向图 $G(V,E)$ 和一个源宿节点对 (s, t) ，一对链路分离路径 $\{P_{d1}, P_{d2}\}$ 与最短路 P_1 之间的关系必然为以下类型之一：

- 1) P_1 本身是 P_{d1} 或 P_{d2} ，即 $P_1 = P_{d1}$ 或 $P_1 = P_{d2}$;
- 2) P_1 与 P_{d1} 和 P_{d2} 都有重叠，即 $P_1 \cap P_{d1} \neq \emptyset, P_1 \neq P_{d1}$ 且 $P_1 \cap P_{d2} \neq \emptyset, P_1 \neq P_{d2}$;
- 3) P_1 仅与 $\{P_{d1}, P_{d2}\}$ 中的一条路径重叠，与另一路径不重叠，即 $P_1 \cap P_{d1} \neq \emptyset, P_1 \neq P_{d1}$ 且 $P_1 \cap P_{d2} = \emptyset$ (或 $P_1 \cap P_{d2} \neq \emptyset, P_1 \neq P_{d2}$ 且 $P_1 \cap P_{d1} = \emptyset$);
- 4) P_1 与 $\{P_{d1}, P_{d2}\}$ 中的每条路径都相互分离，即 $P_1 \cap (P_{d1} \cup P_{d2}) = \emptyset$ 。

引理 1. 给定一有向图 $G(V, E)$ ，对于源-宿节点对 (s, t) ，若有链路分离最短路对的最优解集 $\{P'_1, P'_2\}$ 存在，则该集合必然包含最短路 P_1 本身作为其中一条路径（类型 1），或者两条路径上分别包含部分 P_1 链路（类型 3）。（证明见[8]。）

设 $\{P'_1, P'_2\}$ 为最优解集，记 $O_l = (P'_1 \cup P'_2) \cap (-P_1)$ ，集合 O_l 由并集 $P'_1 \cup P'_2$ 中的 P_1 链路及其对应的 $-P_1$ 链路组成，显然每一对这样的链路都构成一个长度为零的圈（cycle），即 $l(O_l) = 0$ 。我们定义 $P'_1 \cup P'_2$ 与 P_1 的逻辑差集³为 $(P'_1 \cup P'_2) - P_1 = (P'_1 \cup P'_2) \cup (-P_1) \setminus O_l$ 。又因为 $l(-P_1) = -l(P_1)$ ，所以对于最优解集 $\{P'_1, P'_2\}$ 必有

$$l((P'_1 \cup P'_2) - P_1) = l((P'_1 \cup P'_2) \cup (-P_1)) - l(O_l) = l(P'_1) + l(P'_2) + l(-P_1) = l(P'_1) + l(P'_2) - l(P_1),$$

即最优解集 $\{P'_1, P'_2\}$ 与最短路 P_1 的逻辑差集长度等于(3)式中的长度差 Y 。

性质 1. 在所有可能解集中，最优解集 $\{P'_1, P'_2\}$ 与最短路 P_1 的长度差

$$Y = l(P'_1) + l(P'_2) - l(P_1) \tag{3}$$

最小且 $Y \geq 0$ 。

引理 2. 给定一有向图 $G(V,E)$ ， P_1 是源-宿节点对 (s, t) 之间的最短路。将 $G(V,E)$ 中的路径 P_1 替换为 $-P_1$ ，得修正图 $G(V,E)$ 。最优解集 $\{P'_1, P'_2\}$ 与最短路 P_1 之间的逻辑差集形成图 $G(V,E)$ 中由 s 到 t 的最短路。（证明见[8]。）

³在集合论中，差运算为 $P_2 - P_1 = P_2 \setminus (P_1 \cap P_2)$ ，对称差运算为 $P_2 - P_1 = (P_2 \cup P_1) \setminus (P_1 \cap P_2)$ 。本文中的逻辑差运算类似于对称差，但不完全相同。

由引理 1、引理 2 及性质 1 可知, LBA 以及其它基于增广路径的算法[1,18,19]基于最短路构造解集的原因所在。

3.3 LBA 是无环的最优算法

大多数路由算法都建立在非负权重的假设上, 以避免具有负长度的环路出现在路径上。但是 LBA 中引入的负权重不会在路由过程中引起环路。

定理 2: 给定一个有向图 $G(V,E)$, P_1 是源-宿节点对 (s, t) 之间的最短路。将 $G(V,E)$ 中的路径 P_1 替换为 $-P_1$, 得修正图 $G(V,E')$ 。 $G(V,E')$ 中包含负权重链路的环路长度非负。

证明: 不失一般性, 设 $sv_1 \dots v_i v_{i+1} \dots v_n t$ 为 $G(V,E)$ 中由 s 到 t 的最短路 P_1 , 如图 3(a) 所示; $G(V,E)$ 中相应的 $-P_1$ 如图 3(b) 中的虚线所示; $-P_1$ 上的链路 $(v_{i+1} \rightarrow v_i)$ 出现在环路 $P_l = u_i v_{i+1} v_i u_i$ 上。设 P_l 长度为 $l(P_l) = w(u_i \rightarrow v_{i+1}) + w(v_{i+1} \rightarrow v_i) + w(v_i \rightarrow u_i) < 0$ 。因为 $w(v_{i+1} \rightarrow v_i) = -w(v_i \rightarrow v_{i+1})$, 必有 $w(v_i \rightarrow u_i) + w(u_i \rightarrow v_{i+1}) < w(v_i \rightarrow v_{i+1})$ 。因此子路径 $sv_1 \dots v_i u_i v_{i+1}$ 比子路径 $sv_1 \dots v_i v_{i+1}$ 短。与假设 $sv_1 \dots v_i v_{i+1} \dots v_n t$ 为最短路矛盾。?

定理 3. 给定一个有向图 $G(V,E)$, 对于节点对 (s, t) , LBA 可以构造无环的具有最短总长的链路分离路径对。

证明: 令 P_1 为执行 LBA 第 1 步在图 G 中发现的最短路, P_2 为第 3 步在修正图 $G(V,E)$ 中发现的最短路, $\{P'_1, P'_2\}$ 为 LBA 产生的解集。证明分为三个部分: (a) 链路分离性: 由解集的构造过程可知 $P'_1 \cap P'_2 = \emptyset$ 。在 LBA 第 4 步, 从 P_1 与 P_2 的并集中删除 P_2 链路对应的反向链路 P_1 链路所构成的零长度环路, 剩余链路必然构成同一源-宿节点对之间的两条完整的链路分离路径。(b) 最小总长: 若存在另一个解集 $\{P''_1, P''_2\}$ 比 $\{P'_1, P'_2\}$ 的总长短, 那么由引理 2 有 $\{P''_1, P''_2\}$ 与 P_1 的逻辑差集为 $G(V,E)$ 中的最短路, 这与 P_2 是 $G(V,E)$ 中的最短路矛盾。(c) 无环路: 根据定理 2, LBA 无环。因此 LBA 生成的解集必然是最优解。?

4. 多维分离路径问题的困难性

引入 MCP 最短路径算法, 如 SAMCRA 算法 [21], 作为最短路算法, 我们将 LBA 扩展为一个多维算法, 称为 MLBA (Multiple-constrained LBA)。在这一节我们首先简要介绍 SAMCRA 算法的基本思想, 接着通过对 MBLA 算法中存在问题的分析探讨构造精确的多维链路分离 QoS 路由算法的困难所在。

4.1 SAMCRA 简介

SAMCRA [21] 是一个多约束路由精确算法, 该算法基于以下几个基本思想: (a) 非线性路径长度, (b) k -最短路路由, (c) 删除劣势路径 (*non-dominance*)。 (1) 式中定义的非线性长度意味着在多约束路由中, 最短路的子路径不一定是最短路。在中间节点沿 k ($k \geq 1$) 条子路径搜索是使 SAMCRA 具有高搜索效率和解的精确性的另一关键。称 (子) 路径 P_1 相对于 (子) 路径 P_2 处于劣势, 若 $w_m(P_1) \geq w_m(P_2)$, $m = 1, \dots, M$, $M \geq 2$, 且不等号至少对于一个 m 成立。在路径搜索中, 对每条新发现的子路径, 都要将其与队列中的其他路径进行比较, 删除处于劣势的 (子) 路径。这个删除劣势路径操作可以大大地减小搜索空间, 并可以在链路权重非负的假设下保证路由上无环。

4.2 多维测度下路径非线性长度所引起的问题

MLBA 的基本步骤与 LBA 的相似, 特殊之处是 MLBA 采用 SAMCRA 作为最短路算法。当 $M=1$ 时, SAMCRA 等同于 Dijkstra 算法, 则 MLBA 等效为 LBA。这种情况下沿 $-P_1$ 引入的负权重不会导致负长度的环路的出现, SAMCRA 中的删除劣势路径操作保证路径长度大于零的环路不会出现在所选择的最短多约束路径上。当 $M \geq 2$ 时, 易证明含有 $-P_1$ 链路的 (子) 路径仍有非负长度, 定理 2 仍然成立, 但是 MLBA 路由过程可能陷入环路。我们仍以图 4 所示的拓扑来说明。设图 4 中每条链路有 2 维权重矢量, 环路 P_l 的权重矢量为

$$\begin{bmatrix} w_1(P_l) = w_1(u_i \rightarrow v_i) + w_1(v_i \rightarrow v_{i+1}) + w_1(v_{i+1} \rightarrow u_i) \\ w_2(P_l) = w_2(u_i \rightarrow v_i) + w_2(v_i \rightarrow v_{i+1}) + w_2(v_{i+1} \rightarrow u_i) \end{bmatrix}$$

由定理 2 可知, 必有 $l(P_l) > 0$ 。若路径矢量的每个分量均大于零, 即 $w_1(P_l) > 0$ 且 $w_2(P_l) > 0$, 当然有子路径 $sv_1 \dots v_i u_i v_{i+1}$ 相对于子路径 $sv_1 \dots v_i v_{i+1}$ 处于劣势。但是, 若环路 P_l 的路径矢量 $(w_1(P_l), w_2(P_l))$ 中有一个分量小于 0, 那么尽管 $l(P_l) = \max(w_1(P_l), w_2(P_l)) > 0$, 子路径 $sv_1 \dots v_i u_i v_{i+1}$ 相对于子路径 $sv_1 \dots v_i v_{i+1}$ 不处于劣势, 也就无

多约束分离路径算法

法通过删除劣势路径的操作避免环路 P_1 出现在所选择的路径上。所以说 $M \geq 2$ 时, MLBA 路由过程可能陷入循环, 直至一个约束不能得到满足为止。

如第 1 节所述 [18,19] 中引入了一个权重变换 $w'(u \rightarrow v) = w(u \rightarrow v) + d(u) - d(v)$ 作用于每条链路, 这里 $d(u)$ 是最短路树上由根节点 s 到某一节点 u 的距离, 这个变换的目的是保证修正图中最短路树上所有链路都有非负权重。但是, $M \geq 2$ 的多维情况下, 路径非线性长度的影响使得最短路的子路径不一定最短[3,21], 因而这个权重变换不能保证链路权重非负, 也就不能避免环路。

由于矢量运算的特点, 在原始图中所发现的多约束最短路 P_1 和修正图中所发现的路径 P_2 均满足约束矢量, 即 $l(P_1) \leq 1$ 且 $l(P_2) = l((P_1' \cup P_2') - P_1) \leq 1$ 时, 并不一定有 $l(P_1') \leq 1$ 且 $l(P_2') \leq 1$ 。因此 MLBA 的解集不一定是可行解。再者, 修正图中不满足约束的路径 P_3 , $l(P_3) = l((P_1'' \cup P_2'') - P_1) > 1$, 也可能对应一对满足约束的分离路径 P_1' 和 P_2' ——MLBA 可能丢失可行解。

为了考察 MLBA 解集的总长, 我们暂且假设约束足够大, 一对节点间的每个可能路径都是可行的。 $M=1$ 时, 显然 MLBA 所得的解集有最小长度; 但是, $M \geq 2$ 时, 由于权重矢量引入的非线性, MLBA 所得的解集不一定有最小长度。考察 $M \geq 2$ 多维情况下的最优解集 $\{P_1', P_2'\}$ 与最短路的的关系, 由(2)式定义的总长可知, 第 3 节中的引理 1 对于 $M \geq 2$ 仍成立, 而且 $\{P_1', P_2'\}$ 仍然具有性质 1, 但是 $l((P_1' \cup P_2') - P_1) \leq l(P_1') + l(P_2') - l(P_1)$ 时, 引理 2 不一定成立, $(P_1' \cup P_2') - P_1$ 不一定是修正图中的最短路 P_2 。那么基于 P_1 和 P_2 所建立起来的解集不一定有最短长度——MLBA 的解集即使可行, 也不一定是最优解集。

以上分析表明, 多维测度的非线性本质使得 MCLPP/MCNPP 问题错综复杂, 精确算法的构造非常困难。

5. 多约束分离路径算法

在前一节中我们已经说明了将 LBA 向多维扩展的困难。鉴于 MCLPP 的 NP 完备特性和 MLBA 中所存在的问题, 我们提出一组解决 MCLPP/MCNPP 问题的启发式算法——分离多约束路由算法 DIMCRA (Disjoint Multiple Constraints Routing Algorithm), 其中 DIMCRA-L1 和 DIMCRA-L2 为链路分离多约束路由算法, DIMCRA-N 是节点分离多约束路由算法。

5.1 DIMCRA-L1

DIMCRA-L1 (G, s, t): 给定有向图 $G(V, E)$, 约束矢量 \bar{L} , 对于源-宿节点对 (s, t) ,

第 1 步. 执行 SAMCRA 寻找一条最短路径 P_1 ;

第 2 步. 将 P_1 上的所有链路方向取反, 并将其链路权重置为负权重, 即 $w_m(v \rightarrow u) = -w_m(u \rightarrow v)$, $\forall (u \rightarrow v) \in P_1, m = 1, \dots, M$, 得到修正图 $G(V, E')$;

第 3 步. 在修正图 $G(V, E')$ 中执行 SAMCRA 寻找一条满足约束矢量 $2\bar{L}$ 的最短路径 P_2 , 若 P_2 不存在, 算法停止, 路径搜索过程中引入环路避免机制;

第 4 步. 取 P_1 和 P_2 的并集, 从中删去反向链路出现在 P_2 上的 P_1 链路和反向链路出现在 P_1 上的 P_2 链路, 将余下的链路组成两条路径 $\{P_1', P_2'\}$ 。

在 MLBA 中要求 P_2 满足约束 \bar{L} , 实际上, 当 $l(P_2) > 1.0$ 时, 若 P_2 不包含 P_1 链路, 那么 $\{P_1', P_2'\}$ 实际上就是 $\{P_1, P_2\}$, 当然不是可行解集; 但是当 P_2 包含部分 P_1 链路时, $\{P_1', P_2'\}$ 有可能是一个可行解集。另一方面, 当 $\bar{w}(P_2) = \bar{w}(P_1' \cup P_2' \cup (-P_1)) \geq \bar{w}(P_1' \cup P_2') - \bar{w}(P_1) > 2\bar{L}$ 时, 必有 $\bar{w}(P_1' \cup P_2') > 2\bar{L}$, 则 P_1' 和 P_2' 至少中有一条路径是不可行的。因此在 DIMCRA-L1 的第 3 步中, 对 P_2 的约束改为 $2\bar{L}$, 以免忽略某些可行解。同时引入环路避免机制, 防止 (SAMCRA 无法避免的) 包含负权重链路的环路出现在路径上。以图 4(a) 所示拓扑为例说明 DIMCRA-L1 的执行步骤, 此拓扑中每条链路为单向链路且由二维权重矢量界定。要求寻找从节点 a 到节点 b 的两条总长最短的链路分离路径, 并且要求每条路径都服从约束矢量 $\bar{L} = (20, 20)$ 。据定义(1), 由 a 到 b 的最短多约束路径为 $acdb$, 其路径权重矢量为 $(4, 5)$ 。据定义(2), 最短链路分离路径的最优解集为 $\{acb, adb\}$, 两条路径的权重矢量分别为 $(5, 6)$ 和 $(5, 5)$, 最小总长为 $0.3 + 0.25 = 0.55$ 。在此拓扑上运行 DIMCRA-L1。第 1 步, 找到最短路 $P_1 = acdb$ 。第 2 步, 将 P_1 替换为 $-P_1$, 得到图 4(b) 所示的修正图。这里, 每条 $-P_1$ 链路的两个权重分量均设为负值, 例如链路 $d \rightarrow c$ 权重矢量为 $(-1, -1)$ 。第 3 步, 找到修正图中的最短路 $P_2 = adcb$, 权重矢量 $(3, 3) + (-1, -2) + (4, 5) = (6, 6)$ 。第 4 步, 将集合 $O_1 = \{(c \rightarrow d), (d \rightarrow c)\}$ 从 P_1 与 P_2 的并集中删除, 得

多约束分离路径算法

到最优解集 $\{acb, adb\}$ 。RF 算法在此拓扑中所得到的解集为 $\{acdb, aeb\}$ ，路径矢量分别为 $(4,5)$ 和 $(11,9)$ ，总长度为 0.8 。

DIMCRA-L1 所形成的解集不一定总是最优解集，甚至不一定总是可行解集。这是由于 $l(P_1) \leq 1.0$ 和 $l(P_2) = l(P'_1 \cup P'_2 \cup (-P_1)) \leq 2.0$ ，可以保证 $l(P'_1) \leq 2.0$ 和 $l(P'_2) \leq 2.0$ ， $l(P'_1) + l(P'_2) \leq 4.0$ 但是不一定保证 $l(P'_1) \leq 1.0$ 且 $l(P'_2) \leq 1.0$ 。因此 DIMCRA-L1 是解决 MCLPP 问题的近似算法，近似比为 2，即两条路径的总长度小于等于最优解集总长度 2 倍。

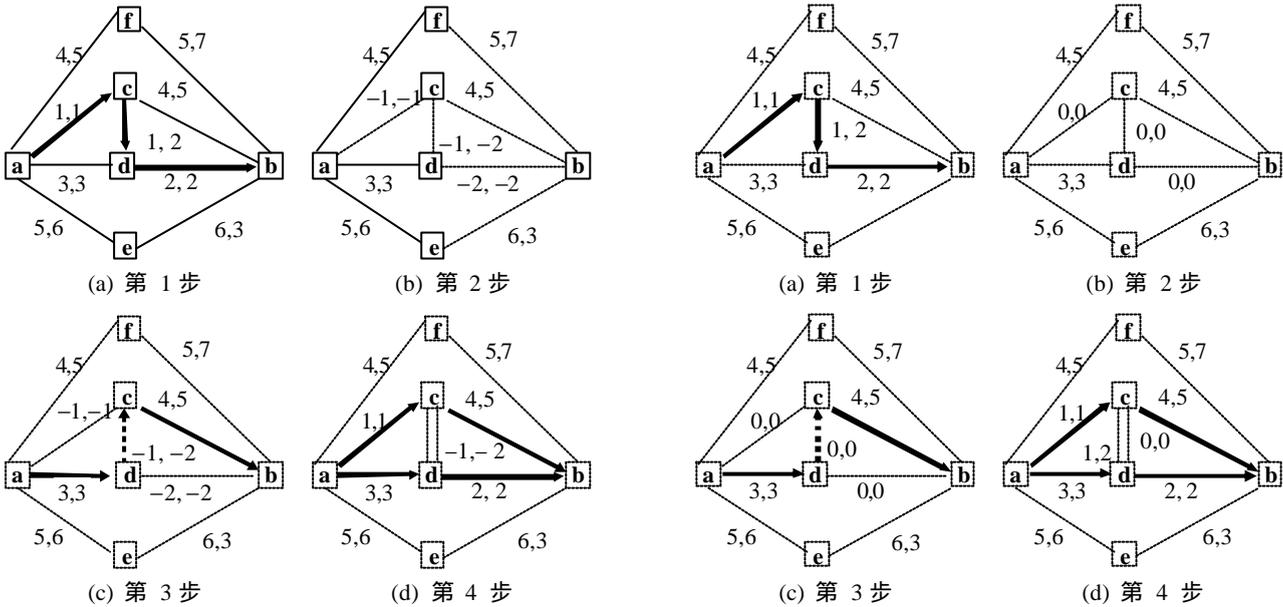


图 4. DIMCRA-L1 的算例

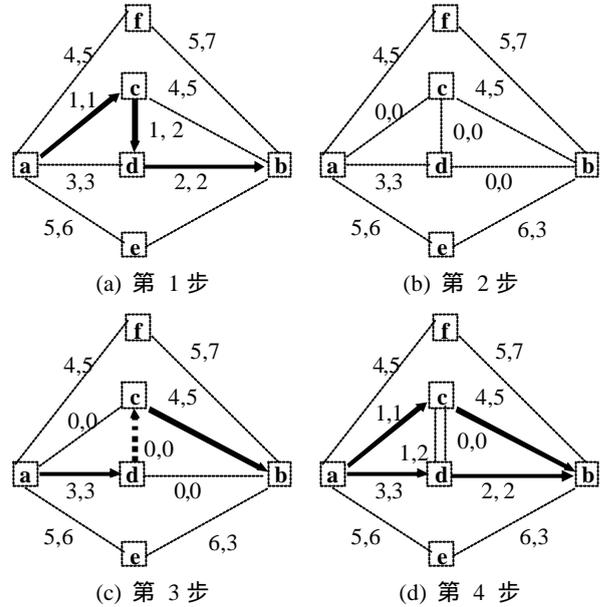


图 5. DIMCRA-L2 的算例 1

5.2 DIMCRA-L2

DIMCRA-L2 (G, s, t): 给定有向图 $G(V, E)$ ，约束矢量 \bar{L} ，对于源-宿节点对 (s, t) ，

第 1 步. 执行 SAMCRA 寻找一条最短路径 P_1 ；

第 2 步. 将 P_1 上的所有链路方向取反，并将其链路权重置为零，即 $w_m(v \rightarrow u) = 0$ ， $\forall (u \rightarrow v) \in P_1$ ， $m = 1, \dots, M$ ，得到修正图 $G(V, E')$ ；

第 3 步. 在修正图 $G(V, E')$ 中执行 SAMCRA 寻找一条满足约束矢量 $2\bar{L}$ 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，从中删去反向链路出现在 P_2 上的 P_1 链路和反向链路出现在 P_1 上的 P_2 链路，将余下的链路组成两条路径 $\{P'_1, P'_2\}$ ；

第 5 步. 检查集合 $\{P'_1, P'_2\}$ 中的每条路径，若路径 P'_i ($1 \leq i \leq 2$) 不满足约束，则从修正图 G 中删去链路集合 $P'_i \setminus (P'_i \cap P_1)$ 得到更新的修正图 $G(V, E')$ ，返回第 3 步；否则，算法停止。

与 DIMCRA-L1 相比，DIMCRA-L2 采用不同的方法得到修正图。在 DIMCRA-L2 的第 2 步，仍然将最短路径上的链路方向置反，但是将反向链路权重置为零而不是负值。这样可以避免因为引入负权重而带来的环路问题，单单依赖 SAMCRA 的删除劣势路径的操作就可避免环路，不必采用专门的环路避免操作。我们设计了 DIMCRA-L2 的第 5 步，来检验第 4 步所构造的解集中的两条路径是否均满足约束。若均满足约束，则 DIMCRA-L2 停止执行；否则回到第 3 步继续搜索。在第 3 步中，若 P_2 不存在，无解，DIMCRA-L2 停止，不会陷入第 3 至 5 步之间的死循环。这样 DIMCRA-L2 的解集必然是可行解集。我们以下列算例说明 DIMCRA-L2 的工作原理。

例 1: 以图 5(a) 中的图例来说明 DIMCRA-L2 的原理。欲在源-宿节点对 (a, b) 之间寻找一对具有最短总长的满足约束 $(20, 20)$ 的链路分离路径。第 1 步，找到最短路 $P_1 = acdb$ 。第 2 步，将 P_1 链路反向并置零权重，

多约束分离路径算法

得到修正图 5(b)。第 3 步，找到修正图中的最短路 $P_2 = adcb$ ，路径权重矢量为 (7,8)。第 4 步，得到两条链路分离 QoS 路径 $\{P'_1, P'_2\} = \{acb, adb\}$ ，如图 5(d)所示。第 5 步，对两条路径分别进行约束检验，二者均服从约束，DIMCRA-L2 停止，得到最优解集 $\{acb, adb\}$ 。对于此例，RF 算法不能得到最优解集。

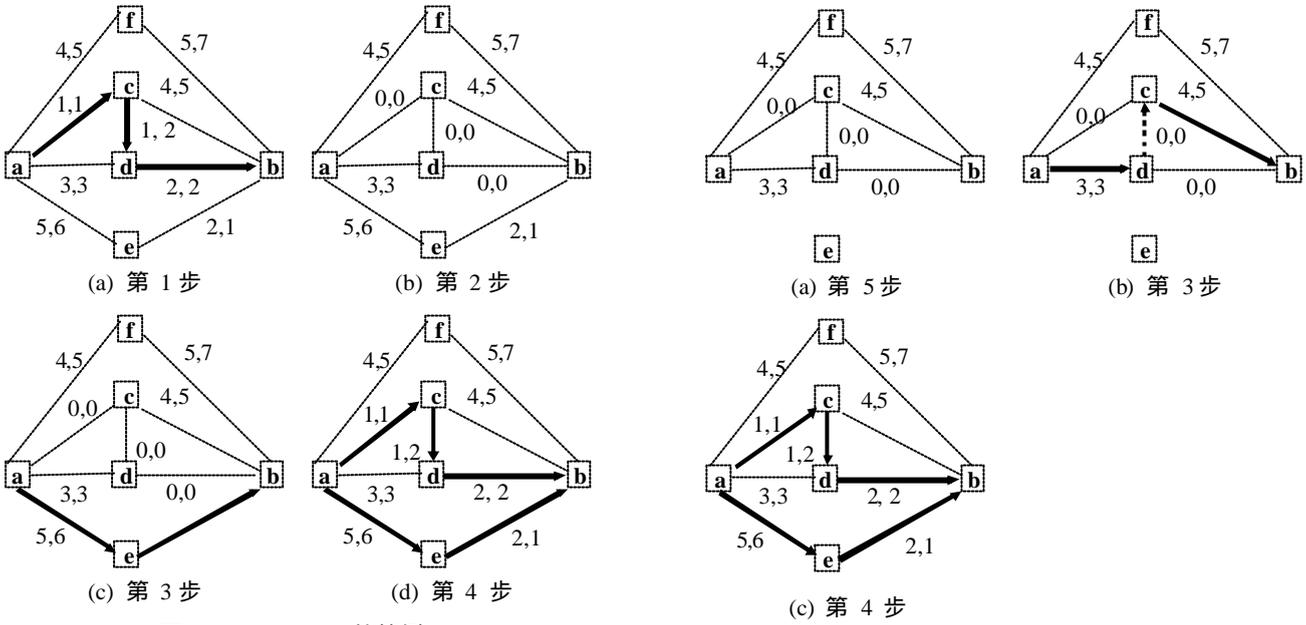


图 6. DIMCRA-L2 的算例 2

图 7. DIMCRA-L2 : 算例 3

例 2：考虑图 6(a)所示的拓扑，除了链路 $e \rightarrow b$ 设置了不同的矢量(2,1)外，与例 1 中的拓扑相同，且要求不变，由图可知本例中的最优解集仍是 $\{adb, acb\}$ ，路径矢量分别为 (5,5) 和 (5,6)，总长为 $5/20 + 6/20 = 0.55$ 。DIMCRA-L2 算法构造出可行解集 $\{P'_1, P'_2\} = \{acdb, aeb\}$ ，见图 6(d)，总长为 $5/20 + 7/20 = 0.6$ 。从总长上来看，DIMCRA-L2 所构造的解集接近最优解集。对于此例，RF 算法得到解集与 DIMCRA-L2 的相同。

例 3：将例 2 中约束矢量改为(6,6)，其他条件和要求不变。运行 DIMCRA-L2 第 1 至 4 步得到与例 2 相同的解集。但是执行第 5 步时，约束检验发现解集 $\{acdb, aeb\}$ 中较长的那条路径 $P'_2 = aeb$ 路径矢量为(7,7)，不符合约束条件。这就是说，当前所构造出的解集不是可行解集。于是从图 6(b)的修正图中删去仅出现在不可行路径 $P'_2 = aeb$ 上的链路 $a \rightarrow c$ 和 $e \rightarrow b$ ，得到更新的修正图 7(a)，DIMCRA-L2 返回第 3 步。重新执行第 3 步，找到当前修正图中的最短路 $P_2 = adcb$ ，见图 7(b)。第 4 步得到解集 $\{acb, adb\}$ ，见图 7(c)。第 5 步检验发现当前解集中的两条路径均服从约束，DIMCRA-L2 停止运行，得到最优解集 $\{acb, adb\}$ 。对于此例，RF 算法既不能得到最优解集，也不能得到可行解集。

DIMCRA-L2 将最短路 P_1 方向取反，在修正图中寻找最短路 P_2 ，并在 P_1 和 P_2 的基础上构造解集的过程保证解集中的两条路径必然是相互分离的。将反向的 P_1 链路置零权重避免了因负权重所引起的环路，而在非负权重图中 SAMCRA 的删除劣势路径操作可以保证所选择的路由上不会出现环路 [28]。这一做法的代价是，虽然仍鼓励选择反向 P_1 链路，但是与将反向的 P_1 链路置负权重的做法相比，这种鼓励的力度(intensity)减小了。

由例 1 可见当最优解集的总长与其他可行解集相比差距较明显时，DIMCRA-L2 可搜索到最优解。如例 2 所示当最优解集的总长接近于其他可行解集时，DIMCRA-L2 不一定能做出最优选择，这说明 DIMCRA-L2 还存在优化的空间。

DIMCRA-L2 算法的约束检验保证其解集必然是可行解。例 3 说明，DIMCRA-L2 的第 5 步通过对当前解集中的每条路径进行约束检验，保证 DIMCRA-L2 返回一个由两条链路分离的多约束路径所组成的可行解集。就其目前的状态而言，DIMCRA-L2 优于 RF 算法，这一点在上述的例子中已经得到说明。DIMCRA-L2 要么给出优于 RF 算法的解，要么是在 RF 算法无解时给出可行解甚至最优解。

5.3 DIMCRA-N 算法

为了保证网络节点故障时业务的不间断性,或者在网络节点间均衡分布业务流,需要提供节点分离路径。采用 RF 法寻找节点多约束分离路径,在原始图中选择第一最短多约束路径 P_1 后,必须删除 P_1 上的全部节点和链路,才能保证路径的节点分离性,而这种删除操作极易造成修正图的不连通致使算法无解。因此采用增广路径的方法进行节点分离多约束路径的构造更有必要性。利用节点分裂的概念[19],我们可以将 DIMCRA-L2 扩展为解决 MCNPP 问题的算法——DIMCRA-N 算法,在一对源宿节点之间建立两条节点分离的多约束路径。

DIMCRA-N (G, s, t): 给定有向图 $G(V, E)$, 约束矢量 \vec{L} , 对于源-宿节点对 (s, t) ,

第 1 步. 将节点 $v \in V, v \neq s, t$, 分裂为由零权重链路 $v' \rightarrow v''$ 连接的两个节点 v' 和 v'' , 对于 $u \in V, u \neq v$, 将 $u \rightarrow v$, 替换为 $u \rightarrow v'$, 将 $v \rightarrow u$ 替换为 $v'' \rightarrow u$, 链路权重不变, 得到修正图 G_1 ;

第 2 步. 在图 G_1 上运行 DIMCRA-L2 算法;

第 3 步. 将 DIMCRA-L2 解集中的分裂节点对 v' 和 v'' 合并为 v , 删除链路 $v' \rightarrow v''$, 得到 (s, t) 间的两条节点分离多约束路径。

DIMCRA-N 算法的第 1 步所进行节点分裂操作,将原始图中除了源宿节点之外的每个节点替换为一对由零权重链路相连接的节点,同时进行相应的链路替换,得到一个修正图 G_1 。第 2 步在修正图 G_1 中运行链路分离路径算法 DIMCRA-L2, 所得到的解集是 G_1 中的一对链路分离多约束路径。DIMCRA-L2 解集的链路分离特性可以保证任何一条链路不会同时出现在解集中的两条路径上。第 3 步将所得到的解集恢复到原始图 G 中,那么原始图中的每个节点就不可能同时出现在解集中的两条路径上,从而保证了节点分离性。

以图 8(a) 的拓扑说明 DIMCRA-N 的运行过程。要求建立源-宿对 (a, b) 之间的一对节点分离多约束路径, 约束矢量 $\vec{L}=(10,10)$ 。执行第 1 步的节点分裂操作后得到图 8(b) 的修正拓扑 G_1 ; 第 2 步运行 DIMCRA-L2 算法得到 G_1 中的链路分离多约束路径对 $\{ac'e'e''b, ad'd''b\}$, 如图 8(c) 所示; 第 3 步得到原始图 G 中的节点分离路径对 $\{aceb, adb\}$, 如图 8(d) 所示, 路径矢量分别为 $(6,4)$ 和 $(8,2)$, 总长度为 1.4, 为最优解集。RF 算法在此拓扑中无解。

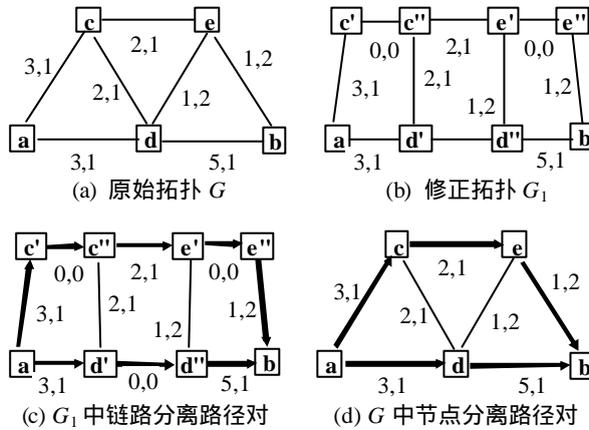


图 8. DIMCRA-N 的算例

6. 结论

在强调可生存性、负荷均衡和网络资源利用率优化的网络设计中,链路/节点分离路径问题是不可避免的一个问题。在 QoS 路由领域中,这一问题尚无论述。由于 QoS 路由中,一条路径的质量是以多个测度来衡量的,由此引入了路径长度的非线性。本文分析了解决 SLPP 问题的增广路径方法及其相对于 RF 方法的优越性,讨论了将增广路径方法直接拓展到多维路由这一方法所存在的问题,分析了构造精确算法的困难性,并提出了一组启发式算法 DIMCRA-L1、DIMCRA-L2 和 DIMCRA-N,解决 $M \geq 2$ 的 MCLPP 问题和 MCNPP 问题。DIMCRA-L1 算法可得到近似比不大于 2 的链路分离多约束路径, DIMCRA-L2 算法则可以得到两条分离路径均满足多维约束的可行解集。DIMCRA-N 算法能够实现节点分离 QoS 路由,得到满足多维约束的节点分离路径对。就路径的总长度而言, DIMCRA-L1、DIMCRA-L2 和 DIMCRA-N 算法的解不一定最优。但是这些算法的性能均优于 RF 算法。我们在前面说明这些算法的基本思想时,都是以 2 维权重矢量为例的,

多约束分离路径算法

但实际上这些算法对任意的 $M > 2$ 均适用,而且这些算法可以很容易地扩展为在一对源宿节点之间发现 K 条链路/节点分离路径算法。这些算法不仅可在要求快速业务恢复能力的网络(如光网络和 MPLS/GMPLS 网络)中用于提供主用路径和保护路径,还可以提供多条分离路径用于业务流的承载,实现负载均衡。如何在保证有效性的前提下将 DIMCRA 改进为可获得最短链路/节点分离多约束路径的精确算法、如何在不能发现节点/链路分离多约束路径时发现最大程度分离路由、是否可以引入局部恢复[2]的概念等问题还有待进一步研究。

参考文献

- [1] R. Bhandari, "Optimal Diverse Routing in Telecommunication Fiber Networks", Proc. IEEE INFOCOM' 94, June 1994, Toronto, Ontario, Canada, Vol.3, pp.1498-1508.
- [2] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi and A. Sprintson, "Algorithms for Computing QoS Paths with Restoration", Proc. of IEEE INFOCOM' 03, April 2003.
- [3] H. De Neve, P. Van Mieghem, "TAMCRA: a tunable accuracy multiple constraints routing algorithm", Computer Communications, vol. 23, No. 7, Mar. 2000, pp 667-679.
- [4] R. Diestel, "Graph Theory", Graduate Texts in Mathematics, 1997, Springer-Verlag New York.
- [5] L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.
- [6] M. R. Garey, D. S. Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness", Freeman, San Francisco, 1979.
- [7] K.P. Gummadi, M.J. Pradeep and C.S.R. Murthy, "An Efficient Primary-Segmented Backup Scheme for Dependable Real-Time Communication in Multihop Networks", ACM/IEEE Transactions on Networking, vol. 11, no. 1, pp. 81-94, February 2003.
- [8] Y. Guo, F. A. Kuipers, P. Van Mieghem, "A Link disjoint paths algorithm for reliable QoS routing", International Journal of Communication Systems, vol. 16, no. 9, pp. 779-798, November 2003.
- [9] K. Kar, M. Kodialam, T. V. Lakshman, "Routing Restorable Bandwidth Guaranteed Connections using Maximum 2-Route Flows", IEEE INFOCOM2002.
- [10] M. Kodialam, T. V. Lakshman, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration", IEEE INFOCOM2000.
- [11] M. Kodialam, T.V. Lakshman, "Restorable Dynamic Quality of Service Routing", IEEE Communications Magazine, pp. 72-81, June 2002.
- [12] S.W. Lee, C. S. Wu, "A k-best paths algorithm for highly reliable communication networks", IEICE Trans. on Commun., Vol. E82-B, No.4, Apr. 1999, pp.586-580.
- [13] W. Liang, "Robust routing in wide-area WDM networks", Proc. of 15th Int'l Parallel and Distributed Processing Symp., San Francisco, April, IEEE Computer Society, 2001.
- [14] C-C Lo and B-W Chuang, "A Novel Approach of Backup Path Reservation for Survivable High-Speed Networks", IEEE Communications Magazine, March 2003.
- [15] R.G. Ogier, V. Rutenburg, N. Shacham, "Distributed algorithms for computing shortest pairs of disjoint paths", IEEE Trans. on Information Theory, Vol. 39, No. 2, Mar. 1993, pp. 443-455.
- [16] E. Oki, N. Yamanaka, "A recursive matrix calculation method for disjoint path search with hop link number constraints", IEICE Trans. Commun., Vol. E78-B, No.5, May 1995, pp. 769-774.
- [17] C.H. Papadimitriou and K. Steiglitz, Combinatorial Optimization - Algorithms and Complexity, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
- [18] J.W. Suurballe, "Disjoint Paths in a Network", Networks, Vol. 4, 1974, pp. 125-145.
- [19] J.W. Suurballe, R.E. Tarjan, "A Quick Method for Finding Shortest Pairs of Disjoint Paths", Networks, Vol. 14 (1984), pp. 325-333.
- [20] N. Taft-Plotkin, B. Bellur, R. Ogier, "Quality-of-Service Using Maximally Disjoint Paths", Proceedings of IWQoS (International Workshop on Quality-of-Service), Jun. 1999.
- [21] P. Van Mieghem, H. De Neve, F. Kuipers, "Hop-by-hop Quality of Service Routing", Computer Networks, Vol. 37, No 3-4, 2001, pp. 407-423.
- [22] Z. Wang, J. Crowcroft, "QoS Routing for supporting Multimedia Applications", IEEE J. Selected Areas in Communications, Vol. 14, No.7, Sep. 1996, pp. 1188-1234.