

基于笼图-快速搜索的移动机器人目标搜索

孙晓莉, 钟发荣

(浙江师范大学 数学与计算机科学学院, 浙江 金华 321000)

摘要:随着信息技术的发展,人工智能已成为全球热点,以移动机器人目标搜索为代表的一系列人工智能应用已逐步影响人类生活的各个方面。为了解决移动机器人目标搜索在特定区域的搜索问题,提出一种基于笼图的快速搜索的移动机器人目标区域搜索模型。该模型为移动机器人目标搜索问题在笼图区域的搜索提供了有效的解决方法。首先通过分治思想研究笼图的性质,根据笼图顶点度和边的关系给出笼图的快速搜索数的下界定理;其次利用笼图的快速搜索数的下界定理进一步确定笼图的快速搜索数;最后针对度为3、围长为 g ($3 \leq g \leq 12$)的笼图,度为4、围长为 g ($3 \leq g \leq 8$)的笼图,提出一种笼图的快速搜索算法。实验结果表明,该算法实现了笼图的快速搜索,能够有效解决在笼图区域的移动机器人目标搜索问题。

关键词:移动机器人目标搜索;人工智能;快速搜索;笼图;搜索策略

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2020)08-0073-08

doi:10.3969/j.issn.1673-629X.2020.08.012

Mobile Robots Target Area Searching Based on Fast Searching of Cage Graph

SUN Xiao-li, ZHONG Fa-rong

(School of Computer Science, Zhejiang Normal University, Jinhua 321000, China)

Abstract: With the development of information technology, artificial intelligence has become a global hot spot. A series of artificial intelligence applications represented by mobile robot target search have gradually affected all aspects of human life. To solve the problem of mobile robot target search in cage areas, we propose a search algorithm for cage graph based on fast searching model. The goal of fast searching is to solve the minimum number of searchers needed to search the intruder. First of all, we study properties of cage graph to get lower bounds on the fast search number which is the minimum number of searchers needed to capture the intruder of cage graph. Then we apply lower bounds on the fast search number of cage graph to give the fast search number of cage graph. Finally, we provide a fast searching algorithm to search cage graph with degree 3 and girth g ($3 \leq g \leq 12$) and degree 4 and girth g ($3 \leq g \leq 8$). The results of three test graphs show that the proposed algorithm can effectively solve the problem of the target search of the mobile robot in the cage graph area.

Key words: mobile robots target area searching; artificial intelligence; fast search; cage graph; search strategy

0 引言

在生物界存在一类常见问题,群狼搜索一只羊。若搜索区域存在道路纵横交错的山洞,为了搜索到羊,群狼该如何利用最少的狼按照某种路线尽早捉到羊呢?移动机器人目标搜索与此类动物捕食问题类似,主要是研究一群作为搜索者的机器人应当如何合作、采取怎样的搜索策略去搜索目标。移动机器人目标搜索是实现救援、防御等任务的重要环节,是多机器人研

究的重要内容之一^[1-4]。所谓搜索问题,即搜索者试图搜索到目标。在某特定区域的移动机器人目标搜索实质是图搜索问题(graph searching)的实际应用案例。图搜索,即一群搜索者在某类图上利用最少的搜索者去搜索目标。因此,考虑将给定区域的移动机器人目标搜索问题的搜索区域抽象为类似结构的图,然后选用合适的图搜索模型,将其转化为图搜索问题进行研究。

收稿日期:2019-09-23

修回日期:2020-01-20

基金项目:国家自然科学基金面上项目(61572442,61272130);浙江省公益性技术应用研究项目(2015C33085);浙江省教育科研项目(Y201226127)

作者简介:孙晓莉(1994-),女,硕士研究生,CCF会员(F3878G),研究方向为图搜索、人工智能;钟发荣,博士,教授,研究方向为并行理论、图搜索。

针对不同的搜索问题,图搜索问题提供了多种模型,如边搜索、点搜索、混合搜索、快速搜索等^[5-9]。考虑到移动机器人目标搜索问题中机器人造价花费高及机器人需避免出现来回往返移动等需求,文中利用快速搜索模型进行研究。Dyer 等人^[10]介绍了快速搜索模型并提出一种用于计算树的快速搜索数的线性时间算法。Stanley 和 Yang^[11]给出一个用于计算 Halin 图的快速搜索数的线性时间算法,并提出一种二次时间算法来计算三次图的快速搜索数。Yang^[12]证明了求解图的快速搜索数是 NP 完全问题,他还证明了判断图 G 的快速搜索数是否等于 G 中奇数顶点数的一半是 NP 完全问题;Dereniowski 等人^[13]描述了一种在快速搜索模型中用两个或三个搜索者就足够搜索到入侵者的图,并证明了图的快速搜索问题是 NP 难问题。Xue 等人^[14]提出完全 k 部图、完全分割图的快速搜索数的下界和上界,并确定了完全二部图的快速搜索数。Xue^[15]还给出了笛卡尔积图的快速搜索数。

目前,在快速搜索模型的研究中仅研究了树、Halin 图、完全 k 部图、完全分割图等快速搜索数,还有大量不同结构图的快速搜索问题没有研究。然而,移动机器人目标搜索等实际问题中的搜索区域结构往往各不相同,当遇到除上述已有成果外的搜索区域时,已经提出的图的快速搜索数及其快速搜索算法不能满足其他区域的移动机器人目标搜索。因此,针对上述问题,考虑到实际生活中存在机器人在类似笼图结构的区域进行目标搜索,根据笼图的特定性质,确定笼图的快速搜索数下界,然后根据下界及搜索策略给出快速搜索数,提出一种基于笼图的快速搜索算法。该算法对求解与笼图类似区域的移动机器人目标搜索问题具有重要意义。

1 基本概念

为了论述方便,引入以下基本概念及符号^[16-18]。

1.1 图论概念

假设 $G = (V_c, E_c)$ 是简单无向图,其中 V_c 和 E_c 分别表示图 G 的顶点集和边集。用 uv 表示连接顶点 u 和 v 的边。如果 $uv \in E_c$,则称 u 和 v 邻接。若 E' 是 E_c 的子集, V' 是 V_c 的子集,则称 $H = (V', E')$ 是 G 的子图。如果顶点集 $S \subseteq V$,那么 $G[S]$ 表示图 G 中由 S 导出的子图。对于 V_c 的子集 X ,称 $N_c(X) = \{u \in V_c \mid \exists v \in X, uv \in E_c\}$ 为 X 的邻接顶点集。若 X 只有一个元素 u ,则用 $N_c(u)$ 表示 u 的邻接顶点集。顶点 u 的邻接顶点集大小,即 u 的邻接顶点数,称为 u 的度或价,用符号 $d(u)$ 或 r 表示,显然有 $d(u) = |N_c(u)| = |\{v \in V_c \mid uv \in E_c\}|$ 。

图 G 的一条通道(walk)是指一个有限的非空序

列 $W = v_0e_1v_1e_2v_2 \cdots e_kv_k$,其中 $v_i \in V_c(0 \leq i \leq k)$, $e_j = v_{j-1}v_j \in E_c(1 \leq j \leq k)$ 。若通道 W 中的边互不相同,则称之为迹(trail)。若迹 W 的长度为正且起点和终点相同,则称之为闭迹(trail)。若通道 W 中的顶点互不相同,则称之为路径(path)。使用 $p = v_1v_2 \cdots v_k$ 表示起点为 v_1 和终点为 v_k 的路径。路径中边的数目称为路径长度。若一条闭迹的起点和内部顶点互不相同,则称之为简单回路,也称为圈(cycle),记为 C 。圈中的边数称为圈的长度,长度为 k 的圈称为 k -圈,记为 C^k 。图中最短圈的长度也称为围长(girth),记为 g 。不在圈上但连接圈中两个顶点的边称为该圈的弦(chord)(如图 1 所示)。经过 G 的每条边一次并且仅一次的路径称为欧拉通路。如果欧拉通路是回路(起点和终点是同一个顶点),则称此回路为欧拉回路(Euler circuit)。

如果对 G 中所有的 $v \in V$,有 $d(v) = k$,则称图 G 是 k 正则的。围长为 g 且具有尽可能少的顶点的 r 正则图称为 (r, g) 笼,记为 $G_{r,g}$ 。 $|V_{G_{r,g}}|$ 表示 (r, g) 笼的顶点数。其中围长 g 为奇数的笼图,其顶点数 $|V_{G_{r,g}}| = 1 + r \sum_{i=0}^{(g-3)/2} (r-1)^i$;围长 g 为偶数的笼图,其顶点数 $|V_{G_{r,g}}| = 2 \sum_{i=0}^{(g-2)/2} (r-1)^i$, $r=2$ 的 $(2, g)$ 笼是 g -圈。 $g=2$ 的 $(r, 2)$ 笼只有两个顶点且由 r 个边连接^[19]。文中主要研究笼图 $G_{3,g}(3 \leq g \leq 12)$ 、 $G_{4,g}(3 \leq g \leq 8)$ 、 $G_{r,4}$ 。除 $(3, 5)$ 笼外,用 v_{ci} 表示通过弦与 v_i 相连的顶点, v_{nei} 表示通过非弦与 v_i 相连的顶点。

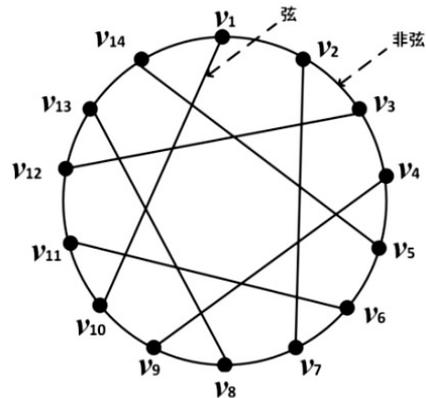


图 1 弦与非弦示意

1.2 快速搜索

快速搜索是一个在连通无向图上进行的搜索问题,该模型由对立的搜索者和入侵者双方组成。搜索初,图 G 上没有搜索者,但图 G 包含一个隐藏在顶点或边上的入侵者,故搜索开始时,假设整个图都是污染的。在搜索过程中,搜索者和入侵者分别在图上移动,其中搜索者只能从一个顶点滑动到与其相邻接的顶点,且搜索者看不见入侵者,只能根据每一次的移动路

径等信息,推测入侵者的位置信息;入侵者可以在任意时刻以任意快的速度沿着一条没有搜索者的路径滑动,去躲避搜索者的搜索^[10-12]。在该模型中,搜索者可以采取两种动作:占据在某一顶点(放置)、从某一顶点滑动到相邻顶点(滑动)。放置动作须发生在滑动动作前。

如果一条边 uv 可能包含入侵者,则称之为污染的,否则称之为干净的。可以采取以下两种动作搜索一条边 uv ,使之成为干净的边:

若顶点 u 包含大于等于两个搜索者,滑动其中一个搜索者从 u 沿边 uv 到 v ;

若顶点 u 包含一个搜索者,并且 uv 是唯一一条与 u 关联的污染的边,则滑动 u 上的搜索者沿边 uv 到 v 。

如果确定某个顶点没有关联污染的边,则称该顶点为干净的,否则称它为污染的。在快速搜索模型中,图中的每条边只允许遍历一次。如果某个顶点只包含一个搜索者,并且与该顶点关联的污染的边的数量大于该顶点上的搜索者的数量,则该顶点上的搜索者不能滑动离开该顶点,否则干净的顶点和边会重新被污染。

当所有的边和顶点都被搜索过,即搜索者搜索到入侵者,搜索结束。图的搜索策略即搜索者所移动的路线的集合。图 G 的快速搜索数,即在图上搜捕到入侵者所需要的最小的搜索者数,用 $fs(G)$ 表示。如果一个策略使用 $fs(G)$ 个搜索者搜捕到入侵者,则称此策略是搜索图 G 的最优搜索策略。

用 V_c 表示干净的顶点的集合, V_z 表示污染的顶点的集合, E_c 表示干净的边的集合, E_z 表示污染的边的集合。 $snum(v)$ 表示顶点 v 上的搜索者数量, $csnum(v)$ 表示可以在顶点 v 上滑动到其他顶点的搜索者数量, $E_z(v)$ 表示与顶点 v 关联的污染的边的集合。

2 笼图的快速搜索

对 $r=2$ 的笼图 $G_{2,g}$,由于 $G_{2,g}$ 是 g -圈,只需要在图中的某一顶点放置两个搜索者,移动其中一个沿圈 C^g 搜索即可搜索到目标,易得 $fs(G_{2,g})=2$ 。

对 $g=2$ 的笼图 $G_{r,2}$,由于 $G_{r,2}$ 只有两个顶点且由 r 个边连接,易得 $fs(G_{r,2})=3$ 。

在本节中,首先研究笼图的性质,利用顶点度和边的关系给出笼图 $G_{3,g}$ ($3 \leq g \leq 12$)、 $G_{4,g}$ ($3 \leq g \leq 8$) 的快速搜索数的下界;其次,根据快速搜索数的下界给出笼图 $G_{3,g}$ ($3 \leq g \leq 12$)、 $G_{4,g}$ ($3 \leq g \leq 8$)、 $G_{r,4}$ 的快速搜索数。

性质 1:对任意 $r=3, 3 \leq g \leq 12, r=4, 3 \leq g \leq 8$

的笼图 $G_{r,g}$,有 $|V_{G_{r,g}}| = \frac{2|E_{G_{r,g}}|}{r}$ 。

2.1 快速搜索数的下界

定理 1:对 $r=3, 3 \leq g \leq 12$ 笼图 $G_{3,g}$,有

$$fs(G_{3,g}) \geq \frac{|E_{G_{3,g}}|}{3} + 2。$$

证明:当 $r=3, 3 \leq g \leq 12$ 时,反证,假设在整个搜索过程中,有 $fs(G_{3,g}) \leq \frac{|E_{G_{3,g}}|}{3} + 1$ 。下面证明采用一个最多包含 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个搜索者的最优搜索策略时,

有 $|E_c| \leq |E_{G_{3,g}}| - 3$ 。根据性质 1,且由于放置动作发生在滑动动作前,故有以下四种基本情况:

(1)若在 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个顶点上放置 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个搜索者,由于每个顶点都关联三条污染的边,易知此时没有搜索者可以滑动,则图中没有任何边被搜索,故有 $|E_c| \leq |E_{G_{3,g}}| - 3$ 。

(2)若在 $\frac{|E_{G_{3,g}}|}{3}$ 个顶点上放置 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个搜索者,可知有一个顶点包含两个搜索者。由于每条边只允许遍历一次,当可滑动的搜索者滑动到一个之前不包含搜索者的顶点时将不能再滑动,因此经过一些滑动动作后,当所有的搜索者都不能再滑动时,搜索便结束,而此时最多可以搜索 $\frac{9|E_{G_{3,g}}|}{10} + 1$ 条边,于是有

$$|E_c| \leq \frac{9|E_{G_{3,g}}|}{10} + 1 \leq |E_{G_{3,g}}| - 3。$$

(3)若在 $\frac{|E_{G_{3,g}}|}{3} - 1$ 个顶点上放置 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个搜索者,考虑最优搜索策略,搜索者的分配如下:对 $r=3, 3 \leq g \leq 12$ (其中 $g \neq 5$) 的笼图,在一个顶点上放置三个搜索者,然后从该顶点的非弦邻接点(当 $g=5$ 时,从该顶点的任意邻接点)开始,将剩余的搜索者分别放置在圈 $C^{|V_c|}$ 上的各个顶点上(当 $g=5$ 时,将剩余的搜索者分别放置在任意不包含搜索者的邻接点上),直到放置完全部搜索者。由于每条边只能遍历一次,因此滑动顶点上的搜索者分别到该顶点的各个邻接点,然后滑动包含两个搜索者的顶点上的搜索者分别到其各个邻接点直到图中所有的搜索者不能继续滑动。此过程结束后最多可以搜索 $|E_{G_{3,g}}| - 3$ 条边,故有 $|E_c| \leq |E_{G_{3,g}}| - 3$ 。

(4)若在 $\frac{|E_{G_{3,g}}| + 3}{6}$ 个顶点上放置 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个搜索者,考虑最优策略,搜索者的分配如下:在一个顶点上放置三个搜索者,然后从该顶点的非弦邻接

点开始(当 $g=5$ 时,从该顶点的任意邻接顶点开始),将剩余的搜索者分别放置在圈 $C^{|V_c|}$ 上的各个顶点上(当 $g=5$ 时,将剩余的搜索者分别放置在任意不包含搜索者的邻接顶点上),直到放置全部搜索者。分别滑动搜索者搜索与其关联的边,直到没有可以滑动的搜索者。根据性质1以及此时最多有 $\frac{|E_{G_{3,g}}|+3}{6}$ 个

顶点包含搜索者,最多可以搜索 $\frac{|E_{G_{3,g}}|+6}{3}$ 条边,故有 $|E_c| \leq \frac{|E_{G_{3,g}}|+6}{3} \leq |E_{G_{3,g}}| - 3$ 。

综上,当最多使用 $\frac{|E_{G_{3,g}}|}{3} + 1$ 个搜索者搜索图 G 时,有 $|E_c| \leq |E_{G_{3,g}}| - 3$ 。这与该策略为最优搜索策略矛盾,故有 $fs(G_{3,g}) \geq \frac{|E_{G_{3,g}}|}{3} + 2$ 。

定理2:对 $r=4, 3 \leq g \leq 8$ 的笼图 $G_{4,g}$, 有:

$$fs(G_{4,g}) \geq \begin{cases} \lfloor \frac{|E_{G_{4,g}}|}{4} \rfloor + 3, 3 \leq g \leq 4 \\ \lfloor \frac{|E_{G_{4,g}}|}{5} \rfloor + 5, 5 \leq g \leq 8 \end{cases}$$

类似定理1的证明可证,不再赘述。

2.2 笼图的快速搜索数

下面给出 $r=3, 3 \leq g \leq 12, r=4, 3 \leq g \leq 8$ 以及 $g=4$ 的笼图的快速搜索数。

引理1:对 $r=3, g=3$ 的笼图,有 $fs(G_{3,3})=4$ 。

证明:根据定理1得, $fs(G_{3,3}) \geq 4$ 。下面采用一个4个搜索者的策略进行搜索($G_{3,3}$ 见图2)。

首先在顶点 v_1 放置搜索者 $\gamma_1, \gamma_2, \gamma_3$, 在顶点 v_4 放置搜索者 γ_4 , 然后进行以下移动动作:

(1)移动 γ_1 从顶点 v_1 沿边 v_1v_4 到顶点 v_4 , 移动 γ_2 从顶点 v_1 沿边 v_1v_3 到顶点 v_3 , 移动 γ_3 从顶点 v_1 沿边 v_1v_2 到顶点 v_2 。

(2)移动 γ_4 从顶点 v_4 沿路径 $v_4v_3v_2v_4$ 到顶点 v_4 。

此时, G 被搜索干净,因此对于(3,3)-笼图,有 $fs(G_{3,3})=4$,引理1成立。

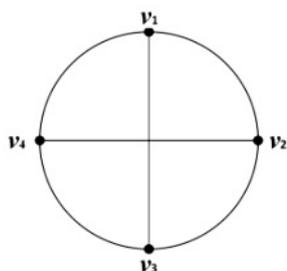


图2 (3,3) 笼

引理2:对 $r=3, g=4$ 的笼图,有 $fs(G_{3,4})=5$ 。

引理3:对 $r=3, g=5$ 的笼图,有 $fs(G_{3,5})=7$ 。

引理4:对 $r=3, g=6$ 的笼图,有 $fs(G_{3,6})=9$ 。

引理5:对 $r=3, g=7$ 的笼图,有 $fs(G_{3,7})=14$ 。

引理6:对 $r=3, g=8$ 的笼图,有 $fs(G_{3,8})=17$ 。

引理7:对 $r=3, g=9$ 的笼图,有 $fs(G_{3,9})=31$ 。

引理8:对 $r=3, g=10$ 的笼图,有 $fs(G_{3,10})=37$ 。

引理9:对 $r=3, g=11$ 的笼图,有 $fs(G_{3,11})=58$ 。

引理10:对 $r=3, g=12$ 的笼图,有 $fs(G_{3,12})=65$ 。

证明类似,不再逐一证明。

下面给出当 $r=3, 3 \leq g \leq 12$ 时,笼图的快速搜索数的定理。

定理3:对任意 $r=3, 3 \leq g \leq 12$ 的笼图,有 $fs(G_{3,g}) = \frac{|E_{G_{3,g}}|}{3} + 2$ 。

根据定理1以及引理1到引理10可证,不再赘述。

引理11:对 $r=4, g=3$ 的笼图,有 $fs(G_{4,3})=5$ 。

证明:根据定理2得, $fs(G_{4,3}) \geq 5$ 。下面采用一个5个搜索者的策略进行搜索($G_{4,3}$ 见图3)。首先在顶点 v_1 放置搜索者 $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ 在顶点 v_5 放置搜索者 γ_5 , 然后进行以下移动动作:

(1)移动 γ_1 从顶点 v_1 沿边 v_1v_5 到顶点 v_5 , 移动 γ_2 从顶点 v_1 沿边 v_1v_4 到顶点 v_4 , 移动 γ_3 从顶点 v_1 沿边 v_1v_3 到顶点 v_3 , 移动 γ_4 从顶点 v_1 沿边 v_1v_2 到顶点 v_2 。

(2)移动 γ_5 从顶点 v_5 沿路径 $v_5v_4v_3v_2v_5$ 到顶点 v_5 ;

(3)移动 γ_1 从顶点 v_5 沿边 v_5v_3 到顶点 v_3 。

(4)移动 γ_4 从顶点 v_2 沿边 v_2v_4 到顶点 v_4 。

此时,图被搜索干净,因此对于(4,3)-笼图,引理成立。

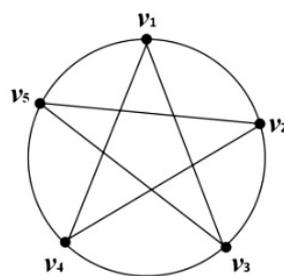


图3 (4,3) 笼

引理12:对 $r=4, g=4$ 的笼图,有 $fs(G_{4,4})=7$ 。

引理13:对 $r=4, g=5$ 的笼图,有 $fs(G_{4,5})=12$ 。

引理14:对 $r=4, g=6$ 的笼图,有 $fs(G_{4,6})=15$ 。

引理15:对 $r=4, g=7$ 的笼图,有 $fs(G_{4,7})=31$ 。

引理16:对 $r=4, g=8$ 的笼图,有 $fs(G_{4,8})=37$ 。

上述引理证明类似,不再逐一列出。

下面给出当 $r=4, 3 \leq g \leq 8$ 时,笼图的快速搜索数的定理。

定理4:对任意 $r=4, 3 \leq g \leq 8$ 的笼图,有:

$$fs(G_{4,g}) = \begin{cases} \lfloor \frac{|E_{G_{4,g}}|}{4} \rfloor + 3, 3 \leq g \leq 4 \\ \lfloor \frac{|E_{G_{4,g}}|}{5} \rfloor + 5, 5 \leq g \leq 8 \end{cases}$$

根据定理 2 以及引理 11 到引理 16 可证,不再赘述。

定理 5:对于 $r \geq 3, g = 4$ 的笼图,有 $fs(G_{r,4}) = 2r - 1$ 。

证明:对 $r \geq 3, g = 4$ 的笼图,给出以下快速搜索策略。

(1)在顶点 v_1 放置 r 个搜索者 $\gamma_1, \gamma_2, \dots, \gamma_r$,在顶点 $v_{|V_c|}$ 放置 $r - 2$ 个搜索者 $\gamma_{r+1}, \gamma_{r+2}, \dots, \gamma_{2r-2}$,在顶点 v_{2r-1} 放置搜索者 γ_{2r-1} 。

(2)对 i 从 1 到 r ,滑动 γ_i 从顶点 v_1 到顶点 v_j ,其中 $v_j \in N_H(v_i), 1 \leq j \leq |V_{G_{r,4}}|$ 。

(3)滑动顶点 $v_{|V_c|}$ 上所有的搜索者从 $v_{|V_c|}$ 分别到其每一个邻接点 v_j ,其中 $v_j \in N_H(v_{|V_c|})$ 。

(4)当 r 为奇数时,滑动搜索者 γ_{2r-1} 从顶点 v_{2r-1} 沿欧拉回路搜索 H ;当 r 为偶数时,滑动搜索者 γ_{2r-1} 从顶点 v_{2r-1} 沿 H 中的非弦边到顶点 v_{2r-1} ,然后滑动搜索者 γ_{2r-1} 从顶点 v_{2r-1} 沿其一条弦到其邻接点搜索直到该搜索者不能再滑动。

(5)若此时 H 仍有污染的边:

若 H 中存在 v_p 满足 $|E_Z(v_p)| = 1$,则滑动 v_p 上的搜索者搜索与其关联的污染的边。若仍存在污染的边,则继续滑动其他能滑动的搜索者搜索 H 。

2.3 笼图的快速搜索算法

下面给出搜索笼图的快速搜索算法,输入为笼图 $G_{r,g}$,输出为放置搜索者的顶点集 V 和移动路径序列 S 。算法大致思路为:(1)在图中某一顶点 v 放置搜索者,并分别移动该顶点上的搜索者到其各个邻接点;(2)此时若有满足 $1 \leq |E_Z(u)| \leq snum(u)$ 的顶点 u ,则滑动 u 上的搜索者沿污染的边搜索;(3)否则,从顶点 v 的一个非弦邻接点 v_i 依次沿圈 $C^{|V_c|}$ 开始(当 $r = 3, g = 5$ 时,从顶点 v 的任意一个邻接点开始),若 $|E_Z(v_i)| \geq 1$ 且 $csnum(v_i) < 1$,则在 v_i 上放置搜索者,使其满足 $|E_Z(v_i)| \leq snum(v_i)$,然后滑动 v_i 上搜索者搜索 v_i 的邻接点;(4)令 v_i 的一个非弦邻接点为新的 v_i (当 $r = 3, g = 5$ 时,令 v_i 的任意一个邻接点为新的 v_i),重复步骤 2 ~ 步骤 4,直到每个顶点都没有污染的边,即图完全干净。下面给出具体搜索算法,算法 SC 为搜索笼图 $G_{3,g} (3 \leq g \leq 12)$ 、 $G_{4,g} (3 \leq g \leq 8)$ 的搜索算法。

算法 SC:

1. 在 v_j 放置搜索者,分别移动 v_j 上的搜索者到其各个邻接点

2. if $r = 3, g = 5$ then
3. 选择 v_j 的一个邻接点 v_p , 令 $i \leftarrow p$
4. else
5. 选择 v_j 的一个邻接点 v_{nej} , 令 $i \leftarrow nej$
6. end if
7. while $E_Z \neq \emptyset$ do
8. while 除了 v_i 有满足 $1 \leq |E_Z(u)| \leq snum(u)$ 的顶点 u do
9. 滑动 u 上的搜索者沿污染的边搜索
10. end while
11. while 除了 v_i 所有的脏点满足 $csnum(v_i) < 1$ and $|E_Z(v_i)| \geq 1$ do
12. if v_i 满足 $snum(v_i) \leq 1$ and $|E_Z(v_i)| \geq 1$ then
13. 在 v_i 上放置 $|E_Z(v_i)| - 1$ 个搜索者,分别移动 v_i 的搜索者到其邻接点
14. end if
15. if v_i 满足 $snum(v_i) \geq 2$ and $|E_Z(v_i)| \geq 1$ then
16. 分别移动 v_i 的搜索者到其邻接点
17. end if
18. if $r = 3, g = 5$ then
19. 令 v_i 的一个邻接点为新的 v_i
20. else
21. 令 v_i 的一个非弦邻接点为新的 v_i
22. end if
23. end while
24. end while

3 实例分析

为保证笼图的快速搜索算法的准确性,本节将对 2.3 节中的算法 SC 进行实验验证。实验的测试例图为(3,5)笼、(3,6)笼、(4,4)笼。具体数据见表 1。其中 $|V_c|$ 为顶点数, $|E_c|$ 为边数, $fs(G)$ 为快速搜索数。

表 1 实验测试例图数据

笼图	$ V_c $	$ E_c $	$fs(G)$
$G_{3,5}$	10	15	7
$G_{3,6}$	14	21	9
$G_{4,4}$	8	16	7

3.1 算法 SC 在笼图 $G_{3,5}$ 上的实验

根据引理 3 可知,搜索笼图 $G_{3,5}$ 需要 7 个搜索者。实验过程如下:在第 0 轮,在顶点 v_1 放置三个搜索者,第 1 轮,移动顶点 v_1 的搜索者去搜索;第 2 轮,在顶点 v_5 放置一个搜索者,移动顶点 v_5 上的搜索者去搜索;第 3 轮,在顶点 v_4 放置一个搜索者,移动 v_4 上的搜索者去搜索;第 4 轮,在顶点 v_3 放置一个搜索者,移动 v_3 上的搜索者去搜索;第 5 轮,移动 v_2 上的搜索者去搜索;第 6 轮,在顶点 v_7 放置一个搜索者,移动 v_7 的搜索者去搜索。下面由表 2 给出搜索过程中搜索者的位置、脏边集合、脏点集合的变化。其中 i 为移动的轮次, V_i 为搜

搜索者每轮所在位置集合, E_Z 为脏边集合(搜索者的移动路径), V_Z 为脏点集合, 且 E_Z 、 V_Z 随轮次 i 变化。

表2 搜索过程中 $G_{3,5}$ 脏点集与脏边集随搜索轮次的变化

i	V_i	E_Z	V_Z
0	$\{v_1, v_5, v_4, v_3, v_7\}$	$E_Z = E_G$	$V_Z = V_G$
1	$\{v_5, v_6, v_4, v_3, v_7, v_2\}$	$E_Z = E_Z - \{v_1 v_6, v_1 v_5, v_1 v_2\}$	$V_Z = V_Z - \{v_1\}$
2	$\{v_{10}, v_4, v_3, v_2, v_7, v_6\}$	$E_Z = E_Z - \{v_5 v_4, v_5 v_{10}\}$	$V_Z = V_Z - \{v_5\}$
3	$\{v_3, v_9, v_{10}, v_6, v_7, v_2\}$	$E_Z = E_Z - \{v_4 v_9, v_4 v_3\}$	$V_Z = V_Z - \{v_4\}$
4	$\{v_8, v_2, v_7, v_6, v_{10}, v_9\}$	$E_Z = E_Z - \{v_3 v_8, v_3 v_2\}$	$V_Z = V_Z - \{v_3\}$
5	$\{v_8, v_2, v_7, v_6, v_{10}, v_9\}$	$E_Z = E_Z - \{v_2 v_7\}$	$V_Z = V_Z - \{v_2\}$
6	$\{v_8, v_2, v_7, v_6, v_{10}, v_9\}$	$E_Z = E_Z - \{v_7 v_{10}, v_{10} v_8, v_8 v_6, v_6 v_9, v_9 v_7\}$	$V_Z = V_Z - \{v_7, v_{10}, v_8, v_6, v_9\}$

图4所示为搜索过程中,脏点和脏边的变化过程及实验结果。其中三角形顶点表示已经搜索过的顶点,虚线表示已经搜索过的边,圆形顶点表示未搜索的顶点,实线边表示未搜索的边。

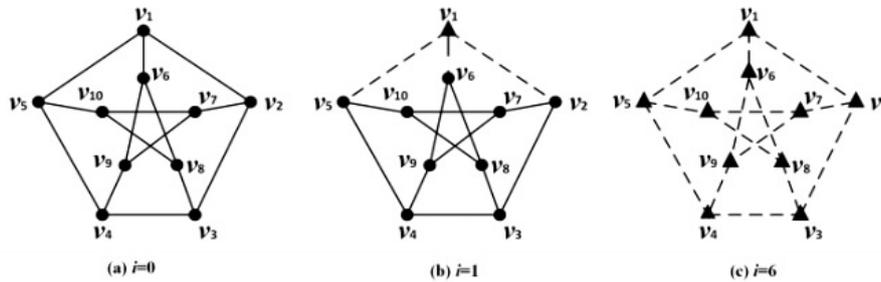


图4 (3,5) 笼的部分搜索过程示意

通过表2和图4可以看出,实验经6轮搜索后,脏点集和脏边集为空集,图4(c)为最后结果,可以看出 $G_{3,5}$ 中的所有顶点皆为三角形、所有边皆为虚线,即 $G_{3,5}$ 中的脏点和脏边全部被搜索完。

3.2 算法 SC 在笼图 $G_{3,6}$ 上的实验

根据引理4可知,搜索笼图 $G_{3,6}$ 需要9个搜索者。实验过程如下:在第0轮,在顶点 v_1 放置三个搜索者;第1轮,移动顶点 v_1 的搜索者去搜索;第2轮,在顶点 v_{14} 放置一个搜索者,移动顶点 v_{14} 上的搜索者去搜索;第3轮,在顶点 v_{13} 放置一个搜索者,移动 v_{13} 上的搜索

者去搜索;第4轮,在顶点 v_{12} 放置一个搜索者,移动 v_{12} 上的搜索者去搜索;第5轮,在顶点 v_{11} 放置一个搜索者,移动 v_{11} 上的搜索者去搜索;第6轮,移动 v_{10} 的搜索者去搜索;第7轮,在顶点 v_9 放置一个搜索者,移动 v_9 的搜索者去搜索;第8轮,移动 v_8 的搜索者去搜索;第9轮,在顶点 v_7 放置一个搜索者,移动 v_7 的搜索者去搜索。下面由表3给出搜索过程中搜索者的位置、脏边集合、脏点集合的变化。其中 i 为移动的轮次, V_i 为搜索者所在位置集合, E_Z 为脏边集合(搜索者的移动路径), V_Z 为脏点集合,且 E_Z 、 V_Z 随轮次 i 变化。

表3 搜索过程中 $G_{3,6}$ 的脏点集与脏边集随搜索轮次的变化

i	V_i	E_Z	V_Z
0	$\{v_1, v_{14}, v_{13}, v_{12}, v_{11}, v_9, v_7\}$	$E_Z = E_G$	$V_Z = V_G$
1	$\{v_2, v_{14}, v_{13}, v_{12}, v_{11}, v_{10}, v_9, v_7\}$	$E_Z = E_Z - \{v_1 v_{10}, v_1 v_2, v_1 v_{14}\}$	$V_Z = V_Z - \{v_1\}$
2	$\{v_2, v_5, v_{13}, v_{12}, v_{11}, v_{10}, v_9, v_7\}$	$E_Z = E_Z - \{v_{14} v_5, v_{14} v_{13}\}$	$V_Z = V_Z - \{v_{14}\}$
3	$\{v_2, v_5, v_8, v_{12}, v_{11}, v_{10}, v_9, v_7\}$	$E_Z = E_Z - \{v_{13} v_8, v_{13} v_{12}\}$	$V_Z = V_Z - \{v_{13}\}$
4	$\{v_2, v_5, v_8, v_3, v_{11}, v_{10}, v_9, v_7\}$	$E_Z = E_Z - \{v_{12} v_3, v_{12} v_{11}\}$	$V_Z = V_Z - \{v_{12}\}$
5	$\{v_2, v_5, v_8, v_3, v_6, v_{10}, v_9, v_7\}$	$E_Z = E_Z - \{v_{11} v_6, v_{11} v_{10}\}$	$V_Z = V_Z - \{v_{11}\}$
6	$\{v_2, v_5, v_8, v_3, v_6, v_{10}, v_9, v_7\}$	$E_Z = E_Z - \{v_{10} v_9\}$	$V_Z = V_Z - \{v_{10}\}$
7	$\{v_2, v_5, v_8, v_3, v_6, v_{10}, v_4, v_7\}$	$E_Z = E_Z - \{v_9 v_4, v_9 v_8\}$	$V_Z = V_Z - \{v_9\}$
8	$\{v_2, v_5, v_8, v_3, v_6, v_{10}, v_4, v_7\}$	$E_Z = E_Z - \{v_8 v_7\}$	$V_Z = V_Z - \{v_8\}$
9	$\{v_2, v_5, v_8, v_3, v_6, v_{10}, v_4, v_7\}$	$E_Z = E_Z - \{v_7 v_2, v_2 v_3, v_3 v_4, v_4 v_5, v_5 v_6, v_6 v_7\}$	$V_Z = V_Z - \{v_7, v_2, v_3, v_4, v_5, v_6\}$

图 5 所示为搜索过程中,脏点和脏边的变化过程及实验结果图。其中图中三角形顶点表示已经搜索过

的顶点、虚线表示已经搜索过的边,圆形顶点表示未搜索的顶点、实线边表示未搜索的边。

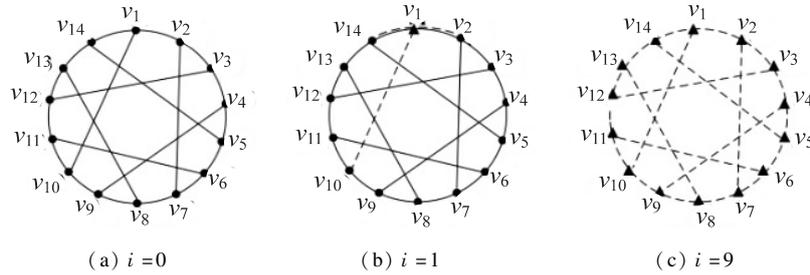


图 5 (3,6) 笼的部分搜索过程示意

通过表 3 和图 5 可以看出,该实验经 9 轮搜索后,脏点集合和脏边集合为空集,其中图 5(c) 为最后结果图,从图 5 可以看出 $G_{3,6}$ 中所有顶点皆为三角形、所有边皆为虚线,即 $G_{3,6}$ 中的脏点和脏边全部被搜索完。

点 v_8 放置 2 个搜索者,移动顶点 v_8 上的搜索者去搜索;第 3 轮,在顶点 v_7 放置一个搜索者,移动 v_7 上的搜索者去搜索;第 4 轮,移动 v_6 上的搜索者去搜索;第 5 轮,移动 v_5 上的搜索者去搜索。下面由表 4 给出搜索过程中搜索者的位置、脏边集合、脏点集合的变化。其中 i 为移动的轮次, V_i 为搜索者所在位置集合, E_Z 为脏边集合(搜索者的移动路径), V_Z 为脏点集合,且 E_Z 、 V_Z 随轮次 i 变化。

3.3 算法 SC 在笼图 $G_{4,4}$ 上的实验

根据引理 12 可知,搜索笼图 $G_{4,4}$ 需要 7 个搜索者。实验过程如下:在第 0 轮,在顶点 v_1 放置三个搜索者;第 1 轮,移动顶点 v_1 的搜索者去搜索;第 2 轮,在顶

表 4 搜索过程中 $G_{4,4}$ 的脏点集与脏边集随搜索轮次的变化

i	V_i	E_Z	V_Z
0	$\{v_1, v_8, v_7\}$	$E_Z = E_G$	$V_Z = V_G$
1	$\{v_6, v_4, v_2, v_8, v_7\}$	$E_Z = E_Z - \{v_1 v_8, v_1 v_6, v_1 v_4, v_1 v_2\}$	$V_Z = V_Z - \{v_1\}$
2	$\{v_5, v_6, v_4, v_3, v_2, v_7\}$	$E_Z = E_Z - \{v_8 v_5, v_8 v_3, v_8 v_7\}$	$V_Z = V_Z - \{v_8\}$
3	$\{v_7, v_6, v_5, v_4, v_3, v_2\}$	$E_Z = E_Z - \{v_7 v_4, v_4 v_3, v_3 v_2, v_2 v_7, v_7 v_6, v_6 v_5, v_5 v_4\}$	$V_Z = V_Z - \{v_7, v_4\}$
4	$\{v_7, v_6, v_5, v_4, v_3, v_2\}$	$E_Z = E_Z - \{v_6 v_3\}$	$V_Z = V_Z - \{v_6, v_3\}$
5	$\{v_7, v_5, v_4, v_3, v_2\}$	$E_Z = E_Z - \{v_5 v_2\}$	$V_Z = V_Z - \{v_5, v_2\}$

图 6 所示为搜索过程中,脏点和脏边的变化过程及实验结果图。其中三角形顶点表示已经搜索过的顶

点,虚线表示已经搜索过的边,圆形顶点表示未搜索的顶点,实线边表示未搜索的边。

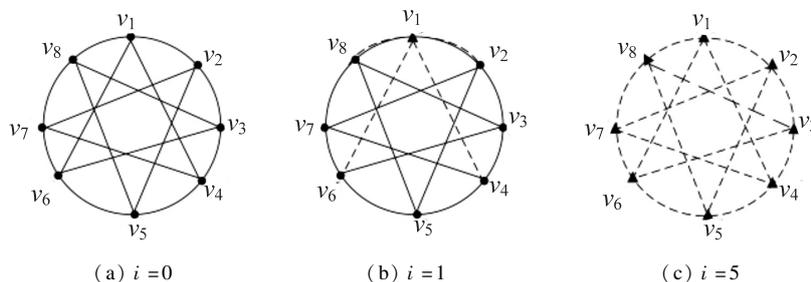


图 6 (4,4) 笼的部分搜索过程示意

通过表 4 和图 6 可以看出,该实验经 5 轮搜索后,脏点集合和脏边集合皆为空集,其中图 6(c) 为最后结果,可见笼图 $G_{4,4}$ 中所有顶点皆为三角形、所有边皆为虚线,即图中所有顶点和边都被搜索过,完成对笼图 $G_{4,4}$ 的搜索。

置搜索者(机器人)的个数,又在每条边搜索的过程中遵循只搜索一次的规则完成在整个区域对目标的搜索。从表 2、表 3、表 4 可以看出脏边一直在减少,且被搜索过的脏边不会再次成为脏边,即搜索者对每条边只搜索了一次,减少了搜索次数;从图 4(c)、图 5(c)、图 6(c) 可见,图中所有顶点皆为三角形、所有边皆为虚线,即都被搜索者搜索过,有效地完成了对笼图 $G_{3,5}$ 、 $G_{3,6}$ 、 $G_{4,4}$ 的搜索。因此,实验表明算法 SC 在经

3.4 总结与分析

在上述三个例图实验中,根据 2.2 节给出的搜索数引理,利用算法 SC 根据顶点关联的污染的边数放

有限轮次后使笼图中的脏点和脏边集合皆为空集,有效地完成了对整个笼图区域的搜索。

4 结束语

针对笼图区域的移动机器人目标搜索问题,对笼图快速搜索建模,通过对笼图性质的探索得到笼图的快速搜索数的下界及笼图的快速搜索数。提出一种基于笼图的快速搜索算法,用于解决笼图形状区域的移动机器人目标搜索问题。在快速搜索模型下对笼图的顶点和边进行搜索,经在三个笼图实例上的模拟验证,该算法可以有效地使机器人在不知道目标具体位置的情况下,逐步搜索到目标。该研究为基于笼图的移动机器人目标搜索问题的求解提供了一种新思路和可行的搜索策略,但仍需进一步在大规模问题上研究。

参考文献:

- [1] 揣迎才,张明清,唐俊,等.智能协作型防火墙仿真模型研究[J].信息工程大学学报,2013,14(1):109-113.
- [2] 曹翔,孙长银.栅格地图中多机器人协作搜索目标[J].控制理论与应用,2018,35(3):273-282.
- [3] 罗盼.室内移动机器人区域语义地图创建及目标搜索方法研究[D].哈尔滨:哈尔滨工业大学,2018.
- [4] 曹凯,高佳佳,高嵩,等.移动机器人在复杂环境中的在线路径规划[J].自动化与仪表,2018,33(9):27-31.
- [5] BONATO A, YANG B. Graph searching and related problems [M]//Handbook of combinatorial optimization. [s. l.]: [s. n.], 2013: 1511-1558.
- [6] PARSONS T D. Pursuit-evasion in a graph [M]//Theory and applications of graphs. Berlin, Heidelberg: Springer, 1978: 426-441.
- [7] MEGIDDO N, HAKIMI S L, GAREY M R, et al. The complexity of searching a graph [J]. Journal of the ACM, 1988, 35(1): 18-44.
- [8] ALSPACH B. Searching and sweeping graphs: a brief survey [J]. Matematiche, 2004, 59(1-2): 5-37.
- [9] KIROUSIS L M, PAPADIMITRIOU C H. Searching and pebbling [J]. Theoretical Computer Science, 1986, 47: 205-218.
- [10] DYER D, YANG B, YAŞAR Ö. On the fast searching problem [C]//International conference on algorithmic applications in management. Berlin, Heidelberg: Springer, 2008: 143-154.
- [11] STANLEY D, YANG B. Fast searching games on graphs [J]. Journal of Combinatorial Optimization, 2011, 22(4): 763-777.
- [12] YANG B. Fast edge searching and fast searching on graphs [J]. Theoretical Computer Science, 2011, 412(12-14): 1208-1219.
- [13] DERENIOWSKI D, DINER Ö Y, DYER D. Three-fast-searchable graphs [J]. Discrete Applied Mathematics, 2013, 161(13-14): 1950-1958.
- [14] XUE Y, YANG B, ZHONG F, et al. Fast searching on complete k-partite graphs [C]//International conference on combinatorial optimization and applications. [s. l.]: [s. n.], 2016: 159-174.
- [15] XUE Y, YANG B. The fast search number of a Cartesian product of graphs [J]. Discrete Applied Mathematics, 2017, 224: 106-119.
- [16] BONDY J A, MURTY U S R. Graph theory with applications [M]. London: Macmillan, 1976.
- [17] DIESTEL R. 图论 [M]. 于青林, 王涛, 王光辉, 译. 北京: 高等教育出版社, 2013.
- [18] 博洛巴. 现代图论 [M]. 北京: 科学出版社, 2001.
- [19] Wong P K. Cages—a survey [J]. Journal of Graph Theory, 1982, 6(1): 1-22.