Advances in Applied Mathematics 应用数学进展, 2025, 14(2), 81-92
Published Online February 2025 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2025.142054

Hans汉斯

完全二部图的线图中的完全独立生成树

赖锦城，何伟骅*

广东工业大学数学与统计学院，广东 广州

收稿日期：2025年1月15日；录用日期：2025年2月9日；发布日期：2025年2月17日

摘　要

完全独立生成树(CISTs)在计算机网络或通信网络的设计中提供了一个重要的架构选择。对于给定图的多个CISTs的构造，已证明其解决方案的实用性和在实际应用中的优化潜力。文章提出了一种高效的算法，用于在完全二部图的线图中构造CISTs，该算法建立了完全二部图的边划分与其线图的顶点划分之间的关联。此外，还进行了实验测试该算法并验证其正确性。

关键词

完全独立生成树，完全二部图，线图，划分

Completely Independent Spanning Trees in the Line Graphs of Complete Bipartite Graphs

Jincheng Lai, Weihua He*

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: Jan. 15th, 2025; accepted: Feb. 9th, 2025; published: Feb. 17th, 2025

Abstract

Completely Independent Spanning Trees (CISTs) provide an important architectural choice in the design of computer networks or communication networks. The construction of multiple CISTs for a given graph has demonstrated the practicality of its solutions and optimization potential in real-world applications. This paper proposes an efficient algorithm for constructing CISTs in the line graphs of complete bipartite graphs, which establishes the correlation between the edge partition of the complete bipartite graph and the vertex partition of its line graph. In addition, experiments

*通讯作者。

文章引用: 赖锦城, 何伟骅. 完全二部图的线图中的完全独立生成树[J]. 应用数学进展, 2025, 14(2): 81-92.
DOI: 10.12677/aam.2025.142054

were conducted to test the performance of the algorithm and to verify its correctness.

Keywords

Completely Independent Spanning Trees, Complete Bipartite Graph, Line Graph, Partition

1. 引言

一个网络可以抽象为图$G\left(V\left(G\right),E\left(G\right)\right)$，其中$V\left(G\right)$表示顶点集，$E\left(G\right)$表示边集，分别代表服务器和它们之间的连接。本文中考虑的图是无向图且是简单图。

对于图 G 和树 T，如果$V\left(T\right)=V\left(G\right)$，$E\left(T\right)\subset E\left(G\right)$，且$\left|E\left(T\right)\right|=\left|V\left(T\right)\right|-1$，则称 T 为 G 的一个生成树。在图 G 中，对于顶点 u 和 v，假设 P_1 和 P_2 是连接顶点 u 到 v 的两条路径。如果 P_1 和 P_2 不共享任何边且不共享任何顶点(除了 u 和 v)，则称 P_1 和 P_2 为内部节点不相交的路径。假设在 G 中有一组生成树$\left\{T_1,T_2,\cdots,T_k\right\}$，对于图 G 中任意两个不同顶点 u 和 v，从 u 到 v 的路径都是内部节点不相交的路径，则这一组生成树被称为完全独立生成树(简称 CISTs)，即在这组树中，任意两棵树的任意一对路径都是内部节点不相交的。

完全独立生成树在设计如计算机网络或通信网络时提供了一个重要的架构选择，用于增强网络的容错性[1]-[3]。例如，如果某一部分的网络出现故障，完全独立生成树确保网络的其它部分仍然能够维持基本的通信功能，从而减少了故障扩散的可能性。此外，完全独立生成树还被运用于 IP 网络中配置保护路由[4]-[8]。当网络密度足够高时，Moinet [9]等人表示可以找到更多的 CIST。因此，在给定的图中构建若干 CISTs，在实际的多个领域中提供了实用的解决方案和优化策略。

在给定图中构造两个 CISTs 已被证明是 NP 难问题[2]。此外，在文献[2]中，Hasunuma 猜测在一个 2n 个顶点连通图 G 中存在 n 个 CISTs。然而，Pétefalvi [10]提出了一个反例来反驳这一猜测。Pai 等人[11]也提出了反例。因此，研究的重点转向了尽可能多地构造 CISTs [12]。Pai 等人[13]证明，在 $K_{m,n}$ 中，当 $m\geq n\geq 4$ 时，存在 $n/2$ 个 CISTs。

我们将$V\left(G\right)$划分为 k 个非空子集$\left(V_1,V_2,\cdots,V_k\right)$，如果满足以下两个条件，则称其为 CIST-划分[14]：

1) 每个子集V_i形成的诱导子图是连通的；

2) 对于任意两个不同的子集V_i和V_j，我们定义一个二部图$B\left(V_i,V_j\right)$，其边集包括所有连接V_i中顶点与V_j中顶点的边。这个二部图$B\left(V_i,V_j\right)$必须满足：每一个连通组件 H，有$\left|E\left(H\right)\right|\geq\left|V\left(H\right)\right|$，这意味着$B\left(V_i,V_j\right)$中不存在仅由树构成的连通组件。

Araki 证明了 k 个 CISTs 的存在实际上等价于 CIST 划分$\left(V_1,V_2,\cdots,V_k\right)$的存在[14]。

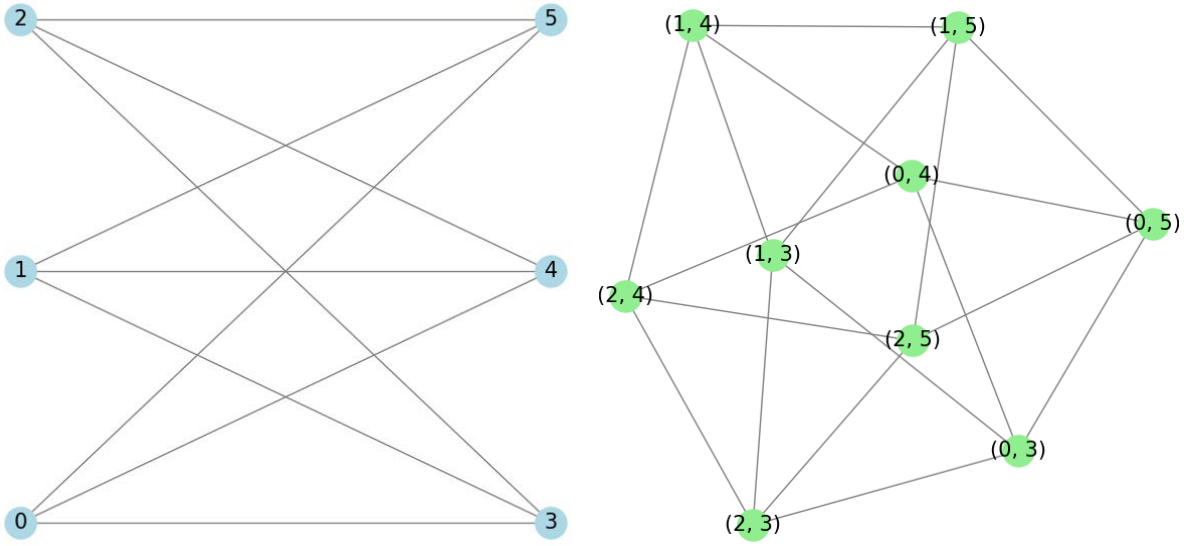
定理1 [14]一个连通图G有k个完全独立生成树，当且仅当存在一个$V\left(G\right)$的CIST划分$\left(V_1,V_2,\cdots,V_k\right)$。

给定一个图 G，令$L\left(G\right)$表示 G 的线图，其构造过程如下(图 1)：

1) 对于图 G 中的每条边$\left(x,y\right)$，在$V\left(L\left(G\right)\right)$中添加一个顶点$x:y$；

2) 如果 G 中有两条边$\left(x,y\right)$和$\left(y,z\right)$相邻，则在$L\left(G\right)$中，顶点$x:y$和$y:z$通过一条边连接。

近年来，线图在理论研究和实际应用中都被证明是一个良好的网络模型。关于线图的研究涉及图论中的各种基础问题，如边不相交的哈密顿回路[15]、可追溯性[16]、生成树的数量[17]等。此外，线图在



(a) (b)

Figure 1. A graph G and its line graph L(G)
图 1. 图 G 及其线图 L(G)

现代网络技术中扮演着重要角色，尤其是在数据中心网络设计中的应用。通过在原始网络的边上部署服务器，例如 SWCube [18]和 BCDC [19]，线图模型为网络设计提供了一个有效的结构框架，从而支持更高效的数据处理和信息流动。对于线图中完全独立生成树(CISTs)的研究也日益受到关注，例如完全图的线图[20]和环面网络的线图[21]。

在本文中，我们研究了完全二部图的线图中的完全独立生成树(CISTs)，并提出了一种在完全二部图的线图中构造多个 CISTs 的算法。本文的其余部分组织如下：第二节，我们介绍了在完全二部图的线图中构造 CIST 划分的算法，并展示了算法的正确性；第三节，我们通过实验应用该算法；第四节，我们给出结论。

2. 完全二部图的线图中的完全独立生成树

根据定理 1，在一个连通图 G 中构造 CISTs 时，我们只需考虑构造 G 的 CIST 划分。由于线图的顶点是由原图的边构成的，因此将线图的顶点划分成 CIST 划分，实质上就是将原图的边划分。

我们提出了一种算法，称为 CIST-P，用于构造 $L\left(K_{m,n}\right)\left(m \geq n \geq 3\right)$ 的 CIST 划分，并证明该算法是正确的。我们将 $K_{m,n}$ 的两个部分标记为 V_a 和 V_b。

算法. CIST-P

输入：完全二部图 $K_{m,n}$。

输出：$L\left(K_{m,n}\right)$ 的 CIST 划分 CP。

步骤 1：将 V_a 和 V_b 中前 n 个顶点之间的边进行划分。初始化 V_a 和 V_b 的顶点为 $\{a_1, a_2, \cdots, a_m\}$ 和 $\{b_1, b_2, \cdots, b_n\}$，分别表示 V_a 和 V_b 的顶点集合。

1: $k = \lfloor n/2 \rfloor$
2: **for** $i = 1$ to k **do**
3: $EP[i] = \varnothing$ /*采用数组的形式存储第 i 组划分包含的边*/

续表

4: **end for**
5: **for** $i=1$ to k **do**
6: $EP[i]=EP[i]\cup\{(a_i,b_i)\}$
7: **for** $i=i+1$ to $i+k$ **do**
8: $EP[i]=EP[i]\cup\{(a_i,b_j)\}$
9: **end for**
10: **for** $j\in[1,i-1]\cup[i+k+1,2k]$ **do**
11: $EP[i]=EP[i]\cup\{(a_{i+k},b_j)\}\cup\{(a_j,b_{i+k})\}$
12: **end for**
13: **end for**

步骤 2：当 n 为奇数时进行调整。

14: **if** n is odd **then**
15: $EP[k+1]=\varnothing$
16: **for** $i=1$ to k **do**
17: $EP[i]=EP[i]-\{(a_{k+i},b_{k+i})\}$
18: $EP[k+1]=EP[k+1]\cup\{(a_{k+i},b_{k+i})\}$
19: $EP[i]=EP[i]\cup\{(a_{k+i},b_{k+1})\}$
20: **end for**
21: **for** $i=1$ to $2k$ **do**
22: $EP[k+1]=EP[k+1]\cup\{(a_{k+1},b_i)\}$
23: **if** $i\in[1,k]$ **then**
24: $EP[k+1]=EP[k+1]\cup\{(a_i,b_{k+1})\}$
25: **end if**
26: **end for**
27: **end if**

步骤 3：构造 V_a 中第 $n+1$ 个顶点到第 m 个顶点与 V_b 的顶点之间的边划分。

28: **for** $i=n+1$ to m **do**
29: **for** $j=1$ to $len(EP)$ **do**
30: /* $len(EP)$ 是 EP 中所有元素的数量，即划分的数量*/
31: $EP[j]=EP[j]\cup\{(a_i,b_j)\}$
32: **if** n is odd **then**
33: **if** j is not equal to $len(EP)$ **then**
34: $EP[j]=EP[j]\cup\{(a_i,b_{j+k+1})\}$
35: **end if**
36: **else**
37: $EP[j]=EP[j]\cup\{(a_i,b_{j+k})\}$
38: **end if**
39: **end for**
40: **end for**

步骤 4：在 $L(K_{m,n})$ 中构造 CIST 划分

41: $CP=\varnothing$
42: **for** each set E in EP **do**
43: $C=\varnothing$
44: **for** each edge (x,y) in E **do**
45: $C=C\cup\{x:y\}$

续表

```
46:     end for
47:     CP = CP ∪ {C}
48:  end for
     END
```

现在我们证明算法 CIST-P 输出的 CP 是 $L\left(K_{m,n}\right)$ 的 CIST 划分。

定理 2 假设 EP 是在算法 CIST-P 的第 3 步中得到的集合，CP 是在算法 CIST-P 的第 4 步中得到的集合。那么，EP 是 $E\left(K_{m,n}\right)$ 的一个划分，而 CP 是 $V\left(L\left(K_{m,n}\right)\right)$ 的一个划分。此外，CP 是 $L\left(K_{m,n}\right)$ 的一个 CIST 划分。

证明：设 $t=\left\lfloor\frac{n+1}{2}\right\rfloor$。

当 n 为偶数时，$t=\frac{n}{2}$。显然，对于 $1\le i\le t$，有 $\left|EP[i]\right|=2(t+1)+2(n-t-1)+2(m-n)=2m$，因此 $\sum_{i=1}^{t}\left|EP[i]\right|=2m\times t=mn$。根据完全二部图的定义，$K_{m,n}$ 有 mn 条边。因此，$\sum_{i=1}^{t}\left|EP[i]\right|=\left|E\left(K_{m,n}\right)\right|=mn$。对于任意 $j\left(1\le j\le t\right)$，以及 $1\le i<j\le t$，有 $EP[i]\cap EP[j]=\varnothing$。因此，$EP$ 是 $E\left(K_{m,n}\right)$ 的一个划分。

当 n 为奇数时，$t=\frac{n+1}{2}$。显然，对于 $1\le i\le t-1$，有 $\left|EP[i]\right|=2t-1+2(n-t-1)+1+2(m-n)=2m-2$。对于 $i=t$，有 $\left|EP[i]\right|=2n-1+(m-n)=m+n-1$。因此，$\sum_{i=1}^{t}\left|EP[i]\right|=\sum_{i=1}^{t-1}\left|EP[i]\right|+\left|EP[t]\right|=(2m-2)\times(t-1)+m+n-1=mn$。因此，$\sum_{i=1}^{t}\left|EP[i]\right|=\left|E\left(K_{m,n}\right)\right|=mn$。对于任意 $j\left(1\le j\le t\right)$，以及 $1\le i<j\le t$，有 $EP[i]\cap EP[j]=\varnothing$。因此，$EP$ 是 $E\left(K_{m,n}\right)$ 的一个划分。

根据线图的定义，因为 EP 是 $E\left(K_{m,n}\right)$ 的一个划分，所以 CP 是 $V\left(L\left(K_{m,n}\right)\right)$ 的一个划分。

设 $t=\left\lfloor\frac{n+1}{2}\right\rfloor$，$V_i=CP[i]$。

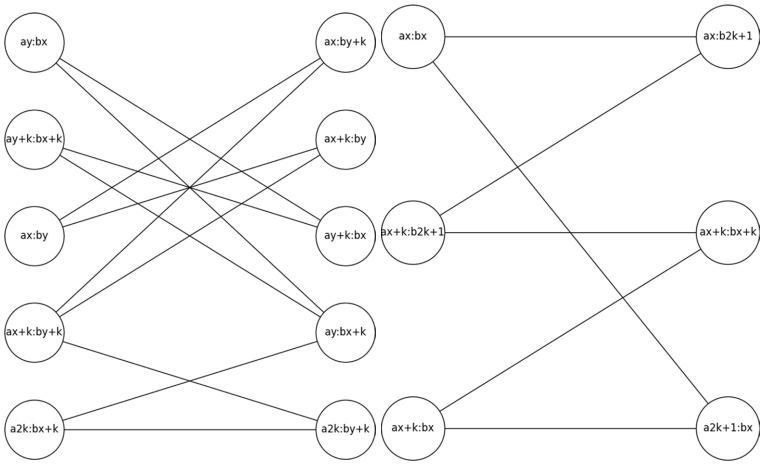
情况 1. 当 n 是奇数时，根据算法 CIST-P 可得：

$$V_i=\begin{cases}\{a_1:b_{i+k},\cdots,a_{i-1}:b_{i+k},a_i:b_i,\cdots,a_i:b_{i+k},\\ a_{i+1}:b_i,\cdots,a_{i+k}:b_i,a_{i+k}:b_1,\cdots,a_{i+k}:b_{i-1},\\ a_{i+k}:b_{i+k+1},\cdots,a_{i+k}:b_{2k+1},a_{i+k+1}:b_{i+k},\cdots,a_{2k}:b_{i+k},\\ a_{2k+2}:b_i,a_{2k+2}:b_{i+k+1},\cdots,a_m:b_i,a_m:b_{i+k+1}\} & \text{for } 1\le i\le k\\ \{a_i:b_{2k+1},\cdots,a_k:b_{2k+1},a_{k+1}:b_{k+1},\\ a_{k+2}:b_{k+2},\cdots,a_{2k}:b_{2k},a_{2k+1}:b_1,\cdots,a_{2k+1}:b_{2k+1},\\ a_{2k+2}:b_k,a_{2k+3}:b_k,\cdots,a_m:b_k\} & \text{for } i=k+1\end{cases}$$

首先，我们证明顶点集 V_i 的诱导子图是连通的。对于 $1\le i\le k$，在 $EP[i]$ 中的边，如 $\left\{\left(a_{2k+2},b_i\right),\left(a_{2k+2},b_{i+k+1}\right),\cdots,\left(a_m,b_i\right),\left(a_m,b_{i+k+1}\right)\right\}$，这些边是连通的，并且与边 $\left(a_i,b_i\right)$ 一起形成一个连通的子图。$EP[i]$ 中的其余边都连接到边 $\left(a_i,b_i\right)$、$\left(a_i,b_{i+k}\right)$ 或 $\left(a_{i+k},b_i\right)$ 之一。由于这三条边是连通的，根据线图的定义，顶点集 $V_i\left(1\le i\le k\right)$ 的诱导子图是连通的。

对于 $i=k+1$，在 $EP[i]$ 中的边，如 $\left(a_{k+i},b_{k+i}\right)\left(1\le i\le k\right)$ 与边 $\left(a_{2k+1},b_{k+i}\right)$ 连接。$EP[i]$ 中的边 $\left(a_{2k+2},b_k\right),\left(a_{2k+3},b_k\right),\cdots,\left(a_m,b_k\right)$ 与边 $\left(a_k,b_k\right)$ 连接。除了边 $\left(a_{k+i},b_{k+i}\right),\left(a_{2k+2},b_k\right),\left(a_{2k+3},b_k\right),\cdots,\left(a_m,b_k\right)$ 之外，所有其他边都与边 $\left(a_{2k+1},b_{2k+1}\right)$ 连接。

因此，顶点集 V_i 的诱导子图是连通的。



(a) $1 \le x, y \le k$ (b) $1 \le x \le k, y = k$

Figure 2. Connected subgraphs of $B\left(V_x, V_y\right)$

图 2. $B\left(V_x, V_y\right)$ 的连通子图

我们用 $B\left(V_x, V_y\right)$ 来表示由 V_x 和 V_y 形成的二部图。接下来，我们证明对于每个二部图 $B\left(V_x, V_y\right)$ (其中 $1 \le x, y \le k+1$，且 $x \ne y$)都没有树形组件。我们分以下情况进行讨论：

情况 1.1 $1 \le x, y \le k$，不失一般性，假设 $x < y$。在 V_x 中的所有顶点都与 V_y 中一个顶点子集 $V_{sy}: \left\{a_x : b_{y+k}, a_{x+k} : b_y, a_{y+k} : b_x, a_y : b_{x+k}, a_y : b_{x+k+1}\right\}$ 相邻。在 V_y 中的所有顶点都与 V_x 中一个顶点子集 $V_{sx}: \left\{a_y : b_x, a_{y+k} : b_{x+k}, a_x : b_y, a_{x+k} : b_{y+k}, a_{x+k} : b_{y+k+1}\right\}$ 相邻。在 V_{sx} 中，顶点 $a_y : b_x, a_{y+k} : b_{x+k}$ 与 V_{sy} 中的顶点 $a_{y+k} : b_x, a_y : b_{x+k}$ 之间存在一个环，形成环 $\left\{a_y : b_x, a_{y+k} : b_x, a_{y+k} : b_{x+k}, a_y : b_{x+k}, a_y : b_x\right\}$。在 V_{sx} 中，顶点 $a_x : b_y, a_{x+k} : b_{y+k}$ 与 V_{sy} 中的顶点 $a_x : b_{y+k}, a_{x+k} : b_y$ 之间也存在一个环，形成 $\left\{a_x : b_y, a_x : b_{y+k}, a_{x+k} : b_{y+k}, a_{x+k} : b_y, a_x : b_y\right\}$。此外，$V_x$ 中的顶点 $a_{x+k} : b_{y+k}$，$a_{x+k} : b_{y+k+1}$，$a_{2k} : b_{x+k}$ 与 V_y 中的顶点 $a_y : b_{x+k}$，$a_y : b_{x+k+1}$，$a_{2k} : b_{y+k}$ 相连接，换句话说，存在一条路径 $\left\{a_{x+k} : b_{y+k+1}, a_{x+k} : b_{y+k}, a_{2k} : b_{y+k}, a_{2k} : b_{x+k}, a_y : b_{x+k}, a_y : b_{x+k+1}\right\}$。因此，二部图 $B\left(V_x, V_y\right)$ 是连通的，连通分支就是其本身，并且没有树状分支。图 2(a)显示了在这种情况下 $B\left(V_x, V_y\right)$ 的局部连通图。

情况 1.2 $1 \le x \le k$，$y = k+1$。在 V_x 中的所有顶点都与 V_y 中一个顶点子集 $V_{sy}: \left\{a_x : b_{2k+1}, a_{x+k} : b_{x+k}, a_{2k+1} : b_x, a_{2k+1} : b_{x+k+1}\right\}$ 相邻。设 V_{x_1} 为 V_x 中一个顶点子集，$V_{x_1} = \left\{a_x : b_x, \cdots, a_x : b_{x+k}\right\}$，并且设 V_{x_2} 为另一个子集，$V_{x_2} = \left\{a_{x+k} : b_1, \cdots, a_{x+k} : b_{x-1}, a_{x+k} : b_{x+k+1}, \cdots, a_{x+k} : b_{2k+1}\right\}$。则 V_y 中的任意顶点总能在 V_{x_1} 或 V_{x_2} 找到一个与之相邻的顶点，换句话说，V_y 中的所有顶点都与 $V_{x_1} \cup V_{x_2}$ 中的某个顶点相邻。

由于 V_{x_1} 中的所有顶点都与 V_{sy} 中的顶点 $a_x : b_{2k+1}$ 相邻，并且 V_{x_2} 中的所有顶点都与 V_{sy} 中的顶点 $a_{x+k} : b_{x+k}$ 相邻。取 V_{x_1} 中的顶点 $a_x : b_x$ 和 V_{x_2} 中的顶点 $a_{x+k} : b_{2k+1}$，设 $V_{sx} = \left\{a_x : b_x, a_{x+k} : b_{2k+1}\right\}$，因此 V_{sx} 与 V_{sy} 中的所有顶点相连，意味着存在一条路径 $\left\{a_{2k+1} : b_{x+k+1}, a_{2k+1} : b_x, a_x : b_x, a_x : b_{2k+1}, a_{x+k} : b_{2k+1}, a_{x+k} : b_{x+k}\right\}$。因此，二部图 $B\left(V_x, V_y\right)$ 是连通的，其连通分支就是其本身。此外，由于 $B\left(V_x, V_y\right)$ 中存在一个环 $\left\{a_{2k+1} : b_x, a_x : b_x, a_x : b_{2k+1}, a_{x+k} : b_{2k+1}, a_{x+k} : b_{x+k}, a_{x+k} : b_x, a_{2k+1} : b_x\right\}$，因此 $B\left(V_x, V_y\right)$ 中没有树状分支。图 2(b)显示了在这种情况下 $B\left(V_x, V_y\right)$ 的局部连通图。

情况 2. 当 n 为偶数时，通过算法 CIST-P 可得

$$V_i=\{a_1:b_{i+k},\cdots,a_{i-1}:b_{i+k},a_i:b_i,\cdots,a_i:b_{i+k},a_{i+1}:b_i,\cdots,a_{i+k}:b_i,a_{i+k}:b_1,\cdots,a_{i+k}:b_{i-1},a_{i+k}:b_{i+k},a_{i+k}:b_{i+k+1},$$
$$\cdots,a_{i+k}:b_{2k+1},a_{i+k+1}:b_{i+k},a_{i+k+2}:b_{i+k},\cdots,a_{2k}:b_{i+k},a_{2k+1}:b_i,a_{2k+1}:b_{i+k},\cdots,a_m:b_i,a_m:b_{i+k}\}$$

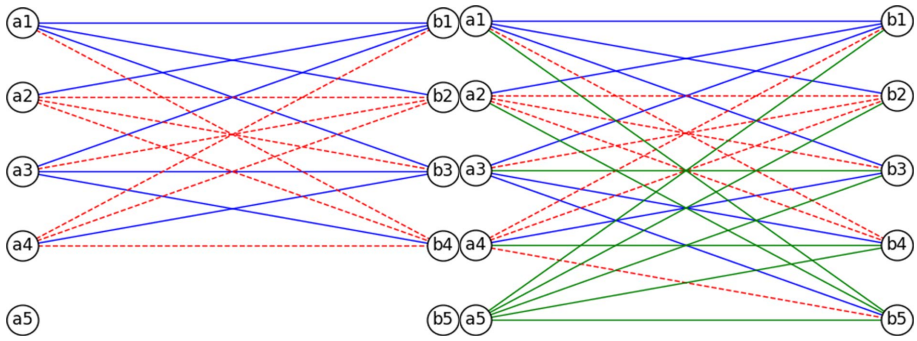
此情况与情况 1.1 相似，V_i 的诱导子图是连通的，并且对于每个二部图 $B(V_x,V_y)(1\le x,y\le k,x\ne y)$ 不存在树形组件。

总之，通过算法 CIST-P 得到的 CP 是 $L(K_{m,n})$ 的 CIST 划分。证毕。□

3. 实验

在本节中，我们通过实验验证了算法 CIST-P。我们将算法 CIST-P 应用于构建 $L(K_{5,5})$ 和 $L(K_{7,5})$ 中的 CIST。

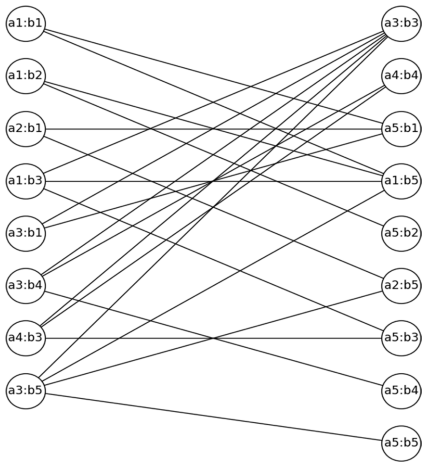
示例 1. 以 $L(K_{5,5})$ 为例。首先，我们将 $K_{5,5}$ 的顶点标记为 $V_a=\{a_1,a_2,a_3,a_4,a_5\}$ 和 $V_b=\{b_1,b_2,b_3,b_4,b_5\}$。对除去与 a_5 和 b_5 关联的边以外的边进行划分，再对与 a_5 和 b_5 关联的边进行划分和调整，如图 3 所示。图表示前两组划分，图 3(b)展示了最后一组边集的划分及划分的调整。算法第三步得到的划分集如下：$\{\{a_1:b_1,a_1:b_2,a_2:b_1,a_1:b_3,a_3:b_1,a_3:b_4,a_4:b_3,a_3:b_5\},\{a_2:b_2,a_2:b_3,a_3:b_2,a_2:b_4,a_4:b_2,a_4:b_1,a_1:b_4,a_4:b_5\},$ $\{a_3:b_3,a_4:b_4,a_5:b_1,a_1:b_5,a_5:b_2,a_2:b_5,a_5:b_3,a_5:b_4,a_5:b_5\}\}$。图 4 展示了 $L(K_{5,5})$ 的 CIST 划分，图 5 展示了 $K_{5,5}$ 对应的边划分。由此 CIST 划分，我们得到了 $L(K_{5,5})$ 的三个完全独立的生成树，如图 6 所示。



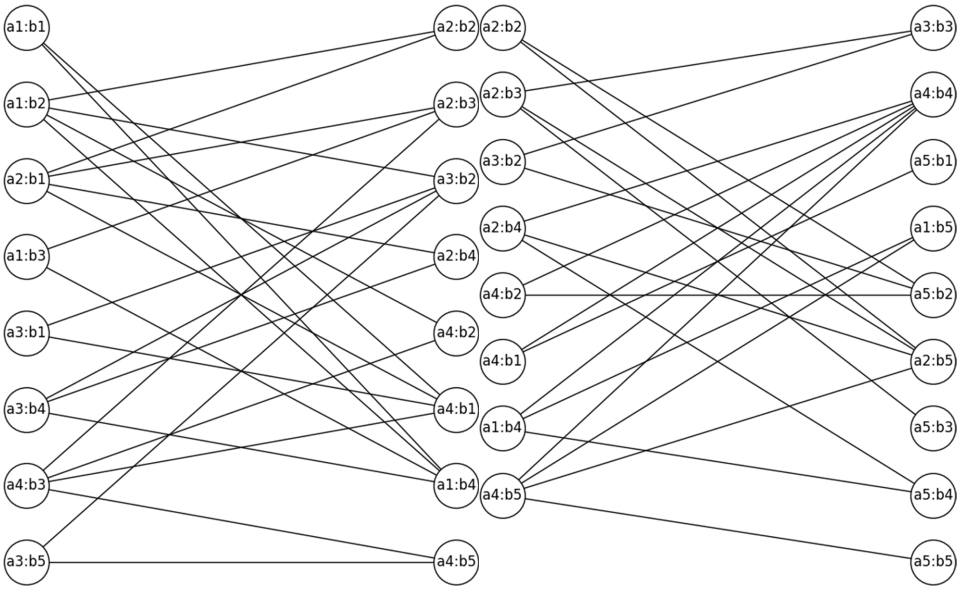
(a) (b)

Figure 3. The edge-partition of $K_{5,5}$

图 3. $K_{5,5}$ 的边划分



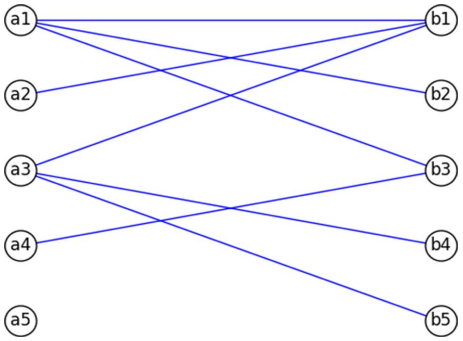
(a)



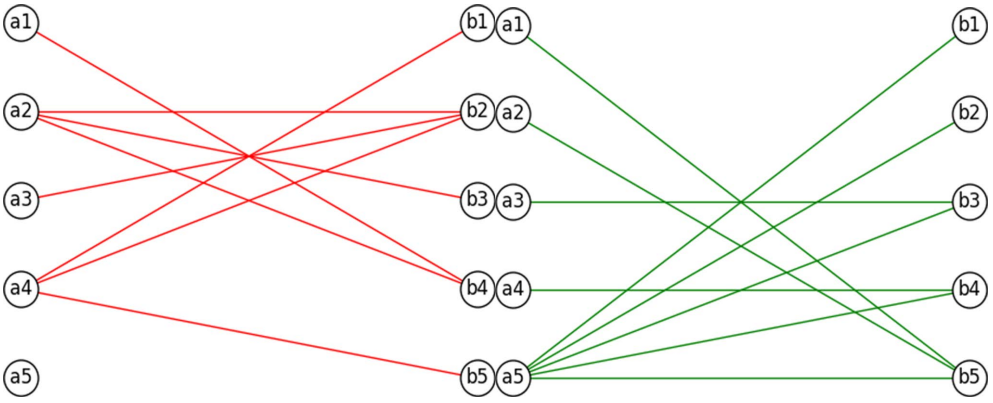
(b) (c)

Figure 4. The CIST-partitions of $L(K_{5,5})$

图 4. $L(K_{5,5})$ 的 CIST 划分



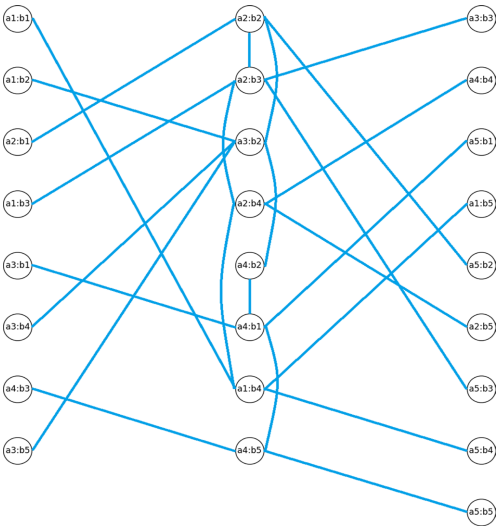
(a)



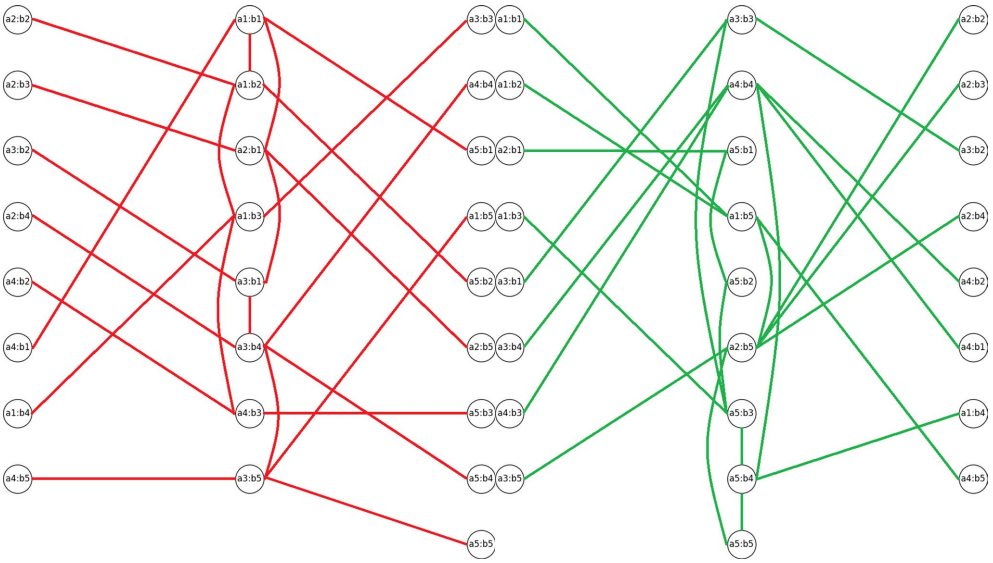
(b) (c)

Figure 5. The edge partition of $K_{5,5}$

图 5. $L(K_{5,5})$ 的边划分



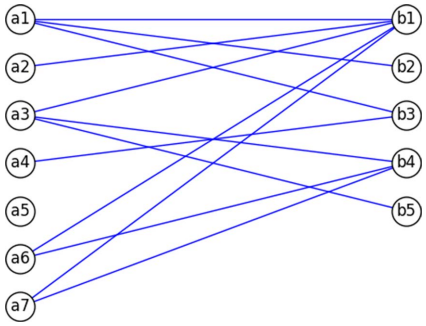
(a) T_1



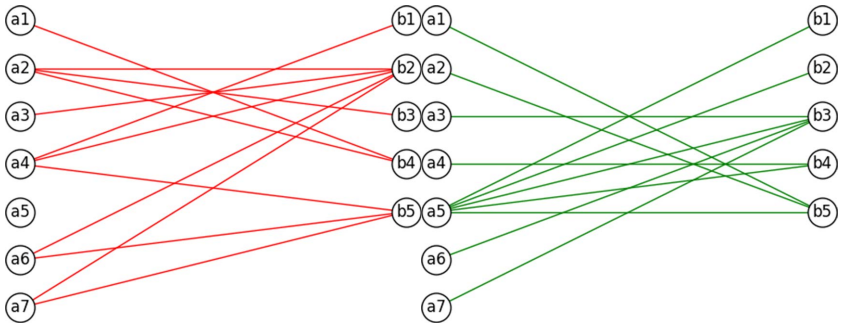
(b) T_2 (c) T_3

Figure 6. The CISTs in $L(K_{5,5})$

图 6. $L(K_{5,5})$ 的完全独立生成树



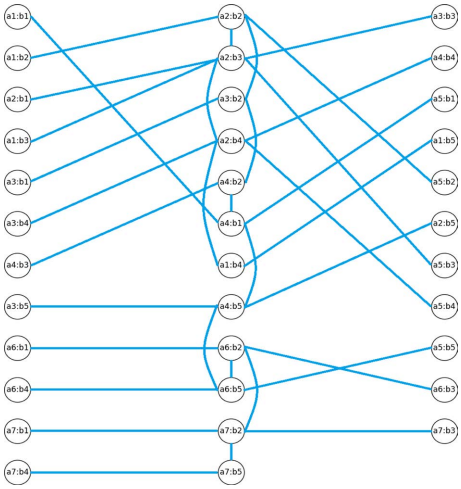
(a)



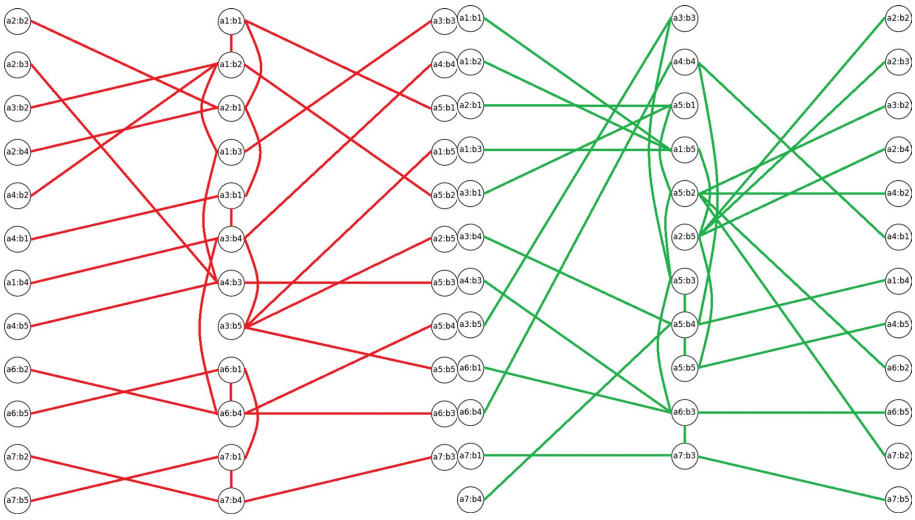
(b) (c)

Figure 7. The edge partition of $K_{7,5}$

图 7. $L\left(K_{7,5}\right)$ 的边划分



(a) T_1



(b) T_2 (c) T_3

Figure 8. The CISTs in $L\left(K_{7,5}\right)$

图 8. $L\left(K_{7,5}\right)$ 的完全独立生成树

示例 2. 以 $L\left(K_{7,5}\right)$ 为例，图 7 展示了 $K_{7,5}$ 的边划分，基于此 CIST 划分，最终得到 $L\left(K_{7,5}\right)$ 的三个完全独立的生成树，如图 8 所示。

4. 结论

在本文中，我们提出了一种算法 CIST-P，用于在 $L\left(K_{m,n}\right)\left(m \geq n \geq 3\right)$ 中构造 CISTs。我们从理论上证明了该算法的正确性，并通过实验验证了算法的有效性。算法适用于所有完全二部图的线图，但算法是否适用于其他类型的图，尚待研究。

基金项目

广东省自然科学基金面上项目(2021A1515012047)。

参考文献

[1] Hasunuma, T. (2001) Completely Independent Spanning Trees in the Underlying Graph of a Line Digraph. *Discrete Mathematics*, **234**, 149-157. https://doi.org/10.1016/s0012-365x(00)00377-0

[2] Hasunuma, T. (2002) Completely Independent Spanning Trees in Maximal Planar Graphs. In: Goos, G., Hartmanis, J., Leeuwen, J. and Kučera, L., Eds., *Graph-Theoretic Concepts in Computer Science*, Springer, 235-245. https://doi.org/10.1007/3-540-36379-3_21

[3] Lin, W., Li, X.-Y., Chang, J.-M. and Jia, X.H. (2023) Constructing Multiple CISTs on BCube-Based Data Center Networks in the Occurrence of Switch Failures. *IEEE Transactions on Computers*, **72**, 1971-1984. https://doi.org/10.1109/TC.2022.3230288

[4] Hussain, Z., Al Azemi, F. and Al Bdaiwi, B. (2024) Completely Independent Spanning Trees in Eisenstein-Jacobi Networks. *The Journal of Supercomputing*, 1-17.

[5] Li, X., Lin, W., Liu, X., Lin, C., Pai, K. and Chang, J. (2022) Completely Independent Spanning Trees on BCCC Data Center Networks with an Application to Fault-Tolerant Routing. *IEEE Transactions on Parallel and Distributed Systems*, **33**, 1939-1952. https://doi.org/10.1109/tpds.2021.3133595

[6] Pai, K., Chang, R. and Chang, J. (2020) A Protection Routing with Secure Mechanism in Möbius Cubes. *Journal of Parallel and Distributed Computing*, **140**, 1-12. https://doi.org/10.1016/j.jpdc.2020.02.007

[7] Pai, K., Chang, R., Wu, R. and Chang, J. (2020) Three Completely Independent Spanning Trees of Crossed Cubes with Application to Secure-Protection Routing. *Information Sciences*, **541**, 516-530. https://doi.org/10.1016/j.ins.2020.05.048

[8] Tapolcai, J. (2012) Sufficient Conditions for Protection Routing in IP Networks. *Optimization Letters*, **7**, 723-730. https://doi.org/10.1007/s11590-012-0455-y

[9] Moinet, A., Darties, B., Gastineau, N., Baril, J. and Togni, O. (2017) Completely Independent Spanning Trees for Enhancing the Robustness in Ad-Hoc Networks. 2017 *IEEE* 13*th International Conference on Wireless and Mobile Computing, Networking and Communications* (*WiMob*), Rome, 9-11 October 2017, 63-70. https://doi.org/10.1109/wimob.2017.8115791

[10] Péterfalvi, F. (2012) Two Counterexamples on Completely Independent Spanning Trees. *Discrete Mathematics*, **312**, 808-810. https://doi.org/10.1016/j.disc.2011.11.015

[11] Pai, K., Yang, J., Yao, S., Tang, S. and Chang, J. (2014) Completely Independent Spanning Trees on Some Interconnection Networks. *IEICE Transactions on Information and Systems*, **97**, 2514-2517. https://doi.org/10.1587/transinf.2014edl8079

[12] Cheng, B., Wang, D. and Fan, J. (2017) Constructing Completely Independent Spanning Trees in Crossed Cubes. *Discrete Applied Mathematics*, **219**, 100-109. https://doi.org/10.1016/j.dam.2016.11.019

[13] Pai, K., Tang, S., Chang, J. and Yang, J. (2013) Completely Independent Spanning Trees on Complete Graphs, Complete Bipartite Graphs and Complete Tripartite Graphs. In: Chang, R.-S., Jain, L.C. and Peng, S.-L., Eds., *Smart Innovation, Systems and Technologies*, Springer, 107-113. https://doi.org/10.1007/978-3-642-35452-6_13

[14] Araki, T. (2013) Dirac's Condition for Completely Independent Spanning Trees. *Journal of Graph Theory*, **77**, 171-179. https://doi.org/10.1002/jgt.21780

[15] Li, H., He, W., Yang, W. and Bai, Y. (2015) A Note on Edge-Disjoint Hamilton Cycles in Line Graphs. *Graphs and Combinatorics*, **32**, 741-744. https://doi.org/10.1007/s00373-015-1606-6

[16] Tian, T. and Xiong, L. (2018) Traceability on 2-Connected Line Graphs. *Applied Mathematics and Computation*, **321**,

463-471. https://doi.org/10.1016/j.amc.2017.10.043

[17] Dong, F. and Yan, W. (2016) Expression for the Number of Spanning Trees of Line Graphs of Arbitrary Connected Graphs. *Journal of Graph Theory*, **85**, 74-93. https://doi.org/10.1002/jgt.22048

[18] Li, D. and Wu, J. (2015) On Data Center Network Architectures for Interconnecting Dual-Port Servers. *IEEE Transactions on Computers*, **64**, 3210-3222. https://doi.org/10.1109/tc.2015.2389847

[19] Wang, X., Fan, J., Lin, C., Zhou, J. and Liu, Z. (2018) BCDC: A High-Performance, Server-Centric Data Center Network. *Journal of Computer Science and Technology*, **33**, 400-416. https://doi.org/10.1007/s11390-018-1826-3

[20] Wang, Y., Cheng, B., Qian, Y. and Wang, D. (2022) Constructing Completely Independent Spanning Trees in a Family of Line-Graph-Based Data Center Networks. *IEEE Transactions on Computers*, **71**, 1194-1203. https://doi.org/10.1109/tc.2021.3077587

[21] Bian, Q., Cheng, B., Fan, J. and Pan, Z. (2022) Completely Independent Spanning Trees in the Line Graphs of Torus Networks. In: Lai, Y.X., *et al*., Eds., *Algorithms and Architectures for Parallel Processing*, Springer International Publishing, 540-553. https://doi.org/10.1007/978-3-030-95391-1_34