

多约束最短链路分离路径精确算法^{*}

熊轲^{1,2+}, 裘正定¹, 郭宇春³, 张宏科⁴, 秦雅娟⁴

¹(北京交通大学 信息科学研究所,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)

⁴(北京交通大学 下一代互联网互联设备国家工程实验室,北京 100044)

Exact Algorithm for Multi-Constrained Shortest Link-Disjoint Paths

XIONG Ke^{1,2+}, QIU Zheng-Ding¹, GUO Yu-Chun³, ZHANG Hong-Ke⁴, QIN Ya-Juan⁴

¹(Institute of Information Science, 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)

⁴(National Engineering Laboratory on Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing 100044, China)

+ Corresponding author: E-mail: kxiong@mail.tsinghua.edu.cn

Xiong K, Qiu ZD, Guo YC, Zhang HK, Qin YJ. Exact algorithm for multi-constrained shortest link-disjoint paths. *Journal of Software*, 2010,21(7):1744–1757. <http://www.jos.org.cn/1000-9825/3589.htm>

Abstract: Finding two link-disjoint QoS paths (primary and backup) between source-destination pairs is one of the most significant schemes to provide reliable QoS routing. Current algorithms for seeking multi-constrained link-disjoint path pair (MCLPP) can not always make sure to find the feasible solutions in networks. To solve this problem, this paper analyzes the properties of the optimal solution of MCLPP problem, and then proposes a design principle for the exact algorithm. Based on the design principle, an exact algorithm called link-disjoint optimal multi-constrained paths algorithm (LIDOMPA) is presented, which is able to find multi-constrained shortest link-disjoint path pair for arbitrary networks. To reduce the complexity, this paper introduces three key concepts: The candidate optimal solution, the constricted constraint vector and the structure-aware non-dominance, which effectively reduce the search space of LIDOMPA without loss of exactness. Extensive experiments show that LIDOMPA outperforms the existing algorithms in terms of the ability of obtaining solutions and achieves acceptable running time overhead.

* Supported by the National Natural Science Foundation of China under Grant No.60772043 (国家自然科学基金); the National Basic Research Program of China under Grant Nos.2007CB307101, 2007CB307106 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z202 (国家高技术研究发展计划(863)); the Program of Introducing Talents of Discipline to Universities of China under Grant No.B08002 (高等学校学科创新引智计划); the Cultivation Fund of the Key Scientific and Technical Innovation Project of the Ministry of Education of China under Grant No.706005 (国家教育部科技创新工程重大项目)

Received 2008-10-20; Revised 2008-11-27; Accepted 2009-01-20

Key words: QoS routing; network reliability; link-disjoint paths; multi-constrained routing; optimal solution

摘要: 在通信的源和目的间寻找两条(主用和备用)链路分离的 QoS 路径是提供可靠 QoS 路由的重要途径. 现有求解多约束链路分离路径对(multi-constrained link-disjoint path pair,简称 MCLPP)的算法难以保证求得存在于任意网络中的可行解和最优解.为解决这一问题,分析了 MCLPP 问题最优解的性质,提出了精确算法的设计原则,在此基础上给出了求解 MCLPP 问题的精确算法(link-disjoint optimal multi-constrained paths algorithm,简称 LIDOMPA 算法),可对任意网络求解客观存在的多约束最短链路分离路径对.为了降低算法的复杂性,引入了候选最优解、紧缩的约束向量和结构化的路径支配 3 种关键方法,在保障算法精确性的同时,有效地降低了 LIDOMPA 的搜索空间.大量的实验结果表明,LIDOMPA 的求解能力优于现有算法,同时可以实现较低的算法执行时间开销.

关键词: 服务质量路由;网络可靠性;链路分离路径;多约束路由;最优解

中图法分类号: TP393 **文献标识码:** A

随着多媒体实时业务的不断兴起和发展,网络的可靠性问题和服务质量(QoS)保障问题已经成为有线网络和无线网络研究共同关注的热点和焦点^[1-4].为通信的一个连接提供满足 QoS 约束的两条链路分离路径,一条作主用,另外一条作备用,当主用路径故障后,将业务流迅速切换到备用路径上传输,被认为是同时提高网络可靠性和 QoS 保障的重要方法^[1-7].此外,寻找满足 QoS 约束的链路分离路径也是实现网络流量工程、负载均衡和拥塞避免的重要手段.

本文研究的多约束链路分离路径对(multi-constrained link-disjoint path pair 简称 MCLPP)问题^[3],目的就是在通信的源和目的之间寻找满足多个 QoS 约束的链路分离路径对.当所求链路分离路径对变成单条路径时,MCLPP 问题就退化为经典的多约束路径(multi-constrained path 简称 MCP)路由问题.然而,MCLPP 问题要比 MCP 问题复杂得多,原因是 MCLPP 问题除了要考虑路径的 QoS 约束以外,还要考虑路径间的结构关系.这里,约束通常是指加性约束,因为在 QoS 的凹性(如带宽等)、加性(如延时、延时抖动等)和乘性(如丢包率等)3 种约束中,加性约束处理起来最为复杂.当加性约束个数大于 1 时,QoS 路由就变成了 NP 问题^[8,9].

近年来,QoS 约束下链路分离路径问题受到了研究者的广泛关注^[1-7],然而大多数研究是针对单个 QoS 约束——延时(delay)展开的,旨在源和目的间找出一对链路分离路径,在满足 delay 约束的同时达到两条路径总的花费(cost)最小.当给定的 delay 约束针对两条路径的端到端延时总和时,问题被称为 DCLDOP-I(delay constrained link disjoint optimal paths),当给定的 delay 约束针对路径对中每条路径的端到端延时时,问题被称为 DCLDOP-II.文献[1]对 DCLDOP-I 和 DCLDOP-II 问题进行了建模,证明了这两种问题同属于 NP 完全问题.文献[4]针对 DCLDOP-I 问题提出了两种近似求解算法.文献[5]研究了总延时受限下的 k 条 cost 最小链路分离路径问题.文献[6]提出了 Min-Min 问题,旨在求解两条满足 QoS 约束的分离路径且满足较短的路径 cost 最小.文献[7]通过求解总延时最小的链路分离路径对来解决单链路失效后的路由恢复问题.

算法的有效性和精确性直接决定着可靠 QoS 路由的发展和部署.然而,现有关于 MCLPP 问题的算法^[3]相对较少且都不够完善,难以保证求得网络中客观存在的可行解.针对此问题,本文从对 MCLPP 问题最优解的性质和构造形式分析入手,设计出了精确算法,可以保证求得任意网络中客观存在的 MCLPP 问题的最优解.

文章的主要贡献如下:1) 深入分析了多约束链路分离问题最优解的性质和构造形式,在此基础上提出了精确算法的设计原则;2) 依据 1)的结论,提出了针对 MCLPP 问题的精确算法 LIDOMPA(link-disjoint optimal multi-constrained paths algorithm);3) 为优化算法性能,在 LIDOMPA 的设计过程中,从解的构造形式、路径长度等限制条件出发,提出了候选最优解、紧缩的约束向量和结构化的路径支配 3 种方法来降低算法的搜索空间和复杂度;4) 理论分析并证明了 LIDOMPA 为求解 MCLPP 问题的精确算法;5) 首次对 MCLPP 问题的现有算法 DIMCRA^[3]进行了实现,通过大量实验,对 DIMCRA 和本文所提算法 LIDOMPA 进行了比较,并讨论了 LIDOMPA 在最坏情况下的算法执行时间.实验结果表明,本文所提算法比现有算法具有更好的求解能力,所用仿真时间开

销较低.

本文第 1 节给出 MCLPP 问题的数学描述和相关工作.第 2 节讨论 MCLPP 最优解的性质和构造形式,并给出精确算法的设计原则.第 3 节介绍 LIDOMPA 算法的步骤和关键处理技术.第 4 节分析 LIDOMPA 是如何在降低搜索空间的同时保证其精确性的.第 5 节通过实验对算法性能进行比较和分析.第 6 节对全文进行总结.

1 问题描述及相关工作

将网络表示为加权有向图 $G(V,E)$, V 代表 G 中节点的集合, E 代表边的集合. $u \rightarrow v$ 表示 G 中节点 u 到 v 的一条加权有向边. $\bar{w}(u \rightarrow v) = [w_1(u \rightarrow v), w_2(u \rightarrow v), \dots, w_m(u \rightarrow v)]$ 表示边 $u \rightarrow v$ 所带的 m 维加权向量. s 和 t 分别代表源和目的节点. P 代表从 s 到 t 的一条路径, P 同时也代表了路径 P 上链路的集合. $V(P)$ 表示路径 P 上节点的集合. 路径 P 的权向量为 $\bar{w}(P) = [w_1(P), w_2(P), \dots, w_m(P)]$, 其中 $w_i(P)$ 表示 P 的第 i 维权重:

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

链路分离路径(link-disjoint paths). 若 $P_a \cap P_b = \emptyset$, 则称 P_a 与 P_b 为链路分离路径.

非线性长度(non-linear length)^[10]. G 为 m 维加权有向图, 给定约束向量 $\bar{C} = [C_1, C_2, \dots, C_m]$, G 中路径 P 的非线性长度定义为

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

式(2)中的非线性长度是一个被约束向量 \bar{C} 归一化的路径长度, $l(P) > 1.0$ 表示路径 P 至少有 1 维权重不满足 \bar{C} 的约束. 本文的目的是寻找一对满足多约束的链路分离路径, 故定义两条路径 P_a 和 P_b 的长度总和为

$$l(P_a) + l(P_b) \quad (3)$$

多约束链路分离路径对(MCLPP)问题. 给定 m 维加权有向图 G 和 m 维约束向量 $\bar{C} = [C_1, C_2, \dots, C_m]$ ($m \geq 2$), MCLPP 问题的目的是在源 s 和目的 t 之间寻找一对路径 $\{P, P'\}$, 要求 $P \cap P' = \emptyset$ 且 $w_i(P) \leq C_i$, $w_i(P') \leq C_i$ ($1 \leq i \leq m$).

满足 MCLPP 问题要求的路径对 $\{P, P'\}$ 称为 MCLPP 问题的可行解. 在实际网络中, s 和 t 之间可能同时存在多组可行解. 而在路径选择时, 往往希望选取路径总长最小的一组. 当 $l(P) + l(P')$ 为可行解中最小时, $\{P, P'\}$ 称为 MCLPP 问题的最优解, 即多约束最短链路分离路径对.

MCLPP 问题由 Guo^[3]等人提出, 作者证明了该问题为 NP 完全问题, 分析强调了链路分离路径算法由 1 维约束向多维约束扩展的难度, 并给出了求解 MCLPP 问题的启发式算法 DIMCRA(link-disjoint multiple constraints routing algorithm)^[3]. 到目前为止, 关于 MCLPP 问题的算法相对较少, 一种是 RF(remove-find)算法^[3], 另一种是 DIMCRA 算法.

RF 算法是求解 MCLPP 的最直观算法, 步骤为: 第 1 步在原图 G 上解出 s, t 间满足 \bar{C} 的最短路径 P_1 ; 第 2 步从 G 上删去 P_1 的所有链路得到修正图 G' ; 第 3 步在 G' 上找出 s, t 间满足 \bar{C} 的最短路径 P_2 , 若 P_2 存在, (P_1, P_2) 为 RF 算法的最终解, 否则无解. RF 算法虽然能够保证所求解一定为链路分离路径, 却难以保证获得任意网络中客

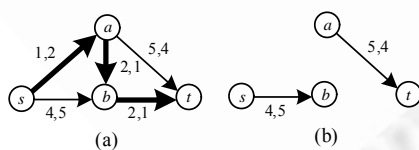


Fig.1 A failed example of RF algorithm
图 1 RF 算法无法求解的算例

观存在可行解. 如图 1 所示, 图 1(a) 为 2 维加权有向网络, $\bar{C} = [7, 7]$. RF 第 1 步在图 1(a) 上计算出最短路径为 $sabt$, 其权向量为 $(5, 4)$, 非线性长度为 $5/7$. 第 2 步从图 1(a) 上删去 $sabt$ 的链路得到如图 1(b) 所示修正图. 第 3 步因无法在图 1(b) 上求得 s 到 t 的 P_2 路径, 算法终止, 返回无解. 实际上, 此例中满足 \bar{C} 的最优解是存在的, 即 $\{sat, sbt\}$, 而 RF 算法并不能成功求解.

DIMCRA 算法由单约束的链路分离路径算法 LBA(link-disjoint version of Bhandari's algorithm)^[3]演化而来, 对 RF 算法有了一定的改进. 给定加权有向图 $G(V,E)$ 和约束向量 \bar{C} , DIMCRA 的步骤为: 第 1 步执行 SAMCRA(self-adaptive multi-constrained routing algorithm)^[11]算法在 G

上找到满足 \bar{C} 的最短路径 P_1 ,若 P_1 不存在,算法终止;第 2 步将 P_1 链路方向取反,并重置取反的链路权重为 0,即 $w_i(v \rightarrow u) = 0, 1 \leq i \leq m, \forall (u \rightarrow v) \in P_1$,得到修正图 G' ;第 3 步在 G' 上执行 SAMCRA 寻找满足约束向量 $2\bar{C}$ 的最短路径 P_2 ,若 P_2 不存在,算法终止;第 4 步取 P_1 和 P_2 的并集 $P_1 \cup P_2$,从中删去反向链路出现在 P_1 上的 P_2 链路和反向链路出现在 P_2 上的 P_1 链路,将余下的链路组成两条路径 $\{P'_1, P'_2\}$;第 5 步检查 $\{P'_1, P'_2\}$ 中的两条路径,若路径 $P'_i (i=1,2)$ 不满足约束,则从 G' 中删去 $P'_i - (P_1 \cap P'_i)$ 集合中的链路,得到新的修正图,返回第 3 步,否则,算法停止。

由于 DIMCRA 的第 2 步采用了反向零权重链路的方法代替了 RF 第 2 步的链路删除操作,在一定程度上保持了网络的连通性,因而能够更好地解决 MCLPP 问题.但是 DIMCRA 同样不能保证对任意结构的网络求解客观存在的可行解.如图 2 所示,图 2(a)为原网络 G ,约束向量 $\bar{C} = [6, 6.5]$. DIMCRA 第 1 步在图 2(a)上求得最短路径 $P_1 = sbct$.第 2 步将 P_1 链路反向并置权重为 0,得到图 2(b)所示修正图.第 3 步在图 2(b)上再次运行 SAMCRA 求得最短路径 $P_2 = scet$.由于 $P_1 \cap P_2 = \emptyset$,第 4 步所求链路分离路径对为 $(sbct, scet)$.第 5 步,因 $scet$ 的路径权向量 $(8, 8)$ 不满足 \bar{C} ,故 $scet - \{scet \cap sbct\}$ 上的链路被删去,得到如图 2(c)所示修正图.然后返回第 3 步,在图 2(c)上求解新的最短路径为 $sabt$. $P_1 \cap sabt = \emptyset$,所以第 4 步所得链路分离路径对为 $\{sbct, sabt\}$.第 5 步,因 $sabt$ 的路径权向量 $(11, 7)$ 不满足 \bar{C} ,故删去 $sabt - \{sabt \cap sbct\}$ 上的链路得到如图 2(d)所示修正图,然后再返回第 3 步.由于无法在图 2(d)上求得 s 到 t 的路径,算法终止,返回无解.实际上,此例中满足 \bar{C} 的最优解存在,即 $\{sbt, sct\}$,而 DIMCRA 并未能成功求解。

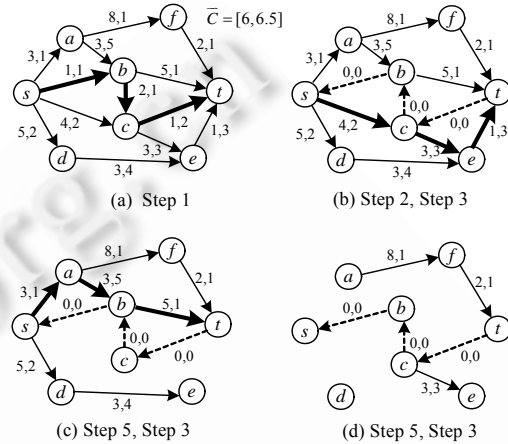


Fig.2 A failed example of DIMCRA
图 2 DIMCRA 无法求解的算例

综上所述, MCLPP 问题的现有算法都不能保证对任意网络求得客观存在的可行解,原因是它们在求解过程中有链路删除操作(如 RF 算法的第 2 步和 DIMCRA 算法的第 5 步),会破坏原网络的连通性,从而导致可行解的丢失.针对此问题,本文对 MCLPP 问题进行了深入研究,提出了求解最优解的精确算法。

2 MCLPP 问题精确算法的设计原则

本节将研究 MCLPP 问题最优解性质和构造形式,并在此基础上讨论精确算法的设计原则。

2.1 文中所用符号定义

定义 1(P_1 路径). P_1 表示原始网络 G 上从源 s 到目的节点 t 之间满足约束 \bar{C} 的最短路径。

定义 2(修正图 G'). G' 表示将 G 上的 P_1 链路反向并置反向链路权重为 0 后的修正图。

定义 3(\bar{P}_1 路径). \bar{P}_1 表示 P_1 的反向路径,由 P_1 的反向零权重链路组成。

定义 4(零 P_1 节点路径). 若 $V(P) \cap V(P_1) = \emptyset^{**}$,称路径 P 为零 P_1 节点路径。

定义 5(零 \bar{P}_1 链路路径). 若 $P \cap \bar{P}_1 = \emptyset$,称路径 P 为零 \bar{P}_1 链路路径。

显然以下结论成立:a) $V(P_1) = V(\bar{P}_1)$; b) $P_1 \cap \bar{P}_1 = \emptyset$; c) 若 P 属于零 P_1 节点路径, P 必属于零 \bar{P}_1 链路路径。

定义 6(P_2 路径). P_2 表示修正图 G' 上从 s 到 t 的路径。

定义 7(共用反向链路集合). $P_1 \bar{\cup} P_2 = (P_1 \cap \bar{P}_2) \cup (P_2 \cap \bar{P}_1)$,其中 \bar{P}_2 为 P_2 的反向路径。

定义 8(共用链路关联节点集合). $V(P_a \cap P_b)$ 表示与集合 $P_a \cap P_b$ 中的链路直接关联的节点的集合。

** 本文用 $V(P) \cap V(P_1) = \emptyset$ 表示,除源和目的节点外, P 和 P_1 无其他共用节点。

定义 9(p_b 子路径和 p_a 子路径). p_b 表示 P_1 路径上距 s 最近的出度大于1的节点与 s 间的 P_1 的子路径; p_a 表示 P_1 路径上距 t 最近的入度大于1的节点到 t 之间的 P_1 的子路径.

如图3所示, $P_1=sabcxyz$, $p_b = abc$, $p_a = yzt$.

定义 10(对称差子路径 **SDSP:symmetric difference sub-paths**). P_a 和 P_b 为 G' 上从源 s 到中间节点 j 间的两条路径.若对称差集合 $P_a \oplus P_b = (P_a - P_b) \cup (P_b - P_a)$ 上所有链路同属一个弱连通分支,则称由 $P_a \oplus P_b$ 上的链路组成的两条路径 $\tilde{P}_a = P_a \cap (P_a \oplus P_b)$ 和 $\tilde{P}_b = P_b \cap (P_a \oplus P_b)$ 为 P_a 和 P_b 的对称差子路径.

如图4例所示,图4(a)为子图 $P_a \cup P_b$,图4(b)为子图 $P_a \oplus P_b$. P_a 和 P_b 的SDSP(symmetric difference sub-paths)为 $\tilde{P}_a = cdt$, $\tilde{P}_b = cet$.

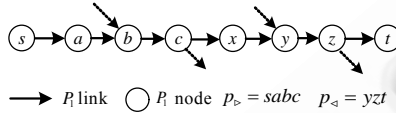


Fig.3 An example of p_b and p_a subpaths

图3 p_b 子路径与 p_a 子路径示例

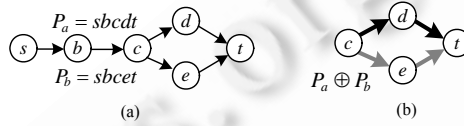


Fig.4 An example of symmetric difference sub-paths

图4 对称差子路径示例

2.2 MCLPP最优解的性质、构造形式及精确算法的设计原则

定理 1. 给定加权有向图 G ,若节点对 (s,t) 间MCLPP问题的最优解 $\{P_1^*, P_2^*\}$ 存在,则 $\{P_1^* \cup P_2^*\} \cap P_1 \neq \emptyset$.

证明:(反证法)假设 $\{P_1^* \cup P_2^*\} \cap P_1 = \emptyset$,则 $P_1^* \cap P_1 = \emptyset$ 和 $P_2^* \cap P_1 = \emptyset$ 成立.因为 $P_1^* \cap P_2^* = \emptyset$,故 P_1^* , P_2^* 和 P_1 两两链路分离.由于 P_1 为最短路径,可得 $l(P_1) + l(P_i^*) < l(P_1^*) + l(P_2^*) (i=1,2)$ 成立.这与 $\{P_1^*, P_2^*\}$ 是最优解矛盾,故假设不成立,定理1得证. □

推论 1. 给定加权有向图 G ,若节点对 (s,t) 间MCLPP问题的最优解 $\{P_1^*, P_2^*\}$ 存在,则 $P_1^* \cap P_1 = \emptyset$,当且仅当 $P_2^* = P_1$ 时成立(P_1^* 和 P_2^* 可以互换).

证明:(必要性证明) $P_1^* \cap P_1 = \emptyset$,假设 $P_2^* \neq P_1$,则最优链路分离路径对为 $\{P_1, P_1^*\}$,这与 $\{P_1^*, P_2^*\}$ 是最优解矛盾,故假设不成立.将 P_1^* 和 P_2^* 互换,上述证明仍成立,故必要性得证.

(充分性证明)因为 $P_1^* \cap P_2^* = \emptyset$,若 $P_1^* = P_1$,则 $P_1 \cap P_2^* = \emptyset$ 成立,将 P_1^* 和 P_2^* 互换,上述证明仍成立,充分性得证.故推论1成立. □

定理1和推论1表明了MCLPP的最优解 $\{P_1^*, P_2^*\}$ 的链路集合必然包含 P_1 的部分或全部链路.

定义 11(候选最优解 **COS:candidate optimal solution**). 给定加权有向图 G , $\{P_1', P_2'\}$ 为节点 s 到 t 间MCLPP问题的可行解,若 $\{P_1' \cup P_2'\} \cap P_1 \neq \emptyset$,则称 $\{P_1', P_2'\}$ 为MCLPP问题的候选最优解.

记候选最优解集合为 S_{cos} ,可行解集合为 $S_{feasible}$,显然 $S_{cos} \subseteq S_{feasible}$.结合定义11,定理1和推论1实际说明了只有COS才可能成为MCLPP问题的最优解.换句话说,要找到最优解,不必搜索所有可行解,只需构造完整的COS解集即可.因此,可以通过只构造COS的方法来降低最优解的搜索空间.由于COS必然包含 P_1 链路,故可用 P_2 路径和 P_1 来构造COS,可能的构造形式由定理2给出.

定理 2. 给定加权有向图 G ,若 P_2 可与 P_1 构造MCLPP的COS, P_2 与 P_1 必然满足以下4种结构关系之一:

- ①: $\bar{P}_1 \cap P_2 = \emptyset$ 且 $V(P_1) \cap V(P_2) = \emptyset$; ②: $\bar{P}_1 \cap P_2 = \emptyset$ 且 $V(P_1) \cap V(P_2) \neq \emptyset$;
- ③: $\bar{P}_1 \cap P_2 \neq \emptyset$ 且 $V(P_1) \cap V(P_2) = V(\bar{P}_1 \cap P_2)$; ④: $\bar{P}_1 \cap P_2 \neq \emptyset$ 且 $V(P_1) \cap V(P_2) \neq V(\bar{P}_1 \cap P_2)$.

定理2的证明见本文附录.图5(a)~图5(d)分别给出了定理2中4种关系的最简单结构示例.如图5(a)所示,

在①的结构上, $P_2 \cup P_1$ 可构成一对链路分离路径,因而最多存在 1 组 COS.在②的结构上, $P_2 \cup P_1$ 上的链路可构造 2^q 组不同的链路分离路径对,因而最多存在 2^q 组 COS,其中 $q = |V(P_1) \cap V(P_2)|$, $|\cdot|$ 表示集合“ \cdot ”中的元素个数.如图 5(b)所示, $V(P_1) \cap V(P_2) = \{b\}$, 可构成 2^1 组链路分离路径对,即 $\{sabct, sbt\}$ 和 $\{sabt, sbct\}$.在③的结构上,从集合 $P_1 \cup P_2$ 中删除反向链路集合 $P_1 \amalg P_2$ 中的链路,余下链路可构成一组链路分离路径对.结构④可通过从 $P_1 \cup P_2$ 上删除 $P_1 \amalg P_2$ 中的链路转换为结构②进行处理.故定理 2 实际上给出了 MCLPP 问题最优解所有可能的构造形式.

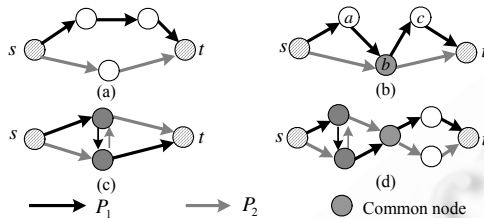


Fig.5 Examples of the possible structure relation between P_1 and P_2

图 5 P_1 与 P_2 路径可能的 4 种结构关系示例

定理 3. 若 P_2 能与 P_1 构造 COS,则 P_2 路径的权向量满足 $\bar{w}(P_2) \leq 2\bar{C} - [\bar{w}(p_b) + \bar{w}(p_s)]$.

定理 3 的证明见本文附录.定理 3 实际上给出了可与 P_1 链路构造 COS 的 P_2 路径的权向量上界.也就是说,超出这个上界的 P_2 不可能和 P_1 组成最优解.由于找到所有的 COS 便可搜索到最优解,而 COS 又由 P_1 和 P_2 构成.因此,只要找到 P_1 及所有可与 P_1 构成 COS 的 P_2 路径,便可得到最优解.为方便起见,此后文中出现的 P_2 均代表可与 P_1 构成 COS 的路径,故 P_2 必满足 a) 是 G' 上 s, t 间的路径且 b) $\bar{w}(P_2) \leq 2\bar{C} - [\bar{w}(p_b) + \bar{w}(p_s)]$.

由本节分析可得 MCLPP 精确算法的设计原则如下:首先应计算最短路径 P_1 ,然后找到能与 P_1 构成 COS 的所有 P_2 路径组成的集合 \mathcal{P} ,按照定理 2 的最优解的构造方法构造 \mathcal{P} 上所有 COS,最后从 COS 中选取长度最短的一组解,即 MCLPP 问题的最优解.

3 LIDOMPA 算法

本节将给出 LIDOMPA 算法的步骤及步骤中搜索 P_2 路径和降低 P_2 搜索空间的关键技术.

3.1 LIDOMPA 的步骤

给定 m 维加权有向图 $G(V, E)$ 和 m 维约束向量 \bar{C} , LIDOMPA 算法的步骤如下:

第 1 步. 运行 SAMCRA 在图 G 上寻找从 s 到 t 满足 \bar{C} 的最短路径 P_1 .若 P_1 不存在,算法终止;

第 2 步. 将 P_1 上的所有链路反向,置反向后的链路权重为 0,得到修正图 G' ;

第 3 步. 在 G' 上运行 M-SAMCRA(modified-SAMCRA),计算 (s, t) 间满足紧缩的约束向量 $\bar{C}' = 2\bar{C} - [w_k(p_b) + w_k(p_s)]$ 的所有未被 SAND 支配掉的 P_2 路径.记这些 P_2 路径组成的路径集合为 S ,若 S 为空,算法停止***.

第 4 步. 将 S 按规则分类:

$$\begin{cases} S_\alpha = \{P \mid P \cap \bar{P}_1 = \emptyset \text{ and } V(P) \cap V(P_1) = \emptyset, P \in S\} \\ S_\beta = \{P \mid P \cap \bar{P}_1 = \emptyset \text{ and } V(P) \cap V(P_1) \neq \emptyset, P \in S\} \\ S_\chi = \{P \mid P \cap \bar{P}_1 \neq \emptyset \text{ and } V(P) \cap V(P_1) = V(P \cap \bar{P}_1), P \in S\} \\ S_\delta \text{ others} \end{cases}$$

第 5 步. 按下述 A, B, C, D 中的步骤分别搜索 $S_\alpha, S_\beta, S_\chi$ 和 S_δ 4 个路径集合上的最优解,选择总长最短的一组作为 MCLPP 问题的最终解.

A) 1) 查找 S_α 上长度最短的路径 P_α ; 2) 检查 P_α 的长度,若 P_α 满足约束 \bar{C} , 则 (P_1, P_α) 为 S_α 上的最优解,反之, S_α

*** ① M-SAMCRA 修正的 SAMCRA 算法,是本文根据 LIDOMPA 算法的需要,由 SAMCRA 算法修改而来,作用是搜索 P_2 路径,第 3.2 节中有相关介绍;② 紧缩的约束向量(constricted constraint vector,简称 CCV)的作用是依据定理 3 从路径长度方面对 P_2 路径的搜索空间进行降低;③ 结构化的路径支配(structure-aware non-dominance,简称 SAND)的作用是进一步降低 P_2 的搜索空间.

上无解;

- \mathcal{B} 1) 对任意 $P_\beta \in S_\beta$, 利用 $P_1 \cup P_\beta$ 集合上的链路构造 $2^{|V(P_1) \cup V(P_\beta)|}$ 对链路分离路径对, 取总长度最小且满足约束 \bar{C} 的一组作为 P_β 与 P_1 构成的最优解; 2) 从所有 P_β 与 P_1 构成的最优解中选择总长度最小的一组作为 S_β 上的最优解;
- \mathcal{C} 1) 对任意 $P_\chi \in S_\chi$, 构造 $P_1 \cup P_\chi$ 并从 $P_1 \cup P_\chi$ 中删去集合 $P_1 \cap P_\chi$ 上的链路, 即 $(P_1 \cup P_\chi) - (P_1 \cap P_\chi)$. 用余下的链路构成一对链路分离路径对 $\{P_\psi, P_\tau\}$. 若 P_ψ 和 P_τ 均满足 \bar{C} 的约束, 则 $\{P_\psi, P_\tau\}$ 为路径 P_χ 与 P_1 构成的 COS; 2) 从所有 P_χ 与 P_1 构成的 COS 中选择总长度最小的一组作为 S_χ 上的最优解;
- \mathcal{D} 1) 对任意 $P_\delta \in S_\delta$, 进行 $(P_1 \cup P_\delta) - (P_1 \cap P_\delta)$ 的处理, 用余下的链路组成 \mathcal{B} 中的结构 (即定理 2②中所示结构), 故处理方法与 \mathcal{B} 中 1) 的处理相同; 2) 从所有 P_δ 与 P_1 构成的 COS 中选择总长度最小的一组作为 S_δ 上的最优解.

上述算法步骤显然满足第 2 节所述精确算法的设计原则. 其中的关键是第 3 步 P_2 路径的计算, 一方面, 能否找到所有 P_2 决定着算法的精确性, 另一方面, 搜索 P_2 的过程及 P_2 的数量直接影响着算法的复杂性. 为了完整求解 P_2 路径的集合并降低搜索空间, 我们充分利用了 SAMCRA 算法的特点和路径支配 (non-dominance) 技术. 由于传统的 SAMCRA 和 non-dominance 并不能直接运用于 LIDOMPA 算法中, 本文对 SAMCRA 进行了修改, 提出了 Modified-SAMCRA, 简称 M-SAMCRA, 对 non-dominance 进行了修正提出了结构化的路径支配 SAND. 下面将详细介绍 M-SAMCRA 和 SAND 的提出原因及原理.

3.2 M-SAMCRA

M-SAMCRA 根据 MCP 问题的精确算法 SAMCRA 算法修改而来. SAMCRA 算法由 Mieghem 等人^[11]提出, 作者在文献[12]中通过引入前向探测 (look-ahead) 方法对 SAMCRA 进行了优化和改进. SAMCRA 有以下几个特点: a) 基于非线性长度, 见第 1 节中的式(2); b) 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; b) 采用 non-dominance 方法降低搜索空间和算法复杂度. 这些特点与我们搜索 P_2 路径的要求很相似, 但是, 1) SAMCRA 只能返回最短路径; 2) SAMCRA 采用的 non-dominance 虽可降低搜索空间, 但会引起 P_2 路径的丢失; 3) SAMCRA 的终止条件是: 当前队列中提取的节点等于目的节点时算法终止, 即找到最短路径算法就终止. 因而 SAMCRA 不能直接用于 LIDOMPA 算法的第 3 步. 为此, 本文对 SAMCRA 作了对应的 3 点修改: 首先, 修改了输出结果, 使其返回值为所有满足要求的 P_2 路径集合; 第二, 提出了结构化的路径支配 (SAND) 来替代传统的 non-dominance, 确保在降低算法复杂度的同时, 保证 P_2 路径不丢失; 第三, 将算法终止条件改为当队列 Queue^[10] 为空时算法终止.

3.3 结构化的路径支配 SAND

在第 3.2 节的 M-SAMCRA 算法中, 对 SAMCRA 所用的 non-dominance 方法也进行了修正. 本节将分析必须修正 non-dominance 的原因, 给出修正后的结构化路径支配方法 SAND, 并通过定理 4 证明 SAND 不会影响 LIDOMPA 算法的精确性. 首先给出传统的 non-dominance 的方法.

路径支配 (non-dominance)^[10]. 给定从源 s 到中间节点 j 之间的两条路径 P_x 和 P_y , 如果 $w_i(P_x) \leq w_i(P_y)$ 对所有的 $1 \leq i \leq m$ 均成立, 则 P_y 可以被 P_x 支配掉.

Non-Dominance 是 SAMCRA 在搜索多约束最短路径中采用的搜索空间降低方法. 如图 6 和图 7 中的例子所示, 若将 non-dominance 运用于 LIDOMPA 算法, 会造成 P_2 路径和 MCLPP 的 COS 的丢失, 破坏算法的精确性. 图 6(a) 为原始网络, $\bar{C} = [6, 6]$, 路径 $sabt$ 为 s 到 t 的最短路径 P_1 . 图 6(b) 为图 6(a) 的修正图 G' . 在 P_2 的搜索过程中, 在图 6(b) 的节点 a 上, 从 s 到 a 的路径 sca 的权向量为 (3, 2), sba 权向量为 (4, 3). 若采用 non-dominance, sba 将被 sca 支配掉, 节点 a 上不再保存路径 sba . 这会导致 P_2 路径 $sbat$ 以及由 $sbat$ 和 P_1 构成的 COS $\{sat, sbt\}$ 的丢失, 而 $\{sat, sbt\}$ 恰恰为图 6 例中的最优解. 在图 7 所示例子中, 若采用 non-dominance, 在修正图图 7(b) 上, 子路径 $sbac$ 会被 sc 支配掉, 从而导致 P_2 路径 $sbact$ 及由 $sbact$ 和 sat 构成的最优解 $\{sact, sbat\}$ 都被丢失掉.

上述两个算例说明, non-dominance 会造成 MCLPP 最优解的丢失, 不能直接用于 LIDOMPA 算法. 原因是传统的 non-dominance 只考虑了路径间的权重大小关系, 并未考虑路径间的结构关系. 而在 MCLPP 问题中, 路径间

的结构关系是必须要考虑的因素.为实现 LIDOMPA 的精确性,本文对 non-dominance 的应用条件进行了限制,提出了结构化的路径支配方法 SAND.

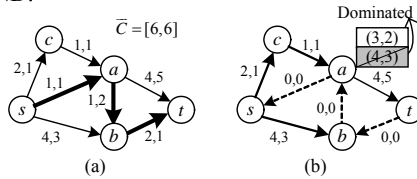


Fig.6 Non-Dominance causes the loss of optimal solution of MCLPP, example-1

图 6 Non-Dominance 造成 MCLPP 最优解丢失的算例-1

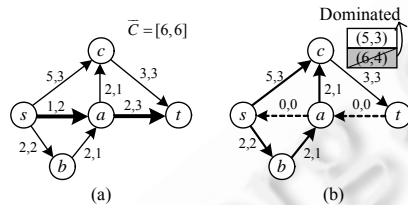


Fig.7 Non-Dominance causes the loss of optimal solution of MCLPP, example-2

图 7 Non-Dominance 造成 MCLPP 最优解丢失的算例-2

结构化的路径支配 SAND. 给定从源节点 s 到中间节点 j 之间的两条路径 P_x 和 P_y , 如果 $w_i(P_x) \leq w_i(P_y)$ 对所有的 $1 \leq i \leq m$ 均成立, 且满足 P_x 和 P_y 的对称差子路径 \tilde{P}_x 为零 \bar{P}_1 链路路径, \tilde{P}_y 为零 P_1 节点路径, 那么 P_y 可以被 P_x 支配掉.

定理 4. SAND 不会破坏 LIDOMPA 的精确性.

定理 4 的证明见附录. 证明过程同时表明, SAND 是通过不断舍弃仅能构造次优解的子路径来降低 P_2 的搜索空间和数目的.

4 LIDOMPA 算法搜索空间降低和精确性保证的分析

本文的目的是设计 MCLPP 问题的精确算法, 算法精确性必然是首要目标, 而算法复杂性也是决定算法设计成败的关键因素. 上一节已给出 LIDOMPA 的步骤, 本节将从理论上分析和讨论 LIDOMPA 是如何在降低搜索空间的同时实现其精确性的. LIDOMPA 第 1 步利用 SAMCRA 算法求解最短路径 P_1 , 第 2 步得到修正图, 第 3 步求解所有未被 SAND 掉的 P_2 路径, 第 4 步、第 5 步利用 P_2 和 P_1 分情况构造 COS 并选取最优解, 这已符合第 2 节精确算法的设计原则. 在 LIDOMPA 设计中, 通过引入以下 3 个概念有效地保证了在不损伤算法精确性的前提下对搜索空间的降低.

1) COS. 定理 1 和推论 1 已表明, 只有 COS 才有可能成为最优解. 因而 LIDOMPA 采用了只构造 COS 的方法, 避免了搜索整个可行解空间, 缩小了最优解的搜索范围. COS 概念的提出和采用, 实际上是从最优解的结构限制条件(最优解链路集合必含 P_1 链路)入手, 对最优解空间进行了有效压缩;

2) CCV. 在 LIDOMPA 第 3 步采用了比 DIMCRA 第 3 步 $2\bar{C}$ 更紧的约束向量 $2\bar{C} - [w_k(p_s) + w_k(p_e)]$. 定理 3 已表明 CCV 不会引起有效 P_2 路径的损失. 采用 CCV 约束, 在 P_2 的搜索过程中, 超出 CCV 约束的子路径将全被丢弃, 这实际上是从路径长度限制入手, 有效降低了 P_2 的搜索和存储空间;

3) SAND. 由 SAND 的定义可知, SAND 是通过路径结构和权重两方面限制同时进一步降低 P_2 搜索空间的. 定理 4 已经证明了 SAND 不会破坏 LIDOMPA 的精确性.

定理 5. LIDOMPA 是求解 MCLPP 问题的精确算法.

证明: 算法的设计原则、算法步骤以及所采用的搜索空间降低方法都保证了 LIDOMPA 的精确性, 因而 LIDOMPA 是求解 MCLPP 问题的精确算法. \square

下面仍以图 2(a) 所示网络为例说明 LIDOMPA 的运行步骤及结果并分析搜索空间降低的过程. 约束 \bar{C} 同样

取[6.5,6],目的是在 s 和 t 之间寻找满足 \bar{C} 的链路分离最短路径对.若运行 LIDOMPA,第 1 步和第 2 步的结果与 DIMCRA 相同,如图 2(a)和图 2(b)所示.图 8(a)~图 8(i)给出了 LIDOMPA 第 3 步的详细处理过程.图 8 中节点附近的小方框代表存储路径的队列,框内数值代表路径的非线性长度.当路由由函数 $EXTRACT_MIN$ ^[11,13]从队列(白色小框表示)中提取出来作当前处理路径时,对应小框就被染成灰色.有灰色线条发出的灰色小框为当前选取路径.灰色带箭头线条表示当前节点向邻居节点传播路径信息.被斜线划掉的灰色小框代表被舍弃掉的路径.

LIDOMPA 算法第 3 步计算得到 $\bar{C}' = 2\bar{C} - [\bar{w}(sb) + \bar{w}(ct)] = [10,10]$.在图 8(b)上,节点 a 将 s 到达自己的路径 sa 通告给节点 b 和 f ,因为 $l(saf)=1.1 > 1.0$,故 saf 被节点 f 舍弃,相同的优化处理在图 8(f)上也有发生.图 8(f)中,因为 $l(sabt) > 1.0$, $sabt$ 被舍弃.在图 8(c)上, scb 权重向量小于 sab ,但因 scb 含有 \bar{P}_1 链路, scb 并未被 sab 支配掉(若采用传统的 non-dominance, scb 会被 sab 支配掉).在图 8(d)上,节点 b 将 scb 通告给 s ,因为发生了环路, $scbs$ 被支配掉.同理,图 8(f)上的 $sabs$ 也被优化掉.在图 8(e)上,因为 $\bar{w}(sde) = (8,6)$, $\bar{w}(sce) = (7,5)$ 且它们的差分子路径 sde 为零 P_1 节点路径, sce 为零 \bar{P}_1 链路路径,故 sde 被 sce 支配掉了.在图 8(h)和图 8(i)中,因为当前提取的路径均在目的节点上,故路径 $scet$ 和 $scbt$ 被放到路径集合 S 中.在图 8(i)中,队列被提取为空(即所有小框均被染灰),故 P_2 路径搜索完毕.LIDOMPA 第 4 步将 $scet$ 划分到子集 S_β 中,将 $scbt$ 划分到子集 S_χ 中.第 5 步,对于 S_β , $scet$ 与 P_1 路径 $sbct$ 可构成两对链路分离路径对 $\{sbct,scet\}$ 和 $\{sbct,scet\}$.由于 $scet$ 和 $sbct$ 不满足约束 \bar{C} , $\{sbct,scet\}$ 和 $\{sbct,scet\}$ 都不为 COS,所以路径集合 S_β 上无解.对于 S_χ 上的 $scbt$, $(scbt \cup P_1) - (scbt \cap P_1)$ 操作后剩余的链路构成一对分离路径对 $\{sbt,scet\}$,经检验, sbt 和 $scet$ 均满足 \bar{C} ,所以 $\{sbt,scet\}$ 为 S_χ 上的最优解,故选择 $\{sbt,scet\}$ 为 LIDOMPA 的最终解.显然,LIDOMPA 所求的 $\{sbt,scet\}$ 本身就是图 2(a)网络上的最优解.

在同样的拓扑和约束下,如图 2 例所述, DIMCRA 调用了 4 次 SAMCRA 算法也未能求得可行解.该算例表明,LIDOMPA 能在 DIMCRA 无法求解的情况下求得最优解.此外, G' 上 s,t 间客观存在 5 条路径,LIDOMPA 的第 3 步通过优化只保留了 2 条 P_2 路径而未失最优解.因此,该算例也展示了 LIDOMPA 通过 CCV 和 SAND 等方法有效地降低了 P_2 的搜索空间和数目而不失算法的精确性.

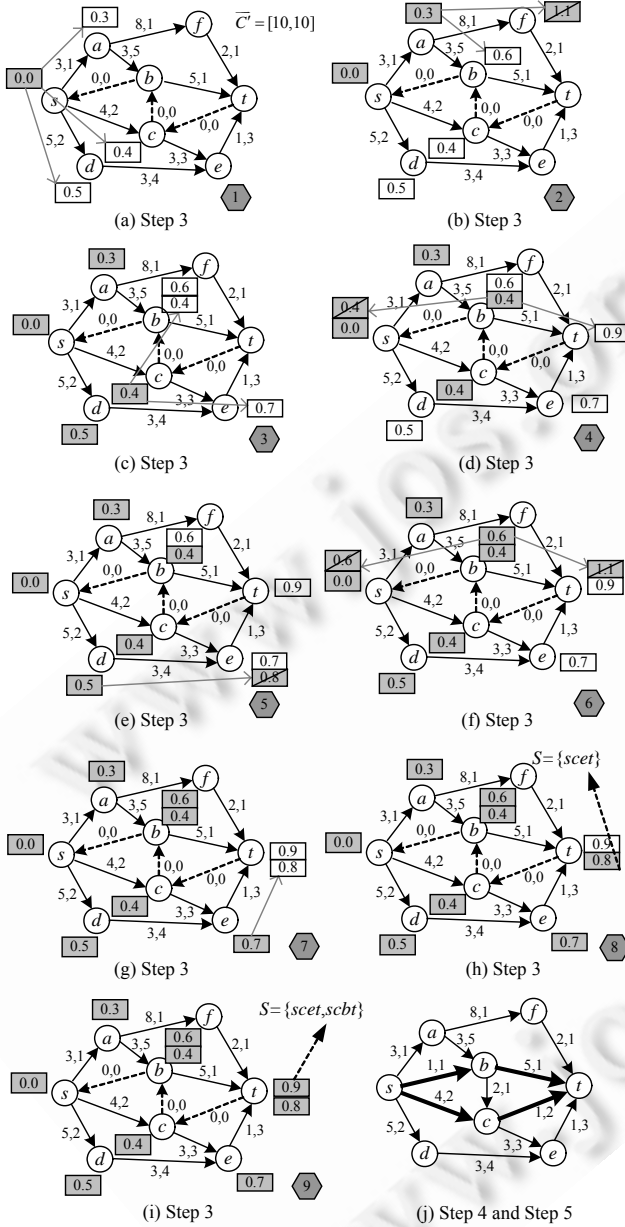


Fig.8 An example of LIDOMPA

图 8 LIDOMPA 算例

5 实验结果与分析

第 5.1 节将通过大量实验对 LIDOMPA 和 DIMCRA 作进一步比较.第 5.2 节将讨论 LIDOMPA 算法在最坏情况下的执行时间开销,第 5.3 节将通过实验验证算法所采用的降低搜索空间方法的有效性.本节实验拓扑全部采用文献[14]的 RGU 模型生成,RGU(random graph with uniformly distributed link weight)图的节点数用 N 表示,链路密度 ρ 取值为 0.2.拓扑中每条链路都带有 m 维的加性权重,每维权重均服从 $[0,1]$ 上的均匀分布.实验所用计算机 CPU 主频为 1.9GHz,内存为 1G.

5.1 LIDOMPA与DIMCRA算法的比较

取 N 为 10,25,50,100,150,200,250,300,350,400,450,500,550,600,700,800,900 和 1000, $C_i=1(1 \leq i \leq m)$,在 $m=2$ 和 $m=3$ 的条件下分别进行实验.每个 N 上生成 10^4 个 RGU.每个 RGU 上运行 LIDOMPA 和 DIMCRA 各 1 次.图 9 给出了两种算法在每个 N 上的成功求解次数.结果显示,无论 $m=2$ 或 $m=3$,LIDOMPA 的求解成功率都明显高于 DIMCRA.关于 LIDOMPA 的精确性已经通过理论得到论证.精确算法是指只要最优解客观存在,就可求得.因为随机生成的 RGU 上很多情况下本身就不存在 MCLPP 的可行解,故图 9 中的求解成功率并不总为 100%.图 10 给出了两种算法在每个 N 上的 10^4 个 RGU 中所求解的平均总长度.结果显示,LIDOMPA 所求解的平均总长度明显小于 DIMCRA 所求解的平均总长度,即 LIDOMPA 所求解优于 DIMCRA.图 11 给出了两种算法的平均执行时间开销.结果显示,LIDOMPA 的时间开销略高于 DIMCRA.这是因为要实现算法的精确性,往往需要以增加一定的复杂性为代价.

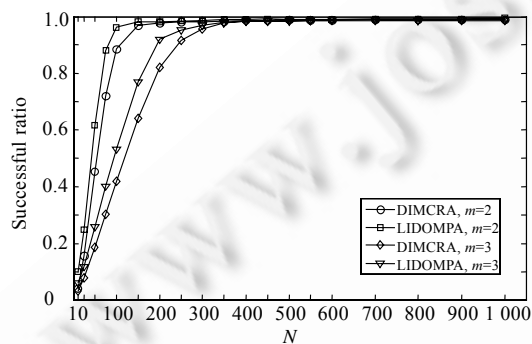


Fig.9 Comparison of the success ratio of obtaining solutions

图 9 算法求解成功率比较

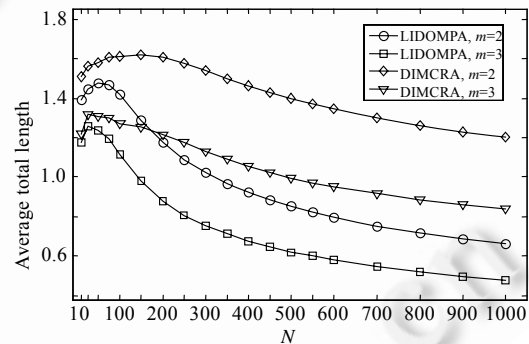


Fig.10 Comparison of the average length of the obtained solutions

图 10 算法所求解的平均长度比较

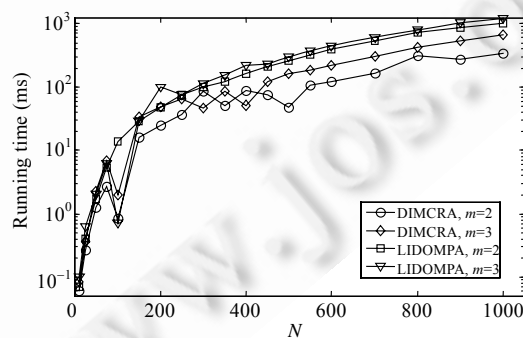


Fig.11 Comparison of running time

图 11 算法求解运行时间比较

5.2 最坏情况下,LIDOMPA算法的执行时间

在以下的实验中,设置 $C_i=N, 1 \leq i \leq m$.即所生成 RGU 上的源、目的间的所有路径均满足约束 \bar{C} ,对应 G' 上

的源、目的间所有的路径也都满足 \bar{C} . 这种情况已可视为最坏情况. 下面的实验将讨论在最坏情况下 LIDOMPA 运行时间开销随 N 和 m 变化的情况.

5.2.1 N 对算法执行时间的影响

图 12 给出了每个 N 上 LIDOMPA 的平均求解时间开销. 结果显示, 算法运行时间随网络节点数 N 的增加而不断增长. 这是因为, N 越大, 通信节点间的路径连接越复杂、数目越多. 然而, 即使在 $m=3, N=1000$ 的情况下, 算法的时间开销也未超过 1.5s.

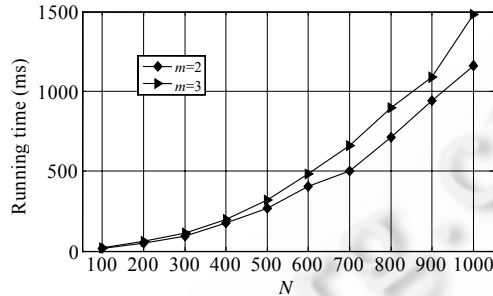


Fig.12 Variation of the running time with N

图 12 平均求解时间随 N 的变化情况

5.2.2 m 对算法执行时间的影响

图 13 给出了不同 m 上 LIDOMPA 的平均运行时间开销. 结果显示在相同 N 下, m 对算法执行时间的影响并不是很大. 在 $m=20, N=500$ 的条件下, 算法时间开销也未超过 1.8s.

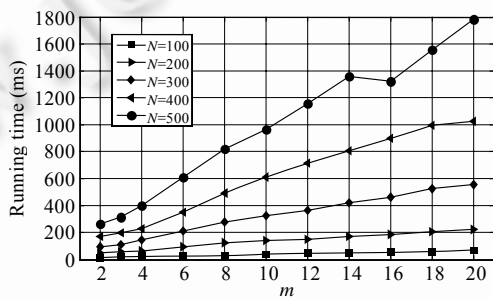
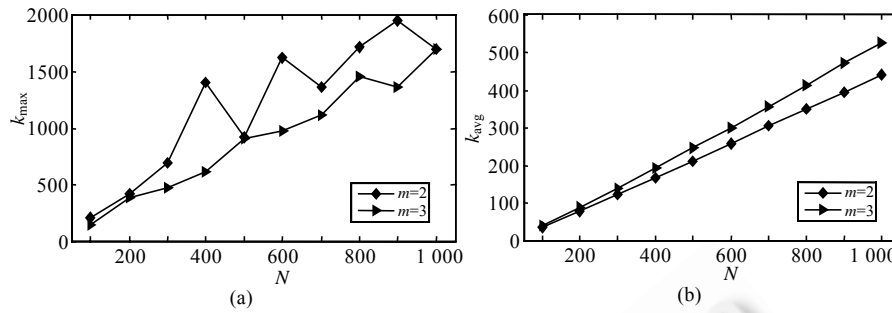
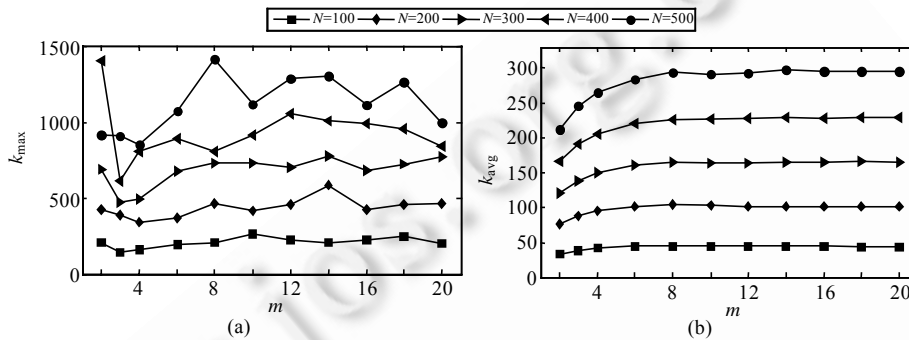


Fig.13 Variation of the running time with m

图 13 平均求解时间随 m 的变化情况

5.3 算法所用降低搜索空间方法的有效性验证

文献[15]给出了 s 和 t 之间路径数上界为 $\lfloor e(N-2)! \rfloor$, 其中 $e \approx 2.718$. 图 14(a)和图 14(b)分别统计了 $m=2$ 和 $m=3$ 条件下, LIDOMPA 在不同 N 上的 10^4 个 RGU 中未被 SAND 掉的 P_2 路径数目的最大值 k_{\max} 和平均值 k_{avg} . 可以看出 k_{\max} 和 k_{avg} 均随 N 的增长呈线性增长. 图 15(a)和图 15(b)分别统计了在 m 从 2 增长到 20 的情况下, LIDOMPA 在不同 N 上的 10^4 个 RGU 中所求得的 P_2 路径数目的最大值 k_{\max} 和平均值 k_{avg} , 可以看出, N 值越大, k_{\max} 和 k_{avg} 越大. 但在相同的 N 上, 随着 m 的增长, k_{\max} 和 k_{avg} 并没有明显的增长. 本节中, k 的变化趋势与第 5.2 节时间开销的增长趋势相似, 这正好验证了 P_2 路径的数目直接影响 LIDOMPA 算法的复杂性的结论. 图 14 和图 15 的结果还表明 $k_{\max} \ll \lfloor e(N-2)! \rfloor$, 即未被 SAND 后的 P_2 路径的数目远远小于 $\lfloor e(N-2)! \rfloor$. 这验证了本文针对 LIDOMPA 算法所提出的搜索空间降低方法对算法复杂度的降低很有效的结论.

Fig. 14 (a) Variation of k_{\max} with N , (b) Variation of k_{avg} with N 图 14 (a) k_{\max} 随 N 的变化情况, (b) k_{avg} 随 N 的变化情况Fig. 15 (a) Variation of k_{\max} with m , (b) Variation of k_{avg} with m 图 15 (a) k_{\max} 随 N 的变化情况, (b) k_{avg} 随 m 的变化情况

上述这些实验说明, LIDOMPA 算法求解性能优于现有算法, 且算法所采用搜索空间降低方法有效地实现了算法复杂度的降低。据文献[16]所述, 路径预计算是高速网络中用于实现 QoS 路由的重要方法。本节的实验结果表明, 即使在最坏情况下, LIDOMPA 算法的时间开销也能够满足路径预计算的要求。因此 LIDOMPA 算法实现了可接受的复杂度, 是一种可行的可靠 QoS 路由算法。

6 总 结

本文对 MCLPP 问题进行了研究, 针对现有算法不能对任意网络求得客观存在的可行解的问题, 提出了精确求解算法 LIDOMPA。本文分析了 MCLPP 问题最优解的性质与构造方法, 提出了候选最优解、紧缩的约束向量以及结构化的路径支配等方法, 可以在不影响算法精确性的前提下, 有效降低算法搜索空间。大量实验结果表明, 本文所提算法的求解能力比现有算法有明显的提高, 且实现了较低的算法复杂度。

References:

- [1] Zhang P, Zhang JW, Li LM, Wang S. Researches on the problem of link disjoint paths with QoS constraints. Journal on Communications, 2006, 7(6):36-42 (in Chinese with English abstract).
- [2] Sprintson A, Yannuzzi M, Orda A, Masip-Bruin X. Reliable routing with QoS guarantees for multi-domain IP/MPLS networks. In: IEEE INFOCOM. Anchorage, 2007. 1820-1828.
- [3] Guo YC, Kuipers F, Mieghem PV. Link disjoint paths algorithm for reliable QoS routing. Int'l Journal of Communication Systems, 2003, 16(9):779-798. [doi: 10.1002/dac.612]
- [4] Peng C, Shen H. An Improved approximation algorithm for computing disjoint QoS paths. In: ICN/ICONS/MCL 2006. Morne: IEEE Computer Society Press, 2006. 10.

[5] Xiao Y, Thulasiraman K, Xue GL. Constrained shortest link-disjoint paths selection: A network programming based approach. IEEE Trans. on Circuits and Systems-I: Regular Papers, 2006,53(5):1174-1187. [doi: 10.1109/TCSI.2006.869907]

[6] Xu DH, Chen Y, Xiong YZ, Qiao CM, He X. On the complexity of and algorithms for finding the shortest path with a disjoint counterpart. IEEE/ACM Trans. on Networking, 2006,14(1):147-158. [doi: 10.1109/TNET.2005.863451]

[7] Naser H, Gong M. Link-Disjoint shortest-delay path-pair computation algorithms for shared mesh restoration networks. In: Proc. of the 12th IEEE Symp. on Computers and Communications, ISCC 2007. Aveiro, 2007. 269-274.

[8] Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman, 1979.

[9] Wang Z, Crowcroft J. Quality-of-Service routing for supporting multimedia applications. IEEE Journal on Selected Areas in Communications, 1996,14(7):1228-1234. [doi: 10.1109/49.536364]

[10] Nevea HD, Mieghe PV. TAMCRA: A tunable accuracy multiple constraints routing algorithm. Computer Communications, 2000,23(7):667-679. [doi: 10.1016/S0140-3664(99)00225-X]

[11] Mieghe PV, Nevea HD, Kuipers F. Hop by hop quality of service routing. Computer Networks, 2001,37(3-4):407-423. [doi: 10.1016/S1389-1286(01)00222-5]

[12] Mieghe PV, Kuipers F. Concepts of exact QoS routing algorithms. IEEE/ACM Trans. on Networking, 2004,12(5):851-863. [doi: 10.1109/TNET.2004.836112]

[13] Cormen TH, Leiserson CE, Rivest RL. Introduction to Algorithms. Cambridge: MIT Press, 2001.

[14] Bollobás B, Erds P, Faudree RJ, Rousseau CC, Schelp RH. Random induced graphs. Discrete Mathematics, 2002,248(1-3): 249-254. [doi: 10.1016/S0012-365X(01)00345-4]

[15] Mieghe PV. Paths in the simple random graph and the Waxman graph. Probability in the Engineering and Informational Sciences, 2001,15(4):535-555.

[16] Orda A, Sprintson A. QoS routing: The precomputation perspective. In: IEEE INFOCOM. Tel Aviv: IEEE Press, 2000. 128-136.

附中文参考文献:

[1] 张品,章坚武,李乐民,王晟.QoS 约束下的链路分离路径问题研究.通信学报,2006,27(6):36-42.

附录

定理 2 证明:任给 G' 上从 s 到 t 的一条路径 P, P 和 \bar{P}_1 间的结构关系可按有无公共链路和公共节点的关系加以分类,分类的结果可由图 16 中的层次结构表示.根据定义, P_2 属于 P 的集合, P_2 和 \bar{P}_1 间的关系也必然属于图 16 所示的 4 种情况.因为 $V(P_1) = V(\bar{P}_1)$,可由图 5 中的分类得出定理 2 中的 4 种关系.故定理 2 成立. \square

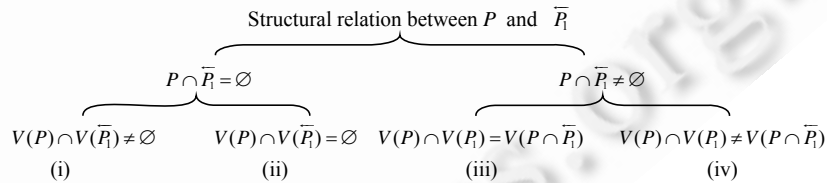


Fig.16 Possible structure relation between P and \bar{P}_1

图 16 P 和 \bar{P}_1 间可能的结构关系

定理 3 证明:因为 P_2 与 P_1 的结构关系必然属于定理 2 所述 4 种之一.将①和②归为一组,即 $\bar{P}_1 \cap P_2 = \emptyset$,将③和④归为一组,即 $\bar{P}_1 \cap P_2 \neq \emptyset$.假设 $\{P'_1, P'_2\}$ 为 P_2 与 P_1 构成的一组 COS,则 $\bar{w}(P'_1) \leq \bar{C}, \bar{w}(P'_2) \leq \bar{C}$.当 $\bar{P}_1 \cap P_2 = \emptyset$ 时, $\{P'_1, P'_2\}$ 由 P_1 与 P_2 的全部链路组成,即 $P'_1 \cup P'_2 = P_1 \cup P_2$.故,

$$w_k(P'_1) + w_k(P'_2) = w_k(P_1) + w_k(P_2) \leq 2C_k \quad (1 \leq k \leq m) \tag{4}$$

因此,

$$w_k(P_2) \leq 2C_k - w_k(P_1) \tag{5}$$

当 $\bar{P}_1 \cap P_2 \neq \emptyset$ 时, $\{P'_1, P'_2\}$ 由 $(P_1 \cup P_2) - (P_1 \amalg P_2)$ 集合上的链路组成, 由于

$$P_1 \cup P_2 = (P'_1 \cup P'_2) \cup (P_1 \amalg P_2) = (P'_1 \cup P'_2) \cup (\bar{P}_1 \cap P_2) \cup (P_1 \cap \bar{P}_2),$$

所以式(6)成立.

$$w_k(P_1) + w_k(P_2) = w_k(P'_1) + w_k(P'_2) + w_k(\bar{P}_1 \cap P_2) + w_k(P_1 \cap \bar{P}_2) \quad (6)$$

根据 G' 和 P_2 的定义可知 $w_k(\bar{P}_1 \cap P_2) = 0$, 代入式(6)有

$$w_k(P_1) + w_k(P_2) = w_k(P'_1) + w_k(P'_2) + w_k(P_1 \cap \bar{P}_2) \quad (7)$$

成立. 因而可得

$$w_k(P_2) = w_k(P'_1) + w_k(P'_2) + w_k(P_1 \cap \bar{P}_2) - w_k(P_1) \leq 2\bar{C} + w_k(P_1 \cap \bar{P}_2) - w_k(P_1) \quad (8)$$

结合式(5)和式(8)可得, 无论 P_2 与 P_1 有结构关系如何, 若 P_2 能与 P_1 构造 COS, P_2 路径的权向量总满足:

$$w_k(P_2) \leq 2\bar{C} + w_k(P_1 \cap \bar{P}_2) - w_k(P_1) \quad (9)$$

令 p_ϕ 为 p_b 与 p_d 间 P_1 的子路径, 则 $P_1 = p_b + p_\phi + p_d$. 所以, $w_k(P_1) = w_k(p_b) + w_k(p_\phi) + w_k(p_d)$. 由于 p_b 与 p_d 的链路不可能出现在 P_2 上, $P_1 \cap \bar{P}_2$ 的最大值只能等于 p_ϕ , 故 $w_k(P_1 \cap \bar{P}_2) \leq w_k(p_\phi)$. 将上述结论代入式(9)可得:

$$w_k(P_2) \leq 2\bar{C} + w_k(p_\phi) - [w_k(p_b) + w_k(p_\phi) + w_k(p_d)] \quad (10)$$

整理得 $w_k(P_2) \leq 2\bar{C} - [w_k(p_b) + w_k(p_d)]$ ($1 \leq k \leq m$) 成立. 故定理 3 得证. \square

定理 4 的证明: \bar{P}_x 为零 P_1 链路路径, 则 $\bar{P}_x \cap \bar{P}_1 = \emptyset$. 假设 P^x 是以 P_x 为子路径的一条 P_2 路径, 根据定理 2, 对最优解构成形式的分析, \bar{P}_x 链路会被完全包含在 P^x 与 P_1 构成的 COS 的链路集合中. 因而, P^x 和 P_1 构成的 COS 可以被分为两类, 记为集合 \mathfrak{M} 和 \mathfrak{A} , 满足 P^x 和 P_1 构成的 COS $\{P_{1x}, P_{2x}\}$ 中, 若 P_{1x} 和 P_{2x} 任意一条路径包含 \bar{P}_x 上所有链路, 则 $\{P_{1x}, P_{2x}\} \in \mathfrak{M}$, 若 P_{1x} 和 P_{2x} 均含有 \bar{P}_x 的部分链路, 则 $\{P_{1x}, P_{2x}\} \in \mathfrak{A}$. 假设 P^y 是以 P_y 为子路径的一条 P_2 路径, $\{P'_1, P'_2\}$ 是由 P_y 和 P_1 构成的 COS. 由于 \bar{P}_y 为零 P_1 节点路径, \bar{P}_y 链路必然被 $\{P'_1, P'_2\}$ 中的一条路径完全包含. 不失一般性, 假设 \bar{P}_y 链路全部被 P'_2 包含, 因为 \bar{P}_x 和 \bar{P}_y 是位于相同节点间的子路径, 将 $\{P'_1, P'_2\}$ 中 P'_2 上的 \bar{P}_y 链路替换为 \bar{P}_x 链路, 可得新的 COS $\{P'_1, P'_2\}$, 且必定满足 $\{P'_1, P'_2\} \in \mathfrak{M}$. 因为 $w_i(P_x) \leq w_i(P_y)$, 根据对称差子路径定义易推得 $w_i(\bar{P}_x) \leq w_i(\bar{P}_y)$, 所以 $l(P'_1) + l(P'_2) \leq l(P'_1) + l(P'_2)$. 也就是说, $\{P'_1, P'_2\}$ 优于 COS $\{P'_1, P'_2\}$. 上述分析说明, 任给一个由 P^y 和 P_1 构成的 COS $\{P'_1, P'_2\}$, 在 \mathfrak{M} 中都有优于 $\{P'_1, P'_2\}$ 的 COS $\{P'_1, P'_2\}$ 存在. 若 P_y 被 P_x 支配掉, 仅损失掉了 p^y 和 $\{P'_1, P'_2\}$, 而更优的 COS $\{P'_1, P'_2\}$ 并未损失. 因此, 用 P_x 支配掉 P_y 并不会影响 LIDOMPA 算法的精确性. \square



熊轲(1981—),男,陕西汉中,博士,主要研究领域为多媒体通信,下一代互联网络.



张宏科(1957—),男,博士,教授,博士生导师,主要研究领域为下一代互联网理论.



裴正定(1944—),男,教授,博士生导师,主要研究领域为信息安全,多媒体通信,无线网络,下一代互联网.



秦雅娟(1963—),女,博士,副教授,博士生导师,主要研究领域为下一代互联网,宽带无线通信.



郭宇春(1968—),女,博士,教授,主要研究领域为网络测量与建模,拓扑,路由,P2P网络,IP网络服务质量.