Discussiones Mathematicae
Graph Theory xx (xxxx) 1–16
doi:10.7151/dmgt.2244

GUARDING A SUBGRAPH AS A TOOL IN PURSUIT-EVASION GAMES

Drago Bokal

Faculty of Natural Sciences and Mathematics
University of Maribor
Maribor, Slovenia

e-mail: drago.bokal@um.si

and

Janja Jerebic

Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia

e-mail: janja.jerebic@um.si

Abstract

Pursuit-evasion games study the number of cops needed to capture the robber in a game played on a graph, in which the cops and the robber move alternatively to neighbouring vertices, and the robber is captured if a cop steps on the vertex the robber is in. A common tool in analyzing this cop number of a graph is a cop moving along a shortest path in a graph, thus preventing the robber to step onto this path. We generalize this approach by introducing a shadow of the robber, the maximal set of vertices from which the cop parries the protected subgraph. In this context, the robber becomes an intruder and the cop becomes the guard. We show that the shadow can be computed in polynomial time, implying polynomial time algorithms for computing both a successful guard as well as a successful intruder, whichever exists. Furthermore, we show that shadow function generalizes the concept of graph retractions. In some cases, this implies a polynomially computable certification of the negative answer to the NP-complete problem of existence of a retraction to a given subgraph.

Keywords: pursuit-evasion game, graph searching, guarding, shadow function, graph retraction.

2010 Mathematics Subject Classification: 05C57, 05C60.

1. Introduction

A pursuit-evasion game in a reflexive graph G is in its most general form played as follows: an intruder is located somewhere in G as well as several searchers. All of them move along the edges of the graph and the aim of the searchers is to capture the intruder by stepping on the same vertex.

According to [2], the pursuit-evasion games in graphs were first considered by Parsons [14] as a model of searching for a lost spelunker in a system of caves. In his model, both the searchers and the intruder could be located either at the vertices or at the edges of the graph and the searchers have no information about the position of the intruder. Nowakowski and Winkler [12] introduced another model, in which the players may be located at the vertices of the graph only and the complete information about the position of the players is available. They characterized the graphs in which a single searcher has a winning strategy, i.e., can always capture the intruder (cop-win graphs). An algorithmic characterisation of finite cop-win digraphs is given in [10].

Several other results on the latter model were published in sequel. The following ones are intimately related to the present paper. Aigner and Fromme [1] showed that three searchers always capture the intruder in a planar graph. This result was generalized by Quilliot [13] who showed that $2g + 3$ searchers have a winning strategy in any graph of genus g. This bound was improved by Schröder [16] to $\left\lfloor \frac{3}{2}g \right\rfloor + 3$. Additionally, Andreae [3] showed that if a connected graph G has no connected minor H, then $|E_H|$ searchers have a winning strategy (in fact, the exact presented bound is even better). Some recent variations were introduced in [9], and a most recent survey of the graph searching and related problems has been published by Bonato and Yang [4].

The results just mentioned have the following technique of guarding in common: at every stage of the game, the searchers are able to capture the intruder instantly after he enters a particular subgraph of the graph, and gradually they manage to extend this guarded territory over the whole graph and thus capture the intruder. However, the technique of guarding was restricted to the use of the following lemma, which initially appeared in [1].

Lemma 1. *Let G be a graph with vertices u and v and P a shortest path between u and v in G. If there are at least two searchers in the game, then after a finite number of moves one of them can guard P from the intruder.*

The idea of this lemma was extended by Chiniforooshan [5], who used the approach of five cops guarding a minimum distance caterpillar to show that cop-number of any n-vertex graph is at most $O\left(\frac{n}{\lg n}\right)$, where a *minimum distance caterpillar* between $u \in V_G$ and $v \in V_G$ is a subgraph H of G such that (1) H is a tree, (2) the path P between u and v in H is a shortest path between u and v in G, and (3) every vertex of H has an edge in H to a vertex in P.

The aim of this paper is to investigate the concept of guarding a subgraph. Besides helping to search the graph, there is independent interest in applications where the whole graph is too large to be searched and the authority is satisfied by fulfilling a weaker goal of preventing the intruders from entering a particular region of the graph. In addition, we establish that guarding a subgraph leads to polynomially computable generalizations of the graph retraction problem, which is not only NP-complete for many specific graphs, but also difficult to classify for which it is NP-complete [7].

During the research presented in this paper, a different guarding game was studied by several groups of authors [8, 9, 11, 15]. The complexity of the cop and robber guarding game was studied by Šámal *et al.* [18, 17]. Contrary to our approach, they restrict the cops to move within the guarded graph, which renders that variant of the game less relevant for the original graph searching game, and we propose to distinguish the two variants as the *regional guarding game*, where the cops are restricted to the region they are guarding, and the *territorial guarding game* where the guards can move over all the territory. A natural generalization of the two games is a *generalized guarding game*, where edges of the graph are used by either the intruders, the cops, or both. We emphasize that territorial guarding game was designed to generalize the guarding a shortest path approach in graph searching, which we find a relevant association of the two problems. However, the two guarding games have much in common, and we conjecture that most of the results about either regional or territorial guarding are valid in the context of the other game as well although they need to be proven separately.

The rest of the paper is organized as follows. The territorial guarding game is introduced formally as an algorithm in Section 2. A characterization when one guard against several intruders has a winning strategy is provided in Section 3; in particular, see Theorem 6. The existence of a unique inclusion-maximal shadow function for every graph G and its subgraph H is established and a polynomial algorithm to find the shadow function is given in Section 4. We apply the shadow function to find a successful guard or a successful intruder, whichever exists. In Section 5, we show that the shadow is a polynomially computable generalization of a retraction to a subgraph, existence of which is in general an NP complete problem. This allows for polynomial certifications of negative answers to the question of existence of a retraction to a given subgraph. The paper closes with observations how the game with several intruders versus several guards can be transformed to the game of a single intruder and a single guard. Aligned with the results of [10], our algorithms can be applied to polynomially solve the problem of guarding a subgraph H of a graph G with k-guards against i-intruders for any fixed k and i, implying fixed-parameter-tractability of the problem of guarding a subgraph with k guards against i intruders.

2. Territorial Guarding a Subgraph – a Formalization

There are several possibilities to extract the concept of guarding from Lemma 1. First we isolate the context in which we want to apply the resulting concept: guards and intruders are located in a reflexive graph G (all the graphs we consider are reflexive, i.e., there is a loop at every vertex which in our formalizatioin allows a player to stay at the occupied vertex), and there is a subgraph H, such that whenever an intruder locates himself on a vertex v of H, there is a guard that can step onto v in the next move. With this goal in mind, we analyze the two possibilities of how the game is started: either the guards or the intruders position themselves first on the graph.

Let us say the guards have the initiative of stepping onto G first. In the case of a single intruder, they need to position themselves in such a way that the graph H is dominated by the vertices of their locations. Having accomplished this, they never need to move, so the game reduces to finding an H-dominating subset of vertices of G. In the case of non-simultaneous positioning of several intruders, the game reduces to variants of protection in graphs, as introduced by Cockayne *et al.* [6], an approach that does not address the dynamics of pursuit-evasion games.

Thus we will focus our attention to the case when the intruders have the initiative of choosing their location on G first. If the guards can afterwards choose just any location, then they step on the intruders and the game is again trivial. But if we constrain the set of vertices, i.e., if the guards can choose to move just to vertices of $S \subsetneq V$, we obtain a nontrivial game that adequately models our problem: intruders are located at some vertices of G, guards at some other, and they start to move alternatively with the aim of the first to step unpunished on a vertex of H and the aim of the others to prevent it. We proceed with a formalization of this game.

Let a graph G, its subgraph H and a set $S \subseteq V_G$ of vertices, denoted by *guard posts*, be given. Let I be the set of intruders, $i = |I|$ their number, Γ the set of guards, and $g = |\Gamma|$ their number. We will model them as memoryless strategies, which we will justify in Proposition 2 at the end of this section. The *territorial guarding game* is played as follows: initially, each *intruder* X_j chooses a vertex v_j in G. After all the intruders have located themselves to $v = (v_1, \dots, v_i)$, each *guard* Y_j chooses a *guard post* $\varphi_j(v) \in S$ for $\varphi_Y : V_G^i \to S$. Now let $v^{(0)} = (v_1, \dots, v_i, \varphi_1(v), \dots, \varphi_g(v))$ be the thus obtained vector of combined positions of all i intruders and all g guards. They continue to move alternatively: first each intruder X_j moves to some neighboring vertex of his position, updating the position vector of the game from $v^{(0)}$ to $v^{(1)}$. Formally, for each $j \in \{1, \dots, i\}$ intruder $X_j \in I$ is a pair (v_j, ι_j), where $v_j \in V_G$ and $\iota_j : V_G \times (V_G \cup \{\infty\})^{i-1} \times V_G^g \to V_G$, such that $\iota_j(x, x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_i, y_1, \dots, y_g)$ is a vertex in G adjacent to x.

First coordinate of the vector $(x, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_i, y_1, \ldots, y_g)$ indicates the position of the intruder X_j, while the other coordinates determine the positions of all remaining $i-1$ intruders and g guards at a given moment of the game. After all the intruders have moved each guard Y_j moves to some neighboring vertex of his position, updating the position vector of the game from $v^{(1)}$ to $v^{(2)}$. Formally, for each $j \in \{1, \ldots, g\}$, a guard $Y_j \in \Gamma$ is a pair $Y_j = (\varphi_j, \gamma_j)$, where φ_j was defined earlier as the initial position of the guard Y_j according to the initial positions of the intruders, and $\gamma_j : V_G \times (V_G \cup \{\infty\})^i \times V_G^{g-1} \to V_G$, such that $\gamma_j(y, x_1, \ldots, x_i, y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_g)$ is a vertex in G adjacent to y. First coordinate of the vector $(y, x_1, \ldots, x_i, y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_g)$ indicates the position of the guard Y_j, while the other coordinates determine the positions of all remaining $g-1$ guards and i intruders at a given moment of the game. A guard can *capture* an intruder by stepping on his vertex, and consequently the intruder is eliminated from the game. In our notation, this is modelled by setting intruder's position to ∞. The aim of the intruders is to enter the specified subgraph H of G without being captured immediately afterwards. Schematically, the game is played as presented by Algorithm 1.

Algorithm 1 The territorial guarding game

```
Set $t := 0$, $n = |V_G|$, $\bar{n} = |V_G \setminus V_H| + 1$.
Set $I$ the set of all $i$ intruders, $\Gamma$ the set of all $g$ guards.
For each intruder $X_j = (v_j, \iota_j) \in I$, set $x_j := v_j$.
For each guard $Y_j = (\varphi_j, \gamma_j) \in \Gamma$, set $y_j := \varphi_j(x_1, \ldots, x_i)$.
**while** $\neg\big(I = \emptyset$ or $\exists j : \big(x_j \in V_H \wedge \forall j' : x_j \neq y_{j'}\big)$ **do**
  Set $\mathcal{C} = \{X_j \mid \exists y_\ell : x_j = y_\ell\}$.
  For each intruder $X_j \in \mathcal{C}$, set $x_j := \infty$.
  Set $I := I \setminus \mathcal{C}$.
  Set $t := t + 1$.
  For each intruder $X_j = (v_j, \iota_j) \in I$,
    set $x'_j := \iota_j(x_j, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_i, y_1, \ldots, y_g)$.
  For each intruder $X_j \in I$, set $x_j := x'_j$.
  For each guard $Y_j = (\varphi_j, \gamma_j) \in \Gamma$,
    set $y'_j := \gamma_j(y_j, x_1, \ldots, x_i, y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_g)$.
  For each guard $Y_j$, set $y_j := y'_j$.
  If $t > n^g \bar{n}^i$, then guards have won. The game will be played indefinitely (see
  Proposition 2).
**end while**
If $I = \emptyset$, then the guards have won,
otherwise, the intruders have won.
```

If the guards can indefinitely prevent the intruders from entering H or capture all the intruders, then the guards *guard* H, and they win the game. Suppose the game is finished in finite time, then the final value of t is the *length* of the game. A set $\mathcal{G}$ of guards is *i-successful* in H, if they guard H from any set $\mathcal{I}$ of less than i intruders. In this case, $|\mathcal{G}|$ guards can *guard* H. A set $\mathcal{I}$ of intruders is *g-successful* in H, if at least one intruder enters H for any set $\mathcal{G}$ of less than g guards. We say that $|\mathcal{I}|$ intruders can *enter* H.

We remarked earlier that the guards and intruders are represented by memoryless strategies. One may remark that they are limited as they only react to the current position of the game. Thus they may not capture all the possible strategies. However, model extensions with memory are not necessary as repeating a position in the game does not alter the chances of either party to win: the game is simply started anew and the winning party could apply the winning moves of the second game already the first time a position occurred (cf. also Proposition 2). Thus the aim of the intruder is to force the guards into a new position by every move. This idea is essential to the following proposition which provides a general upper bound to the length of the game. Note that there are many of those positions in which the intruders may be all captured, hence this general bound could be improved in specific cases.

Proposition 2. *Let G be an arbitrary graph, H its arbitrary subgraph. Let $n = |V_G|$, $\bar{n} = |V_G \setminus V_H| + 1$, and let $i = |I|$ be the prescribed number of intruders and $g = |\Gamma|$ the number of guards. There exists a set of i intruders into H that win against any g guards, if and only if there exists a winning set of i intruders against g guards, such that the intruders enter H in at most $2(n^g\bar{n}^i) + 1$ moves, counting also the initial two moves of positioning the intruders and guards.*

Proof. Let us augment the graph G with a vertex ∞ outside H, having the property that each captured intruder moves to this vertex and stays there and the guards do not move there. Let $P_t = \left(x_1^t, \ldots, x_i^t, y_1^t, \ldots, y_g^t\right)$ be the position of the game after t moves.

Let t_0 be the shortest game in which the intruders enter H. As guards could occupy any vertex of G, and intruders could only occupy vertices of $V(G) \setminus V(H)$ or ∞, there are $\bar{n}^i n^g$ different positions in the game that do not constitute intruder's victory. At each of them, either the intruders or the guards move first. Thus, if the game lasted for $2(n^g\bar{n}^i)$ or more moves, some position is repeated. Hence, if $t_0 > 2(n^g\bar{n}^i)$, we have a contradiction to t_0 being a shortest game for the intruders to enter H. ∎

The rest of the paper, except Sections 6 and 7, is devoted to studying the game in the case of a single intruder and a single guard. We present a polynomial algorithm that computes either a winning guard or a winning intruder, whichever exists.

3. Existence of a Successful Guard

Let G be a connected graph, H its subgraph, and $S \subseteq V_G$ a set of guard posts. All of them are assumed to be fixed throughout the section.

Let $W = x_0 \cdots x_l$ be a walk in G and $Y = (\varphi, \gamma)$ be a guard in G. If the intruder follows the walk W, then the position y_i of the guard Y after i moves is traced with $y_0(W,Y) := \varphi(x_0)$ and $y_i(W,Y) := \gamma(x_i, y_{i-1})$. All possible positions of Y when the intruder has arrived to $x \in V_G$ are captured by the set

$$\Phi_Y(x) := \big\{ y_l(W,Y) \mid W = x_1 \cdots x_{l-1} x \big\}$$

where W runs over all finite walks in $G - H$ which end in x.

Assume that the walk W starts in x and ends in $x' \in V_H$. A vertex $y \in V_G$ is said to *parry* $x \in V_G$ *against the walk* W, if there exists some walk W' of length $|W|$ from y to x'. If $y \in V_G$ parries $x \in V_G$ against any walk W from x to any $x' \in V_H$, we simply say that y *parries* x, and write $y \succeq x$.

Proposition 3. *Let G be a connected graph and H its subgraph. The relation $\succeq$ defined above is a preorder, i.e., it is reflexive and transitive.*

Proof. Let x be an arbitrary vertex of G, x' an arbitrary vertex of H and W any walk from x to x'. By definition, x parry x against W, if there exists some walk W' of length $|W|$ from x to x'. Such a walk exists, namely $W' = W$. It follows that x parries x. Hence, $\succeq$ is reflexive.

Let x, y and z be any vertices of G with the property $y \succeq x$ and $z \succeq y$. Let W be any walk in G from x to arbitrary chosen vertex x' of H. $y \succeq x$ implies that there exists a walk W' of length $|W|$ from y to x'. Furtheremore, $z \succeq y$ implies that there exists a walk W'' of length $|W'| = |W|$ from z to x'. We can conclude that for any walk W in G from x to x' there exists a walk W'' of length $|W|$ from z to x'. Hence, $\succeq$ is transitive. ■

Let $Q : V_G \to 2^{V_G}$ be a function with

$$Q(x) := \{ y \in V_G \mid \forall z \in V_H : d_G(z,y) \leq d_{G-E_H}(z,x) \}.$$

This is precisely the set of vertices y which parry x against any shortest path from x to V_H.

For a set $A \subseteq V_G$ we define $A^+ := A \cup N_G(A)$, where $N_G(A)$ is the G-neighborhood of A. Using the above tools, it is possible to identify the strategies of guarding H.

Lemma 4. *Suppose that $Y = (\varphi, \gamma)$ is a successful guard. Then, for any $x \in V_G$, we have $\Phi_Y(x) \subseteq Q(x)$ and, for any $xx' \in E_G \setminus E_H$, we have $\Phi_Y(x) \subseteq \Phi_Y^+(x')$.*

Proof. Assume that $y \in \Phi_Y(x) \setminus Q(x)$ and let $W = x_1 \cdots x_{l-1}x$ be a walk with $y = y_l(W, Y)$. Let $X = (x_1, \iota_W)$ be an intruder starting at x_1 and following the walk W. After X steps on vertex x, the guard steps on vertex y. As $y \notin Q(x)$, there exists $x' \in V_H$ with $d_G(y, x') > d_{G-E_H}(x, x')$. If the intruder follows some shortest xx' path, he can enter H, a contradiction.

Assume that for $xx' \in E_G \setminus E_H$, $y \in \Phi_Y(x) \setminus \Phi_Y^+(x')$ and let $W = x_1 \cdots x_{l-1}x$, be a walk with $y = y_l(W, Y)$. Let $W' = Wx'$ and let $X = (x_1, \iota_{W'})$ be an intruder starting at x_1 and following the walk W'. Just before he steps on x', the guard is in the vertex y. As $y \notin \Phi_Y^+(x')$, the successful guard cannot step into $\Phi_Y(x')$, a contradiction. ■

Lemma 5. *Let a function $\Phi : V_G \to 2^{V_G} \setminus \{\emptyset\}$ be such that both $\Phi(x) \cap S \neq \emptyset$, $\Phi(x) \subseteq Q(x)$ for any $x \in V_G$, and $\Phi(x) \subseteq \Phi^+(x')$ for any $xx' \in E_G \setminus E_H$. Then there exists a successful guard of H in G.*

Proof. Choose $\varphi(x) \in \Phi(x) \cap S$ and for $y, x \in V_G$ choose $\gamma(x, y) \in \{y\}^+ \cap \Phi(x)$, if the latter set is nonempty, otherwise choose $\gamma(x, y)$ arbitrarily. We claim that the guard $Y = (\varphi, \gamma)$ is a successful guard of H in G. As $Q(x) = \{x\}$ for $x \in V_H$, it is enough to prove that the guard can always step into $\Phi(x)$, where x is the position of the intruder.

This is proved by induction on the number t of moves in the game. If $t = 0$, then $y = \varphi(x) \in \Phi(x)$. Suppose that $y \in \Phi(x)$ at time t and the intruder moves from x to x'. Thus $xx' \in E_G \setminus E_H$, implying $y \in \Phi(x')^+$ and y has a neighbor $y' \in \{y\}^+ \cap \Phi(x')$. ■

Combining Lemmas 4 and 5 yields.

Theorem 6. *Let G be a connected graph, H its subgraph and S a set of guard posts. Then H can be guarded in G with a single guard against a single intruder if and only if there exists a function $\Phi : V_G \to 2^{V_G}$ such that*

(i) *$\Phi(x) \cap S \neq \emptyset$, $\Phi(x) \subseteq Q(x)$ for any $x \in V_G$ and*

(ii) *$\Phi(x) \subseteq \Phi^+(x')$ for any $xx' \in E_G \setminus E_H$.*

4. The Shadow of the Intruder

The inclusion-maximal function Φ satisfying conditions (i) and (ii) of Theorem 6 is called the *shadow function* of H in G, which we assume to be fixed throughout this section. Intuitively, the guard will be able to protect the subgraph H, if he is able to step into the shadow of the intruder. Thus, the subgraph can be guarded, if there is a guard post in the shadow of every vertex of the graph G. On the other hand, the intruder shall avoid casting his shadow into the neighborhood of the guard — as soon as he does, the guard will be able to step into it and prevent him from entering H. This definition is justified by Theorem 11, which establishes that there is a unique inclusion-maximal shadow function for every G and its subgraph H.

A polynomial algorithm to find the shadow function Φ is designed in what follows. For $d \in [0, \mathrm{diam}(G)]$ and $x \in V_G$ we define $D(x,d)$ as the set of all vertices in G on distance d from x and $\bar{D}(x,d)$ as the set of all verices in G on distance less or equal d from x. A variable d_z contains a distance $d_{G-E_H}(z,x)$ and a set V_x contains vertices of H connected to x in $G - E_H$.

Algorithm 2 Given G and its subgraph H, compute the function Q.

```
Using a modification of Floyd-Warshall algorithm, compute the sets
$D(x,d) := \{z \in V_G \mid d_G(z,x) = d\}$ (for $x \in V_G$, $d \in [0, \mathrm{diam}(G)]$).
for each $x \in V_G$ do
  For $d \in [0, \mathrm{diam}(G)]$, set $\bar{D}(x,d) := \bigcup_{0 \le i \le d} D(x,i)$.
  Set $Q(x) = V_G$.
end for
for each $x \in V_G$ do
  Clear all marks.
  Let $q$ be an empty queue.
  Set $d_x = 0$.
  Append $x$ to $q$.
  while $q$ not empty do
    Get $z$ from queue $q$.
    Mark $z$.
    if $z \in V_H$ then
      $Q(x) := Q(x) \cap \bar{D}(z, d_z)$.
    else
      for each unmarked $y \in N_G(z)$ do
        $d_y := d_z + 1$.
        Append $y$ to $q$.
      end for
    end if
  end while
end for
```

Proposition 7. *Algorithm 2 correctly computes the function Q in running time $O(n^2 m)$, where n is the number of vertices and m is the number of edges in G.*

Proof. Observe that

$$\begin{aligned} Q(x) &= \{y \in V_G \mid \forall z \in V_H : d_G(z,y) \le d_{G-E_H}(z,x)\} \\ &= \bigcap_{z \in V_H} \{y \in V_G \mid d_G(z,y) \le d_{G-E_H}(z,x)\} = \bigcap_{z \in V_x} \bar{D}(z, d_z). \end{aligned} \tag{1}$$

Here $V_x \subseteq V_H$ is the set of vertices in H, visited by the algorithm while processing vertex x in the for-loop. Note that if a vertex z of H is not visited by Algorithm 2, then $d_{G-E_H}(x,z) = \infty$ and $\{y \in V_G \mid d_G(z,y) \leq d_{G-E_H}(z,x)\} = V_G$. Correctness follows from (1) and the fact that the breadth-first-search in Algorithm 2 does not continue from the vertices of H.

The running time of Floyd-Warshall algorithm is $O(n^3)$. The sets $\bar{D}(x,d)$ can be computed in time $O(n^3)$. The main loop is executed n times. Every vertex enters the queue at most once, thus the while-loop is executed at most n^2 times. Computing the intersections needs at most $O(n^3)$ steps and the for-loop considers each edge of G at most twice, consuming the dominating computational time $O(n^2m)$. ■

Algorithm 3 Given G and its subgraph H, compute the function Φ of a successful guard.

```
Using Algorithm 2, compute the function $Q$ for $G$ and $H$.
For $x \in V_G$, set $\Phi(x) = Q(x)$.
Let $s$ be an empty stack.
**for** every edge $xx' \in E_G$ **do**
  Push $xx'$ to $s$.
  Push $x'x$ to $s$.
**end for**
**while** $s$ not empty **do**
  Pop $xx'$ from $s$.
  Set $T := \Phi(x) \cap \Phi(x')^+$.
  **if** $T \neq \Phi(x)$ **then**
    **for** every $G - H$ neighbor $u$ of $x$ **do**
      Push $ux$ to $s$.
    **end for**
    Set $\Phi(x) := T$.
  **end if**
**end while**
```

Lemma 8. *Let $\Phi : V_G \to 2^{V_G} \setminus \{\emptyset\}$ be such that $\Phi(x) \subseteq \Phi^+(x')$ for any $xx' \in E_G$. Then, to any walk $W = x_1 \cdots x_t$ in G and any vertex $y_1 \in \Phi(x_1)$, there corresponds a walk $W' = y_1 \cdots y_t$ with $y_i \in \Phi(x_i)$ for $i = 1, \ldots, t$.*

Proof. By induction on t, case $t = 1$ is trivial. Let $W_+ = x_1 \cdots x_t x_{t+1}$. By induction, there is a walk $W' = y_1 \cdots y_t$ with $y_i \in \Phi(x_i)$ for $i = 1, \ldots, t$. Now $x_t x_{t+1} \in E_G$ and $y_t \in \Phi(x_t) \subseteq \Phi^+(x_{t+1})$. Thus there is a vertex $y_{t+1} \in \Phi(x_{t+1})$ adjacent or equal to y_t and we can extend the walk W' to $W'_+ = W' y_{t+1}$. ■

Lemma 9. *Let $\Phi : V_G \to 2^{V_G}$ be the function, returned by Algorithm 3, and let $y \notin \Phi(x)$ for some $x, y \in V_G$. Then y does not parry x.*

Proof. The claim is proved by induction on the number p of edges popped from the stack when y was removed from $Q(x)$. If $p = 0$, then $y \notin Q(x)$ and the claim follows from Lemma 4.

Let y be removed from $Q(x)$ when p edges were popped from the stack and let $\Phi_p(v)$ denote the set $\Phi(v)$ at this stage of the algorithm. Then $y \notin \Phi_p^+(x')$ for some neighbor x' of x. Thus $y' \notin \Phi_{p-1}(x')$ for any $y' \in \{y\}^+$. None of y' parries x' by induction and the claim follows. ■

Lemma 10. *Let $\Phi : V_G \to 2^{V_G}$ be the function computed by Algorithm 3. Then $\Phi(x) \subseteq Q(x)$ for any $x \in V_G$, and $\Phi(x) \subseteq \Phi^+(x')$ for any $xx' \in E_G$. Moreover, if Φ' is any other function with these properties, then $\Phi'(x) \subseteq \Phi(x)$ for any $x \in V_G$.*

Proof. Clearly, $\Phi(x) \subseteq Q(x)$ for any $x \in V_G$. Assume that there is an edge $xx' \in E_G$, such that $\Phi(x) \not\subseteq \Phi^+(x')$. Then, there must have been a neighbor x'' of x', such that $\Phi(x')$ has changed into $\Phi(x') \cap \Phi^+(x'')$ after the edge xx' was popped from the stack. But then xx' was pushed on the stack again and $\Phi(x)$ was recomputed, contradicting the assumption.

Let Φ' be some other function with required properties and $y_1 \in \Phi'(x_1) \setminus \Phi(x_1)$. Let $W = x_1 \cdots x_t$ be any walk in $G - E_H$ with $x_t \in V_H$. By Lemma 8, there exists a walk $W' = y_1 \cdots y_t$ with $y_i \in \Phi'(x_i) \subseteq Q(x_i)$, and $y_t \in Q(x_t) = \{x_t\}$ due to $x_t \in V_H$. This implies that y_1 parries x_1, a contradiction to Lemma 9. ■

The following theorem captures the essence of the function Φ.

Theorem 11. *Let $\Phi : V_G \to 2^{V_G}$ be the function, returned by Algorithm 3, and let $x, y \in V_G$. Then y parries x if and only if $y \in \Phi(x)$.*

Proof. Suppose $y \in \Phi(x)$ and let $W = xx_1 \cdots x_t$ be any walk starting in x and ending in $x_t \in V_H$. By Lemmas 8 and 10, there is a walk $W' = yy_1 \cdots y_t$ of length $|W|$ with $y_i \in \Phi(x_i)$. As $x_t \in V_H$, we have $y_t \in \Phi(x_t) = \{x_t\}$ and the intruder is captured by W'. Thus y parries x.

The converse follows from Lemma 9. ■

Combining Theorem 6 with Lemma 10 we obtain the following characterization result with regards to the existence of a successful guard.

Theorem 12. *Let G be a graph, H its subgraph and S a set of guard posts. Then H can be guarded in G with a single guard against a single intruder if and only if Algorithm 3 produces a function Φ with $\Phi(x) \cap S \neq \emptyset$ for all $x \in V_G$.*

The successful guard can be reconstructed by applying the proof of Lemma 5.

Corollary 13. *Let G be a graph, H its subgraph and $S \subseteq V_G$ a set of guard posts. Existence of a successful guard of H can be verified using Algorithm* 3 *in time $O(nm^2)$, where $n = |V_G|$ and $m = |E_G|$.*

Proof. Computing Q takes $O(n^2m)$ by Proposition 7. As $Q(x) \subseteq V_G$, the set $\Phi(x)$ can change at most n times, thus any edge is pushed onto the stack at most $2n + 2$ times and the while loop is executed $O(nm)$ times. Computing the intersections inside the loop, as well as comparing the sets hence takes altogether at most $O(n^2m)$. The most time consuming part is computing $\Phi^+(x)$. One such operation takes $O(m)$, thus altogether they consume $O(nm^2)$ computational time.

The rest of the claim follows from Theorem 12. ■

For a set $A \subseteq V_G$, we define $A^\oplus := A \cup N_{G-E_H}(A)$. Let Φ be the shadow function of H in G. Define the *intruding function* $\Psi : V_G \times V_G \to 2^{V_G}$ with $\Psi(x, y) = \{x' \in \{x\}^\oplus \mid \Phi(x') \cap \{y\}^+ = \emptyset\}$. The name is justified by the following theorem.

Theorem 14. *Let Φ and Ψ be the shadow and intruding functions of a subgraph H of some graph G. The following statements are equivalent.*

(i) *y parries x,*

(ii) *$y \in \Phi(x)$,*

(iii) *$\Psi(x, y) = \emptyset$.*

Proof. Equivalence of (i) and (ii) is proved in Theorem 11.

Assume that y parries x. This implies that arbitrary $G - E_H$ neighbor x' of x is parried by some $y' \in \{y\}^+$. By the equivalence of (i) and (ii), this implies $y' \in \Phi(x')$ and by the definition we have $x' \notin \Psi(x, y)$. As this holds for all $x' \in \{x\}^\oplus$, we have $\Psi(x, y) = \emptyset$.

The converse implication follows by reversing the preceding argument. ■

Corollary 15. *Let Ψ be the intruding function of the subgraph H of the graph G and let $S \subseteq V_G$ be the set of guard posts. There exists a successful intruder of H if and only if there is a vertex $x \in V_G$ such that $\Psi(x, y) \neq \emptyset$ for every $y \in S$.*

A successful intruder $X = (v, \iota)$ can be reconstructed using the intruding function by setting v to be the vertex from the previous corollary, and choosing $\iota(x, y) \in \Psi(x, y)$ whenever the latter set is nonempty. If it is empty, then $\iota(x, y)$ can be chosen arbitrarily; a successful intruder never enters into such situation. Note also that the intruding function may be constructed in parallel with the shadow function in Algorithms 2 and 3. In order to achieve this, $\Psi(x, y) = \emptyset$ must be set initially and whenever a vertex y is removed from either $Q(x)$ or $\Phi(x)$, appropriate vertices must be added into $\Psi(x, y)$.

5. Shadow and Retractions

Let G be a graph and H its subgraph. A *retraction* of G onto H is a homomorphism $f : G \to H$ that is an identity on V_H. As f preserves edges of G or maps their endpoints to a single edge, it is easy to observe the following proposition. The proposition is not surprising, as it is known [12] that retractions of cop-win graphs are cop-win.

Proposition 16. *Let G be a connected graph, H its subgraph, Φ the shadow function of H in G and $f : G \to H$ a retraction. Then $f(x) \in \Phi(x)$ for every $x \in V_G$.*

Proof. By Theorem 14, it suffices to show that $f(x)$ parries x. Let W be any walk from x to arbitrary chosen vertex x' of H. Consider now the walk $W' = f(W)$. W' starts in $f(x)$ and ends in $f(x') = x'$. Since f preserves the edges of G or maps their endpoints to a single edge, the length of W' is equal to $|W|$. ■

The decision problem of whether there exists a retraction of a given graph G to its subgraph H is NP-complete in general [7]. Feder and Hell prove that for any constraint satisfaction problem $\mathbb{P}$, there exists a fixed reflexive graph H such that the complexity of the existence problem of retraction to H is polynomially equivalent to the complexity of $\mathbb{P}$, which establishes that classification of complexity of retraction problems is difficult. In this light, the shadow function is a polynomially computable generalization of retractions, which can confirm a negative answer to the retraction problem: if there is a vertex $x \in V_G$ such that $\Phi(x) \cap V_H = \emptyset$, then there is no retraction of V_G to H.

6. A Single Intruder Versus Several Guards

Motivated by questions of the referees, we sketch an approach of applying the presented algorithms to the setting of several guards and (in next section), also several intruders. Again let G be a connected graph, H its subgraph, $S \subseteq V_G$ a set of guard posts, and g the number of guards, all of them fixed throughout this section. It will be shown that the game of g guards against a single intruder can be represented as the game of a single guard against a single intruder in a different graph G^g under modified conditions: the metrics in G^g needs to be altered and additional restrictions have to be put on the motion of the intruder.

The *strong product* $\boxtimes_{i=1}^{k} G_i$ of graphs $G_1, \ldots, G_k$ is the graph defined on the Cartesian product of the vertex sets of the factors, two distinct vertices $(u_1, u_2, \ldots, u_k)$ and $(v_1, v_2, \ldots, v_k)$ are adjacent if and only if u_i is equal or adjacent to v_i in G_i for $i = 1, 2, \ldots, k$. The strong product of g copies of G is called the *g-th strong power of* G and will be denoted by G^g.

A position of g guards in G corresponds to a position of a single guard in G^g and vice versa. Similarly, any move of g guards corresponds to some move of the single guard in G^g.

Let G' be the subgraph of G^g that is spanned by the vertices that have all coordinates equal. Then G' is isomorphic to G and any position of the intruder in G corresponds to a unique position of the intruder in G'. The fact of capturing an intruder in G in this setting corresponds to moving the single guard in G^g to a vertex that has at least one coordinate equal to the position of the intruder. Thus the act of capturing in G^g corresponds to the distance 0 according to the following metrics in G^g: $d^*(x, y) = \min_{0 \leq i \leq g} d_G(x_i, y_i)$ for $x = (x_1, \ldots, x_g)$ and $y = (y_1, \ldots, y_g)$. Clearly enough, the set of guard posts $S \subseteq V_G$ in the game of g guards corresponds to the set $S^G \subseteq V_{G^g}$ of vertices with at least one coordinate equal to some vertex in S.

7. Several Intruders Versus Several Guards

Also the game of several intruders versus several guards can be transformed to the game of a single intruder and a single guard. The major difference to the case of a single intruder and several guards is that the guard and the intruder may have limited subgraph of valid positions, and capturing of the guard does not neccessarily mean that the game is finished, but it puts additional restriction on the set of valid moves for the intruder. The set of vertices which get invalid for the intruder depends on the position of the intruder and the guard at the moment of capture.

As in the previous sections, let us fix a graph G, its subgraph H, the set of guard posts S, the number of intruders i, and the number of guards g. Let $m = \max\{i, g\}$ and let $G^m := G \times \cdots \times G$ be the strong product of m copies of G, and let the subgraph $G^g \subseteq G^m$ be spanned by the vertices which have last $m - g + 1$ coordinates equal, and the subgraph G^i be spanned by the vertices which have the last $m - i + 1$ coordinates equal. A position of g guards in G and their moves correspond to a position of a single guard and his moves in G^g and vice versa, and similarly for positions and moves of i intruders.

The act of capturing one of the intruders happens whenever there is some guard positioned at the same position as some intruder. For the single guard and single intruder in G^m, this happens when one of any coordinates of the intruder $X = (x_1, \ldots, x_{i-1}, x_i, \ldots, x_i)$ is equal to some coordinate of the guard $Y = (x_1, \ldots, x_{g-1}, x_g, \ldots, x_g)$, or, in other words, $d^*_G(X, Y) = 0$ and $X \cap Y \neq \emptyset$ for the sets $X = \{x_1, \ldots, x_i\}$ and $Y = \{y_1, \ldots, y_g\}$.

The intruders which are captured can no longer move. Thus after y captures x, the coordinates in $X \cap Y$ get fixed and the intruder can only move in the coordinates $X \setminus Y$. Formally, the graph of allowed moves G^i is limited to $G^i \setminus \Lambda(x, y)$, where $\Lambda(x, y)$ is (vaguely) the set of vertices with coordinates $X \cap Y$ fixed to these values.

Acknowledgement

The authors thank the anonymous referees for detailed and constructive comments, which significantly contributed to the quality of the paper. This work was in part supported by the Ministry of Science of Slovenia under the grant P1-0297 and research projects J1-8130, L7-5459, N1-0057 of ARRS (Slovenia).

References

[1] M. Aigner and M. Fromme, *A game of cops and robbers*, Discrete Appl. Math. **8** (1984) 1–12.
doi:10.1016/0166-218X(84)90073-8

[2] B. Alspach, *Searching and sweeping graphs: A brief survey*, Matematiche (Catania) **59** (2004) 5–37.

[3] T. Andreae, *On a pursuit game played on graphs for which a minor is excluded*, J. Combin. Theory Ser. B **41** (1986) 37–47.
doi:10.1016/0095-8956(86)90026-2

[4] A. Bonato and B. Yang, *Graph searching and related problems*, in: Handbook of Combinatorial Optimization (Springer, New York, 2013) 1511–1558.
doi:10.1007/978-1-4419-7997-1_76

[5] E. Chiniforooshan,*A better bound for the cop number of general graphs*, J. Graph Theory **58** (2008) 45–48.
doi:10.1002/jgt.20291

[6] E.J. Cockayne, P.J.P. Grobler, W.R. Gründlingh, J. Munganga and J.H. van Vuuren, *Protection of a graph*, Util. Math. **67** (2005) 19–32.

[7] T. Feder and P. Hell, *List homomorphisms to reflexive graphs*, J. Combin. Theory Ser. B **72** (1998) 236–250.
doi:10.1006/jctb.1997.1812

[8] F.V. Fomin, P.A. Golovach, A. Hall, M. Mihalák, E. Vicari and P. Widmayer, *How to guard a graph*? Algorithmica **61** (2011) 839–856.
doi:10.1007/s00453-009-9382-4

[9] F.V. Fomin, P.A. Golovach and D. Lokshtanov, *Guard games on graphs: Keep the intruder out*! Theoret. Comput. Sci. **412** (2011) 6484–6497.
doi:10.1016/j.tcs.2011.08.024

[10] G. Hahn and G. MacGillivray, *A note on k-cop, l-robber games on graphs*, Discrete Math. **306** (2006) 2492–2497.
doi:10.1016/j.disc.2005.12.038

[11] H. Nagamochi, *Cop-robber guarding game with cycle robber region*, in: 3rd International Workshop on Frontiers in Algorithmics, Lecture Notes in Comput. Sci. **5598** (Springer-Verlag, Berlin, 2009) 74–84.

[12] R. Nowakowski and P. Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Math. **43** (1983) 235–239.
doi:10.1016/0012-365X(83)90160-7

[13] A. Quilliot, *A short note about pursuit games played on a graph with a given genus*, J. Combin. Theory Ser. B **38** (1985) 89–92.
doi:10.1016/0095-8956(85)90093-0

[14] T.D. Parsons, *Pursuit-evasion in a graph*, in: Theory and Applications of Graphs, Y. Alavi, D.R. Lick (Eds.), Lecture Notes in Math. **642** (Springer-Verlag, Berlin, 1978) 426–441.
doi:10.1007/BFb0070400

[15] T. Reddy, S. Krishna and P. Rangan, *The guarding problem—complexity and approximation*, in: Proceedings of IWOCA, 2009, J. Fiala, J. Kratochvíl, M. Miller (Eds.), Lecture Notes in Comput. Sci. **5874** (2009) 460–470.
doi:10.1007/978-3-642-102217-2$_$-$45

[16] B. Schröder, *The copnumber of a graph is bounded by* $\lfloor \frac{3}{2}\text{genus}(G)\rfloor + 3$, in: Categorical Perspectives, Trends Math. (Birkhäuser, Boston, 2001) 243–263.
doi:10.1007/978-1-4612-1370-3_14

[17] R. Šámal, R. Stolař and T. Valla, *Complexity of the cop and robber guarding game*, in: Proceedings of IWOCA 2011, Lecture Notes in Comput. Sci. **7056** (2011) 361–373.
doi:10.1007/978-3-642-25011-8_29

[18] R. Šámal and T. Valla, *The guarding game is E-complete*, Theoret. Comput. Sci. **521** (2014) 92–106.
doi:10.1016/j.tcs.2013.11.034

Received 20 October 2016
Revised 5 August 2019
Accepted 5 August 2019