Discrete Mathematics 339 (2016) 2804–2811

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

Subdivisions in the Robber Locating game

John Haslegrave [a,*], Richard A.B. Johnson [b], Sebastian Koch [c]

[a] University of Sheffield, Sheffield, UK
[b] University of Memphis, Memphis, TN, USA
[c] University of Cambridge, Cambridge, UK

ARTICLE INFO

Article history:
Received 15 September 2015
Received in revised form 11 April 2016
Accepted 19 May 2016
Available online 15 June 2016

Keywords:
Graph searching
Cops and robbers
Subdivision

ABSTRACT

We consider a game in which a cop searches for a moving robber on a graph using distance probes, which is a slight variation on one introduced by Seager. Carraher, Choi, Delcourt, Erickson and West showed that for any n-vertex graph G there is a winning strategy for the cop on the graph $G^{1/m}$ obtained by replacing each edge of G by a path of length m, if $m \geqslant n$. They conjectured that this bound was best possible for complete graphs, but the present authors showed that in fact the cop wins on $K_n^{1/m}$ if and only if $m \geqslant n/2$, for all but a few small values of n. In this paper we extend this result to general graphs by proving that the cop has a winning strategy on $G^{1/m}$ provided $m \geqslant n/2$ for all but a few small values of n; this bound is best possible. We also consider replacing the edges of G with paths of varying lengths.

1. Introduction

Pursuit and evasion games on graphs have been widely studied, beginning with the introduction by Parsons [11] of a game where a fixed number of searchers try to find a lost spelunker in a dark cave. The searchers cannot tell where the target is, and aim to move around the vertices and edges of the graph in such a way that one of them must eventually encounter him. The spelunker may move around the graph in an arbitrary fashion, and in the worst case may be regarded as an antagonist who knows the searchers' positions and is trying to escape them.

The best-known variant is the classical Cops and Robbers game, introduced by Quillot [12], and independently in a paper of Nowakowski and Winkler [10] (where it is attributed to G. Gabor). Unlike the Lost Spelunker game, Cops and Robbers is played with perfect information, so that at any time each of the agents knows the location of all others. A fixed number of cops take up positions on vertices of a connected graph and a robber then starts on any unoccupied vertex. The cops and the robber take turns: at his turn the robber may move to any adjacent vertex or remain where he is, and at their turn all cops simultaneously make moves of this form. The cops win if at any point one of them reaches the robber's location. On a particular graph G the question is whether a given number of cops have a strategy which is guaranteed to win, or whether there is a strategy for the robber which will allow him to evade capture indefinitely. The *cop number* of a graph is the minimum number of cops that can guarantee to catch the robber.

Early results on this game include those obtained by Nowakowski and Winkler [10], who categorised the graphs of cop number 1, and Aigner and Fromme [1], who showed that every planar graph has cop number at most 3. An important open problem is Meyniel's conjecture, published by Frankl [5], that the cop number of any n-vertex connected graph is at most

* Corresponding author.
E-mail addresses: j.haslegrave@cantab.net (J. Haslegrave), rabjohnson@gmail.com (R.A.B. Johnson), sk629@cam.ac.uk (S. Koch).

http://dx.doi.org/10.1016/j.disc.2016.05.024
0012-365X/

$O(\sqrt{n})$—this has been shown to be true up to a $\log(n)$ factor for random graphs by Bollobás, Kun and Leader [2], following which Łuczak and Prałat improved the error term [9]. More recently several variations on the game have been analysed by Clarke and Nowakowski (e.g. [4]).

In this paper we consider the Robber Locating game, introduced in a slightly different form by Seager [13] and further studied by Carraher, Choi, Delcourt, Erickson and West [3]. Like the Lost Spelunker game, the focus is on locating a hidden target, but, like Cops and Robbers, the target is a robber who moves around the vertices in discrete steps. There is a single cop, who is not on the graph but can probe vertices and receive information about how far away the robber is (in terms of the normal graph distance) from the vertex probed. For ease of reading we shall refer to the cop as female and the robber as male. The robber initially occupies any vertex. Each round consists of a move for the robber, in which he may move to an adjacent vertex or stay where he is, followed by a probe of any vertex by the cop. The cop then wins immediately if she is able to determine the robber's current location from the results of that probe and previous ones. This game may be viewed as a variant of the Sequential Locating game, also studied by Seager [14], with the difference between the two being that in the Robber Locating game the target can move about the graph; in both games the choice of probe made may depend on the results of previous probes. If we instead require all the probes to be chosen at once (with a stationary target), we recover the Graph Locating game, independently introduced by Slater [16] and by Harary and Melter [6].

In games with a stationary target, the searcher can guarantee to win eventually, simply by probing every vertex, and the natural question is the minimum number of probes required to guarantee victory on a given graph G. For the Graph Locating game, this is the *metric dimension* of G, written $\mu(G)$. In the Robber Locating game, by contrast, it is not necessarily true that the cop can guarantee to win in any number of probes. Consequently the primary question in this setting is whether, for a given graph G, the cop can guarantee victory in bounded time on G, or equivalently whether she can catch a robber who has full knowledge of her strategy. We say that a graph is *locatable* if she can do this and *non-locatable* otherwise.

In the game as introduced by Seager there was an additional rule that the robber cannot move to the vertex probed in the previous round (the *no-backtrack condition*). Carraher et al. considered the game without this restriction, as do we, and Seager also considered the version without the no-backtrack rule for trees [15]. A similar game in which the searcher wins only if she probes the current location of the target and receives no information otherwise, but the target must move at each turn, was recently analysed by one of the authors [7].

The main result of Carraher et al. [3] is that for any graph G a sufficiently large equal-length subdivision of G is locatable. Formally, write $G^{1/m}$ for the graph obtained by replacing each edge of G by a path of length m, adding $m-1$ new vertices for each such path. Carraher et al. proved that $G^{1/m}$ is locatable whenever $m \geqslant \min\{|V(G)|, 1+\max\{\mu(G)+2^{\mu(G)}, \Delta(G)\}\}$. In most graphs this bound is simply $|V(G)|$, and they conjectured that this was best possible for complete graphs, i.e. that $K_n^{1/m}$ is locatable if and only if $m \geqslant n$. The present authors [8] showed that in fact $K_n^{1/m}$ is locatable if and only if $m \geqslant n/2$, for every $n \geqslant 11$. In this paper we show that the same improvement may be obtained in general: provided $|V(G)| \geqslant 23$, $G^{1/m}$ is locatable whenever $m \geqslant |V(G)|/2$. This bound is best possible, since $K_n^{1/m}$ is not locatable if $m = (n-1)/2$, and some lower bound on $|V(G)|$ is required for it to hold, since $K_{10}^{1/5}$ is not locatable [8]. These results, and those of Carraher et al., fundamentally depend on taking equal-length subdivisions; in the final section of this paper we show that an unequal subdivision is also locatable provided every edge is subdivided by at least a certain amount.

2. Subdivisions and maximal matchings

Recall that $G^{1/m}$ is the graph obtained by replacing each edge of G with a path of length m through new vertices. Each such path is called a *thread*, and a *branch vertex* in $G^{1/m}$ is a vertex that corresponds to a vertex of G. We write $u \cdots v$ for the thread between branch vertices u and v. We use "a vertex on $u \cdots v$" to mean any of the $m+1$ vertices of the thread, but "a vertex inside $u \cdots v$" excludes u and v.

Our basic strategy to locate the robber on sufficiently large equal-length subdivisions of G is to ensure the following.

(1) Whenever the robber is at a branch vertex, the probe we make reveals that fact.
(2) If the robber ever spends r turns without visiting a branch vertex, we establish which thread he is inside and then can win on the next turn, where r depends only on G.
(3) If, when the robber visits a branch vertex, there is more than one possibility for his location, we can ensure that by the next time he is at a branch vertex we reduce the set of possibilities to a simpler set.

(1) and (2) above are sufficient to ensure that $G^{1/m}$ is locatable for sufficiently large m, since if $m > r$ the robber can only ever visit one branch vertex without being caught, and we can eventually find which one it is. For smaller m the robber may be able to visit several branch vertices, so to get a better bound we need a way to progressively reduce the possibilities as in (3). This reduction is not necessarily to a smaller set but we ensure that only a bounded number of reductions can occur (and, by (2), each takes bounded time) before we reach a singleton set. Note that the reduction occurs by the next time the robber is at a branch vertex, even if this is because he stays at his current branch vertex until the next turn.

To get a bound of close to $|V(G)|/2$, roughly speaking, our strategy is that in between the robber's visits to branch vertices we aim to eliminate possible destinations in pairs. To do this we must probe inside threads, aiming to eliminate both ends of the thread. This approach is simplified in the case of complete graphs, analysed extensively in [8], by the knowledge that any pair of branch vertices has a thread between them. In the general case we need some knowledge of the structure of G.