Dynamic Games and Applications
https://doi.org/10.1007/s13235-021-00385-0

Some Game-Theoretic Remarks on Two-Player Generalized Cops and Robbers Games

Athanasios Kehagias[1] · **Georgios Konstantinidis**[1]

Accepted: 18 March 2021

Abstract

In this paper, we study the two-player *generalized cops and robbers* (GCR) games introduced by Bonato and MacGillivray. Our main goals are to provide: (a) a *game-theoretic* formulation of GCR and (b) a *self-contained game-theoretic* proof that GCR has a value and an optimal strategy profile. To achieve our goals, we first formulate GCR (and CR as a special case) as a zero-sum stochastic game. Then we study a Vertex Labeling (VL) algorithm and prove it computes the value of the GCR game (for every starting "condition") and that the vertex labels can be used to specify a positional deterministic optimal strategy for each player. We also compare our game-theoretic analysis to some of the usual graph theoretic/combinatorial approaches to CR/GCR.

Keywords Cops and robbers · Pursuit evasion

1 Introduction

In this paper, we present a *game-theoretic* analysis of the *two-player generalized cops and robbers* games (henceforth GCR) introduced by Bonato and MacGillivray in [5]. GCR (which includes the "classic" cops and robbers (CR) game as a special case) can be understood as a general framework for *pursuit games on graphs*. Our main goals are:

1. to provide a *game-theoretic* formulation of GCR;
2. to prove in a *game-theoretic* manner that GCR has a value and an optimal strategy profile.

Before presenting our own contribution in more detail, let us briefly discuss previous approaches to CR and GCR. "Classic" CR was introduced in [18] and [19–21] and has been the subject of intense research ever since. A relatively recent review of the literature appears in the excellent book [4]; additional references can be found in [6,8,16]. As already mentioned, Bonato and MacGillivray [5] introduced GCR games, a very broad generalization of classic CR.

In almost all CR/GCR literature, a graph theoretic and/or combinatorial approach is used; as far as we are aware, very few authors deal with the *game-theoretic* aspects (such as value,

✉ Athanasios Kehagias
kehagiat@gmail.com

[1] Aristotle University Thessaloniki, Thessaloniki, Greece

minimax, Nash equilibria, etc.) of classic CR, CR variants or GCR. This has the following consequences[1].

1. Typically, a *payoff function* is not defined for either the CR and GCR games. Similarly, *strategies* are often mentioned without being formally defined. Consequently, the *existence* and *computation* of *optimal strategies* cannot be established in the standard game-theoretic manner.
2. A key concept in CR/GCR is the *capture time*. As we will show in the sequel, this has a simple game-theoretic interpretation: It is the *value* of CR/GCR, if this is appropriately formulated as a zero-sum game. In most of the existing literature, this connection is not pointed out.

In Sect. 4, we will elaborate on the above themes and provide specific examples from three important papers [2,5,9].

We believe that a game-theoretic formulation will put the existing CR/GCR work on a firmer basis. In addition, we hope that the infusion of "classic" game-theoretic concepts into the CR literature will provide new methods of analysis and lead to new insights. We will elaborate on this theme in Sect. 5.

To obtain the desired game-theoretic analysis, we start by formulating GCR (and CR as a special case) as a *zero-sum stochastic game*. The first step to this end is the definition of an appropriate payoff function. Having defined GCR as a stochastic game, we could establish the existence of value and optimal strategies using standard results (e.g., the ones contained in [7]) from stochastic game theory. However we have chosen another route and have obtained a *self-contained* game theoretic solution of GCR.[2]

Namely, we study a *Vertex Labeling* (VL) algorithm (introduced in [9] for CR and then extended in [5] for GCR) and prove some of its properties. We then use these properties to establish that the VL algorithm actually computes the value of the GCR game (for every starting "condition") and that the vertex labels can be used to specify a positional deterministic optimal strategy for each player[3]. We repeat that our analysis is entirely self-contained, with one exception: our Proposition 2.18 is the well-known (and easily proved) theorem about equivalence of value and Nash equilibrium in zero-sum games.

The rest of this paper is organized as follows. In Sect. 2, we present definitions and notations which will be used in the sequel. In Sect. 3, we *solve* the GCR game, i.e., we give a self-contained proof that the VL algorithm computes the value and optimal strategies. In Sect. 4, we discuss the connection of previous analyses of both GCR and classic CR to our own. We summarize and conclude in Sect. 5.

2 Preliminaries

2.1 Informal Description of the GCR Game

We first provide an informal description of the GCR game. This description is inspired by the one given in [5] but differs in several respects. To motivate our description, recall the basic elements of the classic CR game. CR is played on an undirected, simple, finite graph G, in

[1] Let us add that the following remarks apply (to a certain extent) even to the simpler, original version of CR/GCR which is concerned with *winning* (i.e., capturing/evading) rather than *time optimal* strategies.

[2] We mention, for the sake of completeness, that yet another possibility is the formulation and solution of GCR as an *extensive form game*.

[3] It is worth noting that the VL algorithm is actually a special form of *value iteration*.

discrete time steps (*turns*) and involves two players: the *cop* (or *pursuer*) and the *robber* (or *evader*). At every turn, each player is *located* on a vertex v of G, and a *single* player can move to a new location (a vertex v' adjacent to v). The cop wins if he "captures" the robber, i.e., if at some turn they are both located in the same vertex[4].

The *generalized* CR game (GCR game) can be informally described as follows. It involves two *players*, the pursuer and the evader; the game is played in turns; in each turn a single player can (subject to certain restrictions) move between elements of a finite set, resulting in a new "configuration" of player locations; there is a set of target ("capture") configurations; the pursuer wins if a target configuration is reached and the evader wins otherwise.

It is easy to see that the classic CR is a special case of GCR game. As explained in [5] (where details and references are provided) the definition of GCR games also encompasses the following games: distance-k cops and robbers, tandem-win cops and robbers, cops and robbers with traps, eternal domination, revolutionaries and spies, seepage and many more.

2.2 Formal Description of the GCR Game

We give a formal description (similar but not identical to the one in [5]) of the GCR game.

2.2.1 General Rules

A GCR game involves two *players*, P^1 (pursuer) and P^2 (evader); for $n \in \{1, 2\}$, P^n's possible *locations* are the elements of a finite set V.

A *nonterminal game state* (or *position*) is a triple $\left(x^1, x^2, p\right)$ where (for $n \in \{1, 2\}$) $x^n \in V$ indicates P^n's *location* and p indicates the *single* player who currently "has the move" (i.e., can change his location). The set of all nonterminal game states is

$$\overline{S} = V^1 \times V^2 \times \{1, 2\}.$$

We will also use a *terminal state* τ, so that the full state set is $S = \overline{S} \cup \{\tau\}$. The game starts at a prespecified *initial state* $s_0 \in \overline{S}$ and is played in *turns*; in each turn, a *single* player moves by the *movement rules* described in Sect. 2.2.2. A *target set* $S_c \subseteq \overline{S}$ is given; S_c is the set of *capture states*. The set of *noncapture states* is $S_{nc} = \overline{S} \backslash S_c$. So, finally we have the partitions

$$\overline{S} = S_{nc} \cup S_c, \quad S = S_{nc} \cup S_c \cup \{\tau\}.$$

We will also need the sets

$$\forall n \in \{1, 2\} : S^n = \left\{s = \left(x^1, x^2, n\right) : x_1, x_2 \in V\right\};$$

i.e., S^n is the set of states in which P^n has the move. This results in another partition of $\overline{S}$:

$$\overline{S} = S^1 \cup S^2.$$

It is assumed that both players have *perfect information*, i.e., at every turn of the game they have complete knowledge of the way the game has been played so far.

Example 2.1 In the CR game, P^1 is the cop and P^2 is the robber. The location sets are $V^1 = V^2 = V$, the vertex set of a graph $G = (V, E)$. The nonterminal state set is

$$\overline{S} = \left\{\left(x^1, x^2, n\right) : x^1, x^2 \in V, n \in \{1, 2\}\right\}.$$

[4] Note that in the usual CR description time is counted in *rounds*, where each round encompasses (in our terminology) one cop and one robber turn. The two approaches are essentially equivalent.

The capture set S_c is

$$S_c = \left\{ \left(x^1, x^1, n \right) : x^1 \in V, n \in \{1, 2\} \right\}.$$

This is *almost* exactly the classic CR game, with one difference: we omit the classic "placement phase" in which first the cop and then the robber choose initial positions. We assume instead that both initial positions (as well as the first player to move) are given by the prespecified initial state $s_0 = \left(x_0^1, x_0^2, n_0 \right)$.

2.2.2 Movement Rules

At each turn, a player can change his location subject to some movement rules, which are specific to each particular GCR game. For example, in the classic CR game each player can move from his current vertex to any adjacent vertex (or stay in place).

To specify movement rules, we will often use the following standard game-theoretic notation: for each $n \in \{1, 2\}$, we use $-n$ to indicate the index of the "other player." For example, suppose player P^n has location x^n, then the "other" player is P^{-n} and has location x^{-n}.

We have already mentioned that each game state specifies which player makes the next move. Unlike both classic CR and the GCR of [5] , *we do not demand that the players alternate in taking moves*; in making a move, a player may take the game to a state in which he again has the move.

For every state $s = \left(x^n, x^{-n}, n \right)$, there exists a nonempty set of possible next moves for P^n (the player who has the move). To each such move corresponds a unique next state and conversely. Hence we fully describe possible next moves from state s by the set $N(s)$ of possible next states. For instance, if $s = \left(x^n, x^{-n}, n \right) \in \overline{S}$ then every $\left(y^n, y^{-n}, m \right) \in N \left(x^n, x^{-n}, n \right)$ must satisfy two conditions: (a) y^n is a possible next move of P^n from x^n and (b) $y^{-n} = x^{-n}$.

We assume that the GCR game is played for an *infinite* number of turns. That is, even after a capture state is reached, the game will continue ad infinitum, but in a trivial manner. Namely, in every GCR game the only successor of a capture state is the terminal state:

$$s \in S_c \Rightarrow N(s) = \{\tau\}$$

which can only transit into itself

$$N(\tau) = \{\tau\}.$$

In short, the movement rules of a particular GCR game can be encoded by

$$\mathbf{N} = (N(s))_{s \in S},$$

the collection of possible next moves of each state. Note that $\mathbf{N}$ also specifies (implicitly) the state set S which in turn specifies the location sets V^1, V^2.

Example 2.2 Continuing from Example 2.1, in the CR game played on the graph $G = (V, E)$ we have

$$\forall s = \left(x^n, x^{-n}, n \right) \in \overline{S} : N(s) = \left\{ \left(z, x^{-n}, -n \right) : \left\{ x^n, z \right\} \in E \right\} \text{ and } N(\tau) = \{\tau\}.$$

2.2.3 Payoff Function

According to [5], the pursuer wins iff a capture state is reached. We will now express this in terms of a *total payoff function*, which is itself defined in terms of a *turn payoff function*.

Definition 2.3 The *turn payoff function* $q(s)$ specifies the amount gained by the evader (and lost by the pursuer) at every turn of GCR in which the current state is s; it is defined by

$$q(s) = \begin{cases} 1 \text{ iff } s \in S_{nc} \\ 0 \text{ iff } s \in S_c \cup \{\tau\} \end{cases}. \tag{1}$$

Definition 2.4 The *total payoff function* $Q(s_0 s_1 \ldots)$ specifies the amount gained by the evader (and lost by the pursuer) in a play of GCR with state sequence $s_0 s_1 \ldots$; it is defined by

$$Q(s_0 s_1 s_2 \ldots) = \sum_{t=0}^{\infty} q(s_t). \tag{2}$$

The following remarks can be made regarding the significance of the payoff functions.

1. When the game goes through the state sequence $s_0 s_1 \ldots$, clearly $Q(s_0 s_1 s_2 \ldots)$ is the *capture time*, i.e., the number of turns until capture is affected; if $Q(s_0 s_1 s_2 \ldots) = \infty$ then capture is never affected.
2. It is also clear that, by the above payoff functions, GCR is a *zero-sum* game. The pursuer wants to minimize capture time and the evader wants to maximize the same quantity.
3. By the above formulation, the pursuer not only wants to capture the evader in finite time (and hence win); he wants to capture in the *shortest possible* time.

2.2.4 The GCR Game Family

Keeping in mind all of the above, we have the following two definitions.

Definition 2.5 A *GCR game* is a tuple $\left(\mathbf{N}, S^1, S^2, S_c, s_0\right)$ where

1. $\mathbf{N}$ describes the movement rules (and implicitly the sets S, V^1, V^2);
2. S^1 and S^2 describe which player has the move in every state;
3. S_c describes the capture (winning) condition;
4. s_0 is the initial state.

Definition 2.6 A *GCR game family* is a tuple $\left(\mathbf{N}, S^1, S^2, S_c\right)$.

By the above definitions, given the game family $\left(\mathbf{N}, S^1, S^2, S_c\right)$ and a specific initial state $s_0 \in S$, we obtain a particular game $\left(\mathbf{N}, S^1, S^2, S_c, s_0\right)$. In other words, every element of the family $\left(\mathbf{N}, S^1, S^2, S_c\right)$ is a game played by the same rules but from a different initial state. In Sect. 3, we will mostly prove properties of the entire family $\left(\mathbf{N}, S^1, S^2, S_c\right)$ which will also imply corresponding properties of all specific games $\left(\mathbf{N}, S^1, S^2, S_c, s_0\right)$.

Recall that a stochastic game [7] is a sequence of one-turn games; for every one-turn game played, the players receive their *turn payoff* and the next game to be played is selected, depending on the current game and players' actions; the *total payoff* to each player is the sum of his turn payoffs. Obviously, a GCR *game family* is a *stochastic game* (in fact, a special case in which player actions and transitions to the next state are deterministic) and we could obtain the full GCR solution by invoking well-known stochastic games results [7] . However, we will avoid this route since, as mentioned, we want to present a self-contained solution.

At first sight, our definition of GCR games and the one given in [5] differ in several respects. Most importantly, the GCR game of [5] starts with a "placement phase," in which first P^1 and then P^2 chooses his initial location; this yields an initial state s_0 and the remaining part of the Bonato–MacGillivray game is basically our $\left(\mathbf{N}, S^1, S^2, S_c, s_0\right)$ game. We will argue in Sect. 4 that this and several other differences are not significant and that our results also apply to the games defined in [5].

2.3 Additional Game-Theoretic Concepts

We conclude this section by presenting some additional standard game theoretic concepts which will prove useful in the sequel. In what follows we assume that a game family $\left(\mathbf{N}, S^1, S^2, S_c\right)$ has been specified, so it is usually omitted from the notation. Also, in what follows $\mathbb{N}_0$ denotes the set $\{0, 1, 2, ...\}$.

Definition 2.7 A *history* $h = s_0 s_1 ...$ is a finite or infinite sequence of states.

Definition 2.8 We define the following sets of histories

$$\text{the set of finite length histories} : H_* = \{h : h = s_0 s_1 s_2 ... s_T, T \in \mathbb{N}_0, \forall t : s_t \in S\},$$
$$\text{the set of infinite length histories} : H_\infty = \{h : h = s_0 s_1 s_2 ..., \forall t : s_t \in S\}.$$

Definition 2.9 A (*pure* or *deterministic*) *strategy* $\sigma : H_* \to S$ is a function which maps finite histories to *next states*.

We could have defined *randomized* (more precisely: *behavioral*) strategies, but these will not be needed in our analysis, since the GCR game has perfect information. Also, in the context of stochastic games, the usual definition specifies a strategy as a function which maps finite histories to next *moves*. However, since the GCR game evolves deterministically, a move specifies the next state; hence, our definition is sufficient (and more convenient) for our purposes.

Definition 2.10 A strategy σ^m is called *positional* if it depends only on the current state s_t, but neither on previous states nor on current time t, i.e.,

$$\forall h = s_0 s_1 ... s_t, h' = s'_0 s'_1 ..., s'_{t'} : s_t = s'_{t'} = s \Rightarrow \sigma^m(h) = \sigma^m\left(h'\right) = \sigma^m(s).$$

A strategy is called *nonpositional* iff it is not positional.

Notation 2.11 $H\left(\sigma^1, \sigma^2 | \mathbf{N}, S^1, S^2, S_c, s_0\right)$ *denotes the infinite history* $s_0 s_1 s_2 ...$ *generated when: (i) the game is* $\left(\mathbf{N}, S^1, S^2, S_c, s_0\right)$ *and (ii) for* $n \in \{1, 2\}$, P^n *uses* σ^n. *When* $\left(\mathbf{N}, S^1, S^2, S_c\right)$ *is understood from the context, we simply write* $H\left(\sigma^1, \sigma^2 | s_0\right)$.

Notation 2.12 $T\left(\sigma^1, \sigma^2 | \mathbf{N}, S^1, S^2, S_c, s_0\right)$ *denotes the* capture time, *i.e., the first (actually the only) time at which a capture state* $s \in S_c$ *is reached when: (i) the game is* $\left(\mathbf{N}, S^1, S^2, S_c, s_0\right)$ *and (ii) for* $n \in \{1, 2\}$, P^n *uses* σ^n. *When* $\left(\mathbf{N}, S^1, S^2, S_c\right)$ *is understood from the context, we simply write* $T\left(\sigma^1, \sigma^2 | s_0\right)$.

Remark 2.13 It is obvious that

$$T\left(\sigma^1, \sigma^2 | \mathbf{N}, S^1, S^2, S_c, s_0\right) = Q\left(H\left(\sigma^1, \sigma^2 | \mathbf{N}, S^1, S^2, S_c, s_0\right)\right).$$

We conclude by presenting several well-known definitions and facts about zero-sum games, in a notation specific to a GCR game. What follows concerns a specific game $\left(\mathbf{N}, S^1, S^2, S_c\right)$, which we assume known and (for brevity) do not include in the notation.

Definition 2.14 For every s we define the following two quantities

$$\textit{lower value of } \left(\mathbf{N}, S^1, S^2, S_c\right) : T^-(s) = \sup_{\sigma^2} \inf_{\sigma^1} T\left(\sigma^1, \sigma^2 | s\right),$$
$$\textit{upper value of } \left(\mathbf{N}, S^1, S^2, S_c\right) : T^+(s) = \inf_{\sigma^1} \sup_{\sigma^2} T\left(\sigma^1, \sigma^2 | s\right).$$

Proposition 2.15 *For every s we have*

$$T^-(s) = \sup_{\sigma^2} \inf_{\sigma^1} T\left(\sigma^1, \sigma^2 | s\right) \leq \inf_{\sigma^1} \sup_{\sigma^2} T\left(\sigma^1, \sigma^2 | s\right) = T^+(s).$$

Definition 2.16 We say that the game $\left(\mathbf{N}, S^1, S^2, S_c\right)$ has a *value* $\widehat{T}(s)$ iff

$$T^-(s) = \sup_{\sigma^2} \inf_{\sigma^1} T\left(\sigma^1, \sigma^2 | s\right) = \inf_{\sigma^1} \sup_{\sigma^2} T\left(\sigma^1, \sigma^2 | s\right) = T^+(s).$$

in which case we define $\widehat{T}(s)$ to be

$$\widehat{T}(s) = \sup_{\sigma^2} \inf_{\sigma^1} T\left(\sigma^1, \sigma^2 | s\right) = \inf_{\sigma^1} \sup_{\sigma^2} T\left(\sigma^1, \sigma^2 | s\right).$$

Definition 2.17 We say that the game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ has *optimal strategies* $\widehat{\sigma}^1, \widehat{\sigma}^2$ iff

$$\begin{aligned} \forall \sigma^2 &: T\left(\widehat{\sigma}^1, \sigma^2 | s\right) \leq T\left(\widehat{\sigma}^1, \widehat{\sigma}^2 | s\right), \\ \forall \sigma^1 &: T\left(\sigma^1, \widehat{\sigma}^2 | s\right) \geq \widehat{T}\left(\widehat{\sigma}^1, \widehat{\sigma}^2 | s\right). \end{aligned}$$

We also say that $\left(\widehat{\sigma}^1, \widehat{\sigma}^2\right)$ is an *optimal strategy pair* (or *profile*).

Proposition 2.18 *The game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ has value $\widehat{T}(s)$ and optimal strategies $\widehat{\sigma}^1, \widehat{\sigma}^2$ iff*

$$\begin{aligned} \forall \sigma^2 &: T\left(\widehat{\sigma}^1, \sigma^2 | s\right) \leq \widehat{T}(s), \\ \forall \sigma^1 &: T\left(\sigma^1, \widehat{\sigma}^2 | s\right) \geq \widehat{T}(s). \end{aligned}$$

3 Solution of the GCR Game

In Sect. 2.2, we have formulated the GCR game family $\left(\mathbf{N}, S^1, S^2, S_c\right)$ as a two-player, zero-sum stochastic game. By "solving $\left(\mathbf{N}, S^1, S^2, S_c\right)$" we mean proving that, for every $s \in \overline{S}$, the game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ has value and optimal strategies, as well as computing these quantities.

We will now provide a self-contained solution based on a *vertex labeling* algorithm (VL algorithm) obtained by modifying similar algorithms presented in [9] (for the CR game) and [5] (for the GCR game). Our analysis is more detailed than the ones presented in [5,9], mainly because we present in detail the game-theoretic aspects.

Algorithm 1 : The vertex labeling algorithm

Require: The game family $\left(\mathbf{N}, S^1, S^2, S_c\right)$.

1: **for** $s \in \overline{S}$ **do**
2:     **if** $s \in S_c$ **then**
3:         $T^0(s) = 0$
4:     **else**
5:         $T^0(s) = \infty$
6:     **end if**
7: **end for**
8: **for** $i = 1, 2, ..$ **do**
9:     **for** $s \in \overline{S}$ **do**
10:         **if** $T^{i-1}(s) < \infty$ **then**
11:             $T^i(s) = T^{i-1}(s)$
12:         **else if** $s \in S^1$ **then**
13:             $T^i(s) = 1 + \min_{s' \in N(s)} T^{i-1}(s')$
14:         **else if** $s \in S^2$ **then**
15:             $T^i(s) = 1 + \max_{s' \in N(s)} T^{i-1}(s')$
16:         **end if**
17:     **end for**
18: **end for**

The VL algorithm produces, for every $s \in S$, an infinite sequence

$$T^0(s), T^1(s), T^2(s), ...$$

Note that, for any s, the VL algorithm changes $V^i(s)$ at most once (lines 10-11). Hence, either $T^i(s)$ is never changed through the run of the algorithm and then, for all i, $T^i(s) \in \{0, \infty\}$; or $T^i(s)$ starts equal to ∞ and is changed *once* to a finite nonzero value. In other words, the only two forms which $\left(T^i(s)\right)_{i \in \mathbb{N}_0}$ can take are:

1. for every $i \in \mathbb{N}_0$, $T^i(s)$ has the same value (which is either 0 or ∞),
2. there exists some $n \in \mathbb{N}$ such that $T^i(s) = \infty$ for $i \in \{0, ..., n-1\}$ and $T^i(s) = m \in \mathbb{N}$ for $i \in \{n, n+1, ...\}$.

Hence the following are well defined (with the understanding that $\min \emptyset = \infty$).

Definition 3.1 For all $s \in \overline{S}$ we define

$$\overline{T}(s) = \lim_{m \to \infty} T^m(s), \quad \widetilde{T}(s) = \min\{i : T^i(s) < \infty\}.$$

We will show that, for all $s \in \overline{S}$, $\overline{T}(s) = \widetilde{T}(s)$. I.e., if $T^m(s)$ attains a finite value at the i-th iteration of the VL algorithm, this value equals both i and $\lim_{m \to \infty} T^m(s)$. We need the following auxiliary proposition.

Proposition 3.2 *For all $s \in \overline{S}$ and for all $n \in \mathbb{N}_0$ we have*

$$\left(\widetilde{T}(s) = n\right) \Rightarrow \left(\forall i \geq n : T^i(s) = \overline{T}(s) \leq n\right). \tag{3}$$

Proof Let us first prove that, for all $s \in \overline{S}$ and $n \in \mathbb{N}_0$, we have

$$\left(\widetilde{T}(s) = n\right) \Rightarrow T^n(s) \leq n. \tag{4}$$

Clearly (4) holds for $n = 0$; assume it holds for $n \in \{0, 1, ..., k\}$. Now, if for some s we have $\widetilde{T}(s) = k + 1$ then $T^k(s) = \infty$ and $T^{k+1}(s) < \infty$. Using lines 10-16 of the VL algorithm we get:

$$T^{k+1}(s) = 1 + T^k(s')$$

for some $s' \in N(s)$ which satisfies $T^k(s') = m < \infty$. But then $\widetilde{T}(s') \le k$ which, by the inductive assumption, implies $T^k(s') \le k$. Hence $T^{k+1}(s) \le k+1$ and we have proved (4) for every $n \in \mathbb{N}_0$. Now (3) follows from lines 10-11 of the VL algorithm. □

Proposition 3.3 *For every $s \in \overline{S}$ we have: $\overline{T}(s) = \widetilde{T}(s)$.*

Proof We partition $\overline{S}$ into the following two sets

$$S_a = \left\{s : \widetilde{T}(s) = \infty\right\}, \qquad S_b = \left\{s : \widetilde{T}(s) < \infty\right\}.$$

If $s \in S_a$, then $\widetilde{T}(s) = \infty$; then

$$\{i : T^i(s) < \infty\} = \emptyset \Rightarrow \left(\forall i : T^i(s) = \infty\right);$$

hence $\overline{T}(s) = \infty$. We conclude that

$$\forall s \in S_a : \ \overline{T}(s) = \widetilde{T}(s). \tag{5}$$

To complete the proof, we must show:

$$\forall s \in S_b : \ \overline{T}(s) = \widetilde{T}(s). \tag{6}$$

To this end we will show that, for all $n \in \mathbb{N}_0$, we have

$$\forall s \in S_b : \ \widetilde{T}(s) = n \Rightarrow \overline{T}(s) = T^n(s) = T^{n+1}(s) = \ldots = n. \tag{7}$$

Now, (7) clearly holds for $n = 0$. Suppose it holds for $n \in \{0, 1, \ldots, k\}$. Take any state $s \in S_b$ such that $\widetilde{T}(s) = k + 1$. Then we have

$$T^k(s) = \infty, \tag{8}$$
$$T^{k+1}(s) = T^{k+2}(s) = \ldots = \overline{T}(s) = m \le k + 1 < \infty. \tag{9}$$

1. If $s \in S^1$, (8)–(9) imply that there exists some $s' \in N(s)$ such that

   $$T^k(s') = \min_{u \in N(s)} T^k(u) \qquad \text{and} \qquad T^{k+1}(s) = 1 + T^k(s').$$

   Also (8) implies that $\min_{u \in N(s)} T^{k-1}(u) = \infty$ and hence

   $$T^{k-1}(s') = \infty. \tag{10}$$

   But $T^k(s') = T^{k+1}(s) - 1 < \infty$; so $\widetilde{T}(s') = k$ and by the inductive assumption, we have

   $$\overline{T}(s') = T^k(s') = T^{k+1}(s') = \ldots = k \Rightarrow$$
   $$\overline{T}(s) = T^{k+1}(s) = T^{k+2}(s) = \ldots = 1 + T^k(s') = 1 + k.$$

2. If $s \in S^2$, (9) implies that $\max_{u \in N(s)} T^k(u) < \infty$; hence for all $u \in N(s)$ we have $T^k(u) < \infty$. On the other hand, (8) implies that there exists some $s' \in N(s)$ such that

$T^{k-1}\left(s'\right)=\infty$. Hence $\widetilde{T}\left(s'\right)=k$. We also have

$$\begin{aligned}T^{k+1}(s)=m<\infty &\Rightarrow \left(\forall u\in N(s): T^{k}(u)<\infty\right)\\ &\Rightarrow \left(\forall u\in N(s): \widetilde{T}(u)\leq k\right)\\ &\Rightarrow \left(\forall u\in N(s): \overline{T}(u)=T^{k}(u)\leq k\right)\end{aligned}$$

(the last line following from Proposition 3.2). Hence s' achieves the maximum:

$$T^{k}\left(s'\right)=\max_{u\in N(s)} T^{k}(u)=k$$

and so

$$\overline{T}(s)=T^{k+1}(s)=T^{k+2}(s)=...=1+T^{k}\left(s'\right)=1+k.$$

Thus we have proved that (7) holds for every $n\in\mathbb{N}_0$, and we are done. □

We also have the following useful and easily provable proposition.

Proposition 3.4 *The collection $\left(\overline{T}(s)\right)_{s\in\overline{S}}$ satisfies the following:*

$$\forall s\in S^{1}\backslash S_{c}: \overline{T}(s)=1+\min_{s'\in N(s)}\overline{T}\left(s'\right), \tag{11}$$

$$\forall s\in S^{2}\backslash S_{c}: \overline{T}(s)=1+\max_{s'\in N(s)}\overline{T}\left(s'\right), \tag{12}$$

Proof Simply take the limits in lines 13 and 15 of the VL algorithm. □

The system of Eqs. (11)–(12) is the perfect information version of the *optimality equations* which play a central role in stochastic games [7].
We will next show that, for every $s\in\overline{S}$, $\overline{T}(s)$ is the *value* of the game $\left(\mathbf{N},S^1,S^2,S_c,s\right)$. Furthermore, we will use the collection $\left(\overline{T}(s)\right)_{s\in\overline{S}}$ to define *optimal* strategies for P^1 and P^2.

To avoid ambiguities in the definition of the optimal strategies, we modify the functions arg min and arg max as follows. Given the collection of successor states $\mathbf{N}=(N(s))_{s\in\overline{S}}$, we assume that $\overline{S}$ is equipped with a fixed total order. Since also $\overline{S}$ is a finite set, it follows that, for every $s\in\overline{S}$, expressions such as "the *first* element of $N(s)$ such that ..." are uniquely defined. Now, for every function $f:\overline{S}\rightarrow\mathbb{R}\cup\{\infty\}$, we (re)define arg min and arg max:

$$\forall s\in\overline{S}: \arg\min_{s'\in N(s)} f\left(s'\right)=\text{“the } \textit{first} \text{ element } u\in N(s) \text{ s.t. } f(u)=\min_{s'\in N(s)} f\left(s'\right)\text{”},$$

$$\forall s\in\overline{S}: \arg\max_{s'\in N(s)} f\left(s'\right)=\text{“the } \textit{first} \text{ element } u\in N(s) \text{ s.t. } f(u)=\max_{s'\in N(s)} f\left(s'\right)\text{”}.$$

Definition 3.5 Given the game family $\left(\mathbf{N},S^1,S^2,S_c\right)$, with labels $\left(\overline{T}(s)\right)_{s\in S_c}$, we define the following two strategies

$$\begin{aligned}&\text{A pursuer strategy } \widehat{\sigma}^{1}: \forall s\in S^{1}\backslash S_{c}: \widehat{\sigma}^{1}(s)=\arg\min{}_{s'\in N(s)}\overline{T}\left(s'\right),\\ &\text{An evader strategy } \widehat{\sigma}^{2}: \forall s\in S^{2}\backslash S_{c}: \widehat{\sigma}^{2}(s)=\arg\max{}_{s'\in N(s)}\overline{T}\left(s'\right).\end{aligned}$$

Proposition 3.6 *Given the game family $\left(\mathbf{N},S^1,S^2,S_c\right)$, for every $s\in\overline{S}$: $\overline{T}(s)$ is the value of the game $\left(\mathbf{N},S^1,S^2,S_c,s\right)$, and $\widehat{\sigma}^1,\widehat{\sigma}^2$ are optimal positional strategies.*

Proof Assuming $\left(\mathbf{N}, S^1, S^2, S_c\right)$ given and fixed, we drop it from all subsequent notation.

It follows immediately from Definition 3.5 that $\widehat{\sigma}^1, \widehat{\sigma}^2$ are positional strategies. Given Proposition 2.18, to prove the rest of the theorem it suffices to show that

$$\forall s \in S : \forall \sigma^2 : \ T\left(\widehat{\sigma}^1, \sigma^2 | s\right) \leq \overline{T}(s), \tag{13}$$

$$\forall s \in S : \forall \sigma^1 : \ T\left(\sigma^1, \widehat{\sigma}^2 | s\right) \geq \overline{T}(s). \tag{14}$$

Part I Take any s such that $\overline{T}(s) < \infty$.

1. First we show that $\widehat{\sigma}^1$ satisfies (13), by showing that, for every $n \in \mathbb{N}_0$ we have:

$$\forall s : \overline{T}(s) = n \Rightarrow \left(\forall \sigma^2 : T\left(\widehat{\sigma}^1, \sigma^2 | s\right) \leq \overline{T}(s)\right). \tag{15}$$

   Obviously (15) holds when $\overline{T}(s) = 0$ (because then $s \in S_c$ and $T\left(\sigma^1, \sigma^2 | s\right) = 0$ for all σ^1, σ^2). Assume it holds for all $n \in \{0, 1, ..., k\}$ and pick any s such that $\overline{T}(s) = k + 1$.

   (a) Suppose $s \in S^1$. From (11), we have

$$k + 1 = 1 + \min_{s' \in N(s)} \overline{T}(s') \Rightarrow \min_{s' \in N(s)} \overline{T}(s') = k.$$

   Hence, for $\overline{s} = \widehat{\sigma}^1(s) = \arg\min_{s' \in N(s)} \overline{T}\left(s'\right)$ we have $\overline{T}\left(\overline{s}\right) = k$. By the inductive assumption, we then have

$$\forall \sigma^2 : \ T\left(\widehat{\sigma}^1, \sigma^2 | \overline{s}\right) \leq \overline{T}(\overline{s}) = k.$$

   In other words, if P^1 uses $\widehat{\sigma}^1$ in the game $\left(\mathbf{N}, S^1, S^2, S_c, \overline{s}\right)$ then he will achieve capture in at most k moves, no matter how P^2 plays. Since $\overline{s} \in N(s)$, P^1 can also use $\widehat{\sigma}^1$ in the game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$. This means he will move from s to $\overline{s}$ and then will play exactly as in game $\left(\mathbf{N}, S^1, S^2, S_c, \overline{s}\right)$. This ensures capture in at most $k + 1$ moves, no matter how P^2 plays, i.e.,

$$\forall \sigma^2 : \ T\left(\widehat{\sigma}^1, \sigma^2 | s\right) \leq k + 1.$$

   Therefore $\widehat{\sigma}^1$ satisfies (13) for all $s \in S^1$ with $\overline{T}(s) = k + 1$.

   (b) Suppose $s \in S^2$. From relation (12), we have

$$k + 1 = 1 + \max_{s' \in N(s)} \overline{T}(s') \Rightarrow \max_{s' \in N(s))} \overline{T}(s') = k.$$

   Thus, whatever the initial move by player P^2 in game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$, the resulting state $\overline{s}$ will satisfy $\overline{T}\left(\overline{s}\right) \leq k$. Again, by the inductive assumption, we have

$$\forall \sigma^2 : \ T\left(\widehat{\sigma}^1, \sigma^2 | \overline{s}\right) \leq \overline{T}(\overline{s}) \leq k$$

   and hence, in the game $\left(\mathbf{N}, S^1, S^2, S_c, \overline{s}\right)$, P^1 can use $\widehat{\sigma}^1$ and capture in at most k moves (no matter what P^2 plays). We then get, by the same reasoning as above, that

$$\forall \sigma^2 : \ T\left(\widehat{\sigma}^1, \sigma^2 | s\right) \leq k + 1.$$

   Therefore $\widehat{\sigma}^1$ satisfies (13) for all $s \in S^2$ with $\overline{T}(s) = k + 1$.

2. Next we show inductively that $\widehat{\sigma}^2$ satisfies (14). I.e., we show that: for every $n \in \mathbb{N}_0$ we have:

$$\forall s : \overline{T}(s) = n \Rightarrow \left(\forall \sigma^1 : T\left(\sigma^1, \widehat{\sigma}^2 | s\right) \geq \overline{T}(s)\right). \tag{16}$$

Obviously (16) holds when $\overline{T}(s)=0$. Assume it holds for all $n\in\{0,1,...,k\}$ and pick any s such that $\overline{T}(s)=k+1$.

(a) Suppose $s\in S^2$. Similarly to the previous case, from (12) we get $\max_{s'\in N(s)}\overline{T}(s')=k$. Hence for $\overline{s}=\widehat{\sigma}^2(s)=\arg\max_{s'\in N(s)}\overline{T}(s')$ we have $\overline{T}(\overline{s})=k$. By the inductive assumption, we then have

$$\forall\sigma^1: T\left(\sigma^1,\widehat{\sigma}^2|\overline{s}\right)\geq\overline{T}(\overline{s})=k.$$

Hence P^2 using $\widehat{\sigma}^2$ in the game $\left(\mathbf{N},S^1,S^2,S_c,\overline{s}\right)$ can ensure capture will take k turns or more, no matter how P^1 plays. Reasoning as previously, P^2 using $\widehat{\sigma}^2$ in $\left(\mathbf{N},S^1,S^2,S_c,s\right)$ can ensure capture will take $k+1$ turns or more, no matter how P^1 plays, i.e.,

$$\forall\sigma^1: T\left(\sigma^1,\widehat{\sigma}^2|s\right)\geq k+1.$$

Therefore $\widehat{\sigma}^2$ satisfies (14) for all $s\in S^2$ with $\overline{T}(s)=k+1$.

(b) Finally, suppose $s\in S^1$; by an argument similar to that of case 1.b above we get that

$$\forall\sigma^1: T\left(\sigma^1,\widehat{\sigma}^2|s\right)\geq k+1.$$

Therefore $\widehat{\sigma}^2$ satisfies (14) for all $s\in S^1$ with $\overline{T}(s)=k+1$.

We have completed the proof of (13)–(14) for all s such that $\overline{T}(s)<\infty$.

Part II Now take any state s such that $\overline{T}(s)=\infty$. Then we clearly have

$$\forall s\in S:\forall\sigma^2:\ T\left(\widehat{\sigma}^1,\sigma^2|s\right)\leq\overline{T}(s)=\infty;$$

which proves (13). To prove (14), first set $s_0=s$ and then, for any pursuer strategy σ^1, let

$$s_0s_1s_2...=H\left(\sigma^1,\widehat{\sigma}^2|s_0\right).$$

We want to show that $s_0s_1s_2...$ never reaches a capture state. We will actually prove something (apparently) stronger:

$$\forall t\in\mathbb{N}_0,\ s_t\in H(\sigma^1,\widehat{\sigma}^2|s_0):\overline{T}\left(s_t\right)=\infty. \tag{17}$$

Suppose (17) is false and let s_{k+1} be the first element of $s_0s_1s_2...$ with $\overline{T}\left(s_t\right)<\infty$. I.e.,

$$\forall t\leq k:\overline{T}\left(s_t\right)=\infty\text{ and }\overline{T}\left(s_{k+1}\right)=m<\infty. \tag{18}$$

If $s_k\in S^2$, then $s_{k+1}=\widehat{\sigma}^2\left(s_k\right)=\arg\max_{s'\in N(s_k)}\overline{T}\left(s'\right)$ and

$$\infty=\overline{T}\left(s_k\right)=1+\overline{T}\left(s_{k+1}\right)\Rightarrow\overline{T}\left(s_{k+1}\right)=\infty. \tag{19}$$

If $s_k\in S^1$, then clearly (for any σ^1) $s_{k+1}\in N\left(s_k\right)$. Since $\overline{T}\left(s_{k+1}\right)=m<\infty$ we will have

$$\overline{T}\left(s_k\right)=1+\min_{s'\in N(s_k)}\overline{T}\left(s'\right)\leq 1+\overline{T}\left(s_{k+1}\right)=1+m<\infty. \tag{20}$$

In both (19) and (20), we have a contradiction. We conclude that we cannot have $\overline{T}\left(s_t\right)<\infty$ for any t; consequently (17) is true. Hence we have completed the proof of (13)–(14) for all s such that $\overline{T}(s)=\infty$. □

Remark 3.7 It is worth noting that the VL algorithm implements both *Backward Induction* and *Value Iteration*.

The following generalizes the well-known concepts of cop-win and robber-win graphs.

Definition 3.8 The game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ is called *pursuer-win* (*P-win*) iff P^1 has a strategy $\overline{\sigma}^1$ which effects capture for every evader strategy σ^2; $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ is called *evader-win* (*E-win*) iff P^2 has a strategy $\overline{\sigma}^2$ which avoids capture for every pursuer strategy σ^1. In other words

$$\left(\mathbf{N}, S^1, S^2, S_c, s\right) \text{ is } P\text{-}win \text{ iff} : \exists \overline{\sigma}^1 : \forall \sigma^2 : T\left(\overline{\sigma}^1, \sigma^2 | s\right) < \infty, \tag{21}$$

$$\left(\mathbf{N}, S^1, S^2, S_c, s\right) \text{ is } E\text{-}win \text{ iff} : \exists \overline{\sigma}^2 : \forall \sigma^1 : T\left(\sigma^1, \overline{\sigma}^2 | s\right) = \infty \tag{22}$$

Note that the opposite of (21) is

$$\forall \sigma^1 : \exists \sigma^2_{\sigma^1} : T\left(\sigma^1, \sigma^2_{\sigma^1} | s\right) = \infty, \tag{23}$$

(i.e., the evader strategy $\sigma^2_{\sigma^1}$ which ensures no capture takes place will in general depend on the pursuer strategy σ^1) which is *not* equivalent to (22). Hence we cannot *automatically* conclude that a game is either P-win or E-win. But this can be *proved*, as seen next.[5]

Proposition 3.9 *Every* $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ *is either P-win or E-win.*

Proof The game $\left(\mathbf{N}, S^1, S^2, S_c, s\right)$ has value $\overline{T}(s)$ and optimal strategies $\widehat{\sigma}^1, \widehat{\sigma}^2$. We have two cases: either $\overline{T}(s) < \infty$ or $\overline{T}(s) = \infty$. If $\overline{T}(s) < \infty$ then we also have

$$\forall \sigma^2 : T\left(\overline{\sigma}^1, \sigma^2 | s\right) \leq \overline{T}(s) < \infty \Rightarrow \left(\mathbf{N}, S^1, S^2, S_c, s\right) \text{ is } P\text{-win.}$$

If, on the other hand, $\overline{T}(s) = \infty$ then we also have

$$\forall \sigma^1 : T\left(\sigma^1, \overline{\sigma}^2 | s\right) = \overline{T}(s) = \infty \Rightarrow \left(\mathbf{N}, S^1, S^2, S_c, s\right) \text{ is } E\text{-win.}$$

This completes the proof. □

4 Comparison to Other Approaches

Our analysis of Sect. 3 is heavily inspired by [5,9]. Let us compare these two papers and also the related [2] to the game-theoretic approach. These excellent papers have provided the inspiration and foundation for our own approach.

4.1 Hahn and MacGillivray

In [9] Hahn and MacGillivray study, a CR version with two generalizations of the classic game: (i) the game is played on a directed graph and (ii) more than one cops and/or robbers ("k-cop, l-robber") may be involved[6]. On the other hand, following the classic CR formulation, they count time (and thus *capture time*) in *rounds*; one round includes one move by each cop and robber token. While we consider the single cop and single robber case, our own formulation can easily accommodate all of the above.

Next we describe two more substantial differences between Hahn and MacGillivray's approach and our own. These are really differences between the games being studied in each case. Namely, in [9]:

[5] This point is also raised in [10], where an interesting alternative approach is proposed to characterize robber-win graphs.

[6] Let us stress that they still deal with a two-player game: there is a single cop player and a single robber player, but each can control one or more (cop or robber) *tokens*.

1. it is assumed that the two players move alternately and the game always starts with the cop moving first (once again this follows the classic CR game formulation);
2. the game starts with an empty graph, the cop's first move is to place his token on some vertex and the robber's first move is to place his own token; these two moves constitute the "*placement round*".

However both of the above differences can be easily accommodated by our approach. Obviously, removing the "alternating moves" assumption makes our analysis more general. To accommodate the "placement round," we can use a one-round game which consists of two turns: first the cop chooses a vertex x_0^1, then the robber chooses a vertex x_0^2 and then the CR game is played according to our rules. At the game conclusion, the robber gains (the cop loses) $\widehat{T}\left(x_0^1, x_0^2, 1\right)$ payoff units, where $\widehat{T}\left(z^1, z^2, 1\right)$ is the one computed for every vertex pair $\left(z^1, z^2\right)$ by the VL algorithm. Clearly the new game has a value which is

$$\min_{x_0^1} \max_{x_0^2} \widehat{T}\left(x_0^1, x_0^2, 1\right).$$

Hence the solution of our GCR game also provides the solution to Hahn and MacGillivray's (classic) CR game.

One of the main components of [9] is a vertex labeling algorithm very similar to our own, which is used to compute optimal capture times (counted in *rounds*) and strategies. The main properties of this algorithm are established in Lemma 4 of [9]. Both "*strategy*" and "*optimal strategy*" are used informally in [9]. "*Strategy*" is not defined. "*Optimal strategy*" for the cop is defined informally as follows: "a strategy from a configuration c_{xy} [is] *optimal* for the cop if no other strategy gives a win in fewer moves." This needs some clarification: cop strategies *can* give capture time *better than optimal* but only for *some* robber strategies. Similar remarks can be made regarding the definition of optimal robber strategies.

On the other hand, in the game-theoretic approach "*optimal strategy*" is defined by first defining "*strategy*" (as a function from histories to moves) and then providing an optimality criterion, connected with "game value" (which is neither defined nor used in [9]). In the context of CR and GCR, the appropriate definitions are Definitions 2.16 and 2.17 of our Sect. 2.3.

However, the game-theoretic approach can be introduced in [9] with minor modifications. For instance, their statement "*the cop's move will be to an x' ... from which, by the induction hypothesis, the cop can win in $t-1$ rounds*" should be augmented by: "*no matter how the robber plays*" [7]. Similar remarks apply to other parts of [9].

4.2 Bonato and MacGillivray

As already stated, our main inspiration is [5], in which Bonato and MacGillivray generalize the games and results of [9]. In place of "cop" and "robber," they use the terms "pursuer" and "evader." "Alternating moves" and "placement round" are used in the same manner as in [9]. On the other hand, capture is understood in a more general sense; slightly paraphrasing [5], the pursuer wins if, at any time-step, the current position of the game belongs to the subset of *final positions*. Of course this is exactly analogous to our capture set S_c.

[7] Of course this is just a verbal description of the sup and inf conditions of our Definition 2.16.

A vertex labeling algorithm is also provided in [5]; it counts time in turns (not rounds) and is essentially the same as our own VL Algorithm[8]. However, rather than proving directly the properties of their algorithm, the authors proceed in the following manner.

1. They construct, independently of the labeling algorithm, a sequence of orderings $\preceq_0$, $\preceq_1$, ... on pursuer and evader *positions*.
2. They prove that these converge to an ordering $\preceq$.
3. They relate winning and "optimal" game duration to $(\preceq_i)_{i\in\mathbb{N}_0}$ and $\preceq$ (their Theorem 3.1 and Corollary 3.2).
4. Finally they relate state labels to the orderings $(\preceq_i)_{i\in\mathbb{N}_0}$ (Theorem 3.3).

Hence the vertex labeling algorithm is peripheral, rather than central, to the arguments of [5]. "Strategy," "value" and "optimality" are not formally defined. An informal definition of optimality is that "the pursuer's optimal strategy is to move so that the game is over as quickly as possible, and the evader's optimal strategy is to move so the game lasts as long as possible". Similarly to [9], this definition does not clarify the role of the "other" player's strategy. From the game-theoretic point of view, the definition should be modified as follows: "the pursuer's optimal strategy is to move so that the *longest possible* duration of the game is as short as possible" and a similar modification should be applied to the definition of the evader's optimal strategy; these are verbal descriptions of the $\inf_{\sigma^1}\sup_{\sigma^2}$ and $\sup_{\sigma^2}\inf_{\sigma^1}$ conditions on capture time and a condition must be added regarding equality of $\inf_{\sigma^1}\sup_{\sigma^2}$ and $\sup_{\sigma^2}\inf_{\sigma^1}$, as in our Definition 2.16.

4.3 Berarducci and Intrigila

The earliest investigation of time optimal CR strategies that we know of is the one presented in [2] by Berarducci and Intrigila. They *do* provide a definition of strategies, both general and positional. In their Remark 2.2, they apparently assume implicitly that an optimal solution can be found by considering only positional strategies but they actually justify (post facto) this assumption.

Interestingly, the results of [2] are established by using a sequence of sets $W_0, W_1, \ldots$, rather than a labeling algorithm. The sequence is defined inductively: Their W_0 is our capture set S_c and, for each n, W_n is defined (their Definition 2.4) in a manner which strongly resembles our VL Algorithm. The subsequent arguments (contained in their Lemmas 2.5-2.7) resemble the analysis of our VL Algorithm. Their main results are the following.

1. The set W_n is the set of all these starting states from which the cop *can* capture the robber in n moves or less (their Lemma 2.5).
2. The sequence of sets W_n converges to a set W which has the following property: for every starting state $s \in W$ the cop *can* capture the robber in a finite number of moves; for every starting state $s \notin W$ the cop *cannot* capture the robber in a finite number of moves (their Lemma 2.6).
3. Optimal cop and robber strategies are also defined in the proof of Lemma 2.6 and they are, by their definition, positional.

While in the above results Berarducci and Intrigila make no explicit mention of the "other player's" strategy, the use of "can" implies that the cop has a strategy which *guarantees capture no matter how the robber plays*. Similarly, the use of "cannot" implies that the robber has a strategy which *guarantees noncapture no matter how the cop plays*.

[8] There is one caveat: it is never specified in [5] whether once a state achieves a finite label can be subsequently relabeled (it should not); this is probably an oversight in the description.

Keeping the above remarks in mind, the proof of the above results is very close to a game-theoretic one. We believe that the important quantities are not the sets W_n but the sets $U_n = W_n \backslash W_{n-1}$. While not explicitly stated, it follows from their proof that U_n is the set of initial states from which

1. the cop *can* capture the robber in *at most* n rounds, *no matter how the robber plays*;
2. but the robber *can* delay capture for *at least* $n-1$ rounds, *no matter how the cop plays*.

In short, the analysis of [2] respects, at least implicitly, all the relevant game-theoretic considerations and is also related to the previously mentioned vertex labeling algorithm. For example, it is easy to prove that, reverting to our own terminology, the state s belongs to U_n iff $\widehat{T}(s) = n$.

5 Concluding Remarks

We have presented a self-contained game-theoretic solution of the GCR game. Our main tool is the VL algorithm which is based on similar algorithms introduced in [5,9]. Our main contribution is the presentation in greater detail and precision of certain implicit assumptions of [5,9]. Let us also mention that in a longer version of the current paper [14] we have presented an alternative proof of our main result, which is *not* based on the VL algorithm.

We conclude the current paper by listing (i) further generalizations of the GCR game and (ii) well-known families of games which contain GCR as a special case.

Generalizations of GCR The GCR game, as presented in both [5] and the current paper, is a perfect information, two-person, zero-sum game. All of these aspects can be generalized.

1. *Concurrent GCR* A standard assumption of both classic CR and GCR is that a single player moves in each turn of the game. An obvious generalization is to allow both players to move concurrently. In this case, the game no longer has perfect information. It still has a value which, however, will in general be achieved by randomized optimal strategies. An exploration of this direction appears in [15].
2. *Nonzero-sum GCR* By modifying the payoff function, we can obtain a two-player nonzero sum CR game. For example, introducing "energy cost," the robber's payoff could be the capture time minus the distance he has travelled and the cop's payoff could be the negative of the sum of capture time and the distance he has traveled. Another example of a two-player, nonzero-sum GCR game has been presented in [11]; it involves two *selfish* cop players who attempt to catch a "passive" robber (i.e., the robber is not controlled by a player but follows a predetermined path, known to both cop players); the capturing cop player receives a higher payoff than the noncapturing one.
3. *Multiplayer GCR* The classic CR game and practically all its published variants are *two*-player, *zero* -sum games; the same holds for the GCR of [5]. While such games may involve more than one pursuer, all pursuers are controlled by a single player whose payoff is given by a single function. On the other hand, in [12] we have studied an N-player (with $N \geq 2$), *nonzero*-sum version of the classic CR, the so-called *selfish cops and adversarial robber* (SCAR) game. SCAR involves several *selfish* cops, each controlled by a separate player; all cop players share the goal of catching the robber but each cop player has his own payoff function which assigns a higher reward to the player who actually affects the capture; the robber player wants, as in the classic CR game, to delay capture as long as possible. We generalize this approach in [13], introducing *N-player Generalized CR Games*.

Additional Game Families

1. *Stochastic Games* All of the above presented GCR games can be formulated as stochastic games and, using standard results [7], the following can be shown for every GCR game:
   (a) if it is zero-sum, it possesses a value and optimal strategies, which can be computed by the *Value Iteration Algorithm*, a generalization of our VL algorithm;
   (b) if it is nonzero-sum, it possesses at least one *Nash Equilibrium* in deterministic positional strategies [12,13].
2. *Reachability games* The two-player, zero-sum GCR game (understood in the sense of either the current paper or [5]) can also be seen as a special type of *reachability game* [3,17]. In a reachability game, the first player's objective is to bring the game to a target state and the second player's objective is to keep the game away from all target states. This is very similar to the GCR game except that no assumption is made regarding the players' locations. Indeed, a reachability game can be represented by a tuple $\left(\mathbf{N}, S^1, S^2, S\right)$ where $\mathbf{N}$ represents an abstract collection of successor states. Our solution of the GCR game can be applied to any reachability game. However, the "usual" way to solve a reachability game is by constructing a sequence of *attractor sets*; this approach is practically identical to the one used in [2] to solve the classic CR game.
3. *Graphical games* Reachability games are perhaps the simplest example of *infinite perfect information games* [1,3,17,22] which can also be understood as games in which two or more players move a token along the edges of a graph (hence the term "*graphical games*"). Various *infinitary* winning conditions can be used which, in general, depend on some property of the entire game history (for example, Player 1 wins if a certain state is visited infinitely often). In the most general setting, we can have games with any number of players and nonzero-sum winning conditions.

Data availability statement This paper has no associated data and consequently there is no data to be made available. Ath. Kehagias and G. Konstantinidis

References

1. Apt KR, Grädel E (2011) Lectures in game theory for computer scientists. Cambridge University Press, Cambridge
2. Berarducci A, Intrigila B (1993) On the cop number of a graph. Adv Appl Math 14(4):389–403
3. Berwanger D (2013) "Graph games with perfect information." Preprint
4. Bonato A, Nowakowski R (2011) The game of cops and robbers on graphs. American Mathematical Society, USA
5. Bonato A, MacGillivray G (2017) Characterizations and algorithms for generalized Cops and Robbers games. Contributions to Discrete Mathematics, vol.12
6. Chung TH, Hollinger GA, Isler V (2011) Search and pursuit-evasion in mobile robotics. Auton Robots 31:299–310
7. Filar J, Vrieze K (1996) Competitive Markov Decision Processes
8. Fomin FV, Thilikos DM (2008) An annotated bibliography on guaranteed graph searching. Theor Comput Sci 399(3):236–245
9. Hahn G, MacGillivray G (2006) A note on k-cop, l-robber games on graphs. Discret Math 306(19–20):2492–2497
10. Ibragimov G, Luckraz S (2017) On a characterization of evasion strategies for pursuit-evasion games on graphs. J Optim Theory Appl 175(2):590–596
11. Kehagias A, Konstantinidis G (2017) Selfish cops and passive robber: qualitative games. Theor Comput Sci 680:25–35
12. Kehagias A, Konstantinidis G (2019) Selfish cops and active robber: multi-player pursuit evasion on graphs. Theor Comput Sci 780:84–102

13. Kehagias A (2019) Generalized cops and robbers: a multi-player Pursuit game on graphs. Dyn Games Appl 9:1076–1099
14. Kehagias A, Konstantinidis G. “Some Game Theoretic Remarks on Two-Player Generalized Cops and Robbers Games.” arXiv:2007.14758
15. Konstantinidis G, Kehagias Ath (2016) Simultaneously moving cops and robbers. Theor Comput Sci 645:48–59
16. Luckraz S (2019) A survey on the relationship between the game of cops and robbers and other game representations. Dyn Games Appl 9(2):506–520
17. Mazala R (2002) Infinite games. Automata logics, and infinite games. Springer, Berlin, Heidelberg, pp 23-38
18. Nowakowski R, Winkler P (1983) Vertex-to-vertex pursuit in a graph. Discret Math 43:235–239
19. Quilliot A (1978) Thèse de 3ème cycle, Université de Paris VI, pp 131–145
20. Quilliot A (1983) Problemes de jeux, de point fixe, de connectivite et de representation sur des graphes, des ensembles ordonnes et des hypergraphes
21. Quilliot A (1985) A short note about pursuit games played on a graph with a given genus. J Comb Theory Ser B 38:89–92
22. Ummels M (2010) Stochastic multiplayer games: theory and algorithms. Amsterdam University Press, Amsterdam