

图论中闭环问题的研究

王斌君¹ 邵 华^{1,2} 贺滢睿¹ 蔡文哲¹ 李璟颖¹

(中国人民公安大学信息技术与网络安全学院 北京 100038)¹

(公安部第一研究所测试中心 北京 100100)²

摘要 针对图论中闭环的应用问题,提出了结点满足某种特性的最小闭环、最小双闭环的概念,进而提出了“葵花”双闭环的概念,给出了最小闭环、最小双闭环和“葵花”双闭环的形式化定义、算法及算法分析,补充了图论中关于闭环的内容;同时也研究了闭环在视频侦察中的应用问题,为公安机关快速锁定犯罪嫌疑人提供理论和技术支持。

关键词 图,最小闭环,最小双闭环,“葵花”双闭环

中图分类号 TP391 文献标识码 A

Research on Closed Loop in Graph Theory

WANG Bin-jun¹ SHAO Hua^{1,2} HE Ying-rui¹ CAI Wen-zhe¹ LI Jing-ying¹

(College of Information Technology and Cyberspace Security, People's Public Security University of China, Beijing 100038, China)¹

(Test Center, First Research Institute of the Ministry of Public Security of PRC, Beijing 100100, China)²

Abstract Aiming at the problem of closed loop in the graph theory, we put forward the concept of minimum closed loop and minimum double closed loop, on which nodes satisfy certain properties. Then we put forward the concept of “sunflower” double closed loop, which put forward the formal definition, algorithm and algorithm analysis of the minimum closed loop, minimum double closed loop and “sunflower” double closed loop, which supplement the graph theory about the content of the closed loop. The application of closed loop detection in video is also studied, which provides theory and technology support for the police to detect and quickly lock the criminal suspect in the case of video surveillance.

Keywords Graph, Minimum closed loop, Minimum double closed loop, “Sunflower” double loop

1 引言

在人类生活中,许多问题都与图的某个最小闭环有关,例如:在道路交通中,针对某车辆,寻找满足特定时间要求路口的最小闭环,以便无遗漏地围追堵截或设置检查卡口^[1];围绕某案件,查找案发地周围具有监控头路口的最小闭环,以快速锁定犯罪嫌疑人^[2-4]等。这些实际问题都可抽象并归结为寻找图中结点是否具有某种特征的最小闭环问题。在实际应用时,往往还需要计算最小双闭环,例如:某地发生了一起犯罪案件,且确定该案件与车辆有关,那么,如果能计算出包含犯罪现场的具有摄像头路口的最小双闭环,就能根据车辆经过最小双闭环中的时间关系确定哪些是犯罪前进入和犯罪后离开的车辆,进而可以确定嫌疑车辆。计算最小双闭环是为了计算从外围到某个给定点必经的两个结点。

图论是研究图的理论,最早是由 29 岁的欧拉于 1736 年提出,研究包括图的表示及相关的算法,如货郎担问题、最短路径问题、拓扑排序问题等^[5-6]。

但是,现有的图论不包括对闭环的研究。本文从实际问题的研究中抽象出最小闭环、最小双闭环和“葵花”双闭环 3 类图的闭环问题,首次提出了相关的定义和形式化描述,并给出了相关的算法及算法分析。另外,根据实际需要,本文还对闭环理论在视频侦察中的应用进行了研究。

2 闭环概念描述

最小闭环是指对图中某一个给定的结点,寻找满足以下特征的结点集合:图中其他结点与给定结点的所有路径上,要么都包含最小闭环中的某一个结点(注:这些结点被称为闭环的外部结点);要么都不包含最小闭环中的任何结点(注:这些结点被称为闭环的内部结点);并且,该结点集合是最小的,即不存在这样的结点集合,其结点是该最小闭环的内部结点。如图 1(a)所示,三角形的点是给定结点,大、小圆点都是图的结点,而大圆点是具有某个特定性质的结点,闭合光滑曲线所包含的结点是该最小闭环。

另外,在图的很多问题中,还需要使用最小双闭环。最小双闭环是指:对图中某个给定的节点,首先确定一个最小闭环,并称其为内最小闭环;然后,在内最小闭环的外部结点中再寻找一个闭环,称其为外最小闭环,使得外最小闭环的外部结点到达给定结点只包含一个外最小闭环的结点,外最小闭环的内部结点到达给定结点均不包含外最小闭环的结点,且外最小闭环也是最小的。本文将内最小闭环和外最小闭环统称为最小双闭环,如图 1(b)所示。

最小双闭环的一个变种是“葵花”最小双闭环,简称“葵花”双闭环。“葵花”最小双闭环是指:对最小闭环上的每个结点,在最小闭环的外部结点中找出一个该结点的结点最小闭

王斌君(1962—),男,博士,教授,博士生导师,主要研究方向为网络犯罪侦查、网络空间安全, E-mail: wangbinjun@ppsuc.edu.cn;邵 华(1984—),男,博士生,主要研究方向为网络犯罪侦查;贺滢睿(1991—),女,硕士生,主要研究方向为网络犯罪侦查;蔡文哲(1989—),男,硕士生,主要研究方向为网络犯罪侦查;李璟颖 女,硕士生,主要研究方向为网络犯罪侦查。

环,使得任何一个最小闭环的外部结点到该结点的所有路径要么都包含该结点的结点最小闭环中的某一个结点;要么都不包含该结点的结点最小闭环中的任何结点,且该结点的结点最小闭环是最小的。如图1(c)所示。

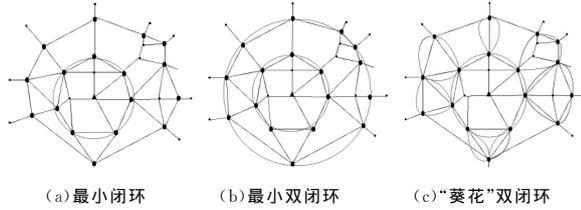


图1 图的闭环

闭环问题的相关概念可形式化定义如下:

设 $G = \langle V, E \rangle$ 是一个图,其中,

- 1) $V = \{v_1, v_2, \dots, v_n\}$ 表示结点的集合, n 是结点的个数;
- 2) $E = \{(v_i, v_j) \mid i, j = 1, 2, \dots, n\}$ 表示连接结点 v_i 和 v_j

的边的集合;

- 3) $w \in V$ 是某个给定的结点;

4) p 表示结点的性质, $\forall v_i \in V (i = 1, 2, \dots, n), v_i(p) = 1$ 表示 v_i 具有性质 $p, v_i(p) = 0$ 则表示 v_i 不具有性质 p ;

- 5) $SP = \{L_{ij} \mid \text{表示 } v_i \text{ 到 } w \text{ 的路径 } j \text{ 上的结点集} \mid v_i \in V, i = 1, 2, \dots, n, j = 1, 2, \dots, m\}$ 。

定义1(最小闭环) 对于给定的图 $G = \langle V, E \rangle$ 和给定结点 w 、性质 p ,如果同时满足下面5个条件,则称 MCL 为最小闭环。

- 1) $MCL = \{m_1, m_2, \dots, m_o \mid m_i \in V, \text{且 } m_i(p) \text{ 为真}, i = 1, 2, \dots, o\}$ 是一个闭环;
- 2) $MCL_I = \{m_{11}, m_{12}, \dots, m_{1l} \mid m_{1i} \in V, i = 1, 2, \dots, l\}$, 且 $\forall v_i \in MCL_I, \forall j, L_{ij} \in SP, L_{ij} \cap MCL = \{\}$, 则称 MCL_I 为闭环 MCL 的内部结点;
- 3) $MCL_O = \{m_{01}, m_{02}, \dots, m_{0k} \mid m_{0i} \in V, i = 1, 2, \dots, k\}$, 且 $\forall v_i \in MCL_O, \forall j, L_{ij} \in SP, L_{ij} \cap MCL \in MCL$, 则称 MCL_O 为闭环 MCL 的外部结点;
- 4) $MCL \cup MCL_I \cup MCL_O = V$;
- 5) 不存在闭环 $MCL' = \{m'_1, m'_2, \dots, m'_z \mid m'_i \in V, m'_i(p) = 1, i = 1, 2, \dots, z\}$, 使得 $\forall v_i \in MCL', \forall j, L_{ij} \cap MCL = \{\}$ 。

定义2(最小双闭环) 对于给定的图 $G = \langle V, E \rangle$ 和给定的结点 w 、性质 p ,记 MCL_1 是给定结点 w 的最小闭环,并称其为内最小闭环, MCL_{11} 和 MCL_{01} 分别对应 MCL_1 确定的内部结点和外部结点;记 MCL_2 是结点 w 的另一个闭环, MCL_{12} 和 MCL_{02} 分别是 MCL_2 确定的内部结点和外部结点,如果同时满足下面5个条件,则称 MCL_2 为外最小闭环,且将 MCL_1 和 MCL_2 统称为最小双闭环。

- 1) $MCL_2 = \{m_{21}, m_{22}, \dots, m_{2o} \mid m_{2i} \in V, \text{且 } m_{2i}(p) \text{ 为真}, i = 1, 2, \dots, o\}$;
- 2) $\forall v_i \in MCL_{12}, \forall j, L_{ij} \in SP \text{ 且 } L_{ij} \cap MCL_2 = \{\}$;
- 3) $\forall v_i \in MCL_{02}, \forall j, L_{ij} \in SP \text{ 且 } L_{ij} \cap MCL_2 \in MCL_2$;
- 4) $MCL_2 \cup MCL_{12} \cup MCL_{02} = V$;
- 5) 且除 MCL_1 外,不存在 $MCL' = \{m'_1, m'_2, \dots, m'_z \mid m'_i \in V, m'_i(p) = 1, i = 1, 2, \dots, z\}$, 使得 $\forall v_i \in MCL', \forall j, L_{ij} \in SP \text{ 且 } L_{ij} \cap MCL_2 = \{\}$ 。

定义3(‘葵花’双闭环) 对于给定的图 $G = \langle V, E \rangle$ 和给定结点 w ,若 MCL 是最小闭环,并称其为基本最小闭环, MCL_I 和 MCL_O 分别是 MCL 确定的内部结点和外部结点,

$\forall v_i \in MCL$,且满足以下条件,则称 MCL_i 为结点 v_i 的一个花瓣。

- 1) $MCL_i = \{m_1, m_2, \dots, m_o \mid m_i \in MCL_O, \text{且 } m_i(p) \text{ 为真}, i = 1, 2, \dots, o\}$ 是一个闭环;
- 2) $MCL_{1i} = \{m_{11}, m_{12}, \dots, m_{1l} \mid m_{1i} \in MCL_O, i = 1, 2, \dots, l\}$, 且 $\forall v_i \in MCL_{1i}, \forall j, L_{ij} \in SP, L_{ij} \cap MCL_i = v_i$, 则称其为闭环 MCL_i 的内部结点;
- 3) $MCL_{0i} = \{m_{E1}, m_{E2}, \dots, m_{Ek} \mid m_{Ei} \in MCL_O, i = 1, 2, \dots, k\}$, 且 $\forall v_i \in MCL_{0i}, \forall j, L_{ij} \in SP \text{ 且 } L_{ij} \cap MCL_i = \{v_i\} \cup \{v \mid v \in MCL_i \text{ 且 } v \neq v_i\}$, 则称其为闭环 MCL_i 的外部结点;
- 4) $MCL_i \cup MCL_{1i} \cup MCL_{0i} = MCL_O$;
- 5) 且除 MCL_i 外,不存在 $MCL' = \{m'_1, m'_2, \dots, m'_z \mid m'_i \in V, i = 1, 2, \dots, z\}$, 使得 $\forall v_i \in MCL', \forall j, L_{ij} \cap MCL = \{\}$ 。

将基本最小闭环 MCL 和其上所有结点的花瓣 MCL_i 统称为‘葵花’双闭环。

3 闭环算法研究

文中算法涉及到的主要数据结构及含义如下:

- 1) SL 表示开启列表,是正在等待判定的访问结点队列;
- 2) CL 为关闭列表,存放已经访问过并且不再访问的结点;
- 3) NL_v 表示结点 v 的所有相邻结点的集合。

算法1 最小闭环

Step1 $MCL = \{\}, CL = \{\}, SL = \{v' \mid v' \in NL_w\}; \text{flag} = 0$ 。

Step2 当 SL 不为空时,重复如下步骤:

Step2.1 任取 $v \in SL, CL = CL \cup \{v\}, SL = SL - \{v\}$;

Step2.2 若 $v(p)$ 为真,则 $MCL = MCL \cup \{v\}$;

Step2.3 若 $v(p)$ 为假,当 v 为边界元素时 $\text{flag} = 1$,转 Step3;否则, $SL = SL \cup \{v' \mid v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step3 若 $\text{flag} = 0$,则 MCL 为最小闭环;否则,没有最小闭环。

该算法在最坏情况下的时间复杂度为 $O(n)$ 。

算法2 最小双闭环

Step1 $MCL = \{\}, CL = \{\}, SL = \{v' \mid v' \in NL_w\}; \text{flag} = 0$ 。

Step2 当 SL 不为空时,重复如下步骤:

Step2.1 任取 $v \in SL, CL = CL \cup \{v\}, SL = SL - \{v\}$;

Step2.2 若 $v(p)$ 为真,则 $MCL = MCL \cup \{v\}$;

Step2.3 若 $v(p)$ 为假,当 v 为边界元素时 $\text{flag} = 1$,转 Step5;否则, $SL = SL \cup \{v' \mid v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step3 $MCL_2 = \{\}; \forall v \in MCL, SL = SL \cup \{v' \mid v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step4 当 SL 不为空时,重复如下步骤:

Step4.1 任取 $v \in SL, SL = SL - \{v\}, CL = CL \cup \{v\}$;

Step4.2 若 $v(p)$ 为真,则 $MCL_2 = MCL_2 \cup \{v\}$;

Step4.3 若 $v(p)$ 为假,当 v 为边界元素时 $\text{flag} = 1$,转 Step5;否则, $SL = SL \cup \{v' \mid v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step5 若 $\text{flag} = 0$,则 MCL 和 MCL_2 为最小双闭环;否则,没有最小双闭环。

该算法在最坏情况下的时间复杂度为 $O(n)$ 。

算法3 ‘葵花’双闭环

Step1 $MCL = \{\}, CL = \{\}, SL = \{v' \mid v' \in NL_w \text{ 且 } v' \notin CL\}; \text{flag} = 0$ 。

Step2 当 SL 不为空时,重复如下步骤:

Step2.1 任取 $v \in SL, CL = CL \cup \{v\}, SL = SL - \{v\}$;

Step2.2 若 $v(p)$ 为真,则 $MCL = MCL \cup \{v\}$;

Step2.3 若 $v(p)$ 为假,当 v 为边界元素时 $\text{flag} = 1$,转 Step5;否则, $SL = SL \cup \{v' \mid v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step3 $MCL' = MCL$ 。

Step4 当 MCL' 不为空时,重复如下步骤:

Step4.1 任取 $v \in MCL'$, $MCL_v = \{v\}$; $SL = \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL\}$; $CL' = CL$ 。

Step4.2 当 SL 不为空时,重复如下步骤:

Step4.2.1 任取 $v \in SL$, $SL = SL - \{v\}$, $CL' = CL' \cup \{v\}$;

Step4.2.2 若 $v(p)$ 为真,则 $MCL_v = MCL_v \cup \{v\}$;

Step4.2.3 若 $v(p)$ 为假,当 v 为边界元素则时 $flag = 1$, 转 Step5; 否则, $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step4.3 $MCL' = MCL' - \{v\}$ 。

Step5 若 $flag = 0$, 则 MCL 和所有的 MCL_v ($\forall v \in MCL$) 构成“葵花”双闭环; 否则, 没有最小双闭环。

该算法在最坏情况下的时间复杂度为 $O(n^2)$ 。

虽然“葵花”双闭环算法在最坏情况下的时间复杂度比双闭环的慢,但是在后续应用中,当检查结点跨越两个闭环所需要的计算量是主要矛盾时,采用“葵花”双闭环算法能取得更高的整体效率。

4 闭环的视频应用研究

公安机关在视频侦查办案过程中,特别是涉及车辆的情况时,需要调用周边的监控录像头来锁定涉案车辆。目前,公安机关在交通路口、重点部位和卡口基本都建设了摄像头。针对某个案件,传统的方式是查阅案件现场周边的摄像头,这种方式针对性不强,并且工作量巨大。通过最小闭环算法,可以无遗漏地查阅出、入案件现场摄像头的视频,再根据闭环上某个摄像头距案发现场的距离和交通拥堵情况估算出时间段,从而进一步减小侦察办案人员检查摄像头中视频的工作量。

针对某些案件,往往需要判定哪些车辆在案发前进入和案发后离开案发现场,这时可采用双闭环算法,并根据时间顺序关系锁定嫌疑车辆。以判定车辆进入为例,在案发前的某时刻,内闭环中发现了某车辆,且在更早的时间在外闭环中也发现了该车辆,则基本可以判定该车辆是犯罪前进入案发现场的嫌疑车辆。但是,采用双闭环算法判定进入和离开车辆的计算量很大,当发现内闭环中出现的车辆后,需要检查所有外闭环的摄像头。而采用本文提出的“葵花”双闭环,在发现内闭环中出现的车辆后,只需要检查其出现结点的花瓣即可,这样可大大减少工作量。

另外,在实际工作中,摄像头有普清和高清之分,在计算机自动识别应用中,高清的识别效果更好,能进一步辨识车辆中的人。但是,高清摄像头少,普清摄像头多。如果在道路交通网的图中,将高清标记为具有某种特性,则有可能找不到高清的闭环。为此,将普清摄像头和高清摄像头都标记为不同类型的某种特性,在算法中,尽量发现高清摄像头,如果找不到,则退而求其次寻找普清摄像头。下面以“葵花”双闭环算法为例来说明算法的修正。

假设: p_1 表示高清, p_2 表示普清; MCL_1 表示最小闭环高清的列表, MCL_2 表示最小双闭环的普清列表,由 MCL_1 和 MCL_2 共同构成最小闭环 MCL 。涉及的其他假设同第 2 节和第 3 节。

算法 4 带高清和普清的“葵花”双闭环算法

Step1 $MCL_1 = \{\}$, $MCL_2 = \{\}$, $CL = \{\}$, $SL = \{v' \mid \forall v' \in NL_w\}$ 。

Step2 当 SL 不为空时,重复如下步骤:

Step2.1 任取 $v \in SL$, $CL = CL \cup \{v\}$, $SL = SL - \{v\}$;

Step2.2 若 $v(p_1)$ 为真,则 $MCL_1 = MCL_1 \cup \{v\}$; 否则,若 $v(p_2)$ 为真,则 $MCL_2 = MCL_2 \cup \{v\}$; 否则, $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL\}$ 。

Step3 $MCL = MCL_1$ 。

Step4 当 MCL_2 不为空时,重复如下步骤://扩展普清节点

Step4.1 $CL' = CL$; 任取 $v \in MCL_2$, $MCL_2 = MCL_2 - \{v\}$; $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL'\}$; $MCL_1' = \{\}$; $flag = 0$ 。

Step4.2 当 SL 不为空时,重复如下步骤:

Step4.2.1 任取 $v \in SL$, $SL = SL - \{v\}$, $CL' = CL' \cup \{v\}$;

Step4.2.2 若 $v(p_1)$ 为真,则 $MCL_1' = MCL_1' \cup \{v\}$ 。

Step4.2.3 若 $v(p_1)$ 为假,当 v 是边界时 $flag = 1$, 转 Step4.3; 否则 $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL'\}$;

Step4.3 若 $flag = 0$, 则 $MCL = MCL \cup MCL_1'$, 否则 $MCL = MCL \cup \{v\}$ 。//最终高清闭环、普清闭环

Step5 $MCL' = MCL$ 。

Step6 当 MCL' 不为空时,重复如下步骤://计算花瓣

Step6.1 任取 $v \in MCL'$, $MCL' = MCL' - \{v\}$; $SL = \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL'\}$; $CL' = CL$; $MCL_1 = \{v\}$, $MCL_2 = \{\}$ 。

Step6.2 当 SL 不为空时,重复如下步骤:

Step6.2.1 任取 $v \in SL$, $SL = SL - \{v\}$, $CL' = CL' \cup \{v\}$;

Step6.2.2 若 $v(p_1)$ 为真,则 $MCL_1 = MCL_1 \cup \{v\}$; 否则,若 $v(p_2)$ 为真,则 $MCL_2 = MCL_2 \cup \{v\}$; 否则, $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL'\}$ 。

Step6.3 $MCL_v = MCL_1$ 。

Step6.4 当 MCL_2 不为空时,重复如下步骤://扩展花瓣的普清节点

Step6.4.1 $CL'' = CL'$; 任取 $v \in MCL_2$, $MCL_2 = MCL_2 - \{v\}$, $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL''\}$; $MCL_1' = \{\}$; $flag = 0$ 。

Step6.4.2 当 SL 不为空时,重复如下步骤:

Step6.4.2.1 任取 $v \in SL$, $SL = SL - \{v\}$, $CL'' = CL'' \cup \{v\}$;

Step6.4.2.2 若 $v(p_1)$ 为真,则 $MCL_1' = MCL_1' \cup \{v\}$; 否则,当 v 是边界时 $flag = 1$, 转 Step6.4.3; 否则 $SL = SL \cup \{v' \mid \forall v' \in NL_v \text{ 且 } v' \notin CL''\}$;

Step6.4.3 若 $flag = 0$, 则 $MCL_v = MCL_v \cup MCL_1'$, 否则 $MCL_v = MCL_v \cup \{v\}$ //最终高清花瓣、普清花瓣

Step7 结束。

结束语 我们在研究图的应用问题时,首次提出了图论中闭环的概念,并结合实际问题,提出了具有某种性质的最小闭环、双闭环和“葵花”双闭环的概念,同时给出了其形式化的定义,研究了相关算法。在此基础上,研究了“葵花”双闭环在视频侦察中应用时涉及高清、普清两种性质的具体问题,并给出了相关解决方案和算法。闭环的相关研究完善了图论的理论,在应用领域有着广泛的前景。

我们将进一步研究闭环的其他形态,并进一步研究闭环的应用问题。另外,本文限于篇幅的关系,只研究了相关的概念和算法,关于闭环算法的实验和“葵花”双闭环在侦查办案中应用的效果等将另文阐述。

参考文献

- [1] 杨敏,牟廉明,吴亚军,等.基于闭集的犯罪嫌疑人快速围堵算法[J].计算机工程与应用,2012,48(29):234-238.
- [2] 郑继明,姚翀.围堵在逃嫌疑犯的优化模型研究[J].科学技术与工程,2012,12(33):8980-8983.
- [3] 陈庆强.凸包算法在街面堵控系统中的应用[D].上海:东华大学,2014.
- [4] 程勇.视频侦查技术应用[M].北京:中国人民公安大学出版社,2012.
- [5] ROSS K A. Discrete Mathematics[M]. Pearson Prentice Hall, 2003.
- [6] 严蔚敏,吴伟民.数据结构(C语言版)[M].北京:清华大学出版社,2009.