Discrete Mathematics 341 (2018) 1032–1035

Contents lists available at ScienceDirect

Discrete Mathematics

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

Note

# The cat and the noisy mouse

Dieter Rautenbach *, Moritz Schneider

*Institut für Optimierung und Operations Research, Universität Ulm, Ulm, Germany*

ARTICLE INFO

*Article history:*
Received 8 July 2017
Received in revised form 9 January 2018
Accepted 9 January 2018
Available online 29 January 2018

*Keywords:*
Pursuit and evasion game

ABSTRACT

We consider a variant of a pursuit and evasion game studied independently by Britnell and Wildon as well as Haslegrave. In their game, a cat has to catch an invisible mouse that moves along the edges of some graph $G$. In our version, the cat receives partial information about its distance to the mouse, and we show that the cat has a winning strategy if and only if $G$ is a forest. Seager proposed a similar game with complete distance information whose rules cause some small yet important differences to the game we consider.



## 1. Introduction

We consider a variant of a pursuit and evasion game independently studied by Britnell and Wildon [2], Komarov and Winkler [6], and Haslegrave [4]. The game is played by two players, called the *cat* and the *mouse* in [4], on a finite graph $G$ known to both, and proceeds in discrete rounds numbered by positive integers. In round $i$, the cat chooses a vertex $c_i$ of $G$ and also the mouse chooses a vertex $m_i$ of $G$. While the choice of the cat is unrestricted among all vertices of $G$, the mouse must move along an edge of $G$, more precisely, the vertices $m_i$ and $m_{i+1}$ have to be distinct yet adjacent for every $i$. The cat *catches* the mouse, and thereby wins the game, in round $i$ as soon as $c_i = m_i$. The mouse wins the game if it can avoid being caught indefinitely, more precisely, the mouse wins if for every positive integer $k$, and every sequence $c_1, \ldots, c_k$ of vertices of $G$, there is a sequence $m_1, \ldots, m_k$ of vertices of $G$ such that $m_{i+1}$ is a neighbor of $m_i$ for every $i \in [k-1]$, and $c_i \neq m_i$ for every $i \in [k]$. As shown by Britnell and Wildon [2], Komarov and Winkler [6], and Haslegrave [4], the cat has a winning strategy in their game played on $G$ if and only if $G$ is a forest that does not contain the tree $T^*$ shown in Fig. 1.

The key feature of their game is the invisibility of the mouse, that is, the cat does not know the position of the mouse until it actually catches it. In the variant that we consider here, the cat receives some information about the distance $\text{dist}_G(c_i, m_i)$ in $G$ between $c_i$ and $m_i$ in every round $i$; what information exactly is specified at the beginning of the next section. Figuratively, the cat estimates its current distance to the mouse using the noise made by the mouse's movement.

Seager [7] introduced a similar game, where the cat, called *cop*, knows its exact distance to the mouse, called *robber*, yet the objective of the cat is to determine the exact position of the mouse at least once. This weaker objective makes the game easier for the cat; knowing where a robber is at some point in time is easier than catching him. There are further differences in Seager's game. The mouse is forbidden to move to the previous position of the cat but it is allowed not to move at all, that is, $m_{i+1}$ is an element of $(\{m_i\} \cup N_G(m_i)) \setminus \{c_i\}$. Allowing $m_{i+1} = m_i$ but not requiring $m_{i+1} \neq c_i$ within Haslegrave's game, leads to a trivial situation where any distance information is useless, because the mouse can avoid capture indefinitely as soon as the graph has at least one edge. Also, requiring $m_{i+1} \in N_G(m_i) \setminus \{c_i\}$ within Haslegrave's game, changes the game considerably; if this modified game is played on $T^*$, and the cat plays $(c_1, \ldots, c_7) = (x, v_1, x, v_2, x, v_3, x)$, then, at some point, the mouse is either caught or has no legal move. In fact, since every second vertex played by the cat is $x$, in order to avoid capture, the mouse must stay in $\{u_j, v_j, w_j\}$ for some $j \in [3]$. Now, when the cat plays $c_i = v_j$, the mouse $m_i$ is in $\{u_j, w_j\}$.

* Corresponding author.
*E-mail addresses:* dieter.rautenbach@uni-ulm.de (D. Rautenbach), moritz-1.schneider@uni-ulm.de (M. Schneider).

https://doi.org/10.1016/j.disc.2018.01.009
0012-365X/

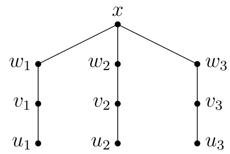


**Fig. 1.** The tree $T^*$.

Furthermore, if the mouse $m_i$ is at $w_j$, then it will be caught on $x$ in the next round, and if it is at $u_j$, then it has no legal move to play. It is unknown what happens to Haslegrave's game if both conditions are imposed, that is, $m_{i+1} \in (\{m_i\} \cup N_G(m_i)) \setminus \{c_i\}$. Altogether, Seager's game shows some small yet important differences to the game that we consider here. For improved bounds, better strategies, and variants of Seager's game refer to [1,5,8].

As shown in [3,7], the cop wins Seager's game on many graphs that contain cycles, while, in our game, the mouse wins as soon as there is any cycle $C$. In fact, the mouse's strategy is to stay on $C$ throughout the game, and, since every vertex of $C$ has two neighbors on $C$, the cat will never be able to guarantee catching the mouse; if the cat chooses a neighbor $c_{i+1}$ of $m_i$, then $m_{i+1}$ can be chosen to be the neighbor of $m_i$ on $C$ that is distinct from $c_{i+1}$.

Our main result is that the cat has a winning strategy on $G$ exactly if $G$ is a forest.

## 2. Result

We consider Haslegrave's game with partial distance information for the cat. In what follows, let $d_i$ denote the distance $\text{dist}_G(c_i, m_i)$ for every positive integer $i$. In the original game, the cat learns in every round $i$, whether $d_i$ is "0" or "*at least* 1". In the version that we consider, the cat receives the following information in every round $i$:

- whether $d_i$ is "0" or "1" or "*at least* 2", and, additionally,
- for $i \geq 2$, whether $d_i$ is "*at most* $d_{i-1}$" or "*bigger than* $d_{i-1}$".

In order to show that the cat has a winning strategy on forests, it suffices to describe a winning strategy on trees; the cat can apply this strategy to the different components of any forest, one by one, and will eventually catch the mouse. As we see below, the maximum number of rounds that the mouse can evade capture when the cat plays our strategy on a tree is bounded in terms of the order of the tree. Therefore, if the cat plays on some component of a forest, and this number of rounds is reached without capture, then the mouse must be in another component.

Let $T$ be a tree. The cat chooses an arbitrary vertex $r$ of $T$ as its *root*. For every vertex $u$ of $T$, let $V_u$ denote the set of vertices that contains $u$ and all its descendants.

We consider the game as a sequence of *transitions* from some round $i$ to some later round $i+j$. In order to show that the cat has a winning strategy, we argue that until it catches the mouse, it can ensure that all these transitions are of one of four different simple types, which allows the cat to make progress. See Fig. 2 for an illustration.

The situation at the beginning of a transition starting with round $i$ is characterized by

- a *reference vertex* $r_i$, and
- two sets $X_i$ and $Y_i$ of vertices.

The set $X_i$ contains all vertices that are not a descendant of $r_i$, which includes $r_i$ itself, as well as all vertices in sets $V_u$ for some children $u$ of $r_i$. Let the children of $r_i$ that do not belong to $X_i$ be the leaves $v_1, \ldots, v_k$ and the non-leaves $w_1, \ldots, w_\ell$. The set $Y_i$ is the possibly empty union of sets $V_u$ for some children $u$ of $w_1$. Let the children of $w_1$ that do not belong to $Y_i$ be the vertices $x_1, \ldots, x_p$. Note that $k$, $\ell$, and $p$ are all non-negative integers that might be 0, and that the cat is free to choose the ordering of the mentioned vertices.

In the round $i$ starting the current transition, the cat chooses $c_i$ equal to the reference vertex $r_i$. Furthermore, exploiting knowledge acquired in earlier rounds, the cat is sure that $m_i$ does not lie in $X_i \cup Y_i$. For $i = 1$, that is, in the very first round, the reference vertex $r_1$ is the root $r$ of $T$, $X_1$ contains only $r_1$, and $Y_1$ is empty. Note that either $c_1 = r_1 = m_1$, in which case the cat wins right away, or $m_1 \notin X_1 \cup Y_1$.

As said above, we consider four types of transitions from some round $i$ to some later round $i+j$ that are characterized by the following conditions.

- **Type 1**
  $d_i = 1, j = 1$, $r_{i+1}$ equals $r_i$, $X_{i+1}$ equals $X_i$, $Y_{i+1} = \emptyset$, and $d_{i+1} \geq 2$.
- **Type 2**
  $d_i \geq 2, j = 1$, $r_{i+1}$ is a child of $r_i$, $X_i$ is a proper subset of $X_{i+1}$, and $Y_{i+1} = \emptyset$.
- **Type 3**
  $d_i \geq 2, j \in \{1, 2\}$, $r_{i+j}$ equals $r_i$, $X_i$ is a proper subset of $X_{i+j}$, and $Y_{i+j} = \emptyset$.