Theoretical Computer Science 413 (2012) 160–175

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

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

Quantitatively fair scheduling

Alessandro Bianco, Marco Faella [*], Fabio Mogavero, Aniello Murano

Università degli Studi di Napoli "Federico II", Napoli, Italy

ARTICLE INFO

Keywords:
Fairness
Polynomial algorithm
Game
Scheduling
Edge-colored graphs
Linear programming

ABSTRACT

We consider finite graphs whose edges are labeled with elements, called *colors*, taken from a fixed finite alphabet. We study the problem of determining whether there is an infinite path where either (i) all colors occur with a fixed asymptotic frequency, or (ii) there is a constant that bounds the difference between the occurrences of any two colors for all prefixes of the path. These properties can be viewed as quantitative refinements of the classical notion of fair path in a concurrent system, whose simplest form checks whether all colors occur infinitely often. Our notions provide stronger criteria, particularly suitable for scheduling applications based on a coarse-grained model of the jobs involved. In particular, they enforce a given set of priorities among the jobs involved in the system. We show that both problems we address are solvable in polynomial time, by reducing them to the feasibility of a linear program. We also consider two-player games played on finite colored graphs where the goal is one of the above frequency-related properties. For all the goals, we show that the problem of checking whether there exists a winning strategy is Co-NP-complete.

1. Introduction

Colored graphs, which are graphs with color-labeled edges, are a model widely used in the field of computer science that deals with the analysis of concurrent systems [15]. For example, they can represent the transition relation of a concurrent program. In this case, the color of an edge indicates which process is making progress along that edge. A basic property of interest for these applications is *fairness*. This property essentially states that, during an infinite computation, each process is allowed to make progress infinitely often [9]. Starting from this core idea, a rich theory of fairness has been developed, as witnessed by the amount of literature devoted to the subject (see, for instance, [7,12,13]). In the abstract framework of colored graphs, the above basic version of fairness asks whether, along an infinite path in the graph, each color occurs infinitely often. Such a requirement does not put any bound on the number of steps that a process needs to wait before it is allowed to make progress. As a consequence, the asymptotic frequency of some color could be zero, even if the path is fair. Accordingly, several authors have proposed stronger versions of fairness. Alur and Henzinger define *finitary fairness* roughly as the property requiring that there is a fixed bound on the number of steps between two occurrences of any given color [1,4]. A similar proposal, supported by a corresponding temporal logic, was made by Dershowitz et al. [8]. On a finitarily fair path, all colors have positive asymptotic frequency. These definitions of fairness treat the frequencies of the relevant events in isolation and in a strictly qualitative manner. They only distinguish between zero frequency (not fair), limit-zero frequency (fair, but not finitarily so), and positive frequency (finitarily fair).

In this article we propose three fairness notions based on quantitative comparisons between competing events. The *bounded-difference* path property requires that there is a numerical constant bounding the difference between the number of occurrences of any two colors, for all prefixes of the path. The *balanced* path property requires that all colors occur with

[*] Corresponding author. Tel.: +39 081679276.
E-mail address: mfaella@na.infn.it (M. Faella).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.06.029

the same asymptotic frequency, i.e., the long-run average number of occurrences for each of them is the same. It is easy to see that bounded-difference paths are special cases of finitarily fair paths. On the other hand, finitarily fair paths and balanced paths are incomparable notions (see Example 1).

We believe that the proposed notions are valuable to some applications, perhaps quite different from the ones in which fairness is usually applied. Both the balance property and the bounded-difference property are probably too strong for the applications where one step in the graph represents a fine-grained transition of unknown length in a concurrent program. In that case, it may be of little interest to require that all processes make progress with the same (abstract) frequency. On the other hand, consider a context where each transition corresponds to some complex or otherwise lengthy operation. As an example, consider the model of a concurrent program where all operations have been disregarded, except the access to a peripheral that can only be used in one-hour slots, such as a telescope, which requires some time for repositioning. Assuming that all jobs have the same priority, it is certainly valuable to find a scheduling policy that assigns the telescope to each job with the same frequency. As a non-computational example, the graph may represent the rotation of cultures on a crop, with a granularity of 6 months for each transition [21]. In that case, we may very well be interested not just in having each culture eventually planted (fairness) or even planted within a bounded amount of time (finitarily fair), but also occurring with the same frequency as any other culture (balanced or bounded-difference).

Sometimes, systems require that some jobs are executed more often than others. In such a situation, it is useful to associate to each job a "priority" representing how often that job should be executed compared to the others. *Priority scheduling* is a problem widely studied in computer science [14], usually with the objective of minimizing the execution time of a given computation. In general, a priority scheduling problem is NP-hard [14] and becomes solvable in polynomial time if there are some restrictions on the nature of the system [5]. We address and solve a new scheduling problem for a system characterized by a finite number of states and infinite computation. We are interested in an execution of the system that spends a given fraction of time on each job. In our framework, the problem translates in looking for a path where each color occurs with some fixed asymptotic frequency. We call it the *frequency-f* problem.

A natural extension consists in introducing a second decision agent in the system, thus switching from graphs to *games*. Games are widely used in computer science to describe the interaction between a system and its environment [11,16,20,22]. Usually, the system is a component that is under the control of its designer and the environment represents all the components the designer has no direct control of. In this context, a game allows the designer to check whether the system can force some desired behavior (or avoid an undesired one) independently of the choices of the other components. In this paper, we address and study *two-player colored games*, i.e., games where the underlying graph is a colored graph and the game is played between two players, which we refer to as player 0 and player 1. In particular, we focus on the goal for player 0 to construct an infinite path which is quantitatively fair in the sense described earlier.

We believe that the game model can be useful in several formal verification contexts. Coming back to the scheduling application, it can be useful in the case the scheduler allows for a certain degree of freedom on the choices of lengthy jobs that have to be executed by some components. More specifically, assume that due to a design issue, the main scheduler can decide which macro-operation has to be executed and then some other scheduler chooses which sub-operation to perform. In this context, our game model allows the designer to check if the main scheduler has the ability to force a fair progress of the activities, independently of the sub-choices of the other scheduler(s). A concrete example is provided in Section 2.3. Our main result shows that, in a game where the goal of player 0 is the construction of a balanced or bounded-difference path, the problem of asking whether this player can always force such a path is Co-NP-complete.

The present paper is an improved and extended version of [2,3], including detailed proofs of all results and the novel result of NP-completeness for the perfectly balanced finite path problem.

Overview. The rest of the paper is organized as follows. In Section 2, we introduce some preliminary notation and the formalization of the problems described above. In Section 3, we introduce basic properties on game graphs which will be used in the next sections. Section 4 establishes, for a given graph, connections between the existence of balanced or bounded-difference paths, as well as certain loop-based properties. In Section 5, we consider systems of linear equations and show that the bounded or frequency-f problem can be reduced to their feasibility. In Section 6, we show that a closely related problem, namely the perfectly balanced finite path problem, is NP-complete. In Section 7, we introduce colored games with balanced, bounded, and frequency goals and show that, in all cases, the problem of deciding whether player 0 has a winning strategy starting from a given node of the game is Co-NP-complete.

2. Preliminaries

2.1. Colored graphs

Let X be a set and i be a positive integer. By X^i we denote the Cartesian product of X with itself i times. By $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ we respectively denote the set of natural, integer, rational, and real numbers. Given a positive integer k, let $[k] = \{1, \dots, k\}$ and $[k]_0 = [k] \cup \{0\}$.

A *k-colored graph* (or simply *graph*) is a pair $G = (V, E)$, where V is a finite set of *nodes* and $E \subseteq V \times [k] \times V$ is a set of *colored edges*. We employ integers as colors for technical convenience. All the results we obtain also hold for an arbitrary set of labels. An edge (u, a, v) is said to be *colored* with a. In the following, we also simply call a k-colored graph a *graph*, when k is clear from the context. For a node $v \in V$ we call $v_{\rightarrow} = \{(v, a, w) \in E\}$ the set of edges exiting from v, and $v_{\leftarrow} = \{(w, a, v) \in E\}$ the set of edges entering in v. For a color $a \in [k]$, we call $E(a) = \{(v, a, w) \in E\}$ the set of edges colored with a. For a node $v \in V$, a *finite v-path* ρ is a finite sequence of edges $(v_i, a_i, v_{i+1})_{i\in\{1,\dots,n\}}$ such that $v_1 = v$. The *length* of ρ is $|\rho| = n$. We denote by $\rho(i)$ the ith edge of ρ, and we sometimes write ρ as $v_1 v_2 \dots v_n$, when the colors are unimportant. A finite path $\rho = v_1 v_2 \dots v_n$ is a *loop* if $v_1 = v_n$. A loop $v_1 v_2 \dots v_n$ is *simple* if $v_i \neq v_j$, for all $1 \leq i < j < n$. An *infinite v-path* is defined analogously, i.e., it is an infinite sequence of edges. Let ρ be a finite path and π be a possibly infinite path, we denote by $\rho \cdot \pi$ the *concatenation* of ρ and π. We denote by ρ^{ω} the infinite path obtained by concatenating ρ with itself infinitely many times, and by $\prod_i \rho_i$ the concatenation of an infinite sequence of finite paths ρ_i. A graph G is *strongly connected* if for each pair (u, v) of nodes there is a finite u-path with last node v and a finite v-path with last node u.

For a finite or infinite path ρ and an integer i, we denote by $\rho^{\leq i}$ (resp., $\rho^{>i}$) the *prefix* (resp., *suffix*) of ρ (resp., not) containing the first i edges. For a color $a \in [k]$, we denote by $\|\rho\|_a$ the number of edges labeled with a occurring in ρ. For two colors $a, b \in [k]$, we denote the difference between the occurrences of edges labeled with a and b in ρ by $\mathit{diff}_{a,b}(\rho) = \|\rho\|_a - \|\rho\|_b$. For all finite paths ρ of G, with a slight abuse of notation let $\mathit{diff}(\rho) = (\mathit{diff}_{1,k}(\rho), \dots, \mathit{diff}_{k-1,k}(\rho))$ (resp. $\|\rho\| = (\|\rho\|_1, \dots, \|\rho\|_k)$) be the vector containing the differences between each color and color k, which is taken as a reference (resp. the number of occurrences of each color). We call $\mathit{diff}(\rho)$ (resp. $\|\rho\|$) the *difference vector* (resp. *color vector*) of ρ[1]. Given a finite set of loops $\mathcal{L} = \{\sigma_1, \dots, \sigma_l\}$ and a tuple of positive natural numbers $c_1, \dots, c_l$, we call *natural linear combination* (in short, *n.l.c.*) of $\mathcal{L}$ with coefficients $c_1, \dots, c_l$ the set $\mathcal{T} = \{(\sigma_1, c_1), \dots, (\sigma_l, c_l)\}$. The *difference value* of $\mathcal{T}$ is the vector of color differences obtained by traversing c_i times each loop σ_i, i.e., $\sum_{i=1}^{l} c_i \mathit{diff}(\sigma_i)$. The *frequency* of $\mathcal{T}$ is the vector of color frequencies obtained by traversing c_i times each loop σ_i, i.e., $\frac{\sum_{i=1}^{l} c_i \|\sigma_i\|}{\sum_{i=1}^{l} c_i |\sigma_i|}$. An infinite path π is *periodic* iff there exists a finite path ρ such that $\pi = \rho^{\omega}$. A loop σ is *perfectly balanced* iff $\mathit{diff}_{a,b}(\sigma) = 0$ for all $a, b \in [k]$. Finally, we denote by $\mathbf{0}$ and $\mathbf{1}$ the vectors containing only 0's and 1's, respectively. We can now define the decision problems that are the subject of the present paper.

The bounded-difference problem. Let G be a k-colored graph. An infinite path ρ in G has the *bounded-difference property* (or, is a *bounded-difference path*) if there exists a number $c \geq 0$, such that, for all $a, b \in [k]$ and $i > 0$,

$$|\mathit{diff}_{a,b}(\rho^{\leq i})| \leq c.$$

The *bounded-difference problem* is to determine whether there is a bounded-difference path in G.

The frequency-f problem. Let G be a k-colored graph, and $f \in \mathbb{Q}^k$ be a vector such that $\sum_{a=1}^{k} f_a = 1$. An infinite path ρ in G has *frequency f* if for all $a \in [k]$,

$$\lim_{i\to+\infty} \frac{\|\rho^{\leq i}\|_a}{i} = f_a.$$

The *frequency-f problem* is to determine whether there is a path with frequency f in G. In the particular case that all components in the frequency vector f are equal to $\frac{1}{k}$, the frequency-f problem is also called *balance problem* and a path with such a frequency is called *balanced*. Observe that all bounded-difference paths are balanced. The following example shows that the converse does not hold.

Example 1. For all $i > 0$, let $\sigma_i = (1 \cdot 2)^i \cdot 1 \cdot 3 \cdot (1 \cdot 3 \cdot 2 \cdot 3)^i \cdot 1 \cdot 3 \cdot 3$. Consider the infinite sequence $\sigma = \prod_i \sigma_i$. Observe that σ is not a bounded-difference path because at the beginning of each σ_i the difference between color 1 and color 3 grows linearly with i. The fact that σ is balanced can be verified by computing the appropriate limits. Moreover, we formally prove it in Example 2.

Notice that the *initialized* versions of the above decision problems, i.e., those obtained by fixing an initial node for the graph, are linear-time reducible to the original ones. It is sufficient to perform a strongly connected component analysis of the graph and then solve the original problem separately on each component. A node satisfies the initialized version of a problem if and only if it can reach a component that satisfies the original version of the problem.

2.2. Games

A *k-colored arena* is a k-colored graph whose set of nodes is partitioned in two sets V_0 and V_1 and contains a starting node. Formally a k-colored arena is a tuple $A = (V_0, V_1, v_{\text{ini}}, E)$, where $V_0 \cap V_1 = \emptyset$, $(V_0 \cup V_1, E)$ is a k-colored graph and v_{ini} is the *initial node*. A *k-colored game* is a pair $G = (A, W)$, where $A = (V_0, V_1, v_{\text{ini}}, E)$ is a k-colored arena and $W \subseteq [k]^{\omega}$ is a set of color sequences called *goal*. We assume that the game is played by two players, referred to as player 0 and player 1. The players construct a path starting at v_{ini} on the arena A, such a path is called *play*. Once the partial play reaches a node $v \in V_0$, player 0 chooses an edge exiting from v and extends the play with this edge; once the partial play reaches a node $v \in V_1$, player 1 makes a similar choice. Player 0's aim is to obtain a play whose color sequence belongs to W, while player 1's aim is the opposite.

[1] The difference vector is related to the Parikh vector [18] of the sequence of colors of the path. Precisely, the difference vector is equal to the first $k - 1$ components of the Parikh vector, minus the kth component.

```
0:   lock();
1:   if (...) {
2:     action();
     } else {
3:     action();
4:     action();
     }
5:   unlock();
   }
```

Fig. 1. A job in a concurrent program.

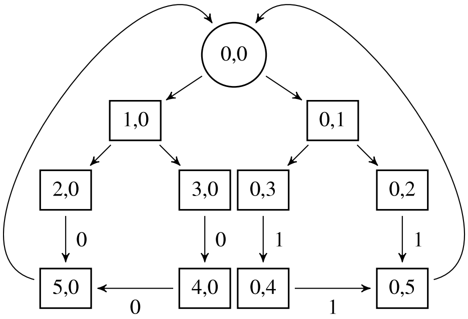

Fig. 2. The non-preemptive scheduling game corresponding to two jobs of the type in Fig. 1.

For $h \in \{0, 1\}$, let $E_h = \{(v, c, w) \in E \mid w \in V_h\}$ be the set of edges ending into nodes of player h. Let ε be the empty word, a *strategy* for player h is a function $\sigma_h : \varepsilon \cup (E^*E_h) \to E$ such that, if $\sigma_h(e_0 \dots e_n) = e_{n+1}$, then the destination of e_n is the source of e_{n+1}, and if $\sigma_h(\varepsilon) = e$, then the source of e is v_{ini}. Intuitively, σ_h fixes the choices of player h for the entire game, based on the previous choices of both players. The value $\sigma_h(\varepsilon)$ is used to choose the first edge in the game. A strategy σ_h is *memoryless* iff its choices depend only on the last node of the play, i.e., for all plays ρ and ρ' with the same last node, it holds that $\sigma_h(\rho) = \sigma_h(\rho')$. An infinite play $\{e_i\}_{i \in \mathbb{N}} \in E^\omega$ is *consistent* with a strategy σ_h iff (i) if $v_{\text{ini}} \in V_h$ then $e_0 = \sigma_h(\varepsilon)$, and (ii) for all $i \in \mathbb{N}$, if $e_i \in E_h$ then $e_{i+1} = \sigma_h(e_0 \dots e_i)$. A strategy for player 0 (resp., player 1) is said *winning* iff the color sequence of all plays consistent with that strategy belong to the goal W (resp., $[k]^\omega \setminus W$). A game is said *determined* iff one of the two players has a winning strategy.

Now we recall some definitions and results developed in [10]. A goal $W \subseteq [k]^\omega$ is said to be *prefix independent* iff for all color sequences $x \in [k]^\omega$, and for all $z \in [k]^*$, we have $x \in W$ iff $zx \in W$. For two color sequences $x, y \in [k]^\omega$, the *shuffle* of x and y, denoted by $x \otimes y$ is the language of all the words $z_1z_2z_3 \dots \in [k]^\omega$, such that $z_1z_3 \dots z_{2h+1} \dots = x$ and $z_2z_4 \dots z_{2h} \dots = y$, where $z_i \in [k]^*$ for all $i \in \mathbb{N}$. A goal W is said to be *convex* iff it is closed w.r.t. the shuffle operation, i.e., for all words $x, y \in W$, it holds that $x \otimes y \subseteq W$.

Theorem 1 (*[10]*). *Let $G = (A, W)$ be a k-colored game such that W is prefix independent and convex. Then, the game is determined. Moreover, if player 1 has a winning strategy, he has a memoryless winning strategy.*

2.3. A scheduling example

Consider two jobs in a concurrent program, both having the structure shown in Fig. 1. Note that the jobs exhibit nondeterministic behavior, due to the unknown (i.e., not explicitly modeled) branching condition on line 1.

Assume that we want to synthesize a scheduler ensuring that the "`action`" function is called with the same asymptotic frequency by the two jobs. The scheduler can decide not to give the lock to a job, but cannot pre-empt them. To this aim, we can produce a game as in Fig. 2, where nodes represent joint configurations of the two jobs. The unique node of player 0 is represented by a circle, while the nodes of player 1 are represented by boxes. Since we are only interested in counting the calls to the `action` function, we only color the edges representing such calls. Clearly, uncolored edges can be represented in our framework by a sequence of two edges, each labeled by a different color. The internal nondeterminism of the jobs is modeled by a move of player 1. The only choice for player 0 (the scheduler) occurs in the node (0, 0), where both jobs are waiting on the `lock` operation, and the scheduler can choose whom to give the lock to.

It is easy to verify that the scheduler has a strategy enforcing the bounded-difference property (hence, the balance property as well): when the game is in (0, 0), it gives the lock to the job that executed the `action` function less times so far. According to this scheduling policy, the difference between the numbers of 0's and 1's along a play is always at most 2, regardless of the choices made by the internal nondeterminism of the jobs. Note that this strategy requires memory. Using a similar strategy, player 0 can also win w.r.t. the frequency-f goal, for all (rational) frequency vectors f.

3. Basic properties

3.1. Connected and overlapping loops

In this section, we assume that $G = (V, E)$ is a finite k-colored graph, i.e., both V and E are finite. Two loops σ and σ' in G are *connected* if there exists a path from a node of σ to a node of σ', and vice versa. A set $\mathcal{L}$ of loops in G is connected if all pairs of loops in $\mathcal{L}$ are connected. Two loops in G are *overlapping* if they have a node in common. A set $\mathcal{L}$ of loops in G is overlapping if, for all pairs of loops $\sigma, \sigma' \in \mathcal{L}$, there exists a sequence $\sigma_1, \ldots, \sigma_n$ of loops in $\mathcal{L}$ such that (i) $\sigma_1 = \sigma$, (ii) $\sigma_n = \sigma'$, and (iii) for all $i = 1, \ldots, n-1$, it holds that (σ_i, σ_{i+1}) are overlapping. Given a set of loops $\mathcal{L}$ in G, the subgraph *induced by* $\mathcal{L}$ is $G' = (V', E')$, where V' and E' are the nodes and the edges, respectively, belonging to a loop in $\mathcal{L}$.

Lemma 1. *Let G be a graph, $\mathcal{L}$ be a set of loops in G, and $G' = (V', E')$ be the subgraph of G induced by $\mathcal{L}$, then the following statements are equivalent:*

1. $\mathcal{L}$ *is* overlapping.
2. *The subgraph G' is strongly connected.*
3. *There exists $u \in V'$ such that for all $v \in V'$ there exists a path in G' from u to v.*

Proof $(1 \Rightarrow 2)$**.** If $\mathcal{L}$ is overlapping, then, for all pairs of loops σ_1, σ_2, there exists a sequence of loops that links σ_1 with σ_2. Thus, from any node of σ_1, it is possible to reach any node of σ_2. Hence, G' is strongly connected.

$[2 \Rightarrow 3]$ Trivial.

$[3 \Rightarrow 2]$ Let $u \in V'$ be a witness for item 3. Let $v, w \in V'$, we prove that there is a path from v to w. We have that u is connected to both v and w. Since all edges in G' belong to a loop, for all edges $(u', \cdot, v')$ along the path from u to v, there is a path from v' to u'. Thus, there is a path from v to u, and so, a path from v to w, through u.

$[2 \Rightarrow 1]$ If G' is strongly connected, for all $\sigma_1, \sigma_2 \in \mathcal{L}$ there is a path ρ in G' from each node of σ_1 to each node of σ_2. This fact holds since G' is induced by $\mathcal{L}$, so, ρ uses only edges of the loops in $\mathcal{L}$. While traversing ρ, every time we move from one loop to the next one, these two loops must share a node. Therefore, all pairs of adjacent loops used in ρ are overlapping. Thus $\mathcal{L}$ is overlapping. □

The above lemma implies that if $\mathcal{L}$ is overlapping then it is also connected, since G' is strongly connected.

3.2. Segmentation and composition

For a path σ and an edge e, denote by $|\sigma|_e$ the number of times that e occurs in σ. We say that a loop is a *composition* of a finite tuple of simple loops $\mathcal{T}$ if it is obtained by using all and only the edges of $\mathcal{T}$ as many times as they appear in $\mathcal{T}$. Formally, the loop σ is a composition of $\mathcal{T} = (\sigma_1, \ldots, \sigma_l)$ if, for all edges e, it holds $|\sigma|_e = \sum_{i=1}^{l} |\sigma_i|_e$). Observe that distinct loops can be a composition of the same tuple of simple loops $\mathcal{T}$.

Given a finite path ρ, we define its *quasi-segmentation* and its *rest*. Intuitively, the idea is to decompose ρ into a sequence of simple loops, plus the remaining simple path that does not form a loop (the *rest*). We proceed by induction on $|\rho|$.

1. The quasi-segmentation is a sequence of simple loops, and the rest is a simple path. The rest is either empty or ending with the last node of ρ.
2. If ρ has length 1 and it is not a loop, then its quasi-segmentation is the empty sequence and its rest is ρ itself.
3. If ρ has length 1 and it is a loop, then its quasi-segmentation is ρ itself and its rest is the empty sequence.
4. If ρ has size n, let $\rho' = \rho^{\leq n-1}$, let $\sigma_1, \ldots, \sigma_n$ be the quasi-segmentation of ρ' and r be its rest. Consider the path r' obtained by extending r with the last edge of ρ (this can be done because the last node of r is the last node of ρ').
   - (a) If r' does not contain a loop, then the quasi-segmentation of ρ is $\sigma_1, \ldots, \sigma_n$ and the rest is r'. Observe that r' ends with last node of ρ.
   - (b) If r' contains a loop σ, this loop is due to the last added edge, i.e., $r' = r''\sigma$. In this case the quasi-segmentation of ρ is $\sigma_1, \ldots, \sigma_n, \sigma$ and the rest is r''. Observe that if r'' is not empty, then it ends with the first node of σ which is also equal to its last node, hence it equal the last node of r', and to the last node of ρ.

The quasi-segmentation of an infinite path ρ is the infinite sequence of loops given by the limit of the quasi-segmentation of $\rho^{\leq n}$, for $n \to +\infty$. An infinite path has no rest.

Lemma 2. *The following statements hold:*

1. *A path ρ is a composition of its quasi-segmentation and its rest. If the rest is not empty, then it starts with the starting node of ρ and ends with the last node of ρ.*
2. *A loop is a composition of its quasi-segmentation.*
3. *A loop of length n containing m distinct nodes is a composition of at least $\lceil \frac{n}{m} \rceil$ simple loops.*

Proof. [1] The proof is by induction on the length of ρ. The base case is trivial both if ρ is a loop and if it is not. For the inductive case: let $n = |\rho|$, $\rho' = \rho^{\leq n-1}$, $\sigma_1, \ldots, \sigma_l$ be the quasi-segmentation of ρ' and r be its rest. By inductive hypothesis,

ρ' is a composition of $\sigma_1, \ldots, \sigma_l$ and r; moreover if r is not empty, it starts with the starting node of ρ'. Consider the path r' obtained by extending r with the last edge of ρ, i.e., $r' = r \cdot \rho(n)$.

1. If r' does not contain a loop, then the quasi-segmentation of ρ is $\sigma_1, \ldots, \sigma_l$ and the rest is r'. Observe that r' starts with the starting node of ρ' which is also the starting node of ρ. Moreover, for all edges $e \neq \rho(n)$ we have $|\rho|_e = |\rho'|_e = |r|_e + \sum_{i=1}^{l} |\sigma_i|_e = |r'|_e + \sum_{i=1}^{l} |\sigma_i|_e$. Also, $|\rho|_{\rho(n)} = |\rho'|_{\rho(n)} + 1 = |r|_{\rho(n)} + 1 + \sum_{i=1}^{l} |\sigma_i|_{\rho(n)} = |r'|_{\rho(n)} + \sum_{i=1}^{l} |\sigma_i|_{\rho(n)}$. Hence, the thesis.
2. If r' contains a loop σ, this loop is due to the last added edge, i.e., $r' = r''\sigma$. In this case the quasi-segmentation of ρ is $\sigma_1, \ldots, \sigma_l, \sigma$ and the rest is r''. Observe that if r'' is not empty, then it starts with the starting node of ρ' which is also the starting node of ρ. Moreover, for all edges $e \neq \rho(n)$ we have $|\rho|_e = |\rho'|_e = |r|_e + \sum_{i=1}^{l} |\sigma_i|_e = |r''|_e + |\sigma|_e + \sum_{i=1}^{l} |\sigma_i|_e$. Also $|\rho|_{\rho(n)} = |\rho'|_{\rho(m)} + 1 = |r|_{\rho(n)} + 1 + \sum_{i=1}^{l} |\sigma_i|_{\rho(n)} = |r'|_{\rho(n)} + \sum_{i=1}^{l} |\sigma_i|_{\rho(n)} = |r''|_{\rho(n)} + |\sigma|_{\rho(n)} + \sum_{i=1}^{l} |\sigma_i|_{\rho(n)}$. Hence, ρ is a composition of the quasi-segmentation and its rest.

[2] Since in a loop the extremes coincide, and since the rest of a quasi-segmentation is a simple path, the rest of a quasi-segmentation of a loop is necessarily empty.

[3] σ is the decomposition of its quasi-segmentation. Since a simple loop of the quasi-segmentation contains at most m nodes, the tuple contains at least $\lceil \frac{n}{m} \rceil$ simple loops. □

4. Graph-theoretic characterizations

In this section, we present some preliminary results that connect the existence of bounded, balanced, and frequency-f paths in a colored graph to the existence of a set of perfectly balanced loops. Later, such a characterization is used to solve the problems that we stated in Section 2.

4.1. Bounded-difference characterization

Given a graph, if there is a perfectly balanced loop σ, it is easy to see that σ^ω is a periodic bounded-difference path. Moreover, if ρ is an infinite bounded-difference path, then there is a constant c such that the absolute value of all color differences is smaller than c. Since both the set of nodes and the possible difference vectors along ρ are finite, we can find two indexes $i < j$ such that $\rho(i) = \rho(j)$ and $\mathit{diff}(\rho^{\leq i}) = \mathit{diff}(\rho^{\leq j})$. So, $\sigma' = \rho(i)\rho(i+1)\ldots\rho(j)$ is a perfectly balanced loop. Thus, the following holds.

Lemma 3. *Given a graph G, the following statements are equivalent:* (i) *there exists a bounded-difference path;* (ii) *there exists a periodic bounded-difference path;* (iii) *there exists a perfectly balanced loop.*

We now prove the following result.

Lemma 4. *Let G be a graph. There exists a perfectly balanced loop in G iff there exists an overlapping set $\mathcal{L}$ of simple loops of G, such that $\mathcal{L}$ has an n.l.c. with difference value* **0**.

Proof (*Only if*). If there exists a perfectly balanced loop σ, by Lemma 2 the loop is the composition of a tuple $\mathcal{T}$ of simple loops. Let $\mathcal{L}$ be the set of distinct loops occurring in $\mathcal{T}$, and for all $\rho \in \mathcal{L}$, let c_ρ be the number of times ρ occurs in $\mathcal{T}$. Since in the computation of the difference vector of a path it does not matter the order in which the edges are considered, we have $\sum_{\rho \in \mathcal{L}} c_\rho \cdot \mathit{diff}(\rho) = \mathit{diff}(\sigma) = \mathbf{0}$. Finally, since the loops in $\mathcal{L}$ come from the decomposition of a single loop σ, we have that $\mathcal{L}$ is overlapping.

[If] Let $\mathcal{L} = \{\sigma_1, \ldots, \sigma_l\}$ be such that $\sum_{i=1}^{l} c_i \cdot \mathit{diff}(\sigma_i) = \mathbf{0}$. We construct a single loop σ such that $\mathit{diff}(\sigma) = \sum_{i=1}^{l} c_i \cdot \mathit{diff}(\sigma_i)$. The construction proceeds in iterative steps, building a sequence of intermediate paths $\rho_1, \ldots, \rho_l$, such that ρ_l is the wanted perfectly balanced loop. In the first step, we take any loop $\sigma_{i_1} \in \mathcal{L}$ and traverse it c_{i_1} times, obtaining the first intermediate path $\rho_1 = \sigma_{i_1}^{c_{i_1}}$. After the jth step, since $\mathcal{L}$ is overlapping, there must be a loop $\sigma_{i_{j+1}} \in \mathcal{L}$ that is overlapping with one of the loops in the current intermediate path ρ_j, say in node v. Then, we *reorder* ρ_j in such a way that it starts and ends in v. Let ρ'_j be such reordering, we set $\rho_{j+1} = \rho'_j \sigma_{i_{j+1}}^{c_{i_{j+1}}}$. One can verify that ρ_l is perfectly balanced. □

The following theorem is a direct consequence of the previous two lemmas.

Theorem 2. *A graph G satisfies the bounded-difference problem iff there exists an overlapping set of simple loops of G having an n.l.c. with difference value* **0**.

4.2. Frequency-f characterization

In the following, by M^T, we denote the transpose of the matrix M and, by $M_{i,j}$, the element of M at its ith row and jth column. The main purpose of this section is to prove the following loop-based characterization for the existence of frequency-f paths.

Theorem 3. *A graph G satisfies the frequency-f problem iff there exists a connected set $\mathcal{L}$ of simple loops having an n.l.c. with frequency f.*

We start the proof with two preliminary lemmas and then we separately prove the two directions of the above theorem, represented by Lemmas 6 and 7.

Lemma 5. *Given a vector $f \in \mathbb{Q}^k$, let $\mathcal{L}$ be a finite set of loops having no n.l.c. with frequency f. Moreover, let $\{\rho_n\}_n$ be an infinite sequence of elements of $\mathcal{L}$, $x(n,a) = \sum_{i=0}^{n} \|\rho_i\|_a$, and $l(n) = \sum_{i=0}^{n} |\rho_i|$. Then, there exists a color $a^* \in [k]$ such that $\lim_{n\to+\infty} \frac{x(n,a^*)}{l(n)} \neq f_{a^*}$.*

Proof. Let $\mathcal{L} = \{\sigma_1, \dots, \sigma_m\}$ and let $g : \mathbb{R}^m \mapsto \mathbb{R}_+$ be defined by $g(c_1, \dots, c_m) = \max_{a\in[k]} \left\{ \left| \frac{\sum_{n=1}^{m} c_n \|\sigma_n\|_a}{\sum_{n=1}^{m} c_n |\sigma_n|} - f_a \right| \right\}$. First, note that g is a continuous function, since it is the maximum of continuous functions. Let $K \subset \mathbb{R}^m$ be the set $\{(c_1, \dots, c_m) \in [0,1]^m \mid \sum_{i=1}^{m} c_i = 1\}$. By Weierstrass theorem, g admits a minimum $M = \min_{x\in K}\{g(x)\}$ on K. We show that $M > 0$. Indeed, if by contradiction $M = 0$, there would be a vector $(c_1, \dots, c_m) \in K$ such that for all $a \in [k]$,

$$\sum_{n=1}^{m} c_n \|\sigma_n\|_a - f_a \sum_{n=1}^{m} c_n |\sigma_n| = M = 0.$$

Since the above linear equality has a (non-negative) solution and contains only rational coefficients, it also has a (non-negative) rational solution. Since it is homogeneous, it also admits a non-negative integer solution with at least one positive component. This solution induces an n.l.c. of $\mathcal{L}$ with frequency f, contradicting the hypotheses.

Now, let $\delta_{i,j}$ be the number of times for which the loop σ_i occurs among the first j elements of the sequence $\{\rho_n\}_n$ and let $c_{i,j} = \delta_{i,j}/j$. Since we have $\sum_{i=1}^{m} \delta_{i,n} = n$, for all $n \in \mathbb{N}$, it follows that $(c_{1,n}, \dots, c_{m,n}) \in K$. Moreover, $x(n,a) = \sum_{i=1}^{m} \delta_{i,n} \cdot \|\sigma_i\|_a = n \cdot \sum_{i=1}^{m} c_{i,n} \cdot \|\sigma_i\|_a$ and $l(n) = \sum_{i=1}^{m} \delta_{i,n} \cdot |\sigma_i| = n \cdot \sum_{i=1}^{m} c_{i,n} \cdot |\sigma_i|$.

For all $n \in \mathbb{N}$, let $Z_n \in \mathbb{R}^k$ be the vector defined by $Z_{n,a} = \left| \frac{\sum_{i=1}^{m} c_{i,n} \|\sigma_i\|_a}{\sum_{i=1}^{m} c_{i,n} |\sigma_i|} - f_a \right|$. Since there is no n.l.c. of $\mathcal{L}$ with frequency f, it holds that for all $n \in \mathbb{N}$ there exists a non-zero element in Z_n. Let $\{a_n\}_n$ be a color sequence such that $Z_{n,a_n} = \max_{a\in[k]}\{Z_{n,a}\} > 0$. Since $\{a_n\}_n$ can assume at most k different values, there exists a color a^* that occurs infinitely often in $\{a_n\}_n$. Let $\{h_t\}_t$ be the index sequence such that $a_{h_t} = a^*$ and there is no $h \in]h_t, h_{t+1}[$ with $a_h = a^*$. Then, consider the subsequence $\{Z_{h_t,a^*}\}_t$ of $\{Z_{n,a^*}\}_n$. It holds that $\lim_{t\to+\infty} Z_{h_t,a^*} \geq M > 0$ and consequently $\lim_{n\to+\infty} Z_{n,a^*} \neq 0$, whenever these limits exist. In conclusion, $\lim_{n\to+\infty} \frac{\sum_{i=1}^{m} c_{i,n} \|\sigma_i\|_{a^*}}{\sum_{i=1}^{m} c_{i,n} |\sigma|} = \lim_{n\to+\infty} \frac{x(n,a^*)}{l(n)} \neq f_{a^*}$. □

Lemma 6. *Let G be a k-colored graph and ρ be an infinite path in G with color frequency $f \in \mathbb{Q}^k$, then there exists a connected set of simple loops having an n.l.c. with frequency f.*

Proof. Since ρ is an infinite path over a finite set of nodes, there exists a non-empty set V' of nodes through which the path passes an infinite number of times. Then, there exists a constant m such that, for all $n \geq m$, it holds that $\rho(n) \in V'$. The path $\pi \triangleq \rho^{\geq m}$ has frequency f, since the frequency-f property is prefix independent. Let $\{\sigma_i\}_i$ be the quasi-segmentation of π and, for all $i \in \mathbb{N}$, let $h(i)$ be the index in π of the node in which σ_i closes itself. So, each time a simple loop closes at step $h(n)$, $\pi^{\leq h(n)}$ is composed of the $n+1$ simple loops $\sigma_0, \dots, \sigma_n$ closed so far, plus the rest r_n. So, for all $a \in [k]$, we have $\|\pi^{\leq h(n)}\|_a = \|r_n\|_a + \sum_{j=1}^{n} \|\sigma_j\|_a$.

Let $x(n,a) = \sum_{i=1}^{n} \|\sigma_i\|_a$, $y(n,a) = \|\pi^{\leq h(n)}\|_a$, and $l(n) = \sum_{i=1}^{n} |\sigma_i|$. Since π has frequency f, we have $\lim_{n\to+\infty} \frac{y(n,a)}{h(n)} = f_a$. Since the rest is a simple path, it has length at most $|V'|$, and we have that $x(n,a) - |V'| \leq y(n,a) \leq x(n,a) + |V'|$. Hence, $|x(n,a) - y(n,a)| \leq |V'|$ and $y(n,a) - |V'| \leq x(n,a) \leq y(n,a) + |V'|$. Moreover, $h(n) = |r_n| + \sum_{j=1}^{n} |\sigma_j|$, thus $l(n) - |V'| \leq h(n) \leq l(n) + |V'|$, so we have that $h(n) - |V'| \leq l(n) \leq h(n) + |V'|$. For all $a \in [k]$, since $\lim_{n\to+\infty} \frac{y(n,a)}{h(n)} = f_a$, then $\lim_{n\to+\infty} \frac{y(n,a)+|V'|}{h(n)-|V'|} = f_a$ and $\lim_{n\to+\infty} \frac{y(n,a)-|V'|}{h(n)+|V'|} = f_a$. Since for all $n \in \mathbb{N}$ such that $h(n) > |V'|$ we have $\frac{y(n,a)-|V'|}{h(n)+|V'|} \leq \frac{x(n,a)}{l(n)} \leq \frac{y(n,a)+|V'|}{h(n)-|V'|}$, we have $\lim_{n\to+\infty} \frac{x(n,a)}{l(n)} = f_a$. By Lemma 5, the set $\mathcal{L}$ of all simple loops in G has an n.l.c. $\mathcal{T}$ with frequency f. Then, the simple loops of $\mathcal{L}$ which occur with a positive coefficient in $\mathcal{T}$ are connected, because they are extracted from the same path π, and have an n.l.c. with frequency f. □

Lemma 7. *If a k-colored graph G contains a set of connected simple loops having an n.l.c. with frequency $f \in \mathbb{Q}^k$, then there exists in G an infinite path ρ with frequency f.*

Proof. Let $\mathcal{L} = \{\sigma_0, \dots, \sigma_{h-1}\}$, and denote by v_i the first node of σ_i in its representation as a cyclic sequence of nodes. For all $i = 0, \dots, h-1$, let π_i be a (possibly empty) path starting in the last node of σ_i and ending in the first node of $\sigma_{(i+1) \bmod h}$. Since $\mathcal{L}$ is connected, it is possible to find such paths. Let $(c_0, c_1, \dots, c_{h-1})$ be the non-negative integers such that $\frac{\sum_{i=0}^{h-1} c_i \|\sigma_i\|}{\sum_{i=0}^{h-1} c_i |\sigma_i|} = f$. We set $Z = \sum_{i=0}^{h-1} c_i \|\sigma_i\|$. Let $n_i = |\sigma_i|$ and $m_i = |\pi_i|$, we set $n = \sum_{i=0}^{h-1} c_i \cdot n_i$ and $m = \sum_{i=0}^{h-1} m_i$.

In order to construct a path with frequency f, we reason as follows. Since in general the loops in $\mathcal{L}$ do not share a node with each other, to move from σ_i to σ_{i+1}, we have to pay a price, represented by the color vector of π_i. In order to make this price disappear in the long-run, we traverse the loops σ_i an increasing number of times: in the first round, we traverse it c_i times, in the second round, $2c_i$ times, and so on. Formally, the construction is iterative and at every round $i > 0$ we add a new cycle ρ_i to the path constructed so far, where $\rho_i = \sigma_0^{ic_0} \pi_0 \sigma_1^{ic_1} \pi_1 \dots \sigma_{h-1}^{ic_{h-1}} \pi_{h-1}$. Notice that ρ_i starts and ends at

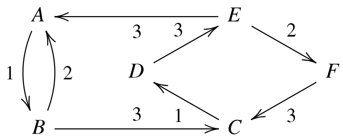

Fig. 3. A 3-colored graph satisfying the balance problem, but not the bounded-difference problem.

node v_0 and contains $m + i \cdot n$ edges. The required infinite path is then $\rho = \rho_1\rho_2 \dots \rho_i \dots$. We now show that this path has frequency f.

Let $l_0 = 0$ and, for all $i > 0$, let $l_i = \sum_{j=1}^{i} |\rho_j| = \sum_{j=1}^{i}(m+i\cdot n) = i\cdot m + \frac{i\cdot(i+1)}{2}n = \frac{i^2}{2}n + O(i)$, so that $\rho^{\leq l_i} = \rho_1 \dots \rho_i$. For all colors a, it holds $\|\rho^{\leq l_i}\|_a = \sum_{j=1}^{i} \|\rho_j\|_a = \sum_{j=1}^{i}(\sum_{q=0}^{h-1} \|\pi_q\|_a) + i(\sum_{q=0}^{h-1} c_q\|\sigma_q\|_a) = \frac{i^2}{2} \cdot (\sum_{q=0}^{h-1} c_q\|\sigma_q\|_a) + O(i)$. Consider an index $j \in \mathbb{N}$, there exists an index $i(j)$ such $l_{i(j)} \leq j \leq l_{i(j)+1}$. Since for all colors a the functions $\|\rho^{\leq j}\|_a$ and $|\rho^{\leq j}|$ are non-decreasing in j, we have that: $\frac{\|\rho^{\leq l_{i(j)}}\|_a}{|\rho^{\leq l_{i(j)+1}}|} \leq \frac{\|\rho^{\leq j}\|_a}{|\rho^{\leq j}|} \leq \frac{\|\rho^{\leq l_{i(j)+1}}\|_a}{|\rho^{\leq l_{i(j)}}|}$. Since $\lim_{j\to+\infty} \frac{\|\rho^{\leq l_{i(j)+1}}\|_a}{|\rho^{\leq l_{i(j)}}|} = \frac{(i+1)^2\cdot(\sum_{q=0}^{h-1} c_q\|\sigma_q\|_a)}{i^2\cdot n} = f_a$, and $\lim_{j\to+\infty} \frac{\|\rho^{\leq l_{i(j)}}\|_a}{|\rho^{\leq l_{i(j)+1}}|} = f_a$, we have $\lim_{j\to+\infty} \frac{\|\rho^{\leq j}\|_a}{|\rho^{\leq j}|} = f_a$. □

Example 2. Consider the graph G in Fig. 3. First note that, up to rotation, there are just three simple loops in it: $\sigma_1 = A \cdot B \cdot A$, $\sigma_2 = C \cdot D \cdot E \cdot F \cdot C$, and $\sigma_3 = A \cdot B \cdot C \cdot D \cdot E \cdot A$. It is easy to see that $\mathit{diff}(\sigma_1) = (1, 1)$, $\mathit{diff}(\sigma_2) = (-1, -1)$, and $\mathit{diff}(\sigma_3) = (-1, -3)$. For all the three overlapping sets of loops ($\{\sigma_1, \sigma_3\}$, $\{\sigma_2, \sigma_3\}$, and $\{\sigma_1, \sigma_2, \sigma_3\}$) there is no way to obtain an n.l.c. with difference value $\mathbf{0}$ and all coefficients different from zero. So, by Theorem 2 there is no bounded-difference path in G. On the other hand, since the connected set of simple loops $\{\sigma_1, \sigma_2\}$ has an n.l.c. (with both coefficients equal to 1) with frequency vector $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, by Lemma 7 we obtain that there is a balanced path in G. In particular, the balanced path which is built in the proof of Lemma 7 is the subject of Example 1.

4.3. 2-colored graphs

In this section we prove that in the case of 2-colored graphs, the bounded-difference problem and the balance problem coincide and we also introduce a faster algorithm for the solution of both problems.

When the graph G is 2-colored, the difference vector is simply a number. So, if $\mathcal{L}$ is a connected set of simple loops having an n.l.c. with difference value zero, then there must be either a perfectly balanced simple loop or two loops with difference vectors of opposite sign. Note that two loops σ, σ' with color differences of opposite sign have the following n.l.c. with difference value zero: $|\mathit{diff}(\sigma')| \cdot \mathit{diff}(\sigma) + |\mathit{diff}(\sigma)| \cdot \mathit{diff}(\sigma') = 0$. If the two loops are connected but not overlapping, we can construct a sequence of adjacent overlapping simple loops connecting them. In this way, we are always able to find a perfectly balanced simple loop or two overlapping simple loops with difference vectors of opposite sign. Therefore, the following holds.

Lemma 8. *Let G be a 2-colored graph. If there is a* connected *set of simple loops of G having an n.l.c. with difference value zero, then there is an* overlapping *set of simple loops of G having an n.l.c. with difference value zero.*

Proof. In a 2-colored graph, the difference vector of any path ρ is simply an integer. Let $\mathcal{L}$ be a connected set of simple loops having an n.l.c. with difference value zero. If $\mathcal{L}$ contains a simple loop σ such that $\mathit{diff}(\sigma) = 0$, then $\{\sigma\}$ is trivially an overlapping set. If $\mathcal{L}$ contains no perfectly balanced simple loop, then it cannot be the case that all difference vectors of the loops have the same sign, because then it would not possible to have an n.l.c. with difference value zero. Thus, let $\mathit{diff}(\sigma) > 0$ and $\mathit{diff}(\sigma') < 0$, for $\sigma, \sigma' \in \mathcal{L}$. If σ and σ' are overlapping, then $\{\sigma, \sigma'\}$ is the overlapping set we are looking for. If σ and σ' are not overlapping, since they are connected, there exist a path ρ_1 from σ to σ' and a path ρ_2 from σ' to σ. So, there exist four indexes i, i', j, j' such that $\rho_1(i)$ is the last node of ρ_1 in σ, $\rho_1(j)$ is the first node of ρ_1 in σ', $\rho_2(i')$ is the last node of ρ_2 in σ', and $\rho_2(j')$ is the first node of ρ_2 in σ. Then, within the loop σ there exists a simple path ρ from $\rho_2(j')$ to $\rho_1(i)$ and, within the loop σ', there exists a simple path ρ' from $\rho_1(j)$ to $\rho_2(i')$. We then set $\rho'_1 = \rho_1(i) \dots \rho_1(j)$ and $\rho'_2 = \rho_2(i'), \dots, \rho_2(j')$. We observe that the pairs of paths (ρ, ρ'_1), (ρ', ρ'_1), (ρ', ρ'_2), and (ρ, ρ'_2) have only one node in common. Moreover, ρ and ρ' have no node in common since σ and σ' are not overlapping. So, the loop $\sigma'' = \rho\rho'_1\rho'\rho'_2$ is not simple iff ρ'_1 and ρ'_2 have a node in common. Now, observe that for two loops π_1 and π_2 with difference vectors of opposite sign, the n.l.c. with coefficients $|\mathit{diff}(\pi_2)|$ and $|\mathit{diff}(\pi_1)|$ has difference value zero, since $|\mathit{diff}(\pi_2)|\mathit{diff}(\pi_1) + |\mathit{diff}(\pi_1)|\mathit{diff}(\pi_2) = 0$. We conclude with the following case analysis.

1. Assume σ'' is simple. Then (σ, σ'') and (σ', σ'') are pairs of overlapping loops. The desired overlapping set $\mathcal{L}'$ of loops having an n.l.c. with difference value 0 is defined by:

$$\mathcal{L}' = \begin{cases} \{\sigma''\} & \text{if } \mathit{diff}(\sigma'') = 0 \\ \{\sigma', \sigma''\} & \text{if } \mathit{diff}(\sigma'') > 0 \\ \{\sigma, \sigma''\} & \text{if } \mathit{diff}(\sigma'') < 0. \end{cases}$$

2. Assume ρ'_1 and ρ'_2 have nodes in common, i.e., there exist two indexes k, k' such that $\rho'_1(k) = \rho'_2(k')$. We can construct two loops $\sigma'_1 = \rho\rho'_1(1)\dots\rho'_1(k)\rho'_2(k'+1)\dots\rho'_2(|\rho'_2|)$ and $\sigma'_2 = \rho'\rho'_2(1)\dots\rho'_2(k')\rho'_1(k+1)\dots\rho'_1(|\rho'_1|)$. The desired overlapping set $\mathcal{L}'$ of loops having an n.l.c. with difference value 0 is defined by:

$$\mathcal{L}' = \begin{cases} \{\sigma'_i\} & \text{if } \mathit{diff}(\sigma'_i) = 0, \text{ for some } i \in \{0, 1\} \\ \{\sigma', \sigma'_2\} & \text{if } \mathit{diff}(\sigma'_2) > 0 \\ \{\sigma, \sigma'_1\} & \text{if } \mathit{diff}(\sigma'_1) < 0 \\ \{\sigma'_1, \sigma'_2\} & \text{if } \mathit{diff}(\sigma'_1) > 0 \text{ and } \mathit{diff}(\sigma'_2) < 0. \quad \square \end{cases}$$

A 2-colored graph can be represented as a *weighted graph*, where color 0 is replaced by weight 1 and color 1 by weight −1. In this framework, a loop is said to be *null* (resp., *positive*, *negative*) if the sum of the values of its weights is zero (resp., positive, negative). Observe that a null (resp. positive, negative) loop contains the same number of occurrences of the two colors (resp. more occurrences of color 0, more occurrences of color 1). Due to the characterization of Lemma 8, the decision problems have an affirmative answer if and only if there exists a null loop or two connected loops of opposite sign. Hence the decision problems can be solved efficiently, as proved by the following result.

Theorem 4. *A 2-colored graph $G = (V, E)$ satisfies the bounded-difference problem iff it satisfies the balance problem. Both problems can be solved in time $\mathcal{O}(|V| \cdot |E|)$.*

Proof. The equivalence between the bounded and the balanced problem in a 2-colored graph is direct consequence of Theorem 2, Lemma 8, and, finally, Theorem 3.

In order to solve the problems, we convert the colored graph into a weighted graph, as explained above. Then, we check whether there exists a null simple loop in the graph. To this purpose, from the weighted graph $G = (V, E)$, we build an enlarged graph by equipping each node with a bounded counter x, which keeps track of the accumulated weight. Due to the fact that we are looking for null simple loops, it is enough to consider $x \in \{-\lfloor |V|/2 \rfloor, \dots, \lfloor |V|/2 \rfloor\}$, since the maximum absolute weight accumulated in such a loop is $\lfloor |V|/2 \rfloor$. Formally, $G' = (V', E')$, where $V' = V \times \{-\lfloor |V|/2 \rfloor, \dots, \lfloor |V|/2 \rfloor\}$ and, for all $(u, x), (v, y) \in V'$, it holds that $((u, x), (v, y)) \in E'$ iff $(u, w, v) \in E$ and $y = x + w$ (where $w \in \{-1, 1\}$ is the weight of the edge). It is easy to see that the following holds: (i) if a node $v \in V$ is part of a perfectly balanced simple loop, there is a non-trivial path in G' from $(v, 0)$ to itself; (ii) if there is a non-trivial path in G' from $(v, 0)$ to itself, the node $v \in V$ is part of a perfectly balanced loop. Now, we use a depth-first search to find a loop from $(v, 0)$ to itself, for each $v \in V$. If such a loop exists, in G there is a null loop containing v and the answer to both decision problems is affirmative. If there is no such loop for any $v \in V$, we proceed to the next phase.

At this point, we decompose the graph in its strongly connected components, by using Tarjan's algorithm in time $\mathcal{O}(|V| + |E|)$. Then, on each of these components, we apply the Bellman–Ford algorithm for both maximum and minimum single-source paths, starting at any node. The two executions of the Bellman–Ford algorithm find, respectively, a positive and negative loop if they exist. If there is a component for which both searches succeed then the balance and bounded-difference problems are solved with a positive answer. Otherwise, the problems are solved with negative answer, since we are sure that there is no way to build a perfectly balanced loop.

Each run of the Bellman–Ford algorithm takes time $\mathcal{O}(|V''| \cdot |E''|)$, where (V'', E'') is a connected component. So, if S is the set of connected components, we have that the whole algorithm runs in time $\mathcal{O}(|V| + |E| + \sum_{(V'',E'') \in S} |V''| \cdot |E''|) \in \mathcal{O}(\sum_{(V'',E'') \in S} |V''| \cdot |E|) = \mathcal{O}(|E| \sum_{(V'',E'') \in S} |V''|) = \mathcal{O}(|E| \cdot |V|)$. $\square$

5. From graphs to linear systems

In this section, we solve in polynomial time the bounded-difference and the frequency-f problem for a rational f, by reducing them to the feasibility of a linear system. The linear system for the frequency-f problem is used as a component for the linear system for the bounded-difference problem. So, for ease of reading, we introduce the solution of the frequency-f problem first.

5.1. Frequency-f linear system

Definition 1. Let $G = (V, E)$ be a k-colored graph, and $f \in \mathbb{Q}^k$ a vector of rationals. We call *frequency-f system* for G the following system of equations on the set of variables $\{x_e \mid e \in E\}$.

$$\begin{array}{lrcl} 1.\ \text{for all } v \in V & \sum_{e \in v_\leftarrow} x_e & = & \sum_{e \in v_\rightarrow} x_e \\ 2.\ \text{for all } a \in [k] & \sum_{e \in E(a)} x_e & = & f_a \sum_{e \in E} x_e \\ 3.\ \text{for all } e \in E & x_e & \geq & 0 \\ 4. & \sum_{e \in E} x_e & > & 0. \end{array}$$

If all components of f are equal to $\frac{1}{k}$, we call the above system the *balance system* for G.

Let $m = |E|$ and $n = |V|$, the frequency-f system has m variables and $m + n + k + 1$ constraints. It helps to think of each variable x_e as a load associated to the edge $e \in E$, and of each constraint as having the following meaning.

1. For each node, the entering load is equal to the exiting load.

2. For all colors $a \in [k]$, the total load on the edges colored by a is equal to f_a times the whole load.
3. Every load is non-negative.
4. The total load is positive.

We prove that the frequency-f problem is equivalent to the feasibility of the previous linear system through the following auxiliary theorem.

Lemma 9. *Given a graph $G = (V, E)$, there exists a vector $x : E \to \mathbb{N}$ such that for each node $v \in V$ it holds $\sum_{e \in v_\leftarrow} x_e = \sum_{e \in v_\to} x_e$ if and only if there exists a sequence of simple loops $\mathcal{T} = \sigma_1, \ldots, \sigma_l$ such that $x_e = \sum_{i=1}^{l} |\sigma_i|_e$.*

Proof (*If*)**.** Let $\mathcal{T} = \sigma_1, \ldots, \sigma_l$ be a sequence of simple loops and let $x_e = \sum_{i=1}^{l} |\sigma_i|_e$. Consider a node $v \in V$, and let $y_{v,i} \in \{0, 1\}$ be a value indicating whether the node v occurs in σ_i. Precisely $y_{v,i} = 1$ if and only if the loop σ_i passes through v. Since each loop passing through v contains only one edge entering in v and only one edge exiting from v, and since a loop non-passing through v contains none of the above, we have for each v, $\sum_{e \in v_\to} x_e = \sum_{i=1}^{l} y_{v,i} = \sum_{e \in v_\leftarrow} x_e$.

[Only if] Let $x : E \to \mathbb{N}$ be a vector such that $\sum_{e \in v_\leftarrow} x_e = \sum_{e \in v_\to} x_e$ for all $v \in V$. We propose an algorithm that computes a sequence of simple loops $\mathcal{T} = \sigma_1, \ldots, \sigma_l$ such that $x_e = \sum_{i=1}^{l} |\sigma_i|_e$. The algorithm is recursive on the value of the sum $\sum_{e \in E} x_e$.

1. For the base case, if $\sum_{e \in E} x_e = 0$ then the empty sequence is the sought sequence of loops.
2. Let $\sum_{e \in E} x_e > 0$. In the recursive step we construct a simple loop ρ and we remove the edges used by this loop from the vector x, thus obtaining a *residual vector* $y_e = x_e - |\rho|_e$ with a lower sum. Notice that since ρ is a simple loop, the residual vector satisfies $\sum_{e \in v_\leftarrow} y_e = \sum_{e \in v_\to} y_e$ for all $v \in V$. The construction is iterative and proceeds as follows:
   (a) In the first step we choose any edge e' such that $x_{e'} > 0$ and we set $\rho = e'$. At this point ρ may already be a loop, in which case the iteration ends. Otherwise, we repeatedly apply the next step.
   (b) Consider the simple path ρ built so far. Let v be the last node in ρ: in ρ there is only one entering edge in v and no exiting edges. Hence, $0 \le \sum_{e \in v_\leftarrow} y_e = (\sum_{e \in v_\leftarrow} x_e) - 1 = (\sum_{e \in v_\to} x_e) - 1 = (\sum_{e \in v_\to} y_e) - 1$. This implies that $\sum_{e \in v_\to} y_e > 0$, hence we can find $e' \in v_\to$ such that $y_{e'} > 0$. Then, update ρ by adding the edge e' at the end. At this point if ρ is a loop the iteration ends, otherwise another iteration of this step is performed.
   (c) Eventually, within $|V|$ steps, the path ρ becomes a simple loop. As a consequence, for the residual vector y defined above, it holds $\sum_{e \in v_\leftarrow} y_e = \sum_{e \in v_\to} y_e$.

   At the end of the iteration we have a simple loop ρ and a residual vector y such that $\sum_{e \in E} y_e < \sum_{e \in E} x_e$ and $\sum_{e \in v_\to} y_e = \sum_{e \in v_\leftarrow} y_e$ for all $v \in V$. We can apply the recursive algorithm to the vector y and the sought sequence of simple loops is obtained by collecting the paths ρ generated along the way. □

Lemma 10. *There exists a set of simple loops in G with an n.l.c. of frequency f iff the frequency-f system for G is feasible.*

Proof (*Only if*)**.** Assume that $\mathcal{L}$ is a set of simple loops having an n.l.c. of frequency f. Let c_σ be the coefficient associated with the loop $\sigma \in \mathcal{L}$. We can construct a vector $x : E \to \mathbb{N}$ that satisfies the frequency-f system: $x_e = \sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|_e = \sum_{i=1}^{l} |\sigma_i|_e$. Let $\mathcal{T} = \sigma_1, \ldots, \sigma_l$ be a sequence of loops obtained by repeating c_σ times each loop σ in $\mathcal{L}$, in arbitrary order. By Lemma 9, x satisfies the first set of constraints of the frequency-f system. The third, fourth and fifth sets are trivially satisfied. For the second set we need to observe that the value $\sum_{e \in E(a)} x_e$ is the sum of edges colored by a color $a \in [k]$: $\sum_{e \in E(a)} x_e = \sum_{e \in E(a)} \sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|_e = \sum_{\sigma \in \mathcal{L}} c_\sigma \sum_{e \in E(a)} |\sigma|_e = \sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|_a$. Due to the natural linear combination of frequency f we have that for all $a \in [k-1]$ it holds that $\sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|_a = f_a \cdot \sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|$, and, hence, the second set of constraints holds.

[If] If the frequency-f system is feasible, since it has integer coefficients, it has a rational solution $x : E \to \mathbb{N}$. By Lemma 9, we can construct a sequence of simple loops $\mathcal{T} = \sigma_1, \ldots, \sigma_l$ such that $\sum_{i=1}^{l} |\sigma_i|_e = x_e$ for all edges $e \in E$. Let $\mathcal{L}$ be the set of these loops and for each $\sigma \in \mathcal{L}$ let c_σ be the number of times σ appears in $\mathcal{T}$. Then $x_e = \sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|_e$. Due to the second set of constraints, and due to the fact that $\sum_{e \in E(a)} x_e = \sum_{\sigma \in \mathcal{L}} c_\sigma |\sigma|_a$ for all colors $a \in [k]$, we have that the natural linear combination of $\mathcal{L}$ with coefficients c_σ for each $\sigma \in \mathcal{L}$ has frequency f. □

The following theorem is an immediate corollary of Theorem 3 and Lemma 10.

Lemma 11. *In a graph G, there exists an infinite path with frequency f iff the frequency-f system for G is feasible.*

Since the feasibility problem for a system of linear equations is solvable in polynomial time in the size of the system (number of constraints and size of the coefficients) [17], we obtain the following.

Theorem 5. *The frequency-f problem is in PTIME.*

Moreover, following the procedure in the [If] proof of Lemma 10 and in the proof of Lemma 7, it is possible to build in polynomial time, from a solution of the linear system, a finite representation of a path in the graph satisfying the frequency-f property.

5.2. Bounded-difference linear system

Definition 2. Let $G = (V, E)$ be a k-colored graph with $m = |E|$, $n = |V|$, and $s_G = \min\{2n + k - 1, n + m\}$. Let $u \in V$ be a node. We call *bounded-difference system* for (G, u) the following system of equations on the set of variables $\{x_e, y_e \mid e \in E\}$.

1-4. The same constraints as in the balance system for G

5. for all $v \in V \setminus \{u\}$	$\sum_{e \in v_{\leftarrow}} y_e - \sum_{e \in v_{\rightarrow}} y_e = \sum_{e \in v_{\rightarrow}} x_e$
6.	$\sum_{e \in u_{\rightarrow}} y_e - \sum_{e \in u_{\leftarrow}} y_e = \sum_{v \in V \setminus \{u\}} \sum_{e \in v_{\rightarrow}} x_e$
7. for all $e \in E$	$y_e \geq 0$
8. for all $e \in E$	$y_e \leq (m \cdot s_G!) x_e$.

The bounded-difference system has $2m$ variables and $3m + 2n + k$ constraints. It helps to think of the vectors x and y as two loads associated to the edges of G. The constraints 1–4 are the same ones used for the balance problem for G and require that x represents a set of simple loops of G having an n.l.c of difference value zero. The constraints 5–8 are *connection constraints*, asking that y represents a connection load, from u to every other node of the simple loops defined by x and carried only on the edges of those loops. Thus, constraints 5–8 ask that the loops represented by x are overlapping, because of Lemma 1.

5 Each node $v \in V \setminus \{u\}$ absorbs an amount of y-load equal to the amount of x-load traversing it. These constraints ensure that the nodes belonging to the x-solution receive a positive y-load.
6 Node u generates as much y-load as the total x-load on all edges, except the edges exiting from u.
7 Every y-load is non-negative.
8 If the x-load on an edge is zero, then the y-load on that edge is also zero. Otherwise, the y-load can be at most $m \cdot s_G!$ times the x-load. More details on the choice of this multiplicative constant follow.

In Lemma 12, we show that if there is a solution x of the balance system, then there is another solution x' whose non-zero components are greater than or equal to 1 and less than or equal to $s_G!$, so that $\sum_{e \in E} x'_e \leq m \cdot s_G!$. In this way, the constraints (8) allow each edge that has a positive x-load to carry as its y-load all the y-load exiting from u.

Lemma 12. *Let $G = (V, E)$ be a k-colored graph, with $|V| = n$, $|E| = m$, and $s_G = min\{2n+k-1, n+m\}$. For all solutions x to the balance system for G there exists a solution x' such that, for all $e \in E$, it holds $(x_e = 0 \Rightarrow x'_e = 0)$ and $(x_e > 0 \Rightarrow 1 \leq x'_e \leq s_G!)$. As a consequence, $1 \leq \sum_{e \in E} x'_e \leq m \cdot s_G!$.*

In order to prove Lemma 12, we first introduce two preliminary results.

Lemma 13. *Let $t \in \mathbb{N}$ be a natural number and $A \in [t]_0^{m \times m}$ be a square matrix, then $|det(A)| \leq t^m m!$. Moreover, if A is not singular then $|\det(A)| \geq 1$.*

Proof. We prove the first statement by induction on m. If $m = 1$ then $|\det(A)| = |a_{1,1}| \leq t$. If the statement holds for $m - 1$, then for any $j \in [m]$ it holds that $\det(A) = \sum_{i=1}^{m} (-1)^{i+j} a_{i,j} \det(M_{i,j})$, where $M_{i,j} \in [t]_0^{(m-1) \times (m-1)}$ is a matrix obtained from A by removing the ith row and the jth column. So, $|\det(A)| \leq |\sum_{i=1}^{m} a_{i,j} \det(M_{i,j})| \leq \sum_{i=1}^{m} |a_{i,j}| |\det(M_{i,j})| \leq \sum_{i=1}^{m} t \cdot t^{m-1} (m-1)! = (tm) t^{m-1} (m-1)! = t^m m!$.

For the second statement, note that if A is not singular, since A has an integer determinant it must be $|\det(A)| \geq 1$. □

Lemma 14. *Let t be a natural number and $A \in [t]_0^{n \times m}$, $A' \in [t]_0^{n' \times m}$, $b \in [t]_0^{n \times 1}$, and $b' \in [t]_0^{n' \times 1}$ be four matrices. Let $S = \{x \in \mathbb{R}^m \mid Ax \geq b, A'x \geq b', x \geq \mathbf{0}\}$ and $M = min\{n + n', n + m\}$. If S is not empty, then there exists a vector $x \in S$ such that $x \in \mathbb{Q}^m$ and every component x_i is less than or equal to $k = M!t^M$.*

Proof. Let $I \in \mathbb{N}^{n \times n}$ be the identity matrix. First, we convert every inequality of the system $Ax \geq B$ in an equivalent equality by adding a new variable: the inequality $\sum_{j=1}^{m} a_{i,j} x_j \geq b_i$ becomes $\sum_{j=1}^{m} a_{i,j} x_j = b_i + y_i$ with $y_i \geq 0$. If we set $C = \left(\begin{smallmatrix} A & -I \\ A' & \mathbf{0} \end{smallmatrix}\right) \in \mathbb{R}^{(n+n') \times (n+m)}$, and $d = \left(\begin{smallmatrix} b \\ b' \end{smallmatrix}\right)$, we can define the set $S' = \{(x, y) \in \mathbb{R}^{m+n} \mid C \cdot (x, y)^T = d, (x, y) \geq \mathbf{0}\}$. It is easy to see that $S = \{x \in \mathbb{R}^m \mid \exists y \in \mathbb{R}^n . (x, y) \in S'\}$, thus in our hypothesis S' is not empty. Let r be the rank of C, we have that $m \leq r \leq M$, since $-I$ is not singular. By a well known result in linear programming (see, for instance, Theorem 3.5 of [17]), the set S' contains a *basic* solution, i.e., there exists a non-singular submatrix $C' \in \mathbb{R}^{r \times r}$ of C, given by the columns $i_1, \ldots, i_r$ and by the rows $j_1, \ldots, j_r$ of C, such that in S there is the point $(z_1, \ldots, z_{m+n}) \in \mathbb{R}^{m+n}$ such that $z' = (z_{i_1}, \ldots, z_{i_r})$ is the unique solution to the system $C'z' = (d_{j_1}, \ldots, d_{j_r})^T = d'$, and for all $j \notin \{i_1, \ldots, i_r\}$ it holds that $z_j = 0$. By Cramer's theorem, for all $k \in [r]$ we have that $z_{i_k} = \det(C'_{i_k}) / \det(C')$ where C'_{i_k} is the matrix obtained from C' by replacing the i_kth column with the vector d'. So, z' and z have components in $\mathbb{Q}$. Since $C', C'_{i_1}, \ldots, C'_{i_r} \in [t]_0^{r \times r}$, by Lemma 13, $|\det(C_i)| \leq r!t^r$. Moreover, since C' is not singular, we have $|\det(C')| \geq 1$. In conclusion, $z_{i_k} = |\det(C_i)| / |\det(C')| \leq (r)!t^r \leq M!t^M$, as requested. □

Now, we are ready to prove Lemma 12.

Proof. Let x be a solution to the balance system for G, and let P (for *positive*) be the set of all edges e such that $x_e > 0$. By construction, P is not empty. We represent the first two sets of equalities of the balance system in matrix form as $Dx = \mathbf{0}$. Then, the set of points satisfying the balance system is $Y = \{y \in \mathbb{R}^m \mid Dy = \mathbf{0}, y \geq \mathbf{0}, \sum_{e \in E} y_e > 0\}$. Now the subset of Y, $Y' = \{y \in Y \mid \forall e \in P : y_e \geq 1 \text{ and } \forall e \notin I : y_e = 0\} = \{y \in Y \mid \forall e \in E \,.\, (x_e > 0 \Rightarrow y_e > 1) \text{ and } (x_e = 0 \Rightarrow y_e = 0)\}$ is not empty. Indeed, the vector $z = x(\min_{e \in P} x_e)^{-1}$ is in Y', since (i) $Dz = (\min_{e \in P} x_e)^{-1} Dx = \mathbf{0}$, (ii) for all $e \in P$, we have $z_e = x_e(\min_{e \in P} x_e)^{-1} \geq 1$, and (iii) for all $e \notin P$, we have $z_e = 0$.

The set of inequalities "$\forall e \in P : y_e \geq 1$" can be represented as the system of linear equations $Fy \geq \mathbf{1}$, with $\mathbf{1} \in \{1\}^{l \times 1}$. Similarly, the set of equalities "$\forall e \notin P : y_e = 0$" can be represented as $F'y = \mathbf{0}$. If we define $D' = \binom{D}{F'} \in \{-1, 0, 1\}^{(2n+k-l-1) \times m}$, we have $Y' = \{y \in \mathbb{R}^m \mid D'y = \mathbf{0}, Fy \geq \mathbf{1}\}$. Since D', F, $\mathbf{1}$, $\mathbf{0}$ all have elements in $\{-1, 0, 1\}$, by Lemma 14, Y contains an element $x' \in \mathbb{Q}^m$ such that for all $i \in [m], x'_i \leq (\min\{2n+k-1, l+m\})! \leq (\min\{2n+k-1, n+m\})! = s_G!$, which concludes the proof. □

The following lemma states that the bounded-difference system can be used to solve the bounded-difference problem.

Lemma 15. *There exists an overlapping set of simple loops in G, passing through a node u and having an n.l.c. of difference value* **0** *iff the bounded-difference system for* (G, u) *is feasible.*

Proof (*Only if*)**.** Let $\mathcal{L}$ be an overlapping set of simple loops having an n.l.c. of difference value $\mathbf{0}$. Let c_σ be the coefficient associated with the loop $\sigma \in \mathcal{L}$ in such linear combination. We start by constructing a solution $x \in \mathbb{R}^m$ to the balance system as follows. Define $h(e, \sigma) \in \{0, 1\}$ as 1 if the edge e belongs to the loop σ, and 0 otherwise. We set $x_e = \sum_{\sigma \in \mathcal{L}} c_\sigma h(e, \sigma)$. We have that x is a solution to the balance system for G, or equivalently that it satisfies constraints 1–4 of the bounded-difference system for (G, u). By Lemma 12, there exists another solution $x' \in \mathbb{R}^m$ to the balance system, such that $x_e = 0 \Rightarrow x'_e = 0$ and $x_e > 0 \Rightarrow 1 \leq x'_e \leq s_G!$. If any loop of the overlapping set $\mathcal{L}$ passes through u, by Lemma 1, there exists a path ρ_v from u to any node v occurring in $\mathcal{L}$. We set $y_e = \sum_{v \in V' \setminus \{u\}} (h(e, \rho_v) \sum_{e \in v_\rightarrow} x'_e)$. Simple calculations show that (x', y) is a solution to the bounded-difference system for (G, u).

[If] If there exists a vector $(x, y) \in \mathbb{R}^{2m}$ satisfying the bounded-difference system, then like we did in the second part of Lemma 10, using x, we can construct a set of simple loops $\mathcal{L}$ having an n.l.c. of frequency f with all components equal to $\frac{1}{k}$. Since all colors occur the same number of times in the n.l.c. ($\frac{1}{k}$ times the number of edges in the n.l.c.), such an n.l.c. has also difference value equal to $\mathbf{0}$. Since $\sum_{e \in u_\rightarrow} y_e - \sum_{e \in u_\leftarrow} y_e = \sum_{v \in V \setminus \{u\}} \sum_{e \in v_\rightarrow} x_e$, we have that u belongs to at least one edge used in the construction of $\mathcal{L}$. If we set $G' = (V', E')$ as the subgraph of G induced by $\mathcal{L}$, we are able to show by contradiction that there is a path in G' from u to every other node of V'. Indeed, if for some $v \in V' \setminus \{u\}$ there is no path in G' from u to v, then there is some load exiting from u that cannot reach its destination using only edges of G'. Since the constraints 8 make it impossible to carry load on edges of G that are not used in $\mathcal{L}$, the connection constraints cannot be satisfied. So, for all $v \in V'$ there is a path in G' from u to v. By Lemma 1, $\mathcal{L}$ is overlapping. □

In order to solve the bounded-difference problem for G, for all $u \in V$ we check whether the bounded-difference system for (G, u) is feasible, by using a polynomial-time algorithm for feasibility of linear systems [17]. This algorithm is used at most n times and it is polynomial in the number of constraints $(2n + 3m + k)$ and in the logarithm of the maximum modulus M of a coefficient in a constraint. In our case, $M = m \cdot s_G!$. Using Stirling's approximation, we have $\log(m \cdot s_G!) = \log(m) + \Theta(s_G \log(s_G))$. Therefore, we obtain the following.

Theorem 6. *The bounded-difference problem is in PTIME.*

6. The perfectly balanced finite path problem

In this section, we introduce an NP-hard problem similar to the bounded-difference problem. Given a k-colored graph G and two nodes u and v, the new problem asks whether there exists a perfectly balanced path from u to v. We call this question the *perfectly balanced finite path problem*. To see that this problem is closely related to the bounded-difference problem, one can note that it corresponds to the statement of item 3 in Lemma 3, by changing the word *loop* to *finite path*. In the following we prove that such a problem is NP-complete.

Theorem 7. *The perfectly balanced finite path problem is NP-hard.*

Proof. We prove the statement by a reduction from 3SAT which is known to be NP-hard [6].

Given a 3SAT formula φ on n variables $x_1, \ldots, x_n$ with k clauses $C_1, \ldots, C_k$, we construct a k-colored graph G such that each color i is associated with the clause C_i. Precisely, for each variable x_j, we construct a subgraph G_j of G with a starting node q_j and an ending node q_{j+1}, as shown in Fig. 4. For $1 \leq j \leq n$, the labels $a_{j,1}, \ldots, a_{j,k_j}$ are the colors corresponding to the clauses in which x_j occurs affirmed and $a'_{j,1}, \ldots, a'_{j,k'_j}$ are the colors of the clauses in which x_j occurs negated. Moreover, the uncolored edges (those leaving from q_j and those entering q_{j+1}) concisely represent a sequence of k edges, each labeled with a different color. Finally, the graph G is obtained by concatenating each graph G_j with G_{j+1}, as they share the node q_{j+1}, for $1 \leq j < n$.

We show that the formula φ is satisfiable iff there exists a perfectly balanced path in G from q_1 to q_{n+1}.

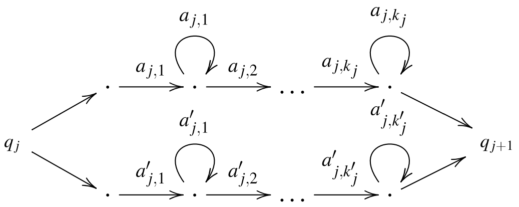

Fig. 4. Proof of Theorem 7: The jth subgraph G_j of G.

First, assume that φ is satisfiable. Then, there exists a truth assignment for the variables that satisfies φ. Using this assignment, we construct a perfectly balanced path in which each color appears exactly $2n + 3$ times. In particular, for all subgraphs G_j, the path takes the upper branch if x_j is assigned true and the lower branch otherwise. For each clause C_i, let L_i be the indexes of the variables that render C_i true, under the given truth assignment. We obtain that the constructed path passes through at least $2n + |L_i|$ non-self-loop edges colored with i. This holds because at each subgraph it passes through the uncolored edges once at the beginning and once at the end. Moreover, for all $j \in L_i$, the path passes through another non-self-loop edge labeled with i in G_j. Since $|L_i| \geq 1$, the path may pass through a self-loop labeled with i at least once in the graph. Thus, by taking $3 - |L_i|$ times one of those self-loops, we get the desired number $2n + 3$ of occurrences of i, for all colors i.

Conversely, assume that there exists a perfectly balanced path from q_0 to q_{n+1}. For all subgraphs G_j the path takes either the upper or the lower branch. Then, there are two possible situations:

1. Each color occurs $2n + l$ times with $l \geq 1$. We define the assignment in the following way: we set x_j to *true* if the path takes the upper branch in the subgraph G_j, and to *false* otherwise. We claim that such assignment satisfies φ. For all colors i the path passes through an i-colored edge α such that it is not a self-loop and it is not a starting or an ending edge of a subgraph G_j (those edges are the first $2n$). Such edge α is on a branch of a subgraph G_j, consequently the assignment for x_j satisfies the clause C_i. Being i arbitrary, all clauses C_i are satisfied by the assignment of the variable.
2. Each color occurs $2n$ times in the path. We define the assignment as follows: we set x_j to *true* if the path takes the lower branch in G_j, and to *false* otherwise. We claim that such assignment satisfies φ. For all colors i there exists at least one variable x_j appearing in the clause C_i. However, the path does not pass through any edge colored with i, except the mandatory edges at the beginning and end of each G_j. Then, in G_j the path takes the branch opposite to the assignment of x_j that makes C_i true Then, the opposite assignment of x_j (the one we choose) makes C_i true. □

We recall a result of integer programming presented in [19].

Lemma 16 ([19]). *Let $A \in \mathbb{Z}^{n\times n}$, $B \in \mathbb{Z}^{n\times 1}$, and $S = \{x \in \mathbb{Z}^n \mid Ax \leq B\}$. If S is not empty then there exists a point $x \in S$ such that the sum of the components of x is bounded by $6n^3\varphi$, where φ is the maximum sum of the coefficients of an inequality of the system $Ax \leq B$.*

Definition 3. Let $G = (V, E)$ be a k-colored graph and $u, w \in V$ be two distinct nodes. We call *perfectly balanced path system* for (G, u, w) the following system of equations on the set of variables $\{x_e, y_e \mid e \in E\}$.

$$\begin{array}{lrcl}
1.\ \text{for all } v \in V \setminus \{u, w\} & \sum_{e\in v_{\rightarrow}} x_e & = & \sum_{e\in v_{\leftarrow}} x_e \\
2. & \sum_{e\in u_{\rightarrow}} x_e & = & 1 + \sum_{e\in u_{\leftarrow}} x_e \\
3. & \sum_{e\in w_{\rightarrow}} x_e & = & -1 + \sum_{e\in w_{\leftarrow}} x_e \\
4.\ \text{for all } a \in [k-1] & \sum_{e\in E(a)} x_e & = & \sum_{e\in E(k)} x_e \\
5.\ \text{for all } e \in E & x_e & \geq & 0 \\
6.\ \text{for all } v \in V \setminus \{u\} & \sum_{e\in v_{\leftarrow}} y_e - \sum_{e\in v_{\rightarrow}} y_e & = & \sum_{e\in v_{\rightarrow}} x_e \\
7. & \sum_{e\in u_{\rightarrow}} y_e - \sum_{e\in u_{\leftarrow}} y_e & = & \sum_{v\in V\setminus\{u\}} \sum_{e\in v_{\rightarrow}} x_e \\
8.\ \text{for all } e \in E & y_e & \geq & 0 \\
9.\ \text{for all } e \in E & y_e & \leq & (6(d-1)^3\varphi)x_e \\
10.\ \text{for all } e \in E & x_e, y_e & \in & \mathbb{Z}.
\end{array}$$

where φ is the maximum sum of the coefficients of any inequality in the first six sets of constraints.

Let $m = |E|$ and $n = |V|$, the perfectly balanced path system has $2m$ variables and $3m + 2n + k$ constraints. It helps to think of the vectors x and y as two integer loads associated to the edges of G. The constraints 1–5 for G, and they ask that x should represent a path from u to w and a set of simple loops such that the latter has an n.l.c. whose difference value is the opposite of the difference vector of the path (constraint 4). Without constraints 6–9, the simple loops could be disconnected from the path from u to w.

The constraints 6–9 are *connection constraints*, asking that y should represent a connection load, from u to every other node of the simple loops defined by x, and carried only on the edges of those loops. Thus, the constraints 6–9 ask that the loops represented by x should be reachable from u, using only edges represented by x, similarly to the bounded-difference system of Section 5. The only difference is the bound in the constraints 9, which is justified by Lemma 16. Observe that Lemma 16 is applicable to the constraints 1–5 that define x.

Lemma 17. *There exists a perfectly balanced path in G from u to w iff the perfectly balanced path system (G, u, w) is feasible.*

The proof of the previous lemma is similar to the proof of Lemma 15. Since the feasibility problem for an integer linear system is in NP, and by Theorem 7, we obtain the following.

Theorem 8. *The perfectly balanced finite path problem is NP-complete.*

7. Colored games with frequency goals

In this section, we study k-colored games having one of the following goals.

- The *bounded-difference* goal W_{bn}, containing all and only the bounded-difference color sequences.
- Let $f \in \mathbb{Q}^k$ be such that $\sum_{a=1}^{k} f_a = 1$ and $f_a \geq 0$ for all $a \in [k]$. The *frequency-f* goal W_f, containing all and only the color sequences with color frequency vector f.
- The *balance* goal W_{bl}, containing all and only the balanced color sequences (i.e. with frequency vector with all equal components).

7.1. Co-NP membership

We prove that the problem of deciding whether there exists a winning strategy for player 0 in the games defined above is in Co-NP.

Lemma 18. *W_{bn}, W_{bl}, and W_f are convex.*

Proof. Let $y, z \in [k]^\omega$ and $x \in y \otimes z$. We prove that if y and z are both balanced (resp., bounded-difference, or frequency-f), then so is x. We have that $x = x_1 \dots x_i \dots$ where $y = x_1 x_3 \dots x_{2k+1} \dots$ and $z = x_2 x_4 \dots x_{2k} \dots$, with $x_i \in [k]^*$ for all i. Also, for all $n \in \mathbb{N}$ there are two indexes n_y, n_z such that $n = n_y + n_z$ and $\mathit{diff}_{a,b}(x^{\leq n}) = \mathit{diff}_{a,b}(y^{\leq n_y}) + \mathit{diff}_{a,b}(z^{\leq n_z})$, for all $a, b \in [k]$. Precisely, if $n = |x_1| + |x_2| + \cdots + |x_{k-1}| + t$, with $t \in \mathbb{N}$, then $x^{\leq n} = x_1 x_2 \dots x_k^{\leq t}$ and, if k is even, we have $n_y = |x_1| + |x_3| + \cdots + |x_{k-1}|$ and $n_z = |x_2| + |x_4| + \cdots + |x_{k-2}| + t$, otherwise we obtain $n_y = |x_1| + |x_3| + \cdots + |x_{k-2}| + t$ and $n_z = |x_2| + |x_4| + \cdots + |x_{k-1}|$. We now distinguish the following cases.

1. (*bounded-difference*) Since y and z are bounded-difference, there exist two constants $C_y, C_z \in \mathbb{N}$ such that for all $a, b \in [k]$ and for all $n > 0$, $|\mathit{diff}_{a,b}(y^{\leq n})| < C_y$ and $|\mathit{diff}_{a,b}(z^{\leq n})| < C_z$. Therefore, let $C_x = C_y + C_z$, for all $a, b \in [k]$ and $n \in \mathbb{N}$ we have $|\mathit{diff}_{a,b}(x^{\leq n})| \leq |\mathit{diff}_{a,b}(y^{\leq n_y})| + |\mathit{diff}_{a,b}(z^{\leq n_y})| \leq C_x$. Hence, the sequence x is bounded-difference.
2. (*frequency-f*) Given that y and z have frequency f, we have that, for all $a \in [k]$ and for all $\varepsilon > 0$, there exists $h(\varepsilon) > 0$ such that for all $n > h(\varepsilon)$, it holds that $\left|\frac{\|y^{\leq n}\|_a}{n} - f_a\right| \leq \varepsilon$ and $\left|\frac{\|z^{\leq n}\|_a}{n} - f_a\right| \leq \varepsilon$. Hence, given $\varepsilon > 0$, let $n > 0$ be such that $n_y \geq h(\varepsilon/2)$ and $n_z \geq h(\varepsilon/2)$. Such n exists, due to the definition of the shuffle operation. For all $n' > n$ we have that: $\left|\frac{\|x^{\leq n'}\|_a}{n'} - f_a\right| = \left|\frac{\|y^{\leq n'_y}\|_a + \|z^{\leq n'_z}\|_a - (n'_y + n'_z) f_a}{n'_y + n'_z}\right| \leq \left|\frac{\|y^{\leq n'_y}\|_a - n'_y \cdot f_a}{n'_y + n'_z}\right| + \left|\frac{\|z^{\leq n'_z}\|_a - n'_z \cdot f_a}{n'_y + n'_z}\right| \leq \left|\frac{\|y^{\leq n'_y}\|_a}{n'_y} - f_a\right| + \left|\frac{\|z^{\leq n'_z}\|_a}{n'_z} - f_a\right| \leq \varepsilon$. So, the color sequence x has frequency vector f.
3. (*balanced*) Since the balance property is equivalent to the frequency-f property with f_a equal to $1/k$ for all colors $a \in [k]$, the thesis holds. □

Now, we can apply Theorem 1 to our goals and obtain the following.

Corollary 1. *Let G be a k-colored game with balance, bounded-difference, or frequency-f goal. The game is determined. Moreover if player 1 has a winning strategy, he has a memoryless winning strategy.*

The fact that memoryless strategies suffice for player 1 easily leads to the following result.

Lemma 19. *Given a k-colored game with balanced, bounded-difference, or frequency-f goal, the problem asking whether there exists a winning strategy for player 1 is in NP, the problem asking whether there exists a winning strategy for player 0 is in Co-NP.*

Proof. By Corollary 1, if player 1 has a winning strategy, he has a memoryless one. The number of memoryless strategies is finite and each one of them can be represented in polynomial space in the size of the problem. So, in polynomial time we can guess a memoryless strategy τ, and verify that it is a winning strategy, using the following algorithm. We construct the sub-arena A', obtained from A by removing all the edges of player 1 that are not used by τ. We have that τ is a winning strategy for player 1 in A iff all the plays on A' are winning for player 1. Thus, player 0 is able to construct a balanced (resp., bounded-difference, frequency-f) path iff there exists a balanced (resp., bounded-difference, frequency-f) path in the graph of A' and this path is reachable from v_{ini}. So, we construct the subgraph A'' of A', obtained by removing all the nodes that are not reachable from v_{ini}. In order to check if there exists a balanced (resp., bounded-difference, frequency-f) path reachable from v_{ini}, it is sufficient to apply the polynomial-time algorithms presented in Section 5.

This concludes the proof that the problem of asking whether there exists a winning strategy for player 1 is in NP. Hence, the complementary problem asking whether there exists a winning strategy for player 0 is in Co-NP. □

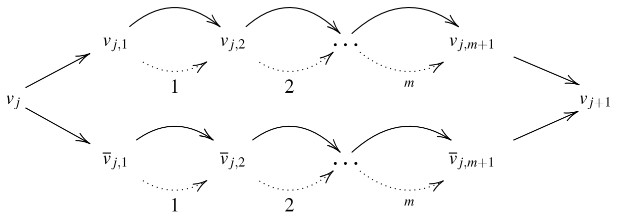

Fig. 5. The jth subgraph A_j of A. The dotted edge from $v_{j,i}$ to $v_{j,i+1}$ is present iff $\psi_i \in \psi(x_j)$, and analogously for the lower branch.

7.2. Co-NP hardness

Lemma 20. *Given a Boolean formula ψ in conjunctive normal form, there exists a k-colored arena A such that the following are equivalent* (i) *ψ is a tautology,* (ii) *there exists a winning strategy for player* 0 *in the game $G = (A, W_{bl})$, and* (iii) *there exists a winning strategy for player* 0 *in the game $G' = (A, W_{bn})$.*

Proof. Let m be the number of clauses of ψ and n be the number of its variables, then we can write $\psi = \wedge_{i=1}^{m}\psi_i$, where each ψ_i is a disjunction of literals. In the following we define $\psi(x)$ as the set of all clauses in which x appears in positive form, and $\psi(\overline{x})$ as the set of all clauses in which x appears negated.

We construct the following $(m+1)$-colored arena $A = (V_0, V_1, v_{\text{ini}}, E)$, where the set of colors corresponds to the set of clauses of ψ with the added control color $m+1$. The description of the arena A makes use of *uncolored edges*, i.e., edges not labeled by any color. Clearly, such an edge can be represented in our framework by a sequence of $m+1$ edges, each labeled by a different color, so that the colors are balanced and do not contribute to the color differences. The arena A is composed by n sub-arenas A_j, one for each variable x_j. Every sub-arena A_j, depicted in Fig. 5, has a starting node v_j, an ending node v_{j+1} and two sequences of nodes: $\{v_{j,i}\}_{i=1}^{m}$, $\{\overline{v}_{j,i}\}_{i=1}^{m}$ where every node is associated with a clause. There is an uncolored edge from v_j to $v_{j,1}$ (resp., $\overline{v}_{j,1}$, and from $v_{j,m+1}$ (resp., $\overline{v}_{j,m+1}$) to v_{j+1}. Moreover, we have that for all $1 \le i \le m$, (i) there is an uncolored edge from $v_{j,i}$ to $v_{j,i+1}$ and from $\overline{v}_{j,i}$ to $\overline{v}_{j,i+1}$, (ii) if $\psi_i \in \psi(x)$ then there is an i-colored edge from $v_{j,i}$ to $v_{j,i+1}$, and (iii) if $\psi_i \in \psi(\overline{x})$ then there is an i-colored edge from $\overline{v}_{j,i}$ to $\overline{v}_{j,i+1}$. We call the sequence $\{v_{j,i}\}_i$ the *upper branch* of A_j and the sequence $\{\overline{v}_{j,i}\}_i$ the *lower branch* of A_j. The arena A is constructed by concatenating the sub-arena A_j with A_{j+1}, as they share node v_{j+1}, and by adding an $(m+1)$-colored edge back from v_n to v_1.

The construction of A is concluded by partitioning the set of nodes as follows: $V_1 = \{v_1, \dots, v_n\}$ and $V_0 = V \setminus V_1$. Intuitively, every sub-arena A_j represents a truth choice for the variable x_j. This choice is made by player 1 with the aim of skipping the passage through some clauses. On the other hand, as soon as there is the chance, player 0 tries to pass through each clause once during a single loop, in order to balance the clauses' colors with the control color $m+1$. Let $G = (A, W_{bl})$ and $G' = (A, W_{bn})$, we now show the correctness of the above construction. In the following, we write $\widetilde{v}_{j,i}$ to mean either $v_{j,i}$ or $\overline{v}_{j,i}$.

[(i) implies (ii) and (iii)] If ψ is a tautology, then the winning strategy for player 0 in both games G and G' may be summarized as follows: for each color i, as soon as there is a chance pass through an edge of color i; then, do not pass through such an edge again, until we pass again through v_1. Formally, the strategy of player 0 is the following: each time the play is in a node $\widetilde{v}_{j,i}$, player 0 chooses to reach $\widetilde{v}_{j,i+1}$ through the i-colored edge iff color i does not appear in the least suffix of the partial play starting with v_1. We observe that during a single loop from v_1 to itself, a strategy of player 1 is a truth assignment to the variables of ψ: precisely for every sub-arena A_j, player 1 chooses to follow the upper branch iff x_j is true. Since ψ is a tautology, any such assignment satisfies ψ, i.e., given such an assignment $a : \{x_1, \dots, x_m\} \to \{T, F\}$, for each clause ψ_i, there exists a variable x such that ψ_i is true also due to the value $a(x)$. This means that player 0 can pass through an i-colored edge at least once during a single loop, and thanks to his strategy, he will pass through such an edge exactly once. Thus, during each loop, the uncolored edges are already perfectly balanced, and the edges added by player 0 are balanced thanks to the last $(m+1)$-colored edge. Thus, during the infinite play, the color differences are always zero when the play is in node v_1. Since the loops from v_1 to itself have bounded length, the color differences are bounded during the play. Thus every infinite play consistent with the strategy is a bounded-difference path and, hence, a balanced path.

[(ii) or (iii) implies (i)] We prove the contrapositive, i.e., if (i) is false then both (ii) and (iii) are false. If ψ is not a tautology, then there is a memoryless winning strategy for player 1 on G and G': player 1 follows a truth assignment of the variables of ψ that does not satisfy ψ. For such an assignment there is an unsatisfied clause ψ_i. So, during a loop from v_1 to itself, if player 1 follows this strategy, player 0 cannot pass through any i-colored edge. Thus, at the end of the loop the color difference between colors i and $m+1$ is increased by one. Every play ρ is an infinite concatenation of simple loops from v_1 to itself. Since these loops have maximum length $l \le |E|$, for all $j \in \mathbb{N}$, we have $\mathit{diff}_{i,m+1}(\rho^{\le j}) \ge \frac{j}{l}$, and thus $\lim_{j\to+\infty} \frac{\mathit{diff}_{i,m+1}(\rho^{\le j})}{j} \ge \frac{1}{l}$. Therefore, every play consistent with that strategy of player 1 is not a balanced path, and hence not a bounded-difference path. □

Theorem 9. *Given a k-colored game G with balanced (resp., bounded-difference, frequency-f) goal, the problem asking whether there exists a winning strategy for player* 0 *is Co-NP-complete.*

Proof. By Lemmas 19 and 20, we have that the problems for the balance and the bounded-difference goal are Co-NP-complete. Since the bounded-difference goal is a special case of frequency-f goal (for $f_i = 1/k$), we have that the frequency-f problem is Co-NP-hard too. Since by Lemma 19 the problem for frequency-f is in Co-NP, it is Co-NP-complete. □

8. Conclusions

We characterized the computational complexity of certain notions of quantitative fairness on graphs and 2-player games. As far as graphs are concerned, the reductions to linear programming pave the way for applications such as the automatic generation of quantitatively fair plans or schedules. On the other hand, the Co-NP-completeness result obtained for games may be regarded as essentially negative. In fact, the algorithm showing membership in (Co-)NP, once converted into a deterministic form, simply suggests to try each one of the (exponentially many) memoryless strategies of player 1 in the game, and solve a linear program to determine whether it is winning. It remains to investigate the possibility of practically efficient algorithms, arising, for instance, from the analysis of the specific properties of the games of interest.

A natural open question arising in this framework is the following: if on a colored graph, or game, there is no bounded-difference nor balanced path, what is the "most balanced" path one can achieve? This problem requires the introduction of a suitable order relation on paths, defining when a path is "more balanced" than another.

References

[1] R. Alur, T. Henzinger, Finitary fairness. ACM Trans. on Programming, Languages and Systems 20 (6) (1998).

[2] A. Bianco, M. Faella, F. Mogavero, A. Murano, Balanced paths in colored graphs, in: MFCS'09, in: LNCS, vol. 5734, Springer, 2009, pp. 149–161.

[3] A. Bianco, M. Faella, F. Mogavero, A. Murano, 2010. Quantitative fairness games. In: QAPL'10. vol. 28 of EPTCS.

[4] K. Chatterjee, T.A. Henzinger, Finitary winning in omega-regular games, in: TACAS'06, in: LNCS, vol. 3920, Springer, 2006, pp. 257–271.

[5] E. Coffman, P. Denning, Operating Systems Theory, Prentice Hall, 1973.

[6] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, second ed., MIT Press, 2001.

[7] L. de Alfaro, 1999. From fairness to chance. ENTCS 22, 55–87.

[8] N. Dershowitz, D. N. Jayasimha, S. Park, Bounded fairness, in: Verification: Theory and Practice, in: LNCS, vol. 2772, Springer, 2003, pp. 304–317.

[9] N. Francez, Fairness, Springer, 1986.

[10] E. Kopczyński, Half-positional determinancy of infinite games, in: ICALP'06, Springer, 2006, pp. 336–347.

[11] O. Kupferman, M. Vardi, P. Wolper, Module checking, Information and Computation 164 (2001) 322–344.

[12] M. Kwiatkowska, Survey of fairness notions, Information and Software Technology 31 (7) (1989) 371–386.

[13] D. Lehmann, A. Pnueli, J. Stavi, Impartiality, justice and fairness: the ethics of concurrent termination, in: ICALP'81, in: LNCS, vol. 115, Springer, 1981, pp. 264–277.

[14] A. Mahammed, A. Al-Massarani, 1993 Priority based scheduling and evaluation of precedence graphs with communication times. Ph.D. Thesis, King Fahd University of Petroleum and Minerals.

[15] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems: Specification, Springer, 1991.

[16] R. McNaughton, Infinite games played on finite graphs, Annals of Pure and Applied Logic 65 (1993) 149–184.

[17] G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, Wiley-Interscience, 1988.

[18] R. Parikh, On context—free languages, Journal of the ACM 13 (4) (1966) 570–581.

[19] A. Schrijver, Theory of Linear and Integer Programming, John Wiley and Sons, 1986.

[20] W. Thomas, On the synthesis of strategies in infinite games, in: STACS'95, in: LNCS, vol. 900, Springer, 1995, pp. 1–13.

[21] Wikipedia, 2009. Crop rotation. http://www.wikipedia.com.

[22] W. Zielonka, Infinite games on finitely coloured graphs with applications to automata on infinite trees, Journal of the Theoretical Computer Science 200 (1–2) (1998) 135–183.