

单平面有向无圈图中最小路覆盖问题的算法研究

管锐*, 梁东岳, 杨卫华

太原理工大学数学学院, 山西 太原

收稿日期: 2023年3月24日; 录用日期: 2023年4月18日; 发布日期: 2023年4月26日

摘要

一个铁路区段在规划时间内的时空网络是一个仅包含一对源与汇的有向无圈平面图。每个列车都可由该网络中的一条有向路表示。本文研究上述网络中的最小路覆盖问题, 即至少用多少条有向路可以覆盖图中所有的边。根据单平面有向无圈图的单源单汇和平面性等结构性质, 本文给出了上述问题的一个时间复杂度为 $O(nk)$ 的精确算法, 这里 n 表示图中顶点数、 k 表示图中最大有向割所包含边的数目。

关键词

最小路覆盖, 有向无圈图, 最大有向割, 精确算法

Research of Minimum Path Covering Problem Algorithm in Single Planar Directed Acyclic Graph

Rui Guan*, Dongyue Liang, Weihua Yang

College of Mathematics, Taiyuan University of Technology, Taiyuan Shanxi

Received: Mar. 24th, 2023; accepted: Apr. 18th, 2023; published: Apr. 26th, 2023

Abstract

The space-time network of a railway section in the planning time is a directed acyclic graph that contains only a pair of sources and sinks. Each train can be represented by a directed path in the network. In this paper, we study the minimum path covering problem in the above net-

*通讯作者。

work, that is, at least how many directed paths can cover all edges in the network. According to the structural properties of single-source, single-sink and planarity, we give an $O(nk)$ time exact algorithm to solve the minimum path covering problem in single planar directed acyclic graph, where n is the number of vertices in graph and k is the number of edges in the maximum directed cut.

Keywords

Minimum Path Covering, Directed Acyclic Graph, Maximum Directed Cut, Exact Algorithm

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

在货运铁路运输当中, 商品在铁路运输中是由火车运输的。火车由机车和火车车厢组成。火车车厢用来装载货物, 机车为火车的运行提供动力。在规划时间内, 一段铁路区间上表示货运需求的时空网络是一个仅包含一对源与汇的平面有向无圈图, 从而列车的行驶路线可由该网络中的一条有向路表示。

在研究铁路上的货运分配问题, 常用的策略是确定铁路通道的一个主要方向, 重点是研究沿这个方向的货物流动。比如, Caprara [1]考虑的是铁路中的南北方向。因此, 我们可将所有的货运走廊分配一个特定的方向, 而不考虑任何沿相反方向移动的路径, 故可用一个平面有向无圈图代表铁路通道, 其弧形集表示铁路线段的集合, 节点集表示线段之间的货运站点的集合。

本文考虑可带重边的有向无圈图。首先给出一些基本的定义和符号。 $D=(V, A)$ 是有向图, V 表示图中顶点组成的集合, A 表示图中有向边组成的集合。对 D 的任意边 $a=(u, v)$, u 为 a 的尾部, v 为 a 的头部。对 D 的任意顶点 v , $\delta^+(v)$ 表示以 v 为尾部的边组成的集合, $\delta^-(v)$ 表示以 v 为头部的边组成的集合。 p 是 D 中的一条带定向的路(圈), 如果其定向与 D 相同, 则称 p 是 D 中的一条有向路(圈)。若有向图 D 不包含任意有向圈称其为有向无圈图。若图 D 可画到平面上使得它的边仅在端点相交则称其为平面图。其它术语可参考文献[2]。

有向无圈图的最小路覆盖问题是找覆盖有向无圈图所有边的有向路集合使得路的数目最少。一种方法是转化为最大匹配问题。第一步是计算有向无圈图 D 的边传递闭包 $TC(D)$, 然后将 D 的最小路覆盖问题转化为 $TC(D)$ 的最小路因子问题。Ntafos 和 Hakimi [3]将最小路因子问题转换为最大二分匹配问题, 而解决最大二分匹配的最好算法其时间复杂度为 $O(n^{2.5})$ [4]和 $O(\sqrt{nm})$ [5], 这里 n 为图中顶点数目、 m 为图中边的数目。另一种方法是将最小路问题转化为具有单位下界的最小可行流问题, 即首先找到最小流, 然后通过分解算法将流分解为有向路。Ahuja [6]通过两次应用任何寻找最大流的算法找到最小可行流。最著名的最大流量算法由 Goldberg [7]和 Orli [8]给出, 从而最小可行流问题在 $O(|V|^2|A|^{0.5})$ 时间和 $O(|V||A|)$ 时间解决, 其中 $|V|$ 是网络的顶点数、 $|A|$ 是网络的边数。虽然有向无圈图的流分解可以在线性参数时间内计算, 但是解决问题的时间复杂度至少是参数的二次方数量级。

本文主要研究单平面有向无圈图的最小路覆盖问题, 并针对问题设计了一个时间复杂度为 $O(nk)$ 的精确算法, 这里 n 为图的顶点数目、 k 为图的最大有向割中边的数目。

2. 相关定义

首先给出算法设计和分析过程中需要用到的一些定义。

定义(平面有向无圈图)设 D 是一个有向无圈图, 如果 D 是一个平面图, 则称 D 是一个平面有向无圈图。

定义(最小路覆盖问题)设 D 是一个有向无圈图, 最小路覆盖问题是找到 D 的一个有向路集合 $\{p_1, \dots, p_i\}$ 其满足 i 最小且 $A(D) = \bigcup_{1 \leq j \leq i} A(p_j)$ 。覆盖 D 的最小路径数记为 $pc(D)$ 。

定义(可比关系)设 D 是一个有向无圈图, 如果存在一条有向路连接 $a' = (u', v')$ 到 $a'' = (u'', v'')$, 则称 a' 可比于 a'' , 记作 $a' < a''$ 。如果不存在一条有向路连接 $a' = (u', v')$ 到 $a'' = (u'', v'')$, 则记作 $a' \not< a''$ 。如果 a' 与 a'' 满足 $a' \not< a''$ 且 $a'' \not< a'$, 则称 a' 与 a'' 不可比, 记作 $a' \times a''$ 。

定义(源 s 与汇 t)设 D 是一个有向无圈图, 如果 $\delta^-(s) = \emptyset$, 则称顶点 s 是源。如果 $\delta^+(t) = \emptyset$, 则称顶点 t 是汇。

定义(面)设 D 是一个有向无圈图, 如果平面被 D 分成有限数量的连通区域并且 F 是这些区域之一, 则称 F 是一个面。如果面 F 是无界的, 则称 F 为外表面。

定义(单平面有向无圈图)设 $D = (V \cup \{s, t\}, A)$ 是一个平面有向无圈图, 如果其只有一个源 s 和一个汇 t 且都位于外表面的边界, 则称 $D = (V \cup \{s, t\}, A)$ 是一个单平面有向无圈图。

图 1 展示了一个单平面有向无圈图的例子。

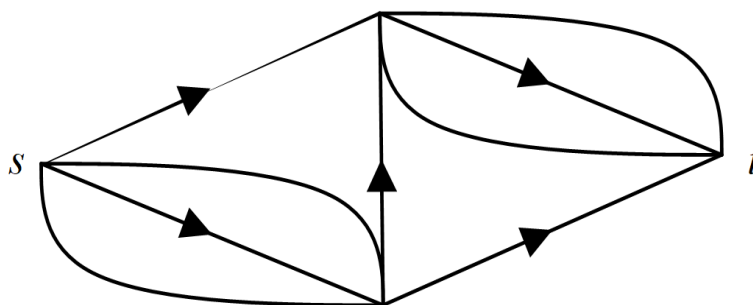


Figure 1. A single planar directed acyclic graph

图 1. 一个单平面有向无圈图

定义(有向割)设 D 是一个有向无圈图, 如果 $S \subset V$ 且 $\delta^-(S) = \emptyset$, 则称 $\delta^+(S)$ 是 D 的一个有向割。此外, D 的最大有向割中边的数目记作 $|\delta^+(D)|$ 。

例如图 1 中 $\delta^+(s)$ 是该图的一个有向割, 其也是一个最大有向割。

定义(界边 boundary arc)设 D 是一个单平面有向无圈图, 如果边 a 是外表面的边界, 则称 a 是一条界边。

定义(界路 boundary path)设 $D = (V \cup \{s, t\}, A)$ 是一个单平面有向无圈图, 如果 p 是一条连接 s 与 t 的路且其每条边都是界边, 则称 p 是一条界路。

定义(反链)设 D 是一个有向图, A_0 是 D 中的一个非空边集合。如果边集 A_0 中任意两个互异的边都不可比, 则称边集 A_0 是 D 的一个反链。

3. 算法设计与分析

本文给出了时间复杂度为 $O(nk)$ 精确算法来解决单平面有向无圈图中的最小路覆盖问题, 这里 n 表示图中顶点数目、 k 表示最大有向割中边的数目。算法是根据图的单源单汇性, 平面性和有向无圈性进行

设计的。本文中算法的设计思路是先找单平面有向无圈的一条界路，然后按照找到的界路进行删边，最后恢复最少的边以保持图中其余边的可变关系不发生改变。

算法 1 是最小路覆盖算法，该算法为主算法。首先调用**算法 1**，对每个顶点 v 建立一个新的标签并附为空。其次判断图中是否存在边未被有向路所覆盖。如果存在未覆盖的边，则需要找一条新的有向路，从而进入**算法 2** 找新的界路。如果不存在未覆盖的边，则算法停止，输出所找到的有向路集合。**算法 1** 的伪代码如下。

Algorithm 1. Minimum path cover algorithm

算法 1. 最小路覆盖算法

Input: A single planar directed acyclic graph $D = (V \cup \{s, t\}, A)$

Output: directed paths set $\{p_1, \dots, p_i\}$

1. **Foreach** $v \in V \cup \{s, t\}$
2. $v \cdot \pi = NIL$
3. $i = 0$
4. **while** $\delta^+(s) \neq \emptyset$
5. $i = i + 1$
6. $p_i = \emptyset$
7. $B_i = \emptyset$
8. $Pf(D, s)$
9. **return** p_1, \dots, p_i

寻路算法是**算法 1** 调用的子算法，其利用单平面有向无圈的结构特点，找出一条界路并依次处理路中的边。首先从源点 s 出发，找一条界边，并将界边的头端点与尾端点关联。其次判断是否存在新的界边与前一个界边相关联。如果存在，则继续调用**算法 2**，然后寻路。否则调用**算法 3**，对有向路中的边做处理。下面给出算法步骤：

Algorithm 2. Path-finding algorithm- $Pf(D, v)$

算法 2. 寻路算法 - $Pf(D, v)$

Input: A single planar directed acyclic graph $D = (V \cup \{s, t\}, A)$ and a vertex

Output: abound arypath p_i

1. **if** $\delta^+(v) \neq \emptyset$
2. find $(v, u) \in \delta^+(v)$ such that (v, u) is a boundary arc
3. $u \cdot \pi = v$
4. $p_i = p_i \cup \{(v, u)\}$
5. $Pf(D, u)$
6. **else** $Delete(D, v)$

删除算法是**算法 2** 调用的子算法，其作用是处理已找到有向路中的边。首先判断是否存在未处理的界边。如果不存在，则有向路中所有的界边都被处理，算法停止。如果存在，则找出相应的界边并将其从图中删除。然后判断被删除的界边是否影响其他边的可比关系，若不影响则继续调用**算法 3**，否则调用**算法 4**。删除算法的伪代码如下。

Algorithm 3. Deletion algorithm- $Delete(D, u)$ **算法 3.** 删除算法 - $Delete(D, u)$

Input: A single planar directed acyclic graph $D=(V \cup \{s, t\}, A)$ and a vertex v

Output: A single planar directed acyclic subgraph D and a vertex u

1. **if** $v \cdot \pi \neq NIL$
2. **find** $u = v \cdot \pi$ and a boundary arc (u, v)
3. $v \cdot \pi = NIL$
4. $A = A - \{(u, v)\}$
5. **if** $|\delta^-(v)| = 0$
6. **if** $|\delta^+(v)| > 0$
7. $A = A \cup \{(u, v)\}$
8. $Delete(D, u)$
9. **else** $|\delta^+(v)| = 0$
10. $B_i = B_i \cup \{(u, v)\}$
11. $Recover(D, u)$
12. **else** $|\delta^-(v)| > 0$
13. **if** $|\delta^+(u)| = 0$
14. **if** $|\delta^-(u)| \geq 1$
15. **if** $|\delta^-(u)| \geq 2$
16. $A = A \cup \{(u, v)\}$
17. $Delete(D, u)$
18. **else** $|\delta^-(u)| = 1$
19. $B_i = B_i \cup \{(u, v)\}$
20. $Recover(D, u)$
21. **else** $|\delta^-(u)| = 0$
22. stop
23. **else** $|\delta^+(u)| > 0$
24. $Delete(D, u)$
25. **else stop**

恢复算法是算法 3 调用的子算法，其作用是恢复最少的界边以保持其它边的可比关系不变。首先判断顶点是否为源点。如果顶点为源点，则恢复界边数为 0，算法停止。如果非源点，则进行判断所恢复的界边集是否能保持可比关系不变。若能保持，则将界边集合添加到图中，然后调用算法 3。若不能，则继续调用算法 4 来寻找满足条件的界边集。算法的伪代码如下。

定理 1 设 $D=(V \cup \{s, t\}, A)$ 是一个单平面有向无圈图，算法 1 可在 $O(n|\delta^+(D)|)$ 时间内找到覆盖 D 的最小有向路集合 $\{p_1, \dots, p_i\}$ ，这里 n 为图中顶点数目、 $i = |\delta^+(D)|$ 。

为方便证明定理 1，我们先给出一些命题及其证明。

命题 2 如果 $D=(V, A)$ 是一个有向无圈图，则 $pc(D) \geq |\delta^+(D)|$ 。

Algorithm 4. Recovery algorithm- $Recover(D, v)$ **算法 4.** 恢复算法 - $Recover(D, v)$

 Input: A single planar directed acyclic graph $D = (V \cup \{s, t\}, A)$ and a vertex v

 Output: A single planar directed acyclic subgraph D and a vertex u

1. **if** $v \cdot \pi \neq NIL$
 2. **if** $|\delta^+(v)| = 0$
 3. **if** $|\delta^-(v)| = 1$
 4. find $u = v \cdot \pi$ and a boundary arc (u, v)
 5. $A = A - \{(u, v)\}$
 6. $B_i = B_i \cup \{(u, v)\}$
 7. $Recover(D, u)$
 8. **else** $|\delta^-(v)| \geq 2$
 9. $A = A \cup B_i$
 10. $B_i = \emptyset$
 11. $Delete(D, v)$
 12. **else** $|\delta^+(v)| > 0$
 13. $B_i = \emptyset$
 14. $Delete(D, v)$
 15. **else stop**
-

证明对于有向无圈图 $D = (V, A)$ 中任意有向割 $\delta^+(S)$, 其满足 $\delta^-(S) = \emptyset$, 故 D 中任意有向路 p 至多包含 $\delta^+(S)$ 中一条边。由于 $pc(D)$ 是最小有向路覆盖数, 所以 $pc(D) \geq |\delta^+(D)|$ 成立。

事实上, 当 D 是有向无圈图时, 其满足 $pc(D) = |\delta^+(D)|$, 这一结论可直接由定理的证明得出。

命题 3 设 $\delta^+(S)$ 是单平面有向无圈图 $D = (V \cup \{s, t\}, A)$ 的一个最大有向割, 如果 p 是一条界路, 那么 $|\delta^+(S) \cap A(p)| = 1$ 成立。

证明令 $\delta^+(S)$ 是单平面有向无圈图 $D = (V \cup \{s, t\}, A)$ 的一个最大有向割。因为 $\delta^-(S) = \emptyset$ 且 D 有唯一源 s 汇 t , 所以 $s \in S$ 且 $t \notin S$ 。由于 p 是连接 s 和 t 的界路并且 $\delta^+(S)$ 将 s 与 t 分割开, 故有向路 p 与 $\delta^+(S)$ 由交集, 即 $|\delta^+(S) \cap A(p)| \geq 1$ 。又因为 $\delta^-(S) = \emptyset$, 所以 $|\delta^+(S) \cap A(p)| \leq 1$ 。综上, $|\delta^+(S) \cap A(p)| = 1$ 成立。

命题 4 如果 D 是一个有向无圈图, 那么其最大有向割中边的数目等于最大反链中边的数目。

证明设 $\delta^+(S)$ 是 D 的一个最大有向割。由于 $\delta^-(S) = \emptyset$, 故 $\delta^+(S)$ 中任意两条边不能在同一条有向路中, 所以 $\delta^+(S)$ 是 D 的一个反链。另一方面, 设 A' 是 D 的一个最大反链, 我们构造边集合 $A'' = \{a \in A - A' : \exists a' \in A', a \prec a'\}$ 且将 A'' 中所有边的顶点记作 V'' 。根据 A'' 的构造可得 $\delta^-(V'') = \emptyset$, 从而 $\delta^+(V'')$ 是一个包含 A' 的有向割。综上, 有向无圈图的最大有向割中边的数目等于其最大反链中边的数目。

命题 5 设 $D = (V \cup \{s, t\}, A)$ 是一个单平面有向无圈图, 如果 D' 是由 D 经过 $Recover(D, v)$ 阶段得到, 那么 D' 中边的可比关系与 D 中相同。

证明在算法的 $Recover(D, v)$ 阶段, 首先判断该顶点是否为源点 s 。如果是 s 则算法停止, 否则对该点的出度 $|\delta^+(v)|$ 进行判断。一方面, 如果 $|\delta^+(v)| > 0$ 则 B_i 是连接两条可比界边的有向路的边集, 并且这条有向路中非端点的度都为 2, 因此删除 B_i 不改变其他边的可比关系。另一方面, 如果 $|\delta^+(v)| = 0$ 则分情

况考虑 v 入度的情况。如果 $|\delta^-(v)| \geq 2$ ，则 v 变为一个断点(入度非零，出度为零)。对于这种情况，我们需要将 B_i 中的边恢复到图中去，才能保证图中边的可比关系不变。如果 $|\delta^-(v)| = 1$ ，则在图中删除相应的界边 (u, v) ，使 v 变为一个孤立点。因为不确定删除 (u, v) 是否会影响其余边的可比关系，所以将其暂存到 B_i 中，然后继续执行 $Recover(D, u)$ 。综上，命题得证。

命题 6 设 $D = (V \cup \{s, t\}, A)$ 是一个单平面有向无圈图，如果 D' 是由 D 经过 $Delete(D, v)$ 阶段得到，那么 D' 中边的可比关系与 D 中相同。

证明在算法的 $Delete(D, v)$ 阶段，首先判断是否存在未处理的的界边 (u, v) 。如果存在则对 v 的入度进行讨论，否则本阶段结束。对于 $|\delta^-(v)| = 0$ 的情况：如果 $|\delta^+(v)| > 0$ ，则恢复边 (u, v) ，从而其他边的可变关系不变；如果 $|\delta^+(v)| = 0$ ，则 v 变为孤立点，然后暂存边 (u, v) 到 B_i 并执行 $Recover(D, u)$ ，由命题 5 知，其余边的可变关系不变。对于 $|\delta^-(v)| > 0$ 的情况，如果 $|\delta^+(u)| > 0$ 则执行 $Delete(D, u)$ ，否则对 u 的入度进行讨论。如果 $|\delta^-(u)| = 0$ ，则 u 为源点，算法停止。如果 $|\delta^-(u)| = 1$ ，则找一条界边 (w, u) 并执行 $A - \{(w, u)\}$ 使得 u 为孤立点，然后执行 $Recover(D, u)$ ，由命题 5 知，其余边的可变关系不变。如果 $|\delta^-(u)| > 1$ ，则 u 为一个断点，故需要恢复 (u, v) ，然后执行 $Delete(D, u)$ 。

下面为不恢复某些边却能保证其余边的可比关系不变的原因。如果 (u, v) 是 $D = (V \cup \{s, t\}, A)$ 中的一条界边且满足 $|\delta^+(u)| \geq 2$ 且 $|\delta^-(v)| \geq 2$ ，那么 $D' = (V \cup \{s, t\}, A - \{(u, v)\})$ 满足 $|\delta^+(u)| \geq 1$ 且 $|\delta^-(v)| \geq 1$ 。由于 $D' = (V \cup \{s, t\}, A - \{(u, v)\})$ 仍是单平面有向无圈图，故存在 $s-v$ 有向路和 $u-t$ 有向路。因为 D' 是一个平面图其边只在顶点处相交，所以上述两条路必在某顶点处相交，即 D' 中存在连接 uv 的有向路，从而以 u 为头部的边和以 v 为尾部的边间的可比关系不变。

定理 1 的证明。根据算法 1，记 $D = D_0, D_1, \dots, D_i$ 为每次 while 产生的单平面有向无圈图，记 p_j 为在 D_{j-1} 中 Visit 阶段找到的界路。要证明 $i = |\delta^+(D)|$ 成立，只需证明 $|\delta^+(D_j)| = |\delta^+(D_{j+1})| + 1$ 对任意的 $0 \leq j \leq i-1$ 成立。

首先证明 $|\delta^+(D_{j+1})| < |\delta^+(D_j)|$ 。假设 $|\delta^+(D_{j+1})| \geq |\delta^+(D_j)|$ 成立，则 D_{j+1} 中存在 $|\delta^+(D_{j+1})|$ 条互不可比的边，不妨记为 $\delta^+(D_{j+1})$ 。由命题 6 知 $\delta^+(D_{j+1})$ 在 D_j 中是互不可比的。根据命题 4 得到 D_j 的最大有向割中边的数目不小 $|\delta^+(D_{j+1})|$ ，从而 $|\delta^+(D_j)| \geq |\delta^+(D_{j+1})| > |\delta^+(D_j)|$ 成立，矛盾。然后证明 $|\delta^+(D_{j+1})| \geq |\delta^+(D_j)| - 1$ 。根据命题 3，如果 $\delta^+(S)$ 是 D_j 中任意最大有向割，则 $|A(p_{j+1}) \cap \delta^+(S)| = 1$ 。由命题 6 知 $\delta^+(S) - A(p_{j+1})$ 是 D_{j+1} 中的反链，故 $|\delta^+(D_{j+1})| \geq |\delta^+(D_j)| - 1$ 成立。综上，当 $0 \leq j \leq i-1$ 时， $|\delta^+(D_j)| = |\delta^+(D_{j+1})| + 1$ 成立。

由命题 2 知 $pc(D) \geq |\delta^+(D)|$ ，所以算法 1 输出的有向路集合 $\{p_1, \dots, p_i\}$ 是最优解，这里 $i = |\delta^+(D)|$ 。

算法的时间复杂度是 $O(n|\delta^+(D)|)$ 。步骤 1 中 for 循环是对图中每个顶点做一次操作，总共是 n 次。while 阶段每次先找一条边界路，然后沿着这条路依次对边做操作，路上每条边至多被删、暂存，恢复各 1 次，所以每阶段操作 $O(n)$ 次。由上面分析知，while 阶段会找到 $|\delta^+(D)|$ 条路，所以 while 循环中所要操作的次数为 $O(n|\delta^+(D)|)$ 。综上，该算法的时间复杂度为 $O(n|\delta^+(D)|)$ 。

为让读者更好的理解算法，我们在图 2 展现出算法在图 1 中的运行过程。首先，判断图中存在未被覆盖的边，则找到一条界路 Path1，图中用红色虚线表示 Path1，删除路中的边后，其他边的可比关系不变，无需恢复边。其次，判断图中存在未被覆盖的边，则找到一条界路 Path2，但删除路中的边后边间的可比关系发生改变，故需恢复第二条边来保持边的可比关系。再次，判断图中存在未被覆盖的边，则找到一条界路 Path3，发现删除路中头部两条边后可能影响边的可比关系，故对此段删除的边暂存，经过我们判断暂存边集不影响剩余边的可比关系，故将暂存边集删除。最后，图只剩一条路，此为界路 Path4，算法结束。为了更加直观展现图 2 中所找的路，我们将路集合展示在图 3 中，该集合也是算法在图 1 中的运行结果。

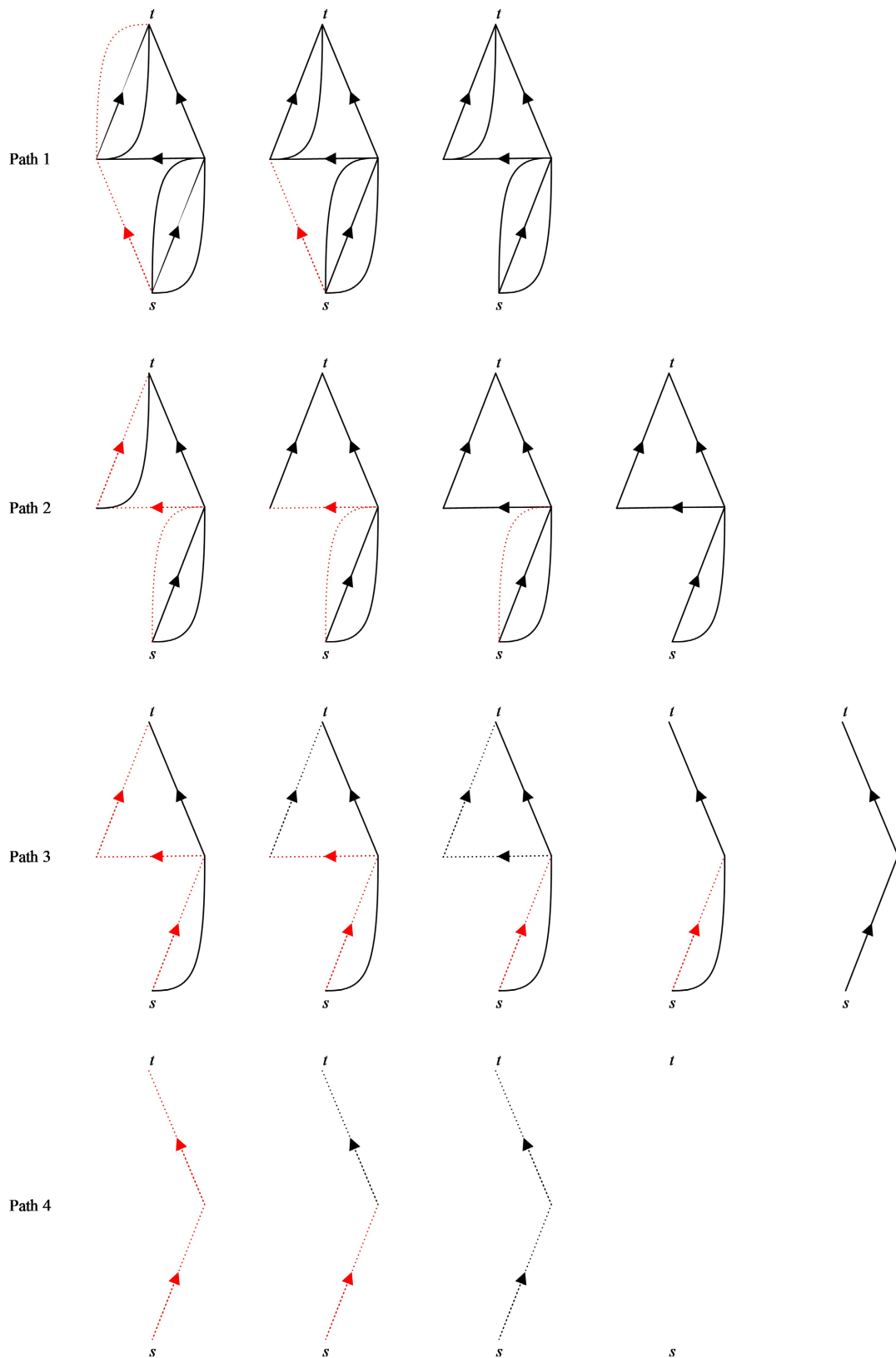


Figure 2. The running process of the algorithm in figure 1
图 2. 算法在图 1 中的运行过程

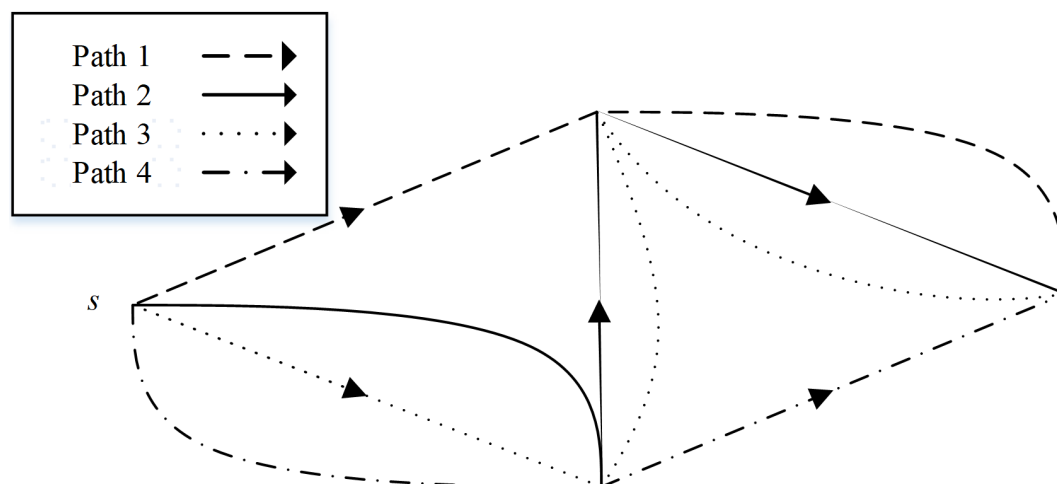


Figure 3. The running result of the algorithm in figure 1
图 3. 算法在图 1 中的运行结果

4. 结论

本文主要针对单平面有向无圈图上的最小路覆盖问题设计了一个时间复杂度为 $O(nk)$ 精确算法，这里 n 代表图的顶点数、 k 代表图中最大有向割的边数。由于所考虑的情况是图可以带重边，所以之前最好算法的时间复杂度是 $O(nm)$ ，这里 m 代表图中边的数目。对于带重边的有向无圈图，其最大有向割中边的数量级要远远小于整个图中边的数量级。该方法是否能够推广到一般有向无圈图的情况仍有待进一步研究。

基金项目

山西省基础研究项目(20210302123097)和山西省自然科学基金(202103021224058)。

参考文献

- [1] Caprara, A., Malaguti, E. and Toth, P. (2011) A Freight Service Design Problem for a Railway Corridor. *Transportation Science*, **45**, 147-162. <https://doi.org/10.1287/trsc.1100.0348>
- [2] Bondy, J.A. and Murty, U.S.R. (1976) Graph Theory with Applications. The Macmillan Press Ltd, Great Britain.
- [3] Ntafos, S.C. and Hakimi, S.L. (1979) On Path Cover Problems in Digraphs and Applications to Program Testing. *IEEE Transactions on Software Engineering*, **SE-5**, 520-529. <https://doi.org/10.1109/TSE.1979.234213>
- [4] Hopcroft, J.E. and Karp, R.M. (1973) An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartitegraphs. *SIAM Journal on Computing*, **2**, 225-231. <https://doi.org/10.1137/0202019>
- [5] Even, S. and Tarjan, R.E. (1975) Network Flow and Testing Graph Connectivity. *SIAM Journal on Computing*, **4**, 507-518. <https://doi.org/10.1137/0204043>
- [6] Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993) Network Flows. Prentice Hall, Hoboken.
- [7] Goldberg, A.V. and Tarjan, R.E. (1988) A New Approach to the Maximum-Flow Problem. *Journal of the ACM*, **35**, 921-940. <https://doi.org/10.1145/48014.61051>
- [8] Orlin, J.B. (2013) Max Flows in $o(nm)$ Time, or Better. *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, 765-774. <https://doi.org/10.1145/2488608.2488705>