

无圈反向超图中Min-Max不相交 反向超路算法设计

余婉霞, 张淑蓉*

太原理工大学数学学院, 山西 太原

收稿日期: 2023年1月14日; 录用日期: 2023年2月8日; 发布日期: 2023年2月17日

摘要

超图与反向超图是计算机网络和通信网络采用的一类重要的网络拓扑结构, 为提高网络以及传输路径容错性, 不相交路径设计与优化成为重要的研究课题。本文在有向无圈的反向超图中, 当每一条反向超弧的尾节点数不超过给定常数 λ 时, 对构建源点与汇点间两条点不相交反向超路(B超路)的Min-Max优化问题进行研究, 使得两条反向超路中权值较大者的值能够达到全局最小。为解决该问题, 首先构造并设计了基于原图的辅助图, 该辅助图也是有向无圈的反向超图, 并将原问题转化为辅助图的多权值函数B超路优化问题。从而设计了伪多项式算法得到该问题的最优解。基于该算法, 进一步给出了近似算法的设计方案得到 $(1 + \epsilon)$ 近似解, 从而有效降低了计算复杂性。

关键词

超图, 路径规划, Min-Max, 无圈反向超图, 不交路径

Design of Min-Max Disjoint Backward Hyperpaths Algorithm in Acyclic Backward Hypergraph

Wanxia Yu, Shurong Zhang*

College of Mathematics, Taiyuan University of Technology, Taiyuan Shanxi

Received: Jan. 14th, 2023; accepted: Feb. 8th, 2023; published: Feb. 17th, 2023

Abstract

Hypergraph and backward hypergraph are an important class of the network topology used in
*通讯作者。

computer networks and communication networks. In order to improve the fault tolerance of networks and transmission paths, the design and optimization of disjoint paths have become important research topics. In this paper, in the directional acyclic backward hypergraph, when the tail node number of each backward hyperarc does not exceed the given constant λ , the Min-Max optimization problem of constructing two node-disjoint backward hyperpaths (B-hyperpaths) from the source node to the sink node is researched, so that the value of the greater weight in this two node-disjoint backward hyperpaths can reach the global minimum. In order to solve this problem, the auxiliary hypergraph based on the original hypergraph is constructed and designed firstly, which is also a directional acyclic backward hypergraph. The original problem is transformed into a B-hyperpath optimization problem with multi-weight function in the auxiliary hypergraph. Thus a pseudo-polynomial algorithm is designed to obtain an optimal solution to the problem. Based on this algorithm, the design scheme of the approximation algorithm is further given to obtain an $(1 + \varepsilon)$ approximate solution which effectively reduce the complexity of calculation.

Keywords

Hypergraphs, Route Planning, Min-Max, Acyclic Backward Hypergraphs, Disjoint Hyperpaths

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

网络中基于特定优化目标的路径设计广泛应用于许多研究领域, 并主要采用赋权图构建网络拓扑结构以及问题优化模型。文献[1]在有向赋权图中设计了多目标路径搜索方式。在网络中, 为了保证传输路径的容错性, 多路径设计具有非常重要的意义。以无线传感网络为背景, 文献[2]设计了智能优化算法构建源节点到目标节点的 k 条不相交路径。在交通网络中, 如果在确定的运输点之间想要找到多条节点不相交的路径, 那么当个别路径故障之后依然能保持交通网络的连通性。文献[3]研究了在有向无圈图和 3 连通平面图中寻找两条不相交路径问题并给出有效算法。其中对不相交路径性能的优化目标也是多样化的, 比如选择的不交路径满足所有路径中最长的路径权值尽可能小, 这类问题就是寻找 Min-Max 不相交路径。文献[4]在有向无圈图中设计出寻找两条不相交路径的近似算法, 问题的解决引用了文献[3]的 PSA 算法构建一般图的辅助图模型, 在该模型下总共考虑了四种优化目标, 分别为 MinMax, Balanced, MinSum-MinMin 和 MinSum-MinMax。在一般图中, Balanced 优化目标下寻找节点不相交路径问题和边不相交路径问题是 NP 完全的。并且 Shiloach [3]等人已经证明在一般图上寻找 Min-Max 优化目标的节点不相交路径问题是 NP 完全的, 在有向无圈图上, 该问题具有伪多项式时间算法。Fleischer [4]等人给出了在有向无圈图上寻找 Min-Max 优化目标下节点不相交路径问题的 $(1 + \varepsilon)$ 近似算法。

然而在复杂的交通运输、网络设计和数据库系统等研究领域, 一般图模型已经满足不了路径设计的需求, 因此基于超图模型的超路径设计也被进一步研究。

超图模型在 1966 年首次被 Berge 提出[5], 超图的超边是包含两个及两个以上点的集合。超图模型被应用于公共交通领域[6] [7]、数据挖掘领域, 比如对于知识图谱的关联规则的挖掘算法就是利用有向超图构建关联规则去发现知识短缺[8]。对于超图的路径优化问题现已设计出一些算法, 比如最短超路搜索算法[9] [10] [11]、动态最短超路搜索算法[12]。关于超图中不相交超路的研究, 文献[13]将超图的连通性与多路径设计问题联系起来并给出了超图中的 Menger 定理。

有向超图中的超边也叫作超弧, 由尾节点集和头节点集两个节点集组成, 方向从尾节点集指向头节点集, 若超弧的头节点个数均为 1 时就称该有向超图为反向超图。文献[14]给出了反向超路的定义并设计了最短反向超树搜索算法。反向超路模型也逐渐被应用于通信网络、分子生物等科学研究领域[15]。本文考虑节点不相交超路径并提出了在无圈反向超图中寻找满足 Min-Max 优化目标的两条点不相交反向超路问题, 其中边赋权函数值为正整数。

本文首先构造了辅助反向超图模型, 将二元点对作为辅助图中新的节点, 基于反向超弧和不相交反向超路的结构特点定义辅助图中的节点关联函数使得在原反向超图中求 Min-Max 两条不相交反向超路问题转化为在辅助图上利用标签集寻找满足一定条件的同源反向超路, 并给出具有伪多项式时间的精确算法。最后通过压缩标签集的容量, 给出 $1+\varepsilon$ 近似算法。在得到有效解的同时降低了计算复杂度。

2. 概念与问题

2.1. 超图相关概念

超图 $\mathcal{H}=(\mathcal{V}, \mathcal{E})$ 是由点集 \mathcal{V} 和超边集 \mathcal{E} 组成, 其中每一条超边包含两个及两个以上节点的集合。有向超图由点集和有向超边集组成, 有向超边就是将一条超边的节点集合分成尾节点集 $T(E)$ 和头节点集 $H(E)$, 方向从尾节点集指向头节点集, 也称为超弧。超弧 E 也被写作 $(T(E), H(E))$ 。当 $H(E)$ 中只有一个点时, 超弧 E 也叫做反向弧。若有向超图中的弧均为反向弧, 那么也称为反向超图。在 \mathcal{H} 中, 所有以节点 v 为头节点的反向弧的集合记作 $BS(v)$, 该集合中的每一条弧称为点 v 的后向弧。

超路是超图中的一个点边集序列, 从 s 到 t 的一条超路表示为 $P_{s,t}=(v_1=s, E_{i_1}, v_2, \dots, E_{i_q}, v_{q+1}=t)$, $s \in T(E_{i_1}), t \in H(E_{i_q}), v_j \in H(E_{i_{j-1}}) \cap T(E_{i_j}), j=2, \dots, q$ 。如果 $t \in T(E_{i_1})$, 则 $P_{s,t}$ 是一个圈[3]。

在反向超图 \mathcal{H} 中, 反向超路(B 超路) $P_{s,t}$ 是从 s 到 t 使用最少的反向弧使得 $P_{s,t}$ 上的每一点 x 都有从 s 到 x 的简单无圈超路。

在赋权反向超图中, 每条超弧 E 都有权值 $W(E)$ 。给定一条反向超路 $P_{s,t}$ 的权值计算方式: 假设沿着路 $P_{s,t}$ 到达点 v 的子路记作 $P_{s,t}(s \rightarrow v)$, 则

$$W(P_{s,s})=0$$

$$W(P_{s,t}(s \rightarrow v))=W(E)+\max\{W(P_{s,t}(s \rightarrow v_i))\}$$

$$v_i \in T(E), E \in BS(v) \cap \mathcal{E}(P_{s,t})。$$

特别地, 当反向弧的权值均为 1 时, 路权记作路的长度 $L(P_{s,t})$ 。

2.2. Min-Max 不相交反向超路问题

Min-Max 不相交反向超路问题: 在反向超图 $\mathcal{H}=(\mathcal{V}, \mathcal{E}, W)$ 中, 每条反向超弧的权值均取正整数, 给定两个节点 $s, t \in \mathcal{V}$, 寻找从 s 到 t 的两条节点不相交反向超路 P_1, P_2 使得两条路的最大权值能够达到最小。

目标函数: $\min\{\max\{W(P_1), W(P_2)\}\}$, 其中 $W(P_i)$ 是反向超路 P_i 的路权。

3. 辅助超图模型

给定有向无圈反向超图 $\mathcal{H}=(\mathcal{V}, \mathcal{E}, W)$, 权值 $\mathcal{H}=(\mathcal{V}, \mathcal{E}, W)$ 取正整数。给定源节点 s 和目标节点 t , 为了寻找从 s 到 t 的不交反向超路, 把 s 记作 v_1 , 把 t 记作 v_n , 按照弧的方向将节点从小到大排序, 序号大的节点不存在弧指向序号小的节点。因为 \mathcal{H} 无圈, 从 t 出发的反向弧不会再回到 t , 所以把从 t 出发的反

向弧均删去成为新 \mathcal{H} 。根据新 \mathcal{H} 构建辅助超图 $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, 令

$$\begin{aligned}\mathcal{V}_2 &= \{\langle i, j \rangle \mid 1 \leq i, j \leq n, i \neq j\} \cup \{\langle s, s \rangle, \langle t, t \rangle\} \\ \mathcal{E}_2 &= \left\{ \left(\left\{ \langle i, j_i \rangle, \langle i, k \rangle \right\} \mid \left\{ \langle v_{j_i} \rangle, v_k \right\} \in \mathcal{E}, \min \{j_i\} \leq i \right) \right. \\ &\quad \left. \cup \left\{ \left(\left\{ \langle i, j \rangle, \langle k, j \rangle \right\} \mid \left\{ \langle v_i \rangle, v_k \right\} \in \mathcal{E}, \min \{i\} \leq j \right) \right\} \right\}\end{aligned}$$

在 \mathcal{E}_2 中, 第一个集合中的超弧记作水平弧, 第二个集合中的超弧记作纵向弧。图 1 给出一个反向超图, 其对应的辅助超图如图 2。

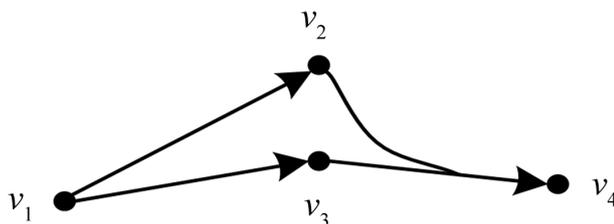


Figure 1. A backward acyclic hypergraph \mathcal{H} with 4 nodes

图 1. 节点个数为 4 的无圈反向超图 \mathcal{H}

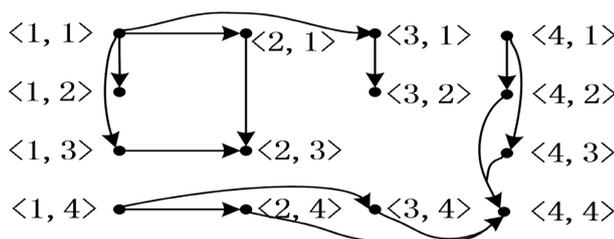


Figure 2. The auxiliary hypergraph of \mathcal{H}

图 2. \mathcal{H} 的辅助超图

定义 1 在 $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ 中, 若反向超路 $P(\langle s, s \rangle, \langle t, t \rangle)$ 上的点满足:

- 对确定的 i , 从 $\langle s, s \rangle$ 沿着路 P 到达任意点 $\langle i, j \rangle$ 的子路上的纵向弧对应在 \mathcal{E} 上的反向弧集合相同。
 - 对确定的 j , 从 $\langle s, s \rangle$ 沿着路 P 到达任意点 $\langle i, j \rangle$ 的子路上的水平弧对应在 \mathcal{E} 上的反向弧集合相同。
- 则 $P(\langle s, s \rangle, \langle t, t \rangle)$ 是一条同源反向超路。

引理 1 在 $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ 中, 给定源节点 s 和图中任意两点 t_1, t_2 , 不相交反向超路 P_{s, t_1}, P_{s, t_2} 存在当且仅当在对应的辅助超图 $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ 中存在同源反向超路 $P^2(\langle s, s \rangle \rightarrow \langle t_1, t_2 \rangle)$ 。

证明: 充分性: 令 P_1, P_2 是两条不交反向超路, 对 $L(P_1) + L(P_2)$ 进行数学归纳。当 $L(P_1) + L(P_2) = 2$ 时, 两条节点不相交 B 超路分别为 $P_1 = (s, t_1)$, $P_2 = (s, t_2)$, 在 $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ 中 $P = \langle s, s \rangle \rightarrow \langle s, t_2 \rangle \rightarrow \langle t_1, t_2 \rangle$ 显然是同源反向超路。

当 $L(P_1) + L(P_2) > 2$, 假设当 $L(P_1) + L(P_2) < k$ 时, 结论成立。记 $E_k = BS(t_1) \cap P_1$, $E_l = BS(t_2) \cap P_2$ 。不妨设 $T(E_k)$ 中顶点的下标的最小值小于 $T(E_l)$ 中顶点的下标的最小值, 记 $T(E_l) = \{v_{x_1}, v_{x_2}, \dots, v_{x_m}\}$ 那么对每一个尾节点 v_x 都满足 $L(P_1) + L(P_2(s \rightarrow v_x)) < L(P_1) + L(P_2)$, 根据数学归纳法, 在 $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ 中存在从 $\langle s, s \rangle$ 到 $\langle t_1, v_{x_i} \rangle$ 的同源反向超路 $P_{x_i}^2$ 。然后令 $E' = (\{\langle t_1, x_1 \rangle, \langle t_1, x_2 \rangle, \dots, \langle t_1, x_m \rangle\}, \langle t_1, t_2 \rangle)$, 根据有向无圈超图中的节点序号的排列顺序, 有 $\min \{x_1, x_2, \dots, x_m\} < \min \{t_1, t_2\} \leq t_1$, 故 $E' \in \mathcal{E}(\mathcal{H}_2)$, 这样我们得到了一条反向超路 $P^2(\langle s, s \rangle \rightarrow \langle t_1, t_2 \rangle) = E' \cup (\cup_{x_i} P_{x_i}^2)$, 因为在 \mathcal{H}_2 对中, 确定的 t_2 只有一个节点 $\langle t_1, t_2 \rangle$, 根据数学归纳法,

$P^2(\langle s, s \rangle \rightarrow \langle t_1, t_2 \rangle)$ 上的水平弧对应在 \mathcal{E} 上的反向弧集合是相同的。故该反向路是一条同源反向超路。

必要性: 若同源反向超路 $P^2(\langle s, s \rangle \rightarrow \langle t_1, t_2 \rangle)$ 存在, 当 $L(P) = 2$, 将 P 中的水平弧对应在 \mathcal{E} 中的反向弧放到 $\mathcal{E}(P_1)$ 中并删去冗余的反向弧即相同的反向弧, 如图 2 中 $(\langle 3, 1 \rangle, \langle 3, 2 \rangle)$ 和 $(\langle 4, 1 \rangle, \langle 4, 2 \rangle)$ 对应在 \mathcal{E} 中的反向弧为同一个 (v_1, v_2) 。同样地, 将 P 中的纵向弧对应在 \mathcal{E} 中的反向弧放到 $\mathcal{E}(P_2)$ 中并删去冗余的反向弧, 得到 $P_1 = (s, t_1)$, $P_2 = (s, t_2)$, $L(P_1) + L(P_2) = 2$ 。

假设当 $L(P) < k$ 时, 根据归纳假设结论成立, 且有 $L(P_1) + L(P_2) = L(P^2)$ 。当 $L(P^2) = k$ 时, 不妨设在该反向路的最后一条弧为 $E' = (\langle t_1, x_1 \rangle, \langle t_1, x_2 \rangle, \dots, \langle t_1, x_m \rangle, \langle t_1, t_2 \rangle)$ 那么 P^2 上的子路 $L(P^2(\langle s, s \rangle \rightarrow \langle t_1, x_i \rangle)) < L(P^2)$, 根据上面 P_1, P_2 的构造方式以及归纳假设, 有 P_1 和 $P_2(s \rightarrow v_{x_i})$ 是两条不交 B 路。根据有向无圈 B 超图的节点排序规律, 由于 $\min\{x_1, x_2, \dots, x_m\} < \min\{t_1, t_2\} \leq t_2$ 。故节点 t_2 也不在 P_1 上, 那么 P_1, P_2 也是两条不相交反向超路。

例如在图 2 中, 显然节点 $\langle 1, 1 \rangle$ 和 $\langle 4, 4 \rangle$ 不连通, 这是由于在 \mathcal{H} 中从 v_1 到 v_4 不存在两条不相交反向超路, 故辅助超图中节点 $\langle 1, 1 \rangle$ 和 $\langle 4, 4 \rangle$ 不存在同源反向超路。

4. Min-Max 不相交反向超路算法

基于上述辅助图的构造和定理, 在原超图中找两条不相交反向超路问题可以转化为在辅助图中寻找同源反向超路问题。所以结合辅助超图给出算法去找从 s 到 t 的 Min-Max 节点不相交反向超路。该算法在限定反向弧的尾节点个数不超过常数 λ 的反向超图上求解。为了记录辅助超图中同源反向超路分别对应的两条不交路的权值, 每到达一个点就记录一个标签 L 存储两条路的权值和前继点, 该标签方法在文献[1]中寻找一般图不相交路径被提出。到达点 $\langle i, j \rangle$ 的标签记为 $L_{i,j}(X, Y, pred)$, X 和 Y 分别用来标记辅助图中到达点 $\langle i, j \rangle$ 的同源反向超路 P 对应到原图中两条不相交反向超路的权值, 分别记作纵向权值和水平权值。值得注意的是这里的纵向权值(水平权值)并不是路 P 上纵向弧(水平弧)的权值的和, 该值与第 1 节中反向超路的权值定义一致。Pred 记录了路 P 上点 $\langle i, j \rangle$ 前继点和前继点的纵向权值和水平权值。节点 $\langle i, j \rangle$ 的所有标签均存储在 $I_{i,t}$ 中。算法 1 计算辅助图中每一个节点的标签集, 当算法结束的时候, 在 $\langle t, t \rangle$ 上的标签集中 $\max\{X, Y\}$ 达到最小的标签对应的同源反向超路就是最优 Min-Max 不相交反向超路对应的同源反向超路。算法 2 计算从 $\langle s, s \rangle$ 到 $\langle t, t \rangle$ 的最优同源反向超路, 会用到算法 1 得到的目标节点 $\langle t, t \rangle$ 的标签集 $I_{t,t}$ 。算法 3 进行路径回溯, 得到 Min-Max 不相交反向超路问题的最优解。

下面 3 小节分别给出这三个算法的具体步骤, 并给出算法的正确性的证明和复杂度。

4.1. 计算标签集

给定反向超图 \mathcal{H} , 利用第 3 节的模型构造得到辅助超图 \mathcal{H}^2 , 在辅助图上执行算法 1 计算每一个顶点的标签值 $(X, Y, pred)$ 。

首先初始化标签集合 $I_{i,j} = \emptyset, 1 \leq i, j \leq n$, 该集合包含点 $\langle i, j \rangle$ 的所有标签 $L_{i,j}$, 同时令源节点的标签集为 $I_{1,1} = \{(0, 0, NULL)\}$ 。接下来按照节点顺序计算每一个点的标签集, 对每一个点 $\langle i, j \rangle$ 寻找它的后向弧即以它为头节点的反相弧并记作 E' 。把后向弧分成水平弧和纵向弧分别进行计算。

如果此刻考虑的后向弧是水平弧, 接着判断该弧的尾节点的个数并记为 α , 将该弧的每一个尾节点 $\langle i, k_1 \rangle, \langle i, k_2 \rangle, \dots, \langle i, k_\alpha \rangle$ 的标签同时取出来一个比较纵向权值, 因为根据同源反向超路的定义, 到达这 α 个节点的纵向权值需要相同, 即对应到原超图 \mathcal{H} 上, 到达节点 i 的反向超路是相同的, 权值也一定相等。所以当判断条件 $X_1 = X_2 = \dots = X_\beta$ 成立时, 追溯两条纵向路是否相同, 如果相同, 那么就可以执行第 11~13 行去计算沿着弧 E' 到达点 $\langle i, j \rangle$ 的水平权值。首先记 $Y' = \max\{Y_{k_\beta}\}$, 再加 E' 的权值, 令 $c = Y' + W(\{v_{k_\beta}, v_j\})$ 赋值给点 $\langle i, j \rangle$ 新标签的水平权值。根据 B 超路权值定义, 为 $\langle i, j \rangle$ 计算新的标签值并放入 $I_{i,j}$ 中。如果节点的后向弧 E' 是纵向弧, 那么尾节点就变成 $\langle k_1, j \rangle, \langle k_2, j \rangle, \dots, \langle k_\alpha, j \rangle$, 处理方法同上。

下面给出算法 1 的伪代码。

Algorithm 1. The calculation of the label set
算法 1. 计算标签集

输入: $\mathcal{H} = (\mathcal{V}, \mathcal{E}, W)$, 常数 λ

输出: $\{I_{i,j} \mid 1 \leq i, j \leq n\}$

1. 构建辅助图 $\mathcal{H}^2 = (\mathcal{V}_2, \mathcal{E}_2)$
2. $I_{i,j} = \emptyset, 1 \leq i, j \leq n$, $I_{1,1} = \{(0, 0, NULL)\}$
3. for $i = 1 : n$
4. for $j = 1 : n$
5. for each $E' = (\{(i, k_\beta)\}, \langle i, j \rangle) \in \mathcal{E}_2$
6. $|T(E')| = \alpha$
7. for each $(X_\beta, Y_{k_\beta}, pred) \in I_{i, k_\beta}, \beta = 1, 2, \dots, \alpha$
8. if $X_1 = X_2 = \dots = X_\beta$
9. $[P_1^{i, k_\beta}, P_2^{i, k_\beta}] = BACK(L_{i, k_\beta}(X_\beta, Y_{k_\beta}, pred))$
10. if $P_1^{i, k_1} = P_1^{i, k_2} = \dots = P_1^{i, k_\alpha}$
11. $(X, Y') = (X_1, \max\{Y_{k_\beta}\})$
12. $c = Y' + W(\{v_{k_\beta}\}, v_j)$
13. $I_{i,j} = I_{i,j} \cup \{(X_1, c, T(E'), \{(X_\beta, Y_{k_\beta})\})\}$
14. end if
15. end if
16. end for
17. end for
18. for each $E' = (\{(k_l, j)\}, \langle i, j \rangle) \in \mathcal{E}_2$
19. $|T(E')| = \alpha$
20. for each $(X_{k_\beta}, Y_\beta, pred) \in I_{k_\beta, j}, \beta = 1, 2, \dots, \alpha$
21. if $Y_1 = Y_2 = \dots = Y_\alpha$
22. $[P_1^{k_\beta, j}, P_2^{k_\beta, j}] = BACK(L_{k_\beta, j}(X_{k_\beta}, Y_\beta, pred))$
23. if $P_2^{k_1, j} = P_2^{k_2, j} = \dots = P_2^{k_\alpha, j}$
24. $(X', Y) = (\max\{X_{k_\beta}\}, Y_1)$
25. $c = X' + W(\{v_{k_l}, v_{k_2}\}, v_i)$
26. $I_{i,j} = I_{i,j} \cup \{(c, Y_1, T(E'), \{(X_{k_\beta}, Y_1)\})\}$
27. end if
28. end if
29. end for
30. end for
31. end for
32. end for

定理 1. 在 \mathcal{H}^2 中, 对任意点 $\langle i, j \rangle$, 如果存在从 $\langle s, s \rangle$ 到 $\langle i, j \rangle$ 的同源反向超路 P , 令 $X = W(P_r)$,

$Y = W(P_H)$, 当算法结束时, 存在 $(X, Y, pred) \in I_{i,j}$ 且 $I_{i,j}$ 中仅包含同源反向超路对应的标签。

证明: 对 $i+j$ 数学归纳, 当 $i+j=2$ 时, $i=j=1$, 在初始化步骤中算法设置了 $I_{1,1} = \{(0,0, NULL)\}$ 。假设 $i+j < k$ 时结论成立。当 $i+j=k$ 时, 若路 P 上的最后一条反向弧的尾节点为 α , 不妨设最后一条反向弧为 $E' = (\langle i, x_\beta \rangle, \langle i, j \rangle)$, 沿着路 P 从 $\langle s, s \rangle$ 到 $\langle i, x_\beta \rangle$ 的反向超路记为 P^{x_β} , 并且记 $W(P_V^{x_\beta}) = X^{\beta+}$, $W(P_H^{x_\beta}) = Y^{\beta+}$, 根据超路权值计算方式得到 $X = X^{\beta+}$, $Y = \max\{Y^{\beta+} \mid \beta=1,2,\dots,\alpha\} + W(E')$ 。因为 $x_\beta < j$, $i+x_\beta < i+j=k$, 由归纳得 $(X^{\beta+}, Y^{\beta+}, pred) \in I_{i,x_\beta}$ 。若满足第 10 行的条件 $P_1^{x_1} = P_1^{x_2} = \dots = P_1^{x_\alpha}$, 执行第 11~13 行, 得到新的标签 $(X, Y, T(E'), \{(X, Y^{\beta+})\})$ 并存入 $I_{i,j}$ 中, 根据第 2 节反向超路权值的计算方式可得 $X = W(P_V), Y = W(P_H)$ 。所以 $I_{i,j}$ 中存在 $(X, Y, pred)$ 。

还需证明 $I_{i,j}$ 中仅包含同源有向 B 路对应的标签。反证法: 假设 $I_{i,j}$ 存在 $(X, Y, pred)$ 对应了一条非同源有向 B 超路 P , 则在 P 上至少存在两个点 $\langle x, y_1 \rangle, \langle x, y_2 \rangle$ (或 $\langle x_1, y \rangle, \langle x_2, y \rangle$) 沿着路 P 到达这两点的纵向弧(或水平弧)不相同, 不妨设为 $\langle x, y_1 \rangle, \langle x, y_2 \rangle$, 沿着路 P 到达这两点的纵向路分别记作 $P_x^{y_1}, P_x^{y_2}$, 根据第 12 行的判断条件, 由于 $P_x^{y_1} \neq P_x^{y_2}$ 并不会执行后面建立新标签的步骤, 故不会放到 $I_{i,j}$ 中, 矛盾。

复杂度: 第 3~4 行需要 n^2 次循环, 第 5 行寻找后向边最多为 m 条边, 第 7 行的循环次数为 $|I_{i,k_1}| |I_{i,k_2}| \dots |I_{i,k_\alpha}|$, 最多不超过 $|I|^2$ 。第 9 行的路径回溯需要 $O(n)$ 的复杂度, 所以算法 1 的复杂度为 $O(n^3 m |I|^2)$, 其中 $|I| = n^2 W'^2$ 是标签集的最大容量, W' 为权值的最大值。

4.2. Min-Max 不相交反向超路算法

算法 2 为 Min-Max 不相交反向超路算法, 该算法为主算法。首先调用算法 1 得到节点 $\langle t, t \rangle$ 的标签集 $I_{t,t}$ 。其次对 $I_{t,t}$ 中每一个标签的水平权值和纵向权值的最大值即 $\max\{X, Y\}$ 进行比较, 如 $\max\{X, Y\}$ 若较小就赋给 $(X^*, Y^*, pred)$, 最终得到的 $(X^*, Y^*, pred)$ 就是最优解 $\min(\max\{X, Y\})$ 。最后调用算法 3 得到 $(X^*, Y^*, pred)$ 对应的两条不相交反向超路 P_1, P_2 。

下面给出算法 2 的具体步骤:

Algorithm 2. Min-Max disjoint backward hyperpaths algorithm

算法 2. Min-Max 不相交反向超路

输入: $\mathcal{H} = (\mathcal{V}, \mathcal{E}, W)$, s, t
 输出: $(X^*, Y^*, pred)$, P^1, P^2

1. 执行计算标签算法(算法 1)
2. $(X^*, Y^*, pred) = (\infty, \infty, NULL)$
3. for each $(X, Y, pred) \in I_{t,t}$
4. if $\max\{X, Y\} < \max\{X^*, Y^*\}$
5. $(X^*, Y^*, pred) = (X, Y, pred)$
6. end if
7. end for
8. $[P_1, P_2] = BACK(L_{t,t}(X^*, Y^*, pred))$

定理 2: 算法 2 输出的 P_1, P_2 是最优解。

证明: 假设最优解为 P_1^*, P_2^* , 权值分别为 $W(P_1^*)$ 和 $W(P_2^*)$ 。根据引理 1, 在 $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ 中存在同源

反向超路 P^* 与之对应。令 $X' = W(P_V^*) = W(P_1^*), Y' = W(P_H^*) = W(P_2^*)$, 根据算法 1 的正确性, $(X', Y', pred) \in I_{i',j'}$ 。若算法 2 没有输出最优解 P^* 对应的标签 $(X', Y', pred)$, 假设输出了另外一条同源反向路 P' 对应的标签 $(X, Y, pred)$, 算法中对标签进行比较后选较小者, 所以 $\max\{X, Y\} < \max\{X', Y'\}$ 。由于 P^* 是最优解, 那么一定有 $\max\{X, Y\} \geq \max\{X', Y'\}$, 得出矛盾。故算法 2 输出的 P_1, P_2 是 Min-Max 不相交反向超路的最优解。

复杂度: 执行算法 1 需要 $O(n^3 m |I|^2)$, 对 $I_{i',j'}$ 中的每一个标签比较, 复杂度为 $O(|I|)$, 最后路径回溯复杂度为 $O(n)$, 以上三个过程没有嵌套关系, 故算法 2 的复杂度为 $O(n^3 m |I|^2)$, 其中 $|I|$ 为标签集的最大容量, $|I| = n^2 W'^2$, W' 为权值的最大值。

4.3. 标签回溯算法

标签回溯算法是算法 1 和算法 2 调用的子算法, 利用标签集得到在 \mathcal{H} 中对应的两条不交反向超路。首先输入已有的标签集、到达的节点 $\langle i, j \rangle$ 和标签 $L_{i,j} = (X, Y, \{i', j'\}, \{X_i^+, Y_j^+\})$ 。现设定队列 N 满足先进先出原则, 初始只有 $L_{i,j}$ 并将其入队。因为要回溯到源节点 $\langle s, s \rangle$, 而 $I_{s,s} = \{(0, 0, NULL)\}$, 只要当出队标签的 $pred \neq NULL$ 说明需要继续回溯。根据前继点判断前继弧是水平弧还是纵向弧, 如果是水平弧即 $i' = i$ 就将 \mathcal{H} 中的反向弧 $(\{j'\}, j)$ 加到 $\mathcal{E}(P_2)$ 中, 如果是纵向弧即 $j' = j$ 就将反向弧 $(\{i'\}, i)$ 加到 $\mathcal{E}(P_1)$ 中。为了从前继点继续追溯, 将每一个 $L_{i',j'} = (X_{i'}^+, Y_{j'}^+, pred)$ 入队到 N , 返回循环直至 N 为空。下面给出算法步骤:

Algorithm 3. Label backtracking algorithm (BACK)

算法 3. 标签回溯算法

输入: $\mathcal{H}_2(\mathcal{V}_2, \mathcal{E}_2)$, $\{I_{i,j} \mid 1 \leq i \leq i_1, 1 \leq j \leq j_1\}$, 标签 $L_{i,j} = (X, Y, \{i', j'\}, \{X_i^+, Y_j^+\})$

输出: $P_1^{i,j}, P_2^{i,j}$

1. $N = \{(X, Y, \{i', j'\}, \{X_i^+, Y_j^+\})\}$
2. while $N \neq \emptyset$
3. $L_{i,j} = (X, Y, \{i', j'\}, \{X_i^+, Y_j^+\})$ 从 N 出队
4. if $\{i', j'\}, \{X_i^+, Y_j^+\} \neq NULL$
5. if $i' = i$
6. if $(\{j'\}, j) \notin \mathcal{E}(P_2)$
7. $\mathcal{E}(P_2) = \mathcal{E}(P_2) \cup \{(\{j'\}, j)\}$
8. end if
9. else
10. if $(\{i'\}, i) \notin \mathcal{E}(P_1)$
11. $\mathcal{E}(P_1) = \mathcal{E}(P_1) \cup \{(\{i'\}, i)\}$
12. end if
13. end if
14. $N = N \cup \{L_{i',j'} = (X_{i'}^+, Y_{j'}^+, pred)\}$
15. end while

复杂度: 根据有向无圈图的排序原则, 到达 v_i 和 v_j 的路上最多有 i' 和 j' 个点, 回溯时最多有 $i' + j' + 2$ 个点入队到 N 且最多仅入队 1 次, 所以循环次数最多为 $i' + j' + 2$ 次, 计算复杂度为 $O(n)$ 。

5. 近似算法

基于上述求 Min-Max 不相交反向超路问题的最优解, 本节通过对路径权值的近似, 降低了算法 1 中每一个节点的标签数量, 进而设计了 $1+\varepsilon$ 近似算法降低了计算复杂性。

首先依照第 3 节构建辅助超图模型。其次, 对每一个节点 $\langle i, j \rangle$, 建立新标签集合 $I_{i,j}$ 并将其新标签记为 $L_{i,j}[a,b] = (X, Y, pred)$, 其中 $a = \lfloor \log_{\delta} X \rfloor + 1$, $b = \lfloor \log_{\delta} Y \rfloor + 1$, $a, b \leq \lfloor \log_{\delta} (nW') \rfloor + 1$, 并且令 $\delta = (1+\varepsilon)^{\frac{1}{2n-2}}$, 当 $\varepsilon > 0$ 时, $\delta > 1$ 。接下来给出新标签计算算法, 具体步骤如下。

在初始化过程中, 令每一个节点 $\langle i, j \rangle$ 的新标签为 $L_{i,j}[a,b] = NULL$, 接下来初始化源节点 $\langle 1, 1 \rangle$ 的新标签为 $L_{1,1}[1,1] = (0, 0, NULL)$ 。

接下来改进算法 1 计算对各点的新标签。算法 1 中第 3~5 行循环条件不变, 当遍历的反向弧为水平弧时, 将第 7 行变为对 $I_{i,k_{\beta}}$ 中的每一个 $L_{i,k_{\beta}}[a,b] = (X_{k_{\beta}}, Y_{k_{\beta}}, pred)$ 进行比较, 8~11 行步骤不变, 第 12 行替换为 $c = \lfloor \log_{\delta} (Y' + W(E')) \rfloor + 1$ 。当 $i = j = t$ 时, 为了删除部分冗余的标签, 如果 $L_{i,t}[a,c] = \emptyset$, 执行第 13 行 $L_{i,j}[a,c] = (X_1, Y' + W(E'), T(E'), \{(X_{k_{\beta}}, Y_{k_{\beta}})\})$ 和 $I_{i,j} = I_{i,j} \cup \{L_{i,j}[a,c]\}$, 否则不更新 $I_{i,j}$ 。当遍历的反向弧为纵向弧时, 将第 20 行变为对 $I_{k_{\beta},j}$ 中每一个 $L_{k_{\beta},j}[a,b] = (X_{k_{\beta}}, Y_{\beta}, pred)$ 进行比较, 21~24 行步骤不变, 第 25 行替换为 $c = \lfloor \log_{\delta} (X' + W(E')) \rfloor + 1$ 。如果 $i = j = t$ 时, 为了删除部分冗余的标签, 如果 $L_{i,t}[c,b] = \emptyset$, 执行第 26 行 $L_{i,j}[c,b] = (X' + W(E'), Y_1, T(E'), \{(X_{k_{\beta}}, Y_{\beta})\})$ $I_{i,j} = I_{i,j} \cup \{L_{i,j}[c,b]\}$, 否则不更新 $I_{i,j}$ 。

由于标签有了变化, 通过执行上述算法, 将反向超路的两个权值存储到了 $a, b \leq \lfloor \log_{\delta} (nW') \rfloor + 1$ 以内的新标签中, 并且在 $I_{i,t}$ 内的标签由于压缩出现重复值, 我们添加了 $L_{i,t}[c,b] = \emptyset$ 条件筛选出冗余的标签, 执行算法 4 在 $I_{i,t}$ 中比较 $L_{i,t}[a,b]$ 存储的 $(X, Y, pred)$, 得到近似解 $(X^*, Y^*, pred)$ 。具体步骤如下:

Algorithm 4. Approximation algorithm for Min-Max target

算法 4. Min-Max 目标近似算法

输入: $\mathcal{H} = (\mathcal{V}, \mathcal{E}, W)$, s, t

输出: $(X^*, Y^*, pred)$, P^1, P^2

1. 执行上述新标签计算算法
2. $(X^*, Y^*, pred) = (\infty, \infty, NULL)$
3. for each $L_{i,t}[a,b] = (X, Y, pred) \in I_{i,t}$
4. if $\max\{X, Y\} < \max\{X^*, Y^*\}$
5. $(X^*, Y^*, pred) = (X, Y, pred)$
6. end if
7. end for
8. $[P_1, P_2] = BACK(L_{i,t}(X^*, Y^*, pred))$

引理 2: 假设在 \mathcal{H}_2 中存在一条从 $\langle s, s \rangle$ 到 $\langle i, j \rangle$ 同源有向 B 超路 P , 令 $X = W(P_V), Y = W(P_H)$, 当近似算法结束时, 节点 $\langle i, j \rangle$ 存在新标签 $L_{i,j}[a,b] = (X', Y', pred) \in I'_{i,t}$ 使得 $X/\delta^{i+j-2} \leq X' \leq \delta^{i+j-2} X$, $Y/\delta^{i+j-2} \leq Y' \leq \delta^{i+j-2} Y$ 。

证明: 对 $i+j$ 进行归纳, 当 $i+j=2$ 时, 由于初始化中在 $I_{1,1}$ 中只有 $(0,0)$, 显然满足结论。当 $i+j < k$ 时, 假设结论成立。令 $i+j=k$, 假设在 \mathcal{H}_2 中存在一条从 $\langle s,s \rangle$ 到 $\langle i,j \rangle$ 同源反向超路 P , 不妨假设超路 P 上点 $\langle i,j \rangle$ 的后向弧为 $E' = (\langle i, x_\beta \rangle, \langle i, j \rangle)$, 沿着 P 到达点 $\langle i, x_\beta \rangle$ 的纵向权值和水平权值分别记为 $W(P_V^{x_\beta}) = X^{\beta+}$, $W(P_H^{x_\beta}) = Y^{\beta+}$, 则可以得到 $X = X^{\beta+}$, $Y = \max\{Y^{\beta+} \mid \beta=1,2,\dots,\alpha\} + W(E')$ 。又因为 $x_\beta < j$, 所以 $i+x_\beta < i+j=k$, 由归纳假设, 节点 $\langle i, x_\beta \rangle$ 的标签集中存在标签 $(X^{\beta'}, Y^{\beta'}, pred)$ 满足

$$X^{\beta+} / \delta^{i+x_\beta-2} \leq X^{\beta'} \leq \delta^{i+x_\beta-2} X^{\beta+}$$

$$Y^{\beta+} / \delta^{i+j-2} \leq Y^{\beta'} \leq \delta^{i+j-2} Y^{\beta+}$$

节点 $\langle i, j \rangle$ 的新标签为 $(X^{\beta'}, Y^{\beta'} + W(E'), \langle i, x_\beta \rangle, \{X^{\beta'}, Y^{\beta'}\})$ 。令 $X' = X^{\beta'}$ 和 $Y' = Y^{\beta'} + W(E')$, 得

$$X' = X^{\beta'} \leq \delta^{i+x_\beta-2} X^{\beta+} \leq \delta^{i+j-2} X,$$

$$\begin{aligned} Y' &= Y^{\beta'} + W(E') \\ &\leq \delta^{i+x_\beta-2} Y^{\beta+} + W(E') \\ &\leq \delta^{i+x_\beta-2} (Y^{\beta+} + W(E')) \\ &\leq \delta^{i+x_\beta-2} Y \\ &\leq \delta^{i+j-2} Y \end{aligned}$$

$$X' = X^{\beta'} \geq X^{\beta+} / \delta^{i+x_\beta-2} \geq X / \delta^{i+j-2}$$

由于 $\delta > 1$ 所以

$$\begin{aligned} Y' &= Y^{\beta'} + W(E') \\ &\geq Y^{\beta+} / \delta^{i+x_\beta-2} + W(E') \\ &\geq (Y^{\beta+} + W(E')) / \delta^{i+x_\beta-2} \\ &\geq Y / \delta^{i+x_\beta-2} \\ &\geq Y / \delta^{i+j-2} \end{aligned}$$

定理 3: Min-Max 不相交反向超路近似算法得到 $1+\varepsilon$ 近似解。

证明: 假设从 $\langle 1,1 \rangle$ 到 $\langle n,n \rangle$ 最优解为 P_1^*, P_2^* , 并且其对应的权值分别为 X^*, Y^* , 近似算法得到的解为 $(X, Y, pred)$ 。根据引理 2, 节点 $\langle n,n \rangle$ 存在标签 $L_{i,j}[a,b] = (X', Y', pred) \in I'_{i,j}$ 使得 $X' \leq \delta^{2n-2} X^*$ 和 $Y' \leq \delta^{2n-2} Y^*$ 。因为 $\delta = (1+\varepsilon)^{\frac{1}{2n-2}}$, 所以 $X' \leq (1+\varepsilon) X^*$, $Y' \leq (1+\varepsilon) Y^*$ 。那么

$$\begin{aligned} \max\{X, Y\} &\leq \max\{X', Y'\} \\ &\leq \max\{(1+\varepsilon) X^*, (1+\varepsilon) Y^*\} \\ &= (1+\varepsilon) \max\{X^*, Y^*\} \end{aligned}$$

因此, 该不相交反向超路近似算法得到了 $1+\varepsilon$ 近似解。

复杂度: Min-Max 不相交反向超路近似算法的复杂度为 $O(n^3 m |I'|^2)$, 其中 $|I'| = (\log_\delta(nW'))^2$, W' 为反向弧权值的最大值。该近似算法的复杂度计算过程与 4.2 节类似, 在此不再赘述。

6. 结语

在本篇论文中, 我们研究了在无圈反向超图中寻找固定源节点到目标节点间的两条点不相交反向超路的设计优化问题, 采用 Min-Max 作为优化目标。通过建立辅助超图模型, 将原反向超图的 Min-Max 不交超路

问题转化为在辅助超图中找一条具有两个边权函数的同源反向超路优化问题。当给定反向弧尾节点个数的上限 λ 时, 给出算法 1 计算辅助超图中目标节点的标签, 从而通过算法 2 获得了满足 Min-Max 目标的同源反向超路, 在算法 2 中调用算法 3 得到不相交反向超路问题的最优解。主算法的复杂度为 $O(n^3 m |I|^\lambda)$, 其中 $|I| = n^2 W'^2$ 。为进一步降低该伪多项式算法的计算复杂性, 本文设计了复杂度为 $O(n^3 m |I'|^\lambda)$ 的 $(1+\varepsilon)$ 近似算法, 其中 $|I'| = (\log_\delta(nW'))^2$ 。之后我们将继续在网络中多路径优化方面进行研究, 包括 Min-Sum 和限制 Min-Sum 等不同的优化目标。该研究可以应用到不同的场景和复杂网络当中, 为提高网络和路径容错性提供解决方案。

参考文献

- [1] 姚春龙, 李旭, 沈岚. 公交出行最优路径搜索的有向赋权图模型[J]. 计算机应用研究, 2013, 30(4): 1058-1063.
- [2] 王雪伟, 刘三阳, 张朝辉. k-不相交路径的容错拓扑控制算法[J]. 吉林大学学报(理学版), 2017, 55(3): 635-640.
- [3] Shiloach, Y. and Perl, Y. (1978) Finding Two Disjoint Paths between Two Pairs of Vertices in a Graph. *Journal of the ACM (JACM)*, **25**, 1-9. <https://doi.org/10.1145/322047.322048>
- [4] Fleischer, R., Ge, Q., Li, J. and Zhu, H. (2007) Efficient Algorithms for k-Disjoint Paths Problems on Dags. In: Kao, M.-Y. and Li, X.-Y., Eds., *International Conference on Algorithmic Applications in Management*, Springer, Berlin, 134-143. https://doi.org/10.1007/978-3-540-72870-2_13
- [5] Berge, C. (1973) *Graphs and Hypergraphs*.
- [6] Noh, H., Hickman, M. and Khani, A. (2012) Hyperpaths in Network Based on Transit Schedules. *Transportation Research Record*, **2284**, 29-39. <https://doi.org/10.3141/2284-04>
- [7] Pretolani, D. (2000) A Directed Hypergraph Model for Random Time Dependent Shortest Paths. *European Journal of Operational Research*, **123**, 315-324. [https://doi.org/10.1016/S0377-2217\(99\)00259-3](https://doi.org/10.1016/S0377-2217(99)00259-3)
- [8] 崔阳, 杨炳儒. 超图在数据挖掘领域中的几个应用[J]. 计算机科学, 2010, 37(6): 220-222.
- [9] Nielsen, L.R., Andersen, K.A. and Pretolani, D. (2005) Finding the K Shortest Hyperpaths. *Computers and Operations Research*, **32**, 1477-1497. <https://doi.org/10.1016/j.cor.2003.11.014>
- [10] Ausiello, G., Italiano, G.F. and Nanni, U. (1998) Hypergraph Traversal Revisited: Cost Measures and Dynamic Algorithms. In: Ausiello, G., Italiano, G.F. and Nanni, U., Eds., *International Symposium on Mathematical Foundations of Computer Science*, Springer, Berlin, 1-16. <https://doi.org/10.1007/BFb0055754>
- [11] Borndörfer, R. and Karbstein, M. (2012) A Note on Menger's Theorem for Hypergraphs.
- [12] Gao, J., Zhao, Q., Ren, W., Swami, A., Ramanathan, R. and Bar-Noy, A. (2014) Dynamic Shortest Path Algorithms for Hypergraphs. *IEEE/ACM Transactions on Networking*, **23**, 1805-1817. <https://doi.org/10.1109/TNET.2014.2343914>
- [13] Frank, A. (2011) *Connections in Combinatorial Optimization*. Oxford University Press, Oxford.
- [14] Gallo, G., Longo, G., Pallottino, S. and Nguyen, S. (1993) Directed Hypergraphs and Applications. *Discrete Applied Mathematics*, **42**, 177-201. [https://doi.org/10.1016/0166-218X\(93\)90045-P](https://doi.org/10.1016/0166-218X(93)90045-P)
- [15] Ritz, A. and Murali, T.M. (2014) Pathway Analysis with Signaling Hypergraphs. In: *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, Newport Beach, 20-23 September 2014, 249-258. <https://doi.org/10.1145/2649387.2649450>