

On the Matching Number of Directed Triangle Trees

Mengying Li, Haixing Zhao

Computer Department, QingHai Normal University, Xining Qinghai
Email: 1223788647@qq.com

Received: May 4th, 2016; accepted: May 27th, 2016; published: May 30th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

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

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



Open Access

Abstract

A matching of a directed graph G is a set of some directed edges without common starting-node and end-node. K -matching of a digraph G is the matching with the k ($k = 1, 2, \dots, n$) edges; k -matching number of a graph G is the number of distinct matchings containing k ($k = 1, 2, \dots, n$) edges. The matching of a graph G refers to the number of all k -matchings. Liu and Barabasi put forward: the number of controllable nodes in directed networks is equal to the number of nodes of directed networks minus the number of edges of the maximum matching. It illustrates that the matching number and controllability of directed networks have a close connection. Thus, the research of the number of all matchings of directed networks is of applied significance. This article mainly studies the counting problems and the extremal problems on the number of matchings in a class of directed triangle trees. We investigate the calculation method and the expression of the matching number in a class of directed triangle trees with n triangles and determine the bounds for the matching number in directed triangle trees with n triangles and the correspond graphs.

Keywords

Complex Networks, Directed Tree, Directed Triangle Trees, Matching, Hosoya Index

有向三角形树的匹配数

李梦英, 赵海兴

青海师范大学计算机学院, 青海 西宁
Email: 1223788647@qq.com

收稿日期: 2016年5月4日; 录用日期: 2016年5月27日; 发布日期: 2016年5月30日

摘要

有向图 G 的一个匹配是由其一组没有公共起点也没有公共终点的有向边构成的集合。图 G 的 k 匹配是指含 k ($k = 1, 2, \dots, n$)条有向边的匹配; 图 G 的 k -匹配数是指含 k ($k = 1, 2, \dots, n$)条有向边的匹配的选择方法数; 图 G 的匹配数指所有 k -匹配数的和。刘和Barabasi等人提出: 有向网络的可控节点数等于有向网络的顶点数减去最大匹配包含的边数。说明有向网络的可控性与有向网络的匹配数有着密切的联系。因此, 研究有向网络的所有匹配数目具有一定的应用意义。这篇文章主要研究一类有向三角形树的所有匹配数的计数问题和极值问题。给出了一类含 n 个三角形的有向三角形树匹配数的计算方法, 以及有向三角形树匹配数的上下界和相应的结构。

关键词

复杂网络, 有向树, 有向三角形树, 匹配, Hosoya指标

1. 引言

随着以互联网[1]为代表的网络信息技术的飞速发展, 人类社会已经逐步进入了复杂网络时代。钱学森给出了复杂网络[2]的一个较为严格的定义: 具有自相似、无标度、自组织、小世界、吸引子中部分或全部性质的网络为复杂网络。如何利用复杂网络让人们的生活更加有效, 成为了人们关注的焦点。基于此, 一些学者开始研究复杂网络的可控性。如果我们想要控制一个系统, 我们首先要确定一个顶点集, 通过在这些顶点集上输入不同的信号完全控制整个网络, 我们称这些节点为驱动节点。我们特别有兴趣于: 确定最小的驱动节点数[3], 记为 ND , 它可以有效地完全控制系统的动力学过程。2011年刘和Barabasi等人[3]给出有向网络匹配的定义及最小输入定理: 指出有向网络的可控节点数等于有向网络的顶点数减去最大匹配包含的边数, 由此开启了研究复杂网络可控性的新篇章。

1971年日本化学家Haruo Hosoya [4]-[6]介绍了基于结构描述的分子图, 他命名了拓扑指标并记为 Z 。他用下面的方式定义了 Z 或 $Z(G)$: 从图 G 中选择 k 条相互独立的边的方法数记为 $m(G, k)$, 对所有图 $m(G, 0) = 1$, 并且 $m(G, 1)$ 等于图 G 的边数, 则:

$$Z = Z(G) = \sum_{k \geq 0} m(G, k)$$

称 Z 或 $Z(G)$ 为Hosoya指标或Hosoya拓扑指标。

本文中我们首先给出有向树、有向三角形树及有向三角形树匹配的概念, 研究有向三角形树的匹配数及其相关性质。其次根据本文的主要内容给出计算一类有向三角形树匹配数的算法。

2. 基本知识

首先给出一些无向图、有向图及有向树的基本定义以及基本的符号表示。

2.1. 无向图的基本概念

一个图 $G = (V, E)$ 是一个三元组[7], 这个三元组包含一个顶点集 $V(G)$, 一个边集 $E(G)$ 和一个关系, 该关系使得每一条边和两个顶点(不一定是不同的点)相关联, 并将这两个顶点称为这条边的端点。设边 e

的两个端点分别为 u 和 v , 则边 e 记为 $e = (u, v) = u = (v, u) = vu$, 称节点 u 和 v 在图 G 中邻接, 记为 $u \sim v$ 或 $v \sim u$, 并称节点 u 和 v 与边 e 关联; 否则称为不邻接, 记为 $u \not\sim v$ 。图 G 中, 与节点 v 相关联的边的数目称为 v 的度, 记 $d_G(v)$ 或 $d(v)$ 。度为 0 的点称为孤立点, 度为 1 的点称为悬挂点。

图 $G = (V, E)$ 的一个匹配 M [7] 是由其一组没有公共端点的不是圈的边构成的集合; M 是由若干条边构成的集合, 故其大小就是边的条数。图 $G = (V, E)$ 的一个极大匹配是不能再通过添加边来使其变大的匹配。一个最大匹配是图 $G = (V, E)$ 的所有匹配中边数达到最大值的匹配, 即含边数最多的匹配。

图 $G = (V, E)$ 的一条路径 P_n [7] 是一个简单图, 其顶点排序使得两个顶点是邻接的当且仅当它们在顶点的序列中是前后相继的。

2.2. 有向图的基本概念

一个有向图 [7] G 是一个三元组, 其中包含一个顶点集 $V(G)$, 一个边集 $E(G)$ 和一个为每有向条边分配一个有序顶点对的函数 f , 有序对的第一个顶点是有向边的尾部, 称为有向边的起点; 第二个顶点是有向边头部, 称为有向边的终点; 它们统称为有向边的端点。设有向边 e 的尾部为 u , 头部为 v , 则有向边 e 记为 $e = (u, v) = uv \neq (v, u) = vu$, 称节点 u 和 v 在图 G 中邻接, 记为 $u \sim v$, 我们说一条有向边就是指一条从其尾部到头部的边。若令 v 是有向图 G 中的顶点, 出度是以 v 为尾部的边的条数, 记为 $d^+(v)$; 入度是以 v 为头部的边的条数, 记为 $d^-(v)$ 。

有向图 G 的一个匹配 M [7] 是由其一组没有公共起点也没有公共终点的有向边构成的集合。匹配 M 中包含的有向边的终点称为匹配节点, 否则为未匹配节点。 M [7] 是由若干条有向边构成的集合。故其大小就是有向边的条数。有向图 G 的一个极大匹配是不能再通过添加有向边来使其变大的匹配。一个最大匹配是有向图 G 的所有匹配中有向边数达到最大值的匹配, 即含有向边数最多的匹配。

有向图 G 的一条途径 [8] 是由顶点 v_i 与有向边 e_i 交错排列的序列 $W = v_1 e_1 v_2 e_2 v_3 \cdots v_{k-1} e_{k-1} v_k$, 简记为 $W = v_1 v_2 \cdots v_k$, 使得对于 $i = 1, 2, \dots, k-1$, W 中的有向边 e_i 的尾是顶点 v_i 、头是顶点 v_{i+1} 。我们也称 W 为一条从 v_1 到 v_k 的途径。若一条途径 W 中任意两条有向边都不相同, 则称 W 为迹。若一条迹中的顶点是互不相同的, 则称 W 为路径 P_n 。

刘和 Barabasi 等人 [3] 给出有向网络的匹配的定义: 有向网络的边的子集 E^* 称为是一个匹配集, 如果 E^* 中任意两条边都没有公共的始点, 也没有公共的终点。如果一个节点是 E^* 中某一条边的终点, 该节点为匹配节点, 否则为未匹配节点。

如果有向图 [9] $G = (V, E)$ 被称为有向树 $T = (V, E)$, 则满足以下三个条件

- 1) 有且仅有一个节点的入度为 0;
- 2) 除树根外的节点的入度为 1;
- 3) 从树根到任一节点有一条连通的有向路。

有向图 [9] $G = (V, E)$, 在不考虑边的方向时是一棵无向树, 则该有向图称为有向树。如图 1。

其中顶点集 $V = \{v_1, v_2, v_3, v_4\}$, 边集 $E = \{(v_1, v_3), (v_3, v_2), (v_3, v_4)\}$; 其中入度为 0 的节点称为树根, 出度为 0 的节点称为树叶; 称节点 v 的前驱元素为 v 的父节点, 称 v 的后继元素为 v 的子节点。事实上, 在一个无环有向图 $G = (V, E)$ 中也可以定义父节点和子节点, 若有向边 $(u, v) \in E$, 则称节点 u 是节点 v 的父节点, 节点 u 是节点 v 的子节点。若有向边 $(u, v) \in E$, 则称 u 是 v 的父节点, v 是 u 的子节点。

有向树中, 入度为 0 的节点称为树根 [9], 出度为 0 的节点称为树叶, 出度不为 0 的节点称为分支节点, 将不是根的分支节点称为内点。称从根节点到某个节点的路径长度为该节点的层或级。于是根节点是第 0 层节点, 其子节点是第 1 层节点, 以此类推。称与父节点在同一层的节点为堂兄弟。称节点的最大层次为树的高度或深度。

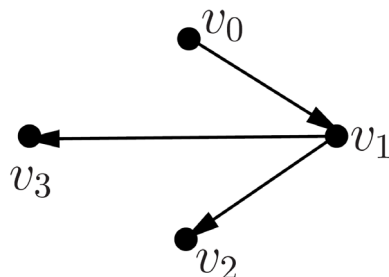


Figure 1. Directed tree $G = (V; E)$
图 1. 有向树 $G = (V; E)$

有向树 $T = (V, E)$ [9]中, $v \in V$, 由节点 v 及其所有后代导出的子图称为有向树 $T = (V, E)$ 的子根树或子树。

有向树 $T = (V, E)$ [9]中, 若 $\max_{v \in V} d^+(v) = m$ 则称 $T = (V, E)$ 是有向 m 叉树。在 m 叉树 $T = (V, E)$ 中, 若对任意节点 v 都有 $d^+(v) = m$ 或 0 , 则称 $T = (V, E)$ 是有向完全 m 叉树, 所有树叶节点所在的层都相同的有向完全 m 叉树称为有向正则 m 叉树或有向满 m 叉树。

3. 一类有向三角形树的匹配数

刘和 Barabasi 等人[3]给出了有向网络匹配的定义以及有向网络可控节点数等于顶点数减去最大匹配包含的边数的结论。自然的问题是研究有向网络的所有匹配数目。本文中主要研究有向三角形树的所有匹配数的计数问题。首先给出与有向树 F 和有向三角形树 D 相关的一些基本定义, 给出一类有向三角形树匹配数的计算方法、表达式, 确定有最大和最小匹配数的有向三角形树的结构, 其次根据本文的主要内容给出计算一类有向三角形树匹配数的算法。

3.1. 有向树的基本定义

有向树 T 被称为有向树 $F = (V, E)$, 则有向树 T 中的有向边的方向均是由上一层中的父节点指向下一层中的子节点的。如图 2。

其中顶点集: $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, 边集: $E = \{v_0v_1, v_0v_2, v_1v_3, v_1v_4, v_2v_5, v_2v_6, v_2v_7\}$; 其中入度为 0 的节点称为树根, 出度为 0 的节点称为树叶; 若有向边 $(u, v) \in E$, 则称 u 是 v 的父节点, v 是 u 的子节点。

有向星图 F^* 是恰好只有一个节点出度为 $n-1$, 其余节点均为叶子节点的 n 个节点的有向树 F 。有向路 F_p 是节点出度为 1 或 0 的有向树 F 。

3.2. 有向三角形树的基本定义

有向三角形树 $D = (V, E)$ 的过程为: $t=1$ 时, 以 v_0 为根节点, 以 v_1, v_2 为左右子节点, 生成两条有向边 v_0v_1, v_0v_2 , 且节点 v_1, v_2 之间生成一条有向边 v_1v_2 (第三边), 这时节点 $\{v_0, v_1, v_2\}$ 构成一个有向三角形, 记为 D_1 ; $t=2$ 时, D_1 恰好包含 3 个节点 $\{v_0, v_1, v_2\} \in V$, 在 D_1 的任意一个节点上粘贴 D_1 的节点 v_0 , 生成有向三角形树 D_2 ; 以此类推, 在有向三角形树 D_i 的任意一个节点上, 粘贴 D_1 的节点 v_0 , 生成有向三角形树 D_{i+1} ; 以时间步生成有向三角形树 D_n 。图 3 是其中一种生成结果。

其中顶点集: $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6\}$, 边集: $E = \{v_0v_1, v_0v_2, v_1v_2, v_1v_3, v_1v_4, v_3v_4, v_2v_5, v_2v_6, v_5v_6\}$; 其中入度为 0 的节点称为树根, 出度为 0 的节点称为树叶; 若有向边 $(u, v) \in E$, 则称 u 是 v 的父节点, v 是 u 的子节点。

有向三角形星图 D_n^* 是在每一时间步都将 D_1 的节点 v_0 粘贴到节点 v_0 而生成的。有向三角形路 $D_{pn}(D_{pn})$

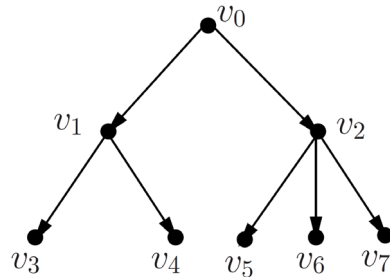


Figure 2. Directed tree F
图 2. 有向树 F

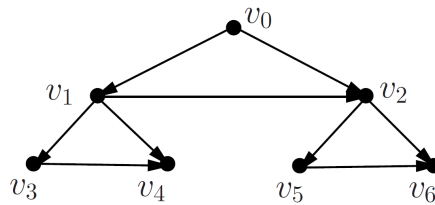


Figure 3. Directed triangle tree D_3
图 3. 有向三角形树 D_3

在每一时间步都将 D_1 的节点 v_0 粘贴到有向三角形树的最左(右)端的节点而生成的。

有向三角形树 D 的匹配数: n 个三角形的有向三角形树 D 的边的子集 E^* 称为是一个匹配集, 如果 E^* 中的任意两条有向边都没有公共的始点, 也没有公共的终点。有向三角形树 D 的一个匹配是由若干条有向边构成的集合, 故其大小就是有向边的条数。有向三角形树 D 的一个极大匹配是不能再通过添加有向边来使其变大的匹配。有向三角形树 D 的一个最大匹配是有向三角形树 D 的所有匹配中有向边数达到最大值的匹配, 即含有向边数最多的匹配。有向三角形树 D 的 k 匹配是指含 k 条有向边的匹配; 有向三角形树 D 的 k 匹配数指含 k ($k=1, 2, \dots, 2n+1$) 条有向边的匹配的选择方法数; 有向三角形树 D 的匹配数指所有 k 匹配数的和, 记为 $Z(D)$; 这里规定有向三角形树 D 的 0 匹配为 1, 即, 空图(记为“ O ”)或只包含孤立点的有向树的匹配数为 1。显然图 3 中有向三角形树 D_3 的 1 匹配数为 9: $v_0v_1, v_0v_2, v_1v_2, v_1v_3, v_1v_4, v_3v_4, v_2v_5, v_2v_6, v_5v_6$; 2-匹配数为 28: $\{v_0v_1, v_1v_3\}, \{v_0v_1, v_1v_4\}, \{v_0v_1, v_3v_4\}, \{v_0v_1, v_1v_2\}, \{v_0v_1, v_2v_5\}, \{v_0v_1, v_2v_6\}, \{v_0v_1, v_5v_6\}, \{v_0v_2, v_1v_3\}, \{v_0v_2, v_1v_4\}, \{v_0v_2, v_3v_4\}, \{v_0v_2, v_2v_5\}, \{v_0v_2, v_2v_6\}, \{v_0v_2, v_5v_6\}, \{v_1v_2, v_3v_4\}, \{v_1v_2, v_2v_5\}, \{v_1v_2, v_2v_6\}, \{v_1v_2, v_5v_6\}, \{v_1v_3, v_3v_4\}, \{v_1v_3, v_2v_5\}, \{v_1v_3, v_2v_6\}, \{v_1v_3, v_5v_6\}, \{v_1v_4, v_2v_5\}, \{v_1v_4, v_2v_6\}, \{v_1v_4, v_5v_6\}, \{v_3v_4, v_2v_5\}, \{v_3v_4, v_2v_6\}, \{v_3v_4, v_5v_6\}, \{v_2v_5, v_5v_6\}$; 同理, 3-匹配数为 36; 4-匹配数为 17; 5-匹配数为 3; 有向三角形树 D_3 的匹配数: 0 匹配数 + 1 匹配数 + 2 匹配数 + 3 匹配数 + 4 匹配数 + 5 匹配数 = 95。

3.3. 一类有向三角形树匹配数的计算方法

设 $D=(V, E)$ 是在有向完全二叉树中的每一个节点的左右子节点上, 添加一条从左子节点到右子节点的有向边所得的图, 它是一类特殊的有向三角形树图。下面给出这一类有向三角形树匹配数的多项式算法。

定理 3.3.1 设 $D=(V, E)$ 是在有向完全二叉树中的每一个节点的左右子节点上, 添加一条从左子节点到右子节点的有向边所得的图, 设 $D=(V, E)$ 是有 n 个三角形的有向三角形树, v_1, v_2 分别是有向三角形树 D 的根节点 v_0 的左、右子节点, 则

$$Z(D) = Z(D_{v_2}) [3Z(D_{v_1}) + 2Z(D_{v_1} - v_1)]$$

这里 D_{v_i} 是以 v_i 为根节点的有向三角形树; $D_{v_i-v_i}$ 是从有向三角形树 D_{v_i} 中删除根节点 v_i 形成的有向子图;

设 v_3, v_4 分别是 v_1 的左、右子节点, 则

$$Z(D_{v_1-v_1}) = Z(D_{v_4})[Z(D_{v_3}) + Z(D_{v_3-v_3})]$$

这里 $D_{v_3-v_3}$ 是 $D_{v_1-v_1}$ 的有向子图。

证明: 用归纳法证明, $n=1$ 时, 有向三角形树 D 仅包含一个三角形, $Z(D)=5$, 定理成立。

假设对小于 n 个三角形的有向三角形树 D 定理成立。设 v_1, v_2 分别是 v_0 的左、右子节点, 见图 4。

则有向三角形树 D 的匹配数可分为以下两类:

第一类: 不包含根节点 v_0 的匹配, 即有向子图 $D-v_0$ 的匹配数 $Z(D-v_0)$, 见图 5。

第二类: 恰好包含根节点 v_0 的有向边 v_0v_1 或 v_0v_2 的匹配, 即有向子图 D_1, D_2 的匹配数 $Z(D_1), Z(D_2)$,

见图 6。

因此

$$Z(D) = Z(D-v_0) + Z(D_1) + Z(D_2)$$

注意到有向子图 $D-v_0$ 的匹配可分为以下两类:

第一类: 不含有向边 v_1v_2 的匹配, 即有向树 D_{v_1} 的匹配数 $Z(D_{v_1})$ 与有向树 D_{v_2} 的匹配数 $Z(D_{v_2})$ 的乘积, 见图 7。

第二类: 恰好含有向边 v_1v_2 的匹配, 即有向子图 D_3 的匹配数 $Z(D_3)$, 见图 7。

因此

$$Z(D-v_0) = Z(D_{v_1})Z(D_{v_2}) + Z(D_3) = Z(D_{v_2})[Z(D_{v_1}) + Z(D_{v_1-v_1})]$$

而有向子图 D_1 的匹配可以分为以下两类:

第一类: 不含有向边 v_1v_2 的匹配, 即有向子图 D_4 的匹配数 $Z(D_4)$, 见图 8。

第二类: 恰好含有向边 v_1v_2 的匹配, 即有向子图 D_5 的匹配数 $Z(D_5)$, 见图 8。

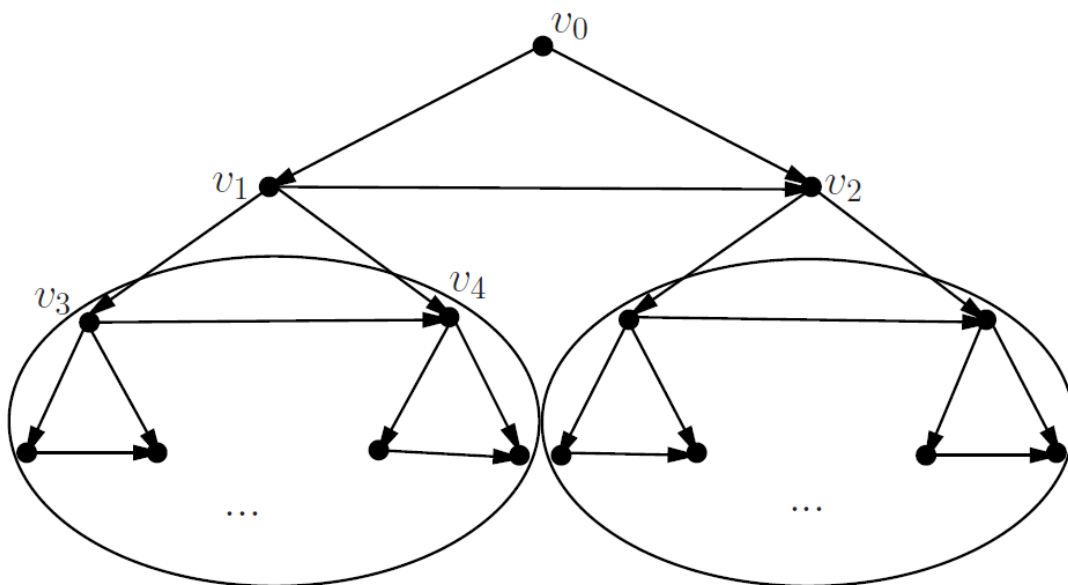


Figure 4. Directed triangle tree D

图 4. 有向三角形树 D

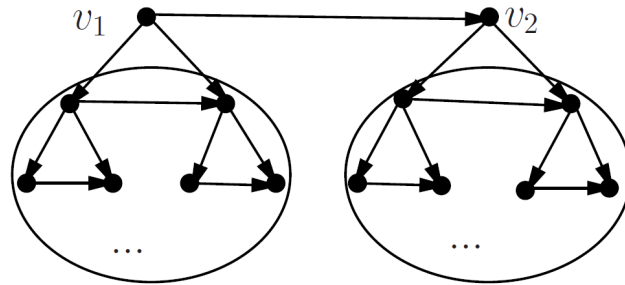


Figure 5. Directed graph $D-v_0$
图 5. 有向子图 $D-v_0$

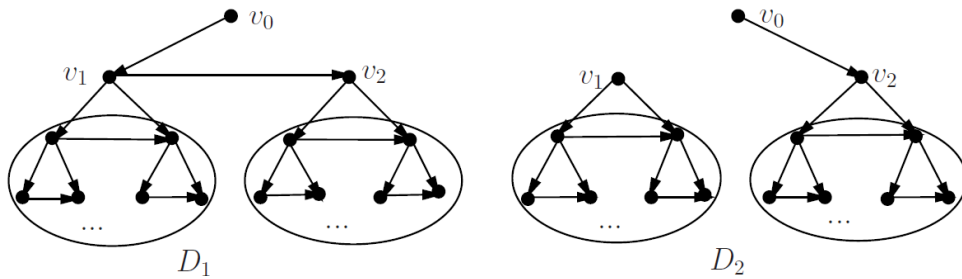


Figure 6. Directed graph D_1 , Directed graph D_2
图 6. 有向子图 D_1 , 有向子图 D_2

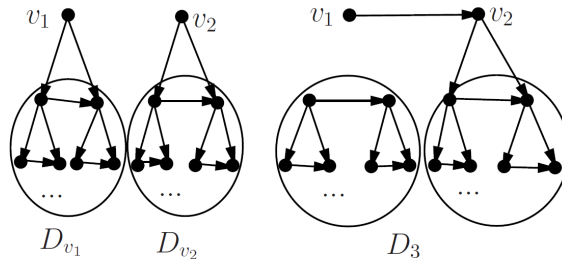


Figure 7. Directed tree D_{v_1} , Directed tree D_{v_2} , Directed graph D_3
图 7. 有向树 D_{v_1} , 有向树 D_{v_2} , 有向子图 D_3

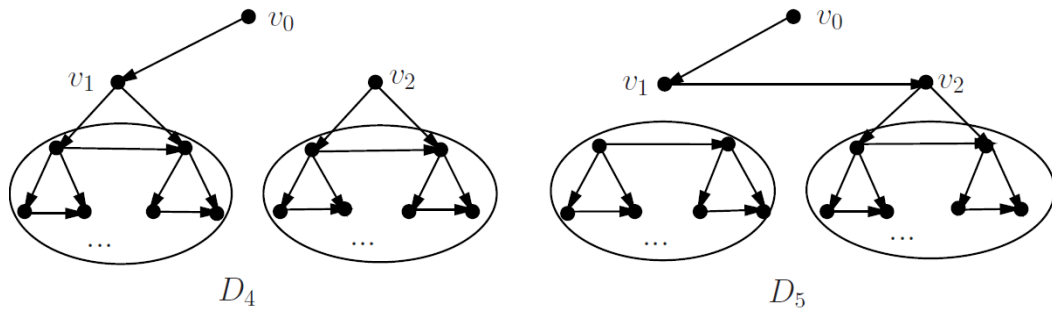


Figure 8. Directed graph D_4 , Directed graph D_5
图 8. 有向子图 D_4 , 有向子图 D_5

因此

$$Z(D_1) = Z(D_4) + Z(D_5) = Z(D_{v_2})[Z(D_{v_1}) + Z(D_{v_1} - v_1)];$$

$$Z(D_2) = Z(D_{v_1})Z(D_{v_2}).$$

则有向三角形树 D 的匹配数

$$Z(D) = 3Z(D_{v_1})Z(D_{v_2}) + 2Z(D_{v_1} - v_1)Z(D_{v_2}) = Z(D_{v_2})[3Z(D_{v_1}) + 2Z(D_{v_1} - v_1)]$$

注意到有向子图 $D_{v_1} - v_1$ 与有向子图 $D - v_0$ 的结构相同, 则

$$Z(D_{v_1} - v_1) = Z(D_{v_3})Z(D_{v_4}) + Z(D_{v_3} - v_3)Z(D_{v_4}) = Z(D_{v_4})[Z(D_{v_3}) + Z(D_{v_3} - v_3)]$$

由归纳假设得: 对有向三角形树 D_{v_3} , D_{v_4} 定理成立。

证毕。

定理 3.3.2 设 D 是含 n 个三角形的有向三角形树, 则

$$2^{n-1}(3*n+2) \leq Z(D) \leq 5^n$$

其中 $Z(D) = 2^{n-1}(3*n+2)$ 当且仅当 D 是有向三角形星 D_n^* ; $Z(D) = 5^n$ 当且仅当 D 是有向三角形路 D_{pn} 。

证明: 用数学归纳法证明。 $N = 1$ 时, 有向三角形树 D 仅包含一个三角形, $Z(D) = 5$, 定理成立。

$N = 2$ 时, 有向三角形树 D 包含两个三角形, 这两个三角形构成的有向三角形树图有三种结构: 有向三角形星图 D_2^* , 有向三角形路 \bar{D}_{p_2} 和 D_{p_2} 。计算得 $Z(D_2^*) = 16$, $Z(\bar{D}_{p_2}) = 17$, $Z(D_{p_2}) = 25$, 定理成立。

假设对小于 n 个三角形的有向三角形树图结论成立。(1) 用数学归纳法证明右边不等式。设 D_n 是具有 n 个三角形的有向三角形树图, 则 D_n 可以在某一个具有 $n-1$ 个三角形的有向三角形树图 D_{n-1} 的任意节点与新 $D_1 = v_0v_1v_2$ 的节点 v_0 粘贴得到。 D_n 和 D_{n-1} 的匹配有如下两种关系:

1) D_{n-1} 的任意一个匹配和 D_1 的任意一个匹配都没有公共始点和终点, 注意到 $Z(D_1) = 5$, 此时有

$$Z(D_n) = Z(D_1)Z(D_{n-1}) = 5Z(D_{n-1})$$

2) D_{n-1} 的某些匹配和 D_1 的某些匹配有公共的始点, 此时有

$$Z(D_n) < Z(D_1)Z(D_{n-1}) = 5Z(D_{n-1})$$

由归纳假设有

$$Z(D_{n-1}^*) \leq Z(D_{n-1}) \leq Z(D_{pn-1})$$

显然 $Z(D_{pn}) = 5Z(D_{pn-1}) = 5^n$, 因此得

$$Z(D_n) \leq Z(D_{pn}) = 5^n$$

且等式成立当且仅当 $D_n = D_{pn}$ 。

(2) 用数学归纳法证明左边不等式。设 D_n 可以在某一个具有 $n-1$ 个三角形的有向三角形树图 D_{n-1} 的任意节点与新 $D_1 = v_0v_1v_2$ 的节点 v_0 粘贴得到, D_n^* 在 D_{n-1}^* 的中心与 $D_1 = v_0v_1v_2$ 的 v_0 粘结得到, 其中 $D_1 = v_0v_1v_2$ 的有向边为 v_0v_1 , v_0v_2 , v_1v_2 。类似(1)的证明考虑 D_n^* 和 D_n 的匹配, 分下面几种情况考虑:

a) D_n^* 的匹配由 v_1v_2 与 D_{n-1}^* 匹配组成, D_n 的匹配由 v_1v_2 与 D_{n-1} 的匹配组成, 根据归纳假设这种情形下 $Z(D_n^*) \leq Z(D_n)$ 。

b) D_n^* 的匹配由 v_0v_1 , v_0v_2 , $v_0v_1v_2$ 分别与 D_{n-1}^* 三角形的第三边生成的匹配组成, D_n 的匹配由 v_0v_1 , v_0v_2 , $v_0v_1v_2$ 分别与 D_{n-1} 三角形的第三边生成的匹配组成, 这种情形下他们匹配数相等。

c) D_n^* 的匹配只能是 v_0v_1 , v_0v_2 , $v_0v_1v_2$, 或者是 D_{n-1}^* 包含中心点 v_0 的匹配, D_n 的匹配由 v_0v_1 , v_0v_2 , $v_0v_1v_2$, 或者 D_{n-1} 三角形的非第三边生成的匹配组成, 或者由 v_0v_1 , v_0v_2 , $v_0v_1v_2$, 和 D_{n-1} 中与 v_0 不相交的 D_{n-1} 三角形的非第三边生成的匹配组成, 这种情形下 $Z(D_n^*) < Z(D_n)$ 。

由以上讨论得:

$$Z(D_n^*) \leq Z(D_n)$$

且等式成立当且仅当 $D_n = D_n^*$ 。下面计算 $Z(D_n^*)$ 。

根据定理 3.3.1 设含 n 个三角形的有向三角形星图 D_n^* 的中心节点为 $v_0, v_1, v_2, \dots, v_{2n}$ 是 v_0 的邻居节点, 见图 9, 则有向三角形星图的匹配可分为以下两类:

第一类: 不包中心节点 v_0 的匹配, 即有向子图 $D_n^* - v_0$ 的匹配数 $Z(D_n^* - v_0)$, 见图 9。

第二类: 恰好包含中心节点 v_0 的有向边 $v_0v_i (i=1, 2, \dots, 2n)$ 的匹配, 即有向子图 $D_i^* (i=1, 2, \dots, 2n)$ 的匹配数 $Z(D_i^*)$, 见图 10。

如图 10 所示, 共有 n 个有向子图 $D_1^*, D_2^*, \dots, D_{2n}^*$, 因此

$$Z(D_n^*) = Z(D_n^* - v_0) + \sum_{i=1}^{2n} Z(D_i^*)$$

注意到 D_i^* 的每一个匹配必须包括有向边 v_0v_i , 当包含的有向边 v_0v_i 是 v_0 的右节点时, D_i^* 的匹配数

$$Z(D_i^*) = 2^n$$

当包含的有向边 v_0v_i 是 v_0 左节点时, D_i^* 的匹配数

$$Z(D_i^*) = 2^{n-1}$$

而有向三角形星图 D_n^* 左节点数等于右节点数等于 n 。从 D_i^* 的一个 k -匹配中删除左节点得到 $D_n^* - v_0$ 的一个 $(k-1)$ -匹配, 反之在 $D_n^* - v_0$ 的 $(k-1)$ -匹配中添加左节点得到 D_i^* 的一个 k -匹配,

则有

$$Z(D_n^* - v_0) = 2^n$$

所以

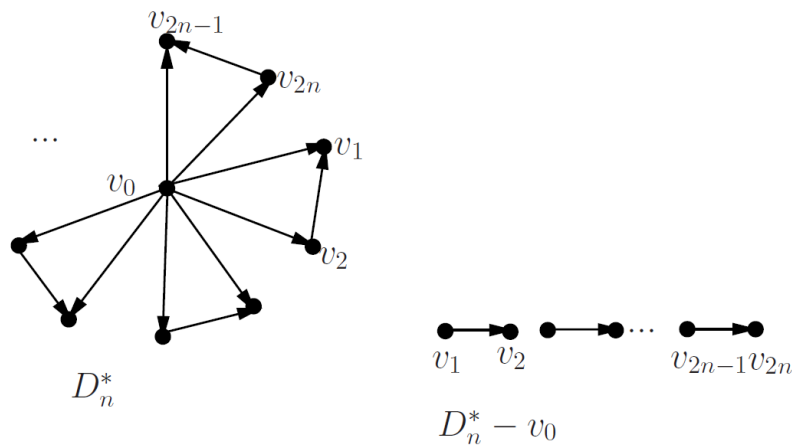


Figure 9. Directed triangle star graph D_n^* , Directed graph $D_n^* - v_0$
 图 9. 有向三角形星图 D_n^* , 有向子图 $D_n^* - v_0$

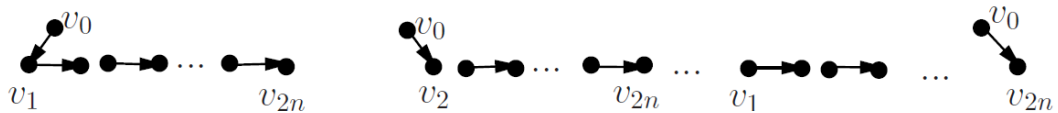


Figure 10. Directed graph D_i^*
 图 10. 有向子图 D_i^*

$$Z(D_n^*) = n(2^n + 2^{n-1}) + 2^n = 2^{n-1}(3n + 2)$$

证毕。

3.4. 有向三角形树匹配数的算法

本小节中, 我们给出有向三角形树匹配数的算法。

设 $D = (V, E)$ 是在有向完全二叉树中的每一个节点的左右子节点上, 添加一条从左子节点到右子节点的有向边所得的图, 设 $D_n = (V, E)$ 是包含 n 个三角形的有向三角形树, 输出 $Z(D_i)$ 值。假设 D_0 表示只含有一个节点的有向图, D_{vl} 和 D_{vr} 分别表示 D_v 的左、右子图。

开始

$$\begin{aligned} Z(O) &= 1, \quad Z(D_0) = 1, \quad Z(P_2) = 2 \\ \text{For each } v \in V \text{ bottom up do} \\ Z(D_v) &:= Z(D_{vr})[3Z(D_{vl}) + 2Z(D_{vl} - v_l)] \\ Z(D_v - v) &:= Z(D_{vr})[Z(D_{vl}) + Z(D_{vl} - v_l)] \end{aligned}$$

显然, 该算法的时间复杂度为 $o(n)$ 。

4. 结论与展望

本文中我们给出了有向网络中一类有向三角形树匹配数的计算方法和计算表达式。得出: 有向三角形路 D_{pn} 的匹配数最大, 有向三角形星图 D_n^* 的匹配数最小。有向三角形树的匹配数与以其左右端子节点为顶点的有向三角形的数目密切相关: 有向三角形树中, 以左端子节点为顶点的有向三角形的数目越多, 有向三角形树的匹配数越小; 有向三角形树中, 以右端子节点为顶点的有向三角形的数目越多, 有向三角形树的匹配数越大。而刘和 Barabasi 等人有结论: 有向路最易控制, 即: 需要的输入最少; 有向星图最难控制, 即: 需要的输入最多; 那么能否得出: 有向三角形树的匹配数越大, 有向三角形树越容易控制的结论; 或者在什么条件下, 有向三角形树的匹配数越大, 有向三角形树可控制性越好, 是值得思考的问题。

基金项目

本文受科技部 973 专项(No. 2010CB334708)、国家自然科学基金项目(No. 61164005)、教育部长江学者与创新团队支持计划(No. IRT1068)、国家自然科学基金项目(批准号: 11551002)、青海省自然科学基金项目(No. 2012-Z-943)资助。

参考文献 (References)

- [1] 汪小帆, 李翔, 陈关荣. 网络科学导论[M]. 北京: 高等教育出版社, 2012.
- [2] 郭世泽, 陆哲明. 复杂网络基础理论[M]. 北京: 科学出版社, 2012.
- [3] Liu, Y.Y., Slotine, J.J. and Barabasi, A.L. (2011) Controllability of Complex Networks. *Nature*, **473**, 167-173. <http://dx.doi.org/10.1038/nature10011>
- [4] 陈关荣. 复杂动态网络环境下控制理论遇到的问题与挑战[J]. 自动化学报. 2013, 39(4): 312-321.
- [5] Wagner, S. (2008) Extremal Trees with Respect to Hosoya Index and Merrifield-Simmons Index. *MATCH Communications in Mathematical and in Computer Chemistry*, **59**, 203-216.
- [6] Wagner, S. and Gutman, I. (2010) Maxima and Minima of the Hosoya Index and Merrifield-Simmons Index A Survey of Result and Techniques. *Acta Applicandae Mathematicae*, **112**, 323-346.

<http://dx.doi.org/10.1007/s10440-010-9575-5>

- [7] West, D.B., 著. 图论导引[M]. 李建中, 骆吉洲, 译. 北京: 机械工业出版社, 2005.
- [8] Jogen, B.J. and Gutin, G., 著. 有向图理论, 算法及其应用[M]. 姚兵, 张忠辅, 译. 北京: 科学出版社, 2009.
- [9] 邓辉文. 离散数学[M]. 北京: 清华大学出版社, 2013.