Fractional Combinatorial Two-Player Games

Frédéric Giroire, Nicolas Nisse, Stéphane Pérennes, Ronan Pardo Soares

Fractional Combinatorial Two-Player Games

Frédéric Giroire, Nicolas Nisse, Stéphane Pérennes, Ronan Soares

RESEARCH REPORT
N° 8371
February 2013
Project-Teams COATI

ISSN 0249-6399 ISRN INRIA/RR--8371--FR+ENG

Fractional Combinatorial Two-Player Games

Frédéric Giroire[*†], Nicolas Nisse[*†], Stéphane Pérennes[*†], Ronan Soares[†*‡]

Project-Teams COATI

Research Report n° 8371 — February 2013 — 18 pages

Abstract: During the last decades, many combinatorial games involving two persons playing on a (directed) graph have received a lot of attention. Some examples of such games are the *Angel problem*, the *Cops and Robbers*, the *Surveillance game*, the *Eternal Dominating Set* and *Eternal Set Cover*. One of the main questions in these games is to decide if a given player has a winning strategy. That is, if it can always win regardless of behaviour of the other player. This question is often NP-hard. In the *Cops and Robbers* game and for the *Surveillance game* this question is PSPACE-complete [Mamino 2012, Fomin *et al.* 2012].
In this paper, we propose a fractional relaxation of these games. That is, we present a framework, based on linear programming techniques, that can be used to model any of the aforementioned games and some of its variants. As far as we know, it is the first time that such combinatorial games have been studied in this way and perspectives are promising.
We also propose an algorithm that decides whether the first player can win the game in at most t turns. Moreover, under a weak assumption which is valid for all the aforementioned games, the fractional game gives us a lower bound for the integral game. For the *Surveillange game* and the *Angel problem* we show that there is, with high probability, a winning strategy which is in a $O(\log n)$ factor of the correspondent fractional parameter against a surfer or angel that follows a particular behaviour. On the other hand, we prove that our framework cannot be used to approximate the classical *Cops and Robber* game.

Key-words: Combinatorial games, Cops and Robber games, linear programming

* Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
† Inria, France
‡ ParGO, Universidade Federal do Ceará, Fortaleza, Brazil

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Jeux Combinatoires Fractionnaires

Résumé :

Mots-clés :

1 Introduction

During the last decades, a huge amount of research has been devoted to the study of combinatorial games involving two players. In particular, many such games are played in graphs and the basic question is to minimize the amount of resources required by one of the player to win regardless of the strategy of the other player. These games arise from a wide range of applications, from the practical ones such as search and rescue, military strategy to trajectory tracking (e.g., *Pursuit-evasion games*), surveillance, monitoring (e.g., *Surveillance games*), etc. to the handling of abstract mathematical and theoretical computer science concepts (e.g., *Cops and Robber games*). Unfortunately, most of the related combinatorial optimization problems are NP-hard or even PSPACE-hard, and few or none approximation results are known.

In this paper, we propose a new and general framework based on linear programming to study two-player games on graphs. We propose a unified definition including (but not limited to) many well studied combinatorial games. We then show how our definition allows us to provide a fractional relaxation (in the sense of linear programming) of these games. As an application, in some cases, we prove that the integrality gap is bounded with high probability. The results we obtained so far show that our approach is promising for obtaining approximation algorithms for some existing games.

1.1 Two-Players games in graphs: towards fractionality

In this section, we give examples of combinatorial games that have attracted much attention during the last years. For two of these games, we describe their *fractional* counterpart to better understand the concept of a fractional game and to illustrate its interest.

Cops and Robber game. One of the most famous combinatorial game is probably the *Cops and Robber game* (see the recent book [4]). In this game, Player $\mathcal{C}$ (the *cop-player*) controls a team of $k \in \mathbb{N}$ cops that aims at capturing a *robber* controlled by Player $\mathcal{R}$ (the *robber-player*) in a graph G. The game proceeds as follows. First, $\mathcal{C}$ places its cops on nodes of the graph, then $\mathcal{R}$ chooses one vertex to place its robber. Then, turn-by-turn, each player moves each of its tokens along at most one edge of G. The cop-player wins if eventually one of its cops occupies the same node as the robber. Otherwise, $\mathcal{R}$ wins. The *cop-number* of a graph G, denoted by $\mathrm{cn}(G)$, is the smallest number k of cops ensuring that $\mathcal{C}$ has a winning strategy [1]. That is, if $\mathcal{C}$ is allowed to use k cops, then $\mathcal{C}$ wins regardless of the behaviour of $\mathcal{R}$. For instance, it is well known (and easy to prove) that 2 cops are necessary and sufficient to capture a robber in a 4-node cycle C_4, i.e., $cn(C_4) = 2$.

Since the seminal work of Nowakowski and Winkler [13] and Quilliot [14] that characterized graphs G with $\mathrm{cn}(G) = 1$, many studies of this game have led to a better understanding of graphs structures, e.g., decomposition of graphs with bounded genus [1, 16] or graphs excluding a fixed minor [2] using shortest paths, tree-decomposition of graphs with small induced cycles [10], structure of random graphs [11], etc. The problem of computing the cop-number has been shown PSPACE-hard in [12] and cannot be approximated up to a ratio $O(\log n)$ in n-node graphs (unless P=NP) [7].

The main principle of fractional games is to relax the constraint that both players have to use their tokens (the cops and the robber) as indivisible (integral) entities. As a first example, let us assume now that the cop-player is allowed at each step (including the initial one) to split its cops into arbitrary small pieces and to move these pieces independently. In that case, the following strategy is clearly winning for Player $\mathcal{C}$: initially $\mathcal{C}$ chooses 3 nodes of C_4 and places 1/2 of cops on each of them and then, at its first turn, $\mathcal{C}$ can move two 1/2 cops to the node occupied by the robber. In other words, Player $\mathcal{C}$ can win in C_4using at most 3/2 cops which is less than $cn(C_4)$. It is clear that, when only Player $\mathcal{C}$ is allowed to split its token, the number of cops necessary for $\mathcal{C}$ to win cannot increase. This actually remains true even if the robber can split itself: intuitively, each cop can be divided according to the same proportion as the robber, and each part of a cop is responsible for a fraction of the robber.

That is, the minimum (fractional) number of cops that are needed to capture the robber in a graph G in the *fractional Cops and Robber game*, where both players can split their token, is a lower bound on $cn(G)$.

Surveillance game. Another (less known) combinatorial game is the *Surveillance game* that have been defined for its applications in telecommunication networks, in particular to model prefetching problems [6, 8]. In this game, the first player $\mathcal{C}$ can mark at most $k \in \mathbb{N}$ nodes of a (directed) graph G at each turn while the second player $\mathcal{R}$ slides one agent on the edges (arcs). More precisely, initially, player $\mathcal{C}$ marks one predetermined (part of the input) node $v_0 \in V(G)$ and $\mathcal{R}$ must place its agent on it. Then, alternatedly, $\mathcal{C}$ marks at most k nodes and $\mathcal{R}$ may move along an edge. $\mathcal{R}$ wins if its agent reaches an unmarked node. Otherwise, after $\left\lceil \frac{|V(G)|-1}{k} \right\rceil$ turns, all nodes are marked, then the game stops and $\mathcal{C}$ wins. The *surveillance number* of a (di)graph G, denoted by $\mathrm{sn}(G)$, is the minimum $k \in \mathbb{N}$ such that player $\mathcal{C}$ has a winning strategy marking at most k vertices per turn. Unfortunately, this game appears to be very hard even in specific graph classes, e.g., deciding whether $\mathrm{sn}(G) \leq 2$ is NP-hard in chordal graphs and deciding whether $\mathrm{sn}(D) \leq 4$ is PSPACE-complete in Directed Acyclic Graphs [6]. While polynomial-time algorithms exist in trees and interval graphs [6], no approximation algorithms (nor inapproximability results) are known for general graphs.

The problem of computing the surveillance number of a graph is NP-hard even when the game is restricted to two turns (Player $\mathcal{R}$ can move at most twice) [6]. Now, consider the relaxed game where Player $\mathcal{C}$ can mark fractions of nodes. That is, at its turn, the only constraint is that the sum of what it has marked is at most k. In that case, we show that the complexity of the problem of computing the surveillance number changes. Indeed, when Player $\mathcal{R}$ may move at most twice (i.e., the game consists of two turns), then the strategy of $\mathcal{C}$ can easily be described as follows: at its first turn, $\mathcal{C}$ marks all the $|N(v_0)|$ neighbors[1] of the starting position v_0 and then marks $k - |N(v_0)|$ nodes at distance 2 of v_0, then whatever be $w \in N(v_0)$ where $\mathcal{R}$ moves its token, $\mathcal{C}$ must be able to mark all unmarked neighbors of w. For any node w at distance 2 of v_0, let us define the real variable $m_w \in [0, 1]$ that indicates the portion of w that must be marked by $\mathcal{C}$ during the first turn. Then, the following linear program (which is a simple instance of hitting set) can compute the *fractional surveillance number* in polynomial time when the game is limited to two turns:

$$
\begin{array}{lcl}
 & \min & k \\
\text{subject to} & & \\
 & \sum_{w \in N_2(v_0)} m_w & \leq k \\
\forall x \in N(v_0) & |N(x) \cap N_2(v_0)| - \sum_{w \in N(x) \cap N_2(v_0)} m_w & \leq k \\
\forall w \in N_2(v_0) & m_w \in [0,1] &
\end{array}
$$

Other examples and common properties. The list of examples of similar games is quite long and include among others: the *Angel problem* [3], the *Eternal Dominating Set* [5], the *Eternal Vertex Cover* [9], and variants of the *Cops and Robbers* game (e.g., [7]) and of the *Surveillance game* [8]. While each of these games has its own specificities, they all share common characteristics. In particular, in each of these games, two adversaries are playing by acting alternatedly on the vertices of a graph (either marking some node or moving some token

[1] Given a graph G and $v \in V(G)$, we set $N(v)$ as the set of neighbors of v and let $N_2(v)$ be the set of nodes at distance exactly two from v.

from one node to another one, etc.) with perfect information on the current state (*configuration*) of the game. Moreover, in all these games, we are interested in optimizing the amount of some resource to be given to one of the player in order to ensure its victory whatever be the strategy of the other player.

Towards fractional games. In the above two examples, note that we only relax the behaviour of Player $\mathcal{C}$, i.e., providing it more power. In what follows, we allow both players to use fraction of tokens. This allows us to design a general algorithm for computing if Player $\mathcal{C}$ has a winning strategy. One of our main result is that, in general games, we can also relax the behaviour of $\mathcal{R}$ without giving it more power.

1.2 Results and Outline of the Paper

In Section 2, we formally define a fractional two-players game on graphs.

In Section 3, we prove our main result which is the design of an algorithm that decides whether player $\mathcal{C}$ has a winning strategy, i.e., can win the game whatever does $\mathcal{R}$. This algorithm, however, has a step that is not polynomial in the size of the input graph.

It is easy to see that player $\mathcal{C}$ has a winning strategy in the integral game only if it has a winning strategy in the semi-fractional game. In Section 4, we prove that, under weak hypothesis, player $\mathcal{C}$ has a winning strategy in the fractional game if, and only if, it has a winning strategy in the semi-fractional game. This shows that, for some games the resources, used by the pursuer are the same in the fractional and in the semi-fractional version. That is, given that the pursuer is playing in a fractional manner, allowing the evader to play in a fractional manner does not help the pursuer.

Then, in Section 5, we focus on some particular games by presenting some results for the *Cops and Robbers* game, the *Angel problem* and the *Surveillance game*. We show that if the fractional parameter of the *Surveillance game* or the *Angel problem* is k then, with high probability, there is a winning strategy for $\mathcal{C}$ that uses at most $O(k \log n)$ if Player $\mathcal{R}$ follows a random walk. In the fractional *Cops and Robbers* game, we prove that $1+\epsilon$, $\epsilon > 0$, cops are enough to capture the robber.

Finally, in Section 6, we conclude this paper with some open questions.

2 Description of a Turn-by-Turn Pursuit-Evasion Game

We want to define a framework that is general enough to model all the aforementioned games. Roughly, we define a game where two players play by moving or adding token on the vertices of the graph. With the help of sets defining what are the possible initial positioning of these tokens, how these tokens can be moved and how each player wins we are able to model each of the aforementioned game.

Let $\mathcal{C}$ (the pursuer) and $\mathcal{R}$ (the evader) be the two players that play on a directed graph $G = (V, E)$ with $n \in \mathbb{N}$ nodes. Let $V = \{1, \dots, n\}$. In order to formally describe a fractional turn-by-turn pursuit-evasion game, or simply combinatorial game, we need to introduce some notation. For any vector $x \in \mathbb{R}^n$ and for any $1 \leq i \leq n$, let x_i be the i^{th} coordinate of x, i.e., $x = (x_1, \dots, x_n)$. The concatenation of two vectors $x, y \in \mathbb{R}^n$ is denoted by $(x, y) = (x_1, \dots, x_n, y_1, \dots, y_n) \in \mathbb{R}^{2n}$. The sum of two vectors $x, y \in \mathbb{R}^n$ is denoted by $x + y = (x_1 + y_1, \dots, x_n + y_n) \in \mathbb{R}^n$.

The game involves two players, $\mathcal{C}$ and $\mathcal{R}$, that play alternatedly on an n-node graph. A *configuration* of the game is represented by a vector $(c, r) \in \mathbb{R}_+^{2n}$ where c and r belong to $\mathbb{R}_+^n$. Intuitively, the i^{th} coordinate of c (resp., of r) represents the amount of tokens of player $\mathcal{C}$ (resp., of player $\mathcal{R}$) on the node $v_i \in V(G)$, $1 \leq i \leq n$. When it is its turn, one player can perform a move, that is, it can modify the current configuration of the game by following some rules described below. Given a configuration $(c, r) \in \mathbb{R}_+^{2n}$, player $\mathcal{C}$ (resp., of player $\mathcal{R}$) can only modify c (resp., r). Before formally describing the game, we introduce some definitions. The moves of the players will be defined by the following operators.

- Let $\mathcal{X}_\mathcal{C} \subseteq \mathbb{R}^n$ and $\mathcal{X}_\mathcal{R} \subseteq \mathbb{R}^n$ be any two convex sets containing 0_n and defined by a polynomial (in n) number of constraints.
- Let Δ_G be a set of left stochastic matrices defined by G as follows: for all $\delta \in \Delta_G$, we have that $\delta \in [0,1]^n \times [0,1]^n$ and if (i,j) is not an arc of G then $\delta_{i,j} = 0$. Note that Δ_G is convex and contains the identity matrix.

To understand the intuition behind any matrix in Δ_G, assume that a player has put some tokens on the vertices of G and let $x \in \mathbb{R}^n$ be the vector representing these tokens, i.e., x_i is the amount of tokens on node $v_i \in V(G)$, $1 \le i \le n$. Then, for any $\delta \in \Delta_G$, $\delta x \in \mathbb{R}^n$ represents the state after some tokens have have moved (depending on δ) along edges of G. More precisely, for any $1 \le i,j \le n$, $\delta_{i,j}$ represents the fraction of tokens initially present in $v_j \in V(G)$ that moved along $\{v_j, v_i\} \in E(G)$ to reach $v_i \in V(G)$.

On the other hand, the vectors in $\mathcal{X}_\mathcal{C}$ and $\mathcal{X}_\mathcal{R}$ will be used to add or remove tokens from nodes of G. For any $y \in \mathcal{X}_\mathcal{C}$ (or $y \in \mathcal{X}_\mathcal{R}$), $x + y$ represents the new state after some tokens have been added or removed to the configuration x. More precisely, for any $1 \le i,j \le n$, y_i is the variation of tokens on node v_i (without considering the movement of the tokens along edges incident to v_i).

Now, let us define some particular configurations that will be used to precise a Fractional game: let $\mathcal{V} \subseteq \mathbb{R}_+^{2n}$ be a non empty polytope with number of facets polynomial in n, $\mathcal{V}$ is called the set of *valid configurations*; let $\mathcal{I} \subseteq \mathcal{V}$ be any non empty set, $\mathcal{I}$ is the set of *initial configurations*; let $\mathcal{W}_\mathcal{C} \subseteq \mathcal{C}$ be a polytope with number of facets polynomial in n, this is the set of *winning configurations for* $\mathcal{C}$; let $\mathcal{W}_\mathcal{R} = \mathbb{R}_+^{2n} \setminus \mathcal{V}$, this is the set of *winning configurations for* $\mathcal{R}$; let $F \in \mathbb{N}$ be the maximum number of turns the game is allowed to last; finally, let *Last* $\in \{\mathcal{C}, \mathcal{R}\}$ be the player that wins if the game lasts more than F turns.

Now, we are ready to formally define the general game with parameters $\{\mathcal{V}, \mathcal{I}, \mathcal{W}_\mathcal{C}, \mathcal{X}_\mathcal{C}, \mathcal{X}_\mathcal{R}, \Delta_G, F, Last\}$.

1. Initially, $\mathcal{C}$ chooses any vector $c_0 \in \mathbb{R}_+^n$ such that there exists $r \in \mathbb{R}_+^n$ with $(c_0, r) \in \mathcal{I}$. Then, $\mathcal{R}$ chooses any vector $r_0 \in \mathbb{R}_+^n$ such that $(c_0, r_0) \in \mathcal{I}$. $(c_0, r_0) \in \mathcal{I}$ is then the *initial configuration* of the game.
   - If $(c_0, r_0) \in \mathcal{W}_\mathcal{C}$, then player $\mathcal{C}$ wins and the game is over.
   - Else, if $F = 0$, then player *Last* wins and the game is over.

   Otherwise, at each turn $t \ge 1$, there are two steps:
2. First, player $\mathcal{C}$ chooses $\delta \in \Delta_G$ and $x \in \mathcal{X}_\mathcal{C}$ such that $y = (\delta c_{t-1} + x, r_{t-1}) \in \mathcal{V}$. Then, player $\mathcal{C}$ moves to the configuration $(c_t, r_{t-1}) = y$.
   - If $(c_t, r_{t-1}) \in \mathcal{W}_\mathcal{C}$, then player $\mathcal{C}$ wins and the game is over after t turns.
3. Otherwise, $\mathcal{R}$ chooses $\delta \in \Delta_G$ and $x \in \mathcal{X}_\mathcal{R}$ such that $y = (c_t, \delta r_{t-1} + x) \notin \mathcal{W}_\mathcal{C}$. Note that, because $I_{n\times n}$ (identity matrix) is in Δ_G and $0_n \in \mathcal{X}_\mathcal{R}$, then there always exists such y. Then, player $\mathcal{R}$ moves to the configuration $(c_t, r_t) = y$.
   - if $y \notin \mathcal{V}$, then player $\mathcal{R}$ wins and the game is over after t turns.
   - else, if $t \ge F$, then player *Last* wins and the game is over.
   - Else, the next turn $t+1$ starts (Goto 2).

A *winning strategy* for player $\mathcal{C}$ consists of a vector c_0 and a function $\sigma : \mathbb{R}^{2n} \to \mathcal{X}_\mathcal{C} \times \Delta_G$ that allows player $\mathcal{C}$ to win whatever be the behavior of player $\mathcal{R}$. That is, player $\mathcal{C}$ chooses c_0 as initial vector, and then, at each turn t, it moves to $(\delta c_{t-1} + x, r_{t-1})$ where $(x, \delta) = \sigma((c_{t-1}, r_{t-1}))$. Following this process, player $\mathcal{C}$ must win in any execution of the game.

3 Algorithm to Compute a Winning Strategy for player $\mathcal{C}$

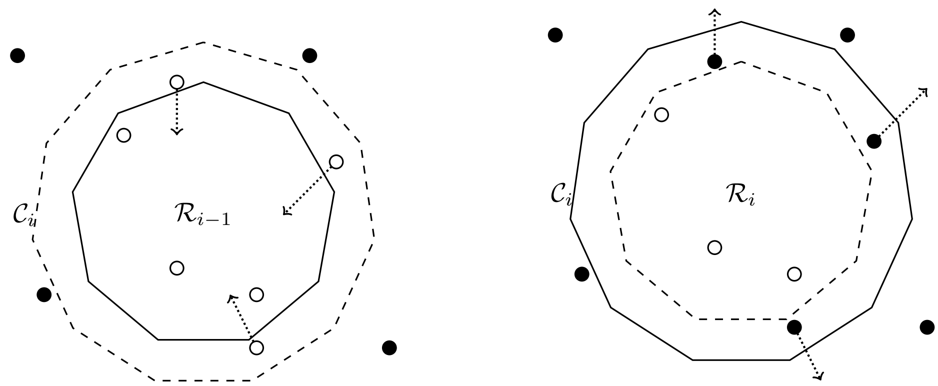
In this section, we describe an algorithm that given a game $\{\mathcal{V}, \mathcal{I}, \mathcal{W}_\mathcal{C}, \mathcal{X}_\mathcal{C}, \mathcal{X}_\mathcal{R}, \Delta_G, F, Last\}$, decides whether there is a winning strategy for player $\mathcal{C}$.

Roughly, this is done by starting with a set C of configurations which are winning for $\mathcal{C}$ in t turns, meaning that starting the game from any configuration in this set, the game can always be won by $\mathcal{C}$ in at most t turns, and computing a set $C' \supseteq C$. This set C' is such that any configuration in C' is winning for $\mathcal{C}$ in at most $t+1$ turns. Then, we iterate this process until we get a set C^* such that any configuration in C^* is winning for $\mathcal{C}$ in at most F turns.

Let us define the following sets.

- For any $t \in \mathbb{N}^*$, let $\mathcal{C}_t \subseteq \mathcal{V}$ be the set of configurations such that, for any configuration $m \in \mathcal{C}_t$, there is a strategy with initial configuration m that allows player $\mathcal{C}$ to win in at most t turns. That is, there is a winning strategy for $\mathcal{C}$ in the game $\{\mathcal{V}, \mathcal{C}_t, \mathcal{W}_\mathcal{C}, \mathcal{X}_\mathcal{C}, \mathcal{X}_\mathcal{R}, \Delta_G, t, Last\}$.
- Let $\mathcal{R}_0 = \mathcal{W}_\mathcal{C}$ and, for any $t \in \mathbb{N}^*$, let $\mathcal{R}_t \subseteq \mathcal{V}$ be the set of configurations m such that for every move of player $\mathcal{R}$ from m to m' we have that $m' \in \mathcal{C}_t$. That is, even when the first player to play is $\mathcal{R}$, we have that $\mathcal{C}$ wins if the starting configuration is one in $\mathcal{R}_t$.

Roughly, $\mathcal{C}_i$ can be obtained by the union of all points in $\mathcal{R}_{i-1}$ and all elements in $(c, r) \in \mathbb{R}^{2n}$ such that there exists $x \in \mathcal{X}_\mathcal{C}$ and $\delta \in \Delta_G$ with $(\delta c + x, r) \in \mathcal{R}_{i-1}$, see Lemma 1. On the other hand, $\mathcal{R}_i$ can be obtained by taking $\mathcal{C}_i$ and removing all $(c, r) \in \mathbb{R}^{2n}$ such that there exists $x \in \mathcal{X}_\mathcal{R}$ and $\delta \in \Delta_G$ with $(c, \delta r + x) \notin \mathcal{C}_i$, see Lemma 3. A scheme for this can be found in Figure 1.



(a) Scheme for obtaining $\mathcal{C}_i$.

(b) Scheme for obtaining $\mathcal{R}_i$.

Figure 1: On (a), a scheme on how $\mathcal{C}_i$, denoted by the dashed polygon, is obtained by taking $\mathcal{R}_{i-1}$, denoted by the full polygon, union some points in $\mathbb{R}^{2n}$. Dashed arrows represent a possible movement for the $\mathcal{C}$. Black points represent configurations such that no movement of $\mathcal{C}$ can move them inside $\mathcal{R}_{i-1}$. The respective scheme for obtaining $\mathcal{R}_i$ is shown in (b).

Starting from $\mathcal{R}_0 = \mathcal{W}_\mathcal{C}$, our algorithm iteratively, for any $0 < t \leq F$, build $\mathcal{C}_t$ from $\mathcal{R}_{t-1}$ and $\mathcal{R}_t$ from $\mathcal{C}_t$. Then, the desired strategy exists if and only if there is $c_0 \in \mathbb{R}^n$ such that for all $r \in \mathbb{R}^n$ with $(c_0, r) \in \mathcal{I}$ then $(c_0, r) \in \mathcal{C}_F$.

Lemma 1 $\mathcal{C}_{t+1} = \{(c, r) \in \mathcal{V} \mid \exists x \in \mathcal{X}_\mathcal{C}, \exists \delta \in \Delta_G, (\delta c + x, r) \in \mathcal{R}_t\}$.

Lemma 2 *Let $t \geq 0$ and assume that $\mathcal{R}_t \subseteq \mathbb{R}_+^{2n}$ is a convex set described by ℓ linear inequalities and $2n$ variables. Then, there is an algorithm that computes a set of linear inequalities describing $\mathcal{C}_{t+1}$.*

Let R be the linear program describing $\mathcal{R}_{t-1}$. The program R has variables c_i and r_i, $1 \leq i \leq n$, that representing the configurations of the game. To describe $\mathcal{C}_t$, we add to R new variables c'_i, $i \leq n$, describing the configuration of $\mathcal{C}$ at this turn, variables x_i ($i \leq n$) representing the amount of marks that must be added at node i by $\mathcal{C}$ at this turn and variables $a_{i,j}$ representing the amount of marks that the $\mathcal{C}$ moves from vertex j to vertex i. Then, we add to R the system of inequalities describing $\mathcal{X}_{\mathcal{C}}$ and the system of inequalities describing Δ_G. This guarantees that c can be obtained by a move $x \in \mathcal{X}_{\mathcal{C}}$ and $a \in \Delta_G$ on c'. Note that up to this point R can be obtained in polynomial time on n and ℓ. In fact, R is a system of inequalities describing $\mathcal{C}_t$ where x_i and $a_{i,j}$ are auxiliary variables. In order to eliminate all the auxiliary variables of R, we apply the well known Fourier-Motzkin elimination method [15] on each variable x_i, $a_{i,j}$ and c_i. After this step, R has only $2n$ variables given by c'_i and r_i, R is a system of inequalities describing $\mathcal{C}_i$ and, due to the method used, a number of inequalities that might not be polynomial in n and ℓ.

Lemma 3 $\mathcal{R}_t = \{(c, r) \in \mathbb{R}^{2n} \mid \forall x \in \mathcal{X}_{\mathcal{R}}, \forall \delta \in \Delta_G, (c, \delta r + x) \in \mathcal{C}_t\} \cap \mathcal{V}$.

Lemma 4 *Let $t \geq 0$ and assume that $\mathcal{C}_t \subseteq \mathbb{R}^{2n}_+$ is a set described by ℓ linear inequalities and $2n$ variables. Then, there is an algorithm polynomial in ℓ and n that computes a set of ℓ linear inequalities and $2n$ variables describing $\mathcal{R}_t$.*

A state (c, r) is in $\mathcal{R}_t$ if for every possible move (x, δ) of $\mathcal{R}$ the state $(c, \delta r + x)$ satisfies all inequalities describing $\mathcal{C}_t$. Let $A_i(c, r) \leq b_i$ be an inequality in the linear program describing $\mathcal{C}_t$. The main tool used in the proof of Lemma 3 is that there are specific elements $\delta_i \in \Delta_G$ and $x_i \in \mathcal{X}_{\mathcal{R}}$ such that if $A_i(c, \delta_i r + x_i) \leq b_i$ then $A_i(c, \delta r + x) \leq b_i$ for all other elements $\delta \in \Delta_G$ and $x \in \mathcal{X}_{\mathcal{R}}$. In other words, for each inequality describing $\mathcal{C}_t$ there is a "best" move for $\mathcal{R}$ in order to violate this inequality. Another important property of these specific elements is that they can be found in time polynomial in n. The linear program for $\mathcal{R}_t$ is obtained by taking each inequality, $A_i(c, r) \leq b_i$, describing $\mathcal{C}_t$ and rewriting it in the following manner. Let (B_1, B_2) be equal to A_i. We can rewrite $A_i(c, r) \leq b_i$ as $B_1 c + B_2 r \leq b_i$. Then, in the linear program for $\mathcal{R}_t$, the inequality $A_i(c, r) \leq b_i$ is replaced by $B_1 c + B_2 \delta_i r \leq b_i - x_i$.

Hence, by applying Lemma 2 and Lemma 4 successively, we are able to construct $\mathcal{C}_F$ from $\mathcal{R}_0 = \mathcal{W}_{\mathcal{C}}$. However, this construction might take more than polynomial time in F and n, due to the Fourier-Motzkin elimination.

4 Semi-Fractional and Integral Games

In this section, we define the semi-fractional and integral games related to the general fractional game studied above. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{I}, \mathcal{W}_{\mathcal{C}}, \mathcal{X}_{\mathcal{C}}, \mathcal{X}_{\mathcal{R}}, \Delta_G, F, Last\}$ be a fractional game as defined in Section 2. The corresponding *integral game* is defined by $\mathcal{G}_{int} = \{\mathcal{V}, \mathcal{I} \cap \mathbb{N}^{2n}, \mathcal{W}_{\mathcal{C}} \cap \mathbb{N}^{2n}, \mathcal{X}_{\mathcal{C}} \cap \mathbb{N}^n, \mathcal{X}_{\mathcal{R}} \cap \mathbb{N}^n, \Delta_G \cap \mathbb{N}_{n \times n}, F, Last\}$, and and the *semi-fractional game* by $\mathcal{G}_{sf} = \{\mathcal{V}, \mathcal{I} \cap (\mathbb{R}^n \times \mathbb{N}^n), \mathcal{W}_{\mathcal{C}}, \mathcal{X}_{\mathcal{C}}, \mathcal{X}_{\mathcal{R}} \cap \mathbb{N}^n, (\Delta_G^{\mathcal{C}} = \Delta_G, \Delta_G^{\mathcal{R}} = \Delta_G \cap \mathbb{N}_{n \times n}), F, Last\}$, where the rules of the game are exactly the same as in Section 2. Note that we distinguished the two sets $\Delta_G^{\mathcal{C}}$ and $\Delta_G^{\mathcal{R}}$. Indeed, the game proceeds as before, but player $\mathcal{R}$ is constrained to move only on integral configurations in both games, while $\mathcal{C}$ is only constrained to move only on integral configuration in the integral game.

The next Lemma directly follows from the definition of the games.

Lemma 5 *Let $\mathcal{G}$ be a fractional game.*
$\mathcal{C}$ has a winning strategy in $\mathcal{G}$ only if it has a winning strategy in $\mathcal{G}_{sf}$.
$\mathcal{C}$ has a winning strategy in $\mathcal{G}_{int}$ only if it has a winning strategy in $\mathcal{G}_{sf}$.

We prove that under a small extra assumption, fractional and semi-fractional games are equivalent. Intuitively, assume that $\mathcal{C}$ can win against any integral strategy of $\mathcal{X}_\mathcal{R}$. Now assume that $\mathcal{X}_\mathcal{R}$ can split each of its tokens into two half-tokens following two distinct strategies. Then, $\mathcal{C}$ will also use half-tokens to win against the strategy of the first half-tokens of $\mathcal{X}_\mathcal{R}$, and the second half of the tokens of $\mathcal{C}$ will win against the strategy followed by the remaining half-tokens of $\mathcal{X}_\mathcal{R}$. By convexity of the moves of $\mathcal{C}$, this is a valid strategy. There is also another proof, based on the fact that if the polytopes $\mathcal{X}_\mathcal{R}$, $\mathcal{X}_\mathcal{C}$ and $\mathcal{W}_\mathcal{C}$ have integral coordinates, then δ_i and x_i in the Lemma 4 have also integral coordinates. Meaning that the "best" move for $\mathcal{R}$ is integral, when playing in the fractional game.

Theorem 1 *If all the vertices of the polytopes $\mathcal{X}_\mathcal{R}$, $\mathcal{X}_\mathcal{C}$ and $\mathcal{W}_\mathcal{C}$ have integral coordinates and if $\mathcal{I} \subseteq \mathbb{R}^n \times \mathbb{N}^n$, then: player $\mathcal{C}$ has a winning strategy in $\mathcal{G}$ if and only if it has a winning strategy in $\mathcal{G}_{sf}$.*

5 Applications in Combinatorial Games

In this section, we discuss how to model some turn-by-turn pursuit-evasion games with the framework given in Section 2, while also studying the gap between the fractional and integral parameters of such games.

5.1 Surveillance Game

The classical *Surveillance game* also fits our framework. Consider an observer that can mark at most k vertices at each turn and assume that the game is played on a graph $G = (V, E)$ with $V = \{1, \dots, n\}$ where the initial vertex is vertex 1. Then, the *fractional Surveillance game* can be defined with the help of the following sets: $\mathcal{I} = \{(c, r) \mid c_1 = 1, r_1 = 1, \forall i \in V(G) \backslash \{1\}, c_i = 0, r_i = 0\}$, the only possible initial state is the one where the initial vertex is completely marked and the surfer is entirely contained in it; $\mathcal{V} = \{(c, r) \in \mathbb{R}_+^{2n} \mid \forall i \in V(G), c_i \geq r_1, \sum r_i = 1\}$, the surfer does not win the game until it is able to escape the marked area; $\mathcal{X}_\mathcal{C} = \{x \in \mathbb{R}_+^n \mid \sum_{i=1}^n x_i \leq k\}$, the observer is allowed to mark at most k vertices of the graph during its turn; $\Delta_\mathcal{C} = \{I_{n \times n}\}$, the observer might not move marks along edges of the graph; $\mathcal{X}_\mathcal{R} = \{(0, \dots, 0)\}$ and $\Delta_\mathcal{R} = \Delta_G$, the surfer moves only by sliding its tokens along edges of the graph; Since the observer can mark k vertices per turn, after $F = \lceil \frac{n}{k} \rceil$ rounds all vertices are marked. *Last* $= \mathcal{C}$, again, if the game lasts F rounds, the observer wins; finally, the observer also wins if all vertices are marked, $\mathcal{W}_\mathcal{C} = \{(c, r) \in \mathbb{R}_+^{2n} \mid \forall i \in V(G), c_i = 1\}$.

Theorem 2 *If $\mathcal{C}$, the observer, wins the* fractional Surveillance game *with k marks in an n-node graph, then $\mathcal{C}$ wins the* Surveillance game *with high probability if it is allowed to use $O(k \log n)$ marks against a blind $\mathcal{R}$.*

In other words, if $\mathcal{C}$ wins the fractional Surveillance game *against a surfer following a random walk (or a predetermined path that is unknown to $\mathcal{C}$) with k marks, then it has a high probability of winning against an integral surfer following the same random walk (or path) with $O(k \log n)$ marks.*

The proof of Theorem 2 closely follows that of the $\log n$ approximation for set cover in [17]. Roughly, by considering the amount of marks put on a vertex in the fractional game as a probability of marking this vertex in the integral game, we get a strategy that is winning with high probability.

5.2 Angel problem

Given a graph G, let Δ_G^a be the set of matrices that can be obtained by multiplying any a-tuple of matrices in Δ_G and let $N^s(i)$ be the set of vertices of $V(G) \setminus \{i\}$ that are at distance at most s from i.

The *Angel problem* game where a devil that can mark, or "eat", at most k vertices and an angel that can move along at most s edges at each turn can be modeled with the following sets, we assume that the game is played on a graph $G = (V, E)$ with $V = \{1, \dots, n\}$ and that the initial vertex is vertex 1: $\mathcal{I} = \{(c_1, \dots, c_n, r_1, \dots, r_n) \mid c_1 = 0, r_1 = 1, \forall i \in V(G) \setminus \{1\}, c_i = 0, r_i = 0\}$, the only possible initial state is the one where the surfer is entirely contained in the initial vertex and no other vertex is marked, or eaten; $\mathcal{V} = \{(c, r) \in \mathbb{R}_+^{2n} \mid \sum r_i = 1\}$, the amount of angel does not change during the game; $\mathcal{X_C} = \{x \in \mathbb{R}_+^n \mid \sum_{i=1}^n x_i \leq k\}$ and $\Delta_\mathcal{C} = \{I_{n\times n}\}$, the devil can only eat vertices of the graph and it may not "move" tokens through edges of it; $\mathcal{X_R} = \{(0, \dots, 0)\}$, $\Delta_\mathcal{R} = \Delta_G^s$, the angel may move by sliding through at most s edges during its turn; $Last = \mathcal{R}$, the angel wins if it is able to survive long enough; $\mathcal{W_C} = \{(c, r) \in \mathbb{R}_+^{2n} \mid \forall i \in V(G), \sum_{j\in N^s(i)} c_j \geq r_i\}$, the devil wins if it is able to "eat" all vertices around the angel; finally, $F \in \mathbb{N}$.

The proof of Theorem 3 is very similar with the proof of Theorem 2, since both games have a similar set of rules for the moves of the players.

Theorem 3 *If $\mathcal{C}$, the devil, wins the* fractional Angel problem *game with k marks in an n-node graph, then the devil wins the* Angel problem *game with high probability if it is allowed to use $O(k \log n)$ marks against a blind $\mathcal{R}$.*

In other words, if the devil wins the fractional Angel problem *against an angel following a random walk (or a predetermined path that is unknown to $\mathcal{C}$) with k marks, then it has a high probability of winning against an integral angel following the same random walk (or path) with $O(k \log n)$ marks.*

5.3 Cops and Robbers

The classical *Cops and Robbers* game fits our framework. Indeed, consider the *Cops and Robbers* game played with $k > 0$ cops on a graph $G = (V, E)$ of order n. This game can be defined using the following sets: $\mathcal{I} = \mathcal{V} = \{(c, r) \in \mathbb{R}_+^{2n} \mid \sum r_i = 1, \sum c_i = k\}$, $\mathcal{X_R} = \mathcal{X_C} = \{(0, \dots, 0)\}$, $\Delta_\mathcal{C} = \Delta_\mathcal{R} = \Delta_G$, $Last = \mathcal{R}$ and $\mathcal{W_C} = \{(c, r) \in \mathcal{V} \mid \forall i \in V, c_i \geq r_i\}$. While we can limit F to be at most n^{k+1}, since there are at most n^{k+1} possible configurations for the integral game, we leave F undefined. That is, $F = \infty$.

Let $\text{fcn}(G)$ be the smallest k such that the cops have a finite winning strategy, i.e., they can win in a finite number of steps whatever the robber does.

Claim 1 *For any graph G, $1 \leq \text{fcn}(G) \leq$ fractional domination number(G)*

Proof. Clearly, from their definition, $1 \leq \text{fcn}(G)$. To see that $\text{fcn}(G) \leq$ fractional dominating number(G), let S be a fractional dominating set of G. Assume that $V(G) = \{1, \dots, n\}$. Then, let s_i be the amount of vertex v_i that is on S. Hence, for all $v \in V$, $\sum_{i\in N[v]} s_i = 1$. Therefore, by placing s_i cops on each vertex i during its positioning, we have that the robber can be captured by the cops in their next move. ■

Lemma 6 *There are graphs G such that* $\text{fcn}(G) > 1$.

Proof. Consider any graph containing a cordless cycle with four nodes. Consider any fractional strategy with one cop. The robber chooses first a node v such that $N[v]$ contains less than 1 cop. Then, there is a node, not in $N[v]$, where there is at least $\epsilon > 0$ cops. During the next step and remaining on the cycle, the robber can maintain a distance at least one between itself and a proportion $\epsilon' > 0$ cops of these ϵ cops. ■

Theorem 4 *For any graph G and for any $\beta > 0$, $\text{fcn}(G) \leq 1 + \beta$. Moreover, there is a finite winning strategy that allows the cops to capture the robber in a linear number of turns.*

The idea behind the proof of Theorem 4 is that, once a small amount of cops is on the same vertex as the robber, this small amount can follow the robber until the end of the game. Then, by repeatedly spreading the cops uniformly through all the vertices of the graph we can guarantee that at least a small amount of cops is on the same vertex of the robber. This small amount of cops then is dedicated to follow the robber while the rest of the cops repeat process recursively.

6 Conclusion

Although the proof in this paper are restrict to games where both players are allowed to slide and add tokens, the results also hold for games in which: players can not slide tokens ($\Delta_G = \{I_{n\times n}\}$); Δ_G is different for each player (that is, one player may slide tokens along edges, while the other may not); several types of tokens for each player, or several cops/several robbers; both players can move more than once in their turn (for example, both cops and robbers with speed $s > 1$ in the *Cops and Robbers* game); tokens are on edges instead of vertices.

We finish this paper with some open questions. A first open question is the complexity of fractional pursuit-evasion games. Our algorithm, due to the elimination step, is not polynomial. However, it seems that this elimination creates several redundant constraints. For example, in the *Surveillance game*, if $F = 2$ then the system of inequalities describing $\mathcal{C}_2$ can is roughly the same as the fractional set cover which can be solved in polynomial time.

Another open question is how big can the gap between the fractional and the integral *Cops and Robbers* game when the robber has speed more than one.

Albeit, the approximation results for the *Angel problem* and the *Surveillance game* help us win the integral game against a blind opponent based on a winning fractional strategy, they do not help us bound the gap between the fractional and integral parameters of these games.

References

[1] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8:1–12, 1984.

[2] T. Andreae. On a pursuit game played on graphs for which a minor is excluded. *J. Comb. Theory, Ser. B*, 41(1):37–47, 1986.

[3] E. R. Berlekamp, J. H. Conway, and R. K. Guy. Winning ways for your mathematical plays. *Games in Particular*, 2, 1982.

[4] A. Bonato and R. Nowakovski. *The game of Cops and Robber on Graphs*. American Math. Soc., 2011.

[5] A. P. Burger, E. J. Cockayne, W. R. GgÃ¼ndlingh, C. M. Mynhardt, J. H. Van Vuuren, and W. Winterbach. Infinite order domination in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 50:179–194, 2004.

[6] F. V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To satisfy impatient web surfers is hard. In *6th Int. Conf. on Fun with Algorithms (FUN)*, volume 7288 of *LNCS*, pages 166–176, 2012.

[7] F. V. Fomin, P. A. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theor. Comput. Sci.*, 411(7-9):1167–1181, 2010.

[8] Frédéric Giroire, Dorian Mazauric, Nicolas Nisse, Stéphane Pérennes, and Ronan P. Soares. Connected surveillance game. In *20th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Lecture Notes in Computer Science. Springer, 2013.

[9] W.F. Klostermeyer and C.M. Mynhardt. Graphs with equal eternal vertex cover and eternal domination numbers. *Discrete Mathematics*, 311(14):1371–1379, 2011.

[10] A. Kosowski, B. Li, N. Nisse, and K. Suchan. k-chordal graphs: From cops and robber to compact routing via treewidth. In *39th Int. Col. on Automata, Languages, and Programming (ICALP (2))*, volume 7392 of *LNCS*, pages 610–622, 2012.

[11] T. Luczak and P. Pralat. Chasing robbers on random graphs: Zigzag theorem. *Random Struct. Algorithms*, 37(4):516–524, 2010.

[12] M. Mamino. On the computational complexity of a game of cops and robbers. *CoRR*, abs/1202.6043, 2012.

[13] R. J. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Maths*, 43:235–239, 1983.

[14] A. Quilliot. *Problèmes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes.* Thèse de doctorat d'état, Université de Paris VI, France, 1983.

[15] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley Series in Discrete Mathematics & Optimization. John Wiley & Sons, 1998.

[16] B. S. W. Schröder. The copnumber of a graph is bounded by $\lfloor \frac{3}{2} genus(g) \rfloor + 3$. *Categorical perspectives (Kent, OH, 1998), Trends in Mathematics*, pages 243–263, 2001.

[17] V.V. Vazirani. *Approximation Algorithms.* Springer, 2004.

A Proofs

Lemma 7 *For all $t \in \mathbb{N}$:*

$$\mathcal{C}_{t+1} = \{(c,r) \in \mathcal{V} \mid \exists x \in \mathcal{X}_\mathcal{C}, \exists \delta \in \Delta_G, (\delta c + x, r) \in \mathcal{R}_t\}.$$

Proof. Let $R = \{(c,r) \in \mathbb{R}^{2n} \mid \exists x \in \mathcal{X}_\mathcal{C}, \delta \in \Delta_G, (\delta c + x, r) \in \mathcal{R}_t\} \cap \mathcal{V}$.

For any $m = (c,r) \in R \subseteq \mathcal{V}$, we show that there is a strategy for $\mathcal{C}$ to win the game in at most $t+1$ turns starting from m. Indeed, by definition of R, there are $x \in \mathcal{X}_\mathcal{C}$ and $\delta \in \Delta_G$ such that $c' = \delta c + x$ with $(c', r) \in \mathcal{R}_t \subseteq \mathcal{V}$. Then, in configuration (c,r), player $\mathcal{C}$ moves to (c', r). Since $(c', r) \in \mathcal{R}_t$, for any move of player $\mathcal{R}$, say it moves to (c', r'), then $(c', r') \in \mathcal{C}_t$ by definition of $\mathcal{R}_t$. Finally, by definition of $\mathcal{C}_t$, there is a strategy that allows $\mathcal{C}$ to win in at most t turns starting from (c', r'). Hence, $R \subseteq \mathcal{C}_{t+1}$.

Reciprocally, let $(c,r) \in \mathcal{C}_{t+1} \subseteq \mathcal{V}$. By definition, there is a strategy σ that allows $\mathcal{C}$ to win in at most $t+1$ turns starting from (c,r). Let $(x,\delta) = \sigma((c,r)) \in \mathcal{X}_\mathcal{C} \times \Delta_G$ and $c' = \delta c + x$. Since σ is winning, whatever be the move (c', r') of player $\mathcal{R}$ from (c', r), player $\mathcal{C}$ wins in at most t turns starting from (c', r'). Hence, $(c', r) \in \mathcal{R}_t$. Therefore, $(c,r) \in R$ and $\mathcal{C}_{t+1} \subseteq R$. ■

Lemma 8 *Let $t \geq 0$ and assume that $\mathcal{R}_t \subseteq \mathbb{R}_+^{2n}$ is a convex set described by ℓ linear inequalities and $2n$ variables. Then, there is an algorithm that computes a set of linear inequalities describing $\mathcal{C}_{t+1}$.*

Proof. Let us consider the following convex set R.

$$\begin{array}{lll}
(c', r) = (c'_1, \dots, c'_n, r_1, \dots, r_n) \in \mathbb{R}_+^{2n} & & \\
\text{Subject to} & & \\
(c', r) \in \mathcal{V} & & (1) \\
c_i = x_i + a_{i,i} + \sum_{1 \leq j \leq n, \{i,j\} \in E(G)} a_{i,j} & \forall 1 \leq i \leq n & (2) \\
(c_1, \dots, c_n, r_1, \dots, r_n) \in \mathcal{R}_t & & (3) \\
(x_1, \dots, x_n) \in \mathcal{X}_\mathcal{C} & & (4) \\
c'_i = a_{i,i} + \sum_{1 \leq j \leq n, \{i,j\} \in E(G)} a_{j,i} & \forall 1 \leq i \leq n & (5.a) \\
a_{i,j} \geq 0 & \forall 1 \leq i,j \leq n & (5.c) \\
a_{j,i} = 0 & \forall i \neq j \in [1,n]^2, \{j,i\} \notin E(G) & (5.b)
\end{array}$$

$\mathcal{V}$ and $\mathcal{X}_\mathcal{C}$ are convex sets defined by polynomial (in n) number of linear inequalities. Therefore, $p(n) = O(n^k)$ for some fixed k.

By hypothesis, $\mathcal{R}_t$ is a convex set defined by ℓ linear inequalities. Since there are at most $O(n^2)$ linear equations (2) and (5) with $O(n^2)$ new variables, the above linear program has a total of $\ell + O(\max\{p(n), n^2\})$ linear inequalities and $O(n^2)$ variables.

Moreover, given the set of inequalities defining $\mathcal{V}$, $\mathcal{X}_\mathcal{C}$ and $\mathcal{R}_t$, the above set of inequalities can be computed in time $O(\ell + \max\{p(n), n^2\})$. Note that if ℓ can be bounded by a polynomial in n and t then R can be constructed in polynomial-time (in n and t).

Now, let us show that $\mathcal{C}_{t+1}$ can be described by the above system of linear inequalities by projecting R over the variables $c'_1, \dots, c_n$ and $r_1, \dots, r_n$. That is, (c', r) belongs to R projected into $c'_1, \dots, c_n$ and $r_1, \dots, r_n$ if and only if $(c', r) \in \mathcal{C}_{t+1}$. Indeed,

(c', r) belongs to R projected into $c'_1, \dots, c_n$ and $r_1, \dots, r_n$ if and only if there exist values of c_i, x_i and $a_{i,j}$, for $1 \leq i,j \leq n$, such that $(c', c, x, a, r) \in R$.

$\Leftrightarrow$ $(c', r) \in \mathcal{V}$ and there exist $x = (x_1, \dots, x_n) \in \mathcal{X}_\mathcal{C}$ and $A = [a_{i,j}]_{1 \leq i,j \leq n} \in \mathbb{R}^+_{n \times n}$ such that $(A1_n + x, r) \in \mathcal{R}_t$, where $1_n = (1, \dots, 1) \in \mathbb{R}^n$, and for all $i \leq n$, $a_{i,i} + \sum_{1 \leq j \leq n, \{i,j\} \in E(G)} a_{j,i} = c_i$ and $a_{j,i} = 0$ for any $j \neq i$, $\{j,i\} \notin E(G)$.

$\Leftrightarrow$ $(c', r) \in \mathcal{V}$ and there exist $x \in \mathcal{X}_\mathcal{C}$ and $\delta = [\alpha_{i,j}]_{1 \leq i,j \leq n} = [\frac{a_{i,j}}{c_j}]_{1 \leq i,j \leq n} \in \Delta_G$ such that $(\delta c' + x, r) \in \mathcal{R}_t$.

$\Leftrightarrow$ $(c', r) \in \mathcal{C}_{t+1}$, by Lemma 1.

The set R, however, has several variables that are auxiliary. It is necessary to eliminate the variables c_i, x_i and $a_{i,j}$ for all $1 \leq i,j \leq n$. For this we successively use the the Fourier-Motzkin elimination method [15] on these variables. See below for a brief discussion on how this method works. Since there are $2n+n^2$ variables in total that we want to eliminate, R' obtained after eliminating all c_i, x_i and $a_{i,j}$ variables might have a number of linear inequalities that is not polynomial in the size of R. Therefore, the size of R' is not polynomial in n and t, even if ℓ is.

Since there are only $2n$ variables in R' and that R' is a projection of R into $c'_1,\dots,c'_n$ and $r_1,\dots,r_n$, we have that $R' = \mathcal{C}_{t+1}$. ■

Remark 1 *For the sake of completeness we briefly illustrate the Fourier-Motzkin elimination method. Let $A_{\ell\times n}x \leq b$ be a system of linear inequalities. Assume that we want to eliminate the last variable of the vector x from this system. Let x_n be this variable. We first rewrite all inequalities such that $a_{i,1} \neq 0$ in order to isolate x_n. That is, every inequality such that the coefficient x_n is not 0 is rewritten as (Type 1) $x_n \geq$ "something" or (Type 2) $x_n \leq$ "something". Note that the coefficient of x_n is 1. Then, there are two cases to consider:*

- *There are only inequalities of the form $x_n \geq$ "something" or there are only inequalities of the form $x_n \leq$ "something". In this case we simply remove all these inequalities from $Ax \leq b$.*
- *If there are both types of inequalities, then we combine each inequality of (Type 1) with each inequality of (Type 2). That is, for each pair of inequalities $x_n \leq$ "somethingA" and $x_n \geq$ "somethingB", we add the inequality "somethingB" $\leq$ "somethingA" to $Ax \leq b$. Then, we remove all inequalities such that the coefficient of x_n is non-zero.*

This method guarantees that, after eliminating a variable, the result is a system of inequalities $A'_{\ell'\times n-1}x' \leq b'$ such that $A'x' \leq b'$ has a solution if and only if $Ax \leq b$ has a solution. Moreover, if x' is a solution for $A'_{\ell'\times n-1}x' \leq b'$ then there is $x = (x_1,\dots,x_n)$ with $(x_1,\dots,x_{n-1}) = (x'_1,\dots,x'_{n-1})$ such that x is a solution to $Ax \leq b$. In other words, we this process projects the set described by $Ax \leq b$ into its first $n-1$ variables.

While we remove some inequalities, a single execution of this method, however, might add $\ell^2/4$ new inequalities, where ℓ is the number of initial inequalities. Hence, in order to eliminate d variables, we might add $4(\ell/4)^{2^d}$ inequalities.

Lemma 9 *Let $t \geq 0$ and assume that $\mathcal{C}_t \subseteq \mathbb{R}^{2n}_+$ is a convex set described by ℓ linear inequalities and $2n$ variables. Then, there is a polynomial-time algorithm in ℓ and n that computes a set of at most ℓ linear inequalities and $2n$ variables describing $\mathcal{R}_t$.*

Proof. By the hypothesis, there exist $A \in \mathbb{R}^{\ell\times 2n}$ and $b = (b_1,\dots,b_\ell) \in \mathbb{R}^\ell$ such that $\mathcal{C}_t = \{m \in \mathbb{R}^{2n}_+ \mid Am \leq b\}$.

For any $1 \leq i \leq \ell$, let $(z_{i,1},\dots,z_{i,n},a_{i,1},\dots,a_{i,n})$ be the i^{th} row of A. Let $A_i = (a_{i,1},\dots,a_{i,n})$.

- Let $b'_i = \max_{x\in\mathcal{X}_\mathcal{R}}\{A_i x\}$ and let $X_i \in \text{argmax}_{x\in\mathcal{X}_\mathcal{R}}\{A_i x\}$. This is computable in polynomial-time in n since $\mathcal{X}_\mathcal{R}$ is a convex set defined by a polynomial number of constraints.
- For any $u \in V(G)$, let $u_i \in \text{argmax}_{v\in N(u)}\{a_{i,v}\}$. Let $\delta_i = [\alpha_{v,u}]_{1\leq u,v,\leq n}$ such that, for any $1 \leq v,u \leq n$, $\alpha_{v,u} = 1$ if $v = u_i$ and $\alpha_{v,u} = 0$ otherwise. Clearly, $\delta_i \in \Delta_G$.

Let us consider the following convex set R.

$$(c,r) = (c_1,\dots,c_n,r_1,\dots,r_n) \in \mathbb{R}^{2n}_+$$

Subject to

$$(c,r) \in \mathcal{V}$$
$$(z_{i,1},\dots,z_{i,n},A_i\delta_i)\cdot(c,r) \leq b_i - b'_i \qquad ,\forall 1 \leq i \leq \ell$$

Since $\mathcal{V}$ is a convex set defined by a size polynomial in n and $\mathcal{C}_t$ is a convex set described ℓ linear inequalities, the above linear system has size polynomial in ℓ and n and can be computed in polynomial-time (in ℓ and n).

It remains to show that:

Claim 2 $R = \mathcal{R}_t$.

Let $(c,r) \in \mathcal{R}_t$. By Lemma 3, $(c,r) \in \mathcal{V}$ and, for any $\delta \in \Delta_G$ and $x \in \mathcal{X}_\mathcal{R}$, $(c, \delta r + x) \in \mathcal{C}_t$. Then, for any $1 \leq i \leq \ell$, $(c, \delta_i r + X_i) \in \mathcal{C}_t$. In other words, $A(c, \delta_i r + X_i)^T \leq b$. In particular, $(z_{i,1}, \ldots, z_{i,n}, A_i) \cdot (c, \delta_i r + X_i) = (z_{i,1}, \ldots, z_{i,n}, A_i\delta_i) \cdot (c,r) + A_i X_i = (z_{i,1}, \ldots, z_{i,n}, A_i\delta_i) \cdot (c,r) + b'_i \leq b_i$. Hence, $(c,r) \in R$.

Let $(c,r) \in R$. Then, $(c,r) \in \mathcal{V}$. Let $\delta = [\alpha'_{i,j}]_{1 \leq i,j \leq n} \in \Delta_G$ and $x \in \mathcal{X}_\mathcal{R}$. We show that $(c, \delta r + x) \in \mathcal{C}_t$. More precisely, we show that $A \cdot (c, \delta r + x) \leq b$. Let $1 \leq i \leq \ell$. Then,

$$(z_{i,1}, \ldots, z_{i,n}, A_i) \cdot (c, \delta r + x) = (z_{i,1}, \ldots, z_{i,n})c + A_i \delta r + A_i x.$$

Since $X_i \in \text{argmax}_{x \in \mathcal{X}_\mathcal{R}} \{A_i x\}$, we have $b'_i = A_i X_i \geq A_i x$. Hence,

$$(z_{i,1}, \ldots, z_{i,n}, A_i) \cdot (c, \delta r + x) \leq (z_{i,1}, \ldots, z_{i,n})c + A_i \delta r + b'_i.$$

Moreover, because $(c,r) \in R$, for any $1 \leq i \leq \ell$, $(z_{i,1}, \ldots, z_{i,n}, A_i\delta_i) \cdot (c,r) \leq b_i - b'_i$. Hence,

$$(z_{i,1}, \ldots, z_{i,n})c + A_i \delta_i r \leq b_i - b'_i.$$

To show that $(z_{i,1}, \ldots, z_{i,n}, A_i) \cdot (c, \delta r + x) \leq b_i$, it remains to prove that $A_i \delta r \leq A_i \delta_i r$

On the one hand,

$$A_i \delta r = \sum_{1 \leq j \leq n} a_{i,j} \sum_{1 \leq k \leq n} \alpha'_{j,k} r_k = \sum_{1 \leq k \leq n} r_k \sum_{1 \leq j \leq n} a_{i,j} \alpha'_{j,k}.$$

Since, for any $1 \leq k \leq n$, $\sum_{1 \leq j \leq n} \alpha'_{j,k} = 1$ and for all $1 \leq j,k \leq n$, $\alpha'_{j,k} \geq 0$ and $r_k \geq 0$, we get that $A_i \delta r \leq \sum_{1 \leq k \leq n} r_k \max_{1 \leq j \leq n} a_{i,j}$.

On the other hand,

$$A_i \delta_i r = \sum_{1 \leq k \leq n} r_k \sum_{1 \leq j \leq n} a_{i,j} \delta_{j,k}.$$

Recall that, by definition, there is exactly one $1 \leq j \leq n$ such that $\delta_{j,k} = 1$, and such that $a_{i,j} = \max_{1 \leq j' \leq n} a_{i,j'}$, and $\delta_{j,k} = 0$ for all the $(n-1)$ other values of j. Therefore, $A_i \delta_i r = \sum_{1 \leq k \leq n} r_k \max_{1 \leq j \leq n} a_{i,j}$. Thus, we got the result, i.e., for any $1 \leq i \leq \ell$, $(z_{i,1}, \ldots, z_{i,n}, A_i) \cdot (c, \delta r + x) \leq b_i$.

Therefore, $A(c, \delta r + x)^T \leq b$ and, by Lemma 3, $(c,r) \in \mathcal{R}_t$. Hence, $R = \mathcal{R}_t$. ■

Lemma 10 *Let $\mathcal{G}$ be a fractional game.*

- *Player $\mathcal{C}$ has a winning strategy in $\mathcal{G}$ only if it has a winning strategy in $\mathcal{G}_{sf}$.*
- *Player $\mathcal{C}$ has a winning strategy in $\mathcal{G}_{int}$ only if it has a winning strategy in $\mathcal{G}_{sf}$.*

Proof. Indeed, any winning strategy in $\mathcal{G}$ (resp., in $\mathcal{G}_{int}$) is a winning strategy in $\mathcal{G}_{sf}$. Indeed, the possible moves and initial configurations of $\mathcal{C}$ in $\mathcal{G}$ are still possible in $\mathcal{G}_{sf}$ while the moves (and initial configurations) of $\mathcal{R}$ are more constrained in $\mathcal{G}_{sf}$. On the other hand, the possible moves and initial configurations of $\mathcal{C}$ in $\mathcal{G}_{int}$ are still possible in $\mathcal{G}_{sf}$ while the moves and initial configurations of $\mathcal{R}$ remains the same in $\mathcal{G}_{int}$ and $\mathcal{G}_{sf}$. ■

Theorem 2 *If all the vertices of the polytopes $\mathcal{X}_\mathcal{R}$, $\mathcal{X}_\mathcal{C}$ and $\mathcal{W}_\mathcal{C}$ have integral coordinates and if $\mathcal{I} \subseteq \mathbb{R}^n \times \mathbb{N}^n$, then:*

Player $\mathcal{C}$ has a winning strategy in $\mathcal{G}$ if and only if it has a winning strategy in $\mathcal{G}_{sf}$.

Proof. By previous lemma, it is sufficient to prove that if $\mathcal{C}$ has a winning strategy in $\mathcal{G}_{sf}$ then it has a winning strategy in $\mathcal{G}$.

For any $1 \leq t \leq F$, $\mathcal{C}_t$ is defined as in Section 3 as the set of configurations from which $\mathcal{C}$ wins in at most t turns in the fractional game. Let $\mathcal{C}_t^{sf} \subseteq \mathcal{C}_t \cap (\mathbb{R}^n \times \mathbb{N}^n)$ be the set of configurations from which $\mathcal{C}$ wins in at most t turns in the semi-fractional game.

Let $\mathcal{R}_0^{sf} = \mathcal{W}_\mathcal{C} \cap (\mathbb{R}^n \times \mathbb{N}^n) = \mathcal{R}_0 \cap (\mathbb{R}^n \times \mathbb{N}^n)$ and, for any any $1 \leq t \leq F$, let $\mathcal{R}_t$ is defined as in Section 3 as the set of configurations from which player $\mathcal{R}$ can only enter in $\mathcal{C}_t$ in the fractional game, i.e., in one (fractional) move. Let $\mathcal{R}_t^{sf} \subseteq \mathcal{V} \cap (\mathbb{R}^n \times \mathbb{N}^n)$ be the set of configurations from which player $\mathcal{R}$ can only enter in $\mathcal{C}_t^{sf}$ in the semi-fractional game, i.e., in one integral move.

Given $X \subseteq \mathbb{R}^{2n}$, let CH$(X)$ be the convex hull of X.

We prove by induction on t that, for any $1 \leq t \leq F$, $\mathcal{C}_t = \mathrm{CH}(\mathcal{C}_t^{sf})$ and $\mathcal{R}_t = \mathrm{CH}(\mathcal{R}_t^{sf})$.

Let $t \geq 0$, and assume for purpose of induction that $\mathcal{R}_t = \mathrm{CH}(\mathcal{R}_t^{sf})$. This is true for $t = 0$ by definition and because the vertices of $\mathcal{W}_\mathcal{C}$ have integral coordinates. By a proof as the one of Lemma 1, $\mathcal{C}_{t+1}^{sf} = \{(c,r) \in \mathcal{V} \cap (\mathbb{R}^n \times \mathbb{N}^n) \mid \exists x \in \mathcal{X}_\mathcal{C}, \exists \delta \in \Delta_G, (\delta c + x, r) \in \mathcal{R}_t^{sf}\}$. Therefore, because $\mathcal{R}_t = \mathrm{CH}(\mathcal{R}_t^{sf})$ by induction, $\mathcal{C}_{t+1}^{sf} = \{(c,r) \in \mathcal{V} \cap (\mathbb{R}^n \times \mathbb{N}^n) \mid \exists x \in \mathcal{X}_\mathcal{C}, \exists \delta \in \Delta_G, (\delta c + x, r) \in \mathcal{R}_t\}$. And thus, $\mathrm{CH}(\mathcal{C}_{t+1}^{sf}) = \mathcal{C}_{t+1}$ by Lemma 1 and because the vertices of $\mathcal{X}_\mathcal{C}$ and Δ_G have integral coordinates.

Let $t > 0$, and assume for purpose of induction that $\mathcal{C}_t = \mathrm{CH}(\mathcal{C}_t^{sf})$. This is true for $t = 1$ by above paragraph. By the same proof as the one of Lemmas 1 and 3, $\mathcal{R}_t^{sf} = \{(c,r) \in \mathcal{V} \cap (\mathbb{R}^n \times \mathbb{N}^n) \mid \forall x \in \mathcal{X}_\mathcal{R} \cap \mathbb{N}^n, \forall \delta \in \Delta_G^R = \Delta_G \cap \mathbb{N}_{n \times n}, (c, \delta r + x) \in \mathcal{C}_t^{sf}\}$. Therefore, by induction, $\mathcal{R}_t^{sf} = \{(c,r) \in \mathcal{V} \cap (\mathbb{R}^n \times \mathbb{N}^n) \mid \forall x \in \mathcal{X}_\mathcal{R} \cap \mathbb{N}^n, \forall \delta \in \Delta_G^R = \Delta_G \cap \mathbb{N}_{n \times n}, (c, \delta r + x) \in \mathcal{C}_t\}$.

The proof is then the similar as the one of Lemma 4. Recall that $\mathcal{C}_t = \{x \in \mathbb{R}_+^{2n} \mid Am \leq b\}$, and $A_i = (a_{i,1}, \ldots, a_{i,n})$ where $(z_{i,1}, \ldots, z_{i,n}, a_{i,1}, \ldots, a_{i,n})$ be the i^{th} row of A, for any $1 \leq i \leq \ell$.

Note first that, because the vertices of $\mathcal{X}_\mathcal{R}$ have integral coordinates, $b'_i = \max_{x \in \mathcal{X}_\mathcal{R}} \{x \cdot A_i\} = \max_{x \in \mathcal{X}_\mathcal{R} \cap \mathbb{N}^n} \{x \cdot A_i\}$ and therefore, there is $X_i \in \mathbb{N}^n$ such that $X_i \in \mathrm{argmax}_{x \in \mathcal{X}_\mathcal{R}} \{x \cdot A_i\}$. Let also $\delta_i \in \Delta_G \cap \mathbb{N}_{n \times n}$ as defined in the proof of Lemma 4.

By the same proof as the one of Lemma 4, it can be proved that $\mathcal{R}_t^{sf}$ is defined by

$$\begin{array}{lll} & (c,r) = (c_1, \ldots, c_n, r_1, \ldots, r_n) \in \mathbb{R}^n \times \mathbb{N}^n & \\ \text{Subject to} & & \\ & (c,r) \in \mathcal{V} & \\ & (z_{i,1}, \ldots, z_{i,n}, A_i \delta_i) \cdot (c,r) \leq b_i - b'_i & , \forall 1 \leq i \leq \ell \end{array}$$

Therefore, because $X_i \in \mathbb{N}^n$ (for any $1 \leq i \leq \ell$) and $\delta_i \in \Delta_G \cap \mathbb{N}_{n \times n}$, we get that $\mathcal{R}_t = \mathrm{CH}(\mathcal{R}_t^{sf})$.

Hence, $\mathcal{C}_t = \mathrm{CH}(\mathcal{C}_t^{sf})$. This is easy to conclude because $\mathcal{I} \subseteq \mathbb{R}^n \times \mathbb{N}^n$. ■

Theorem 3 *If $\mathcal{C}$, the observer, wins the* fractional Surveillance game *with k marks in an n-node graph, then $\mathcal{C}$ wins the* Surveillance game *with high probability if it is allowed to use $O(k \log n)$ marks against a blind $\mathcal{R}$.*

In other words, if $\mathcal{C}$ wins the fractional Surveillance game *against a surfer following a random walk (or a predetermined path that is unknown to $\mathcal{C}$) with k marks, then it has a high probability of winning against an integral surfer following the same random walk (or path) with $O(k \log n)$ marks.*

Proof. The proof of Theorem 2 closely follows that of the $\log n$ approximation for set cover in [17].

Assume that $\mathcal{C}$, the observer, and $\mathcal{R}$, the surfer, play the integral *Surveillance game* on a graph G, that $\mathrm{fsn}(G,v) \leq k$, that $V = \{1, \ldots, n\}$ and that the initial vertex is vertex 1. The initial state of the game is (c', r') such that: if $i \neq 1$ then $c'_i = r'_i = 0$ and if $i = 1$ then $c_i = r_i = 1$. Since, from Section 4, we have that the number of marks necessary for the observer does not change by restricting the surfer to play in an integral manner, assume, moreover, that the surfer moves in an integral manner. That is, in order to move, the surfer chooses a matrix in $\delta \in \Delta_G \cap \mathbb{N}^n$. Since the initial state of the game we have the surfer entirely on vertex 1, this guarantees that the surfer remains integral during all the game.

In the following we describe the strategy of the observer. Let (c,r) be the current state of the game, which is (c', r') on the first turn of the observer. During each turn t of the observer, let the vector $x^t \in \mathcal{X}_\mathcal{C}$ be the the amount of marks used by the observer, in the *fractional Surveillance game*, when the initial state is given by (c, v). That is, $x^t = (x_1, \ldots, x_n)$ is the amount of marks the observer would place on the vertices of G in order to win against the surfer in the *fractional Surveillance Game*. Then, in the integral game, the observer marks a vertex i if among $O(\log n)$ independent random tests with probability x_i at least one of them is a success.

We want to measure the probability that the observer loses, using this strategy, against any strategy for the surfer in the integral game. Let A_i^t be the event that $r_i > c_i$ at step t of the game. In other words, A_i^t is the event that the observer has lost to the surfer because of vertex i at step t.

Then, $P(A_i^t) \leq (x_i^1)^{c\log n}(x_i^2)^{c\log n}\cdots(x_i^t)^{c\log n}$. Since $\text{fsn}(G,v) \leq k$ we have that $\sum_{i=1}^{t} x_i^t = 1$. Therefore, from a simple calculus manipulation, $P(A_i^t)$ is minimum when $x_i^1 = x_i^2 = \cdots = x_i^t = 1/t$. Hence, $P(A_i^t) \leq (\frac{1}{t})^{tc\log n} \leq (\frac{1}{e})^{c\log n}$, where e is the base of the natural logarithm.

Then, the probability that the observer loses the game is given by $P(\bigcup_{t=1}^{F}\bigcup_{i=1}^{n} A_i^t)$. Therefore, $P(\bigcup_{t=1}^{F}\bigcup_{i=1}^{n} A_i^t) \leq n^2(\frac{1}{e})^{c\log n}$. Let $c \geq 3$, then $P(\bigcup_{t=1}^{F}\bigcup_{i=1}^{n} A_i^t) \leq \frac{1}{n}$. Therefore, the observer wins the game with high probability.

Moreover, the expected cost of this strategy is given by $\text{fsn}(G,v)c\log n = O(k\log n)$. ■

Theorem 4 *If $\mathcal{C}$, the devil, wins the* fractional Angel problem *game with k marks in an n-node graph, then the devil wins the* Angel problem *game with high probability if it is allowed to use $O(k\log n)$ marks against a blind $\mathcal{R}$.*

In other words, if the devil wins the fractional Angel problem *against an angel following a random walk (or a predetermined path that is unknown to $\mathcal{C}$) with k marks, then it has a high probability of winning against an integral angel following the same random walk (or path) with $O(k\log n)$ marks.*

Proof. Assume that $\mathcal{C}$, the devil, and $\mathcal{R}$, the angel, play the integral *Angel problem* game on a graph G, that $\text{fang}(G,v) \leq k$, that $V = \{1,\ldots,n\}$ and that the initial vertex is vertex 1. The initial state of the game is (c',r') such that: if $i \neq 1$ then $c_i' = r_i' = 0$ and if $i = 1$ then $c_i = 0$ and $r_i = 1$. Since, from Section 4, we have that the number of marks necessary for the devil does not change by restricting the angel to play in an integral manner, assume, moreover, that the angel moves in an integral manner. That is, in order to move, the angel chooses a matrix in $\delta \in \Delta_G^s \cap \mathbb{N}^n$. Since the initial state of the game we have the angel entirely on vertex 1, this guarantees that the angel remains integral during all the game.

In the following we describe the strategy of the devil. Let (c,r) be the current state of the game, which is (c',r') on the first turn of the devil. During each turn t of the devil, let the vector $x^t \in \mathcal{X}_\mathcal{C}$ be the the amount of marks used by the devil, in the *fractional Angel problem*, when the initial state is given by (c,v). That is, $x^t = (x_1,\ldots,x_n)$ is the amount of marks the devil would place on the vertices of G in order to win against the angel in the *fractional Angel problem*. Then, in the integral game, the devil marks a vertex i if among $O(\log n)$ independent random tests with probability x_i at least one of them is a success.

We want to measure the probability that the devil does not win at step t, using this strategy, against any strategy for the angel in the integral game. Let $A_{i,j}^t$ be the event that, there is $j \in N^a(i)$ such that $r_i > c_j$ at step t of the game. In other words, $A_{i,j}^t$ is the event that the devil does not win against the angel because of the amount of angel at vertex i at step t.

Then, $P(A_{i,j}^t) \leq \sum_{j\in N^a(i)}(x_j^1)^{c\log n}(x_j^2)^{c\log n}\cdots(x_j^t)^{c\log n}$. Since $\text{fang}(G,v) \leq k$ we have that $\sum_{l=1}^{t} x_j^l = 1$. Therefore, from a simple calculus manipulation, $P(A_{i,j}^t)$ is minimum when for all $j \in N^a(i)$ we have $x_j^1 = x_j^2 = \cdots = x_j^t = 1/t$. Hence, $P(A_i^t) \leq \sum_{j\in N^a(i)}(\frac{1}{t})^{tc\log n} \leq n(\frac{1}{t})^{tc\log n} \leq n(\frac{1}{e})^{c\log n}$, where e is the base of the natural logarithm.

Then, the probability that the devil loses the game is given by $P(\bigcup_{t=1}^{F}\bigcup_{i=1}^{n} A_{i,j}^t)$. Therefore, $P(\bigcup_{t=1}^{F}\bigcup_{i=1}^{n} A_{i,j}^t) \leq n^3(\frac{1}{e})^{c\log n}$. Let $c \geq 4$, then $P(\bigcup_{t=1}^{F}\bigcup_{i=1}^{n} A_{i,j}^t) \leq \frac{1}{n}$. Therefore, the devil wins the game with high probability.

Moreover, the expected cost of this strategy is given by $\text{fang}(G,v)c\log n = O(k\log n)$. ■

Theorem 5 *For any graph G and for any $\beta > 0$, $\text{fcn}(G) \leq 1+\beta$. Moreover, there is a (fractional) winning strategy that allows the cops to capture the robber in a linear number of turns.*

Proof. If $G = K_n$, the result is trivial so let us assume that G has minimum degree $\delta < n-1$. Let us define the following strategy. First, the $k = x_0 = 1+\beta$ cops places themselves uniformly at each node (i.e., x_0/n at each node). Then, the robber places itself at some node v. Then, $\delta x_0/n$ cops at the neighbours of v goes to v. At this step, there are $y_1 = (1+\delta)x_0/n$ cops at the same node v as the robber. The remaining amount of the cops is $x_1 = x_0 - y_1 = (1 - \frac{1+\delta}{n})k$.

By induction on $t \geq 0$, assume that $y_t = k - x_t$ cops occupy the same node v as the robber and it is the turn of the robber. Moreover, the remaining amount of cops is $x_t = (1 - \frac{1+\delta}{n})^t k$. Now, the robber moves to a node w adjacent to v. Then, the y_t cops on v move to w and there are two cases to consider:

1. if the x_t remaining cops are not uniformly placed (i.e., x_t/n at each node), they move to achieve such a position. This can be done, in one step, by moving the cops along a spanning tree of G rooted in w, where each vertex moves to its parent an amount of cops that is proportional to the number of its descendants in the spanning tree.
2. else, $\delta x_t/n$ cops at the neighbours of w goes to w. Moreover, before this move, except the y_t cops there are also x_t/n cops at w. Therefore, after this step, there are $y_{t+1} = (1+\delta)x_t/n + y_t$ cops at the same node w as the robber.
   Hence,
   $$x_{t+1} = x_t - \frac{(1+\delta)\, x_t}{n} = x_t \left(1 - \frac{1+\delta}{n}\right) = k \left(1 - \frac{1+\delta}{n}\right)^{t+1}$$
   and
   $$y_{t+1} = k - x_{t+1}.$$

The result follows, essentially, from the fact that $\lim_{t\to\infty} y_t = 1$ when $\beta = 0$ and that there exists $t > 0$ such that $y_t \geq 1 + \beta$ when $\beta > 0$. ■

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399