

Directed Grabbing Games or How to Politely Grab the Maximum Number of Olives in a Reception

Jean-Claude Bermond 

Université Côte d’Azur, Inria, CNRS, I3S, Sophia Antipolis, France

Michel Cosnard 

Université Côte d’Azur, Inria, CNRS, I3S, Sophia Antipolis, France

Frédéric Havet 

Université Côte d’Azur, Inria, CNRS, I3S, Sophia Antipolis, France

Takako Kodate 

Department of Information and Mathematical Sciences, Tokyo Woman’s Christian University, Japan

Stéphane Pérennes 

Université Côte d’Azur, Inria, CNRS, I3S, Sophia Antipolis, France

Abstract

We introduce and study the directed grabbing game, a directed variation of the graph grabbing game played by two players Alice and Bob on a weighted acyclic digraph D . Alice plays first and then they play alternately. At a given odd (resp. even) move, Alice (resp. Bob) chooses a sink, that is a vertex of out-degree 0, grabs the weight (olives) on it and then removes the vertex. The aim of each player is to grab a maximum weight, i.e. a maximum number of olives. This game is inspired by the behaviour that guests are expected to adopt during a reception or cocktail party.

We first consider the case where hors d’oeuvre are arranged on slightly spaced parallel lines, such that politeness allows one to take the first hors d’oeuvre from each line. This corresponds to the directed grabbing game on a union of disjoint directed paths. We give an algorithm that, given a weighted digraph D of order n which is the union of q disjoint directed paths, computes an optimal play in $O(n \log q)$ time.

Then we consider the “pissaladière case” where the digraph D is a directed $(p \times q)$ -grid. We show that, depending on the parity of pq , one player, called Content, has a strategic advantage. Specifically, Content is Alice when pq is odd and Bob when pq is even. We present some strategies that enable Content to remove some large sets of vertices (of order $pq/2$) in directed grids. We then derive that Content can remove any given vertex that is not in the border of the grid. Finally, in the case where each vertex contains either zero or one olive, we prove that Content can secure the grabbing of around one third of the olives.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases grabbing games, paths, directed grids

Digital Object Identifier 10.4230/LIPIcs.FUN.2026.8

Funding French government, through the UCAjedi Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-15-IDEX-01.

Takako Kodate: International Academic Exchange Fund from Tokyo Woman’s Christian University.

1 Introduction

Graph grabbing and graph sharing games have been introduced by P. Winkler in [5] and have since attracted a lot of attention (see the references in [2]). In such games, two players Alice and Bob play alternatively on a connected graph with non-negative weights assigned to the vertices. There are some differences according to the rule taken to grab a vertex. The aim of each player is to grab a maximum weight.



© Jean-Claude Bermond, Michel Cosnard, Frédéric Havet, Takako Kodate, and Stéphane Pérennes; licensed under Creative Commons License CC-BY 4.0

13th International Conference on Fun with Algorithms (FUN 2026).

Editor: John Iacono; Article No. 8; pp. 8:1–8:19



Leibniz International Proceedings in Informatics

LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A standard rule is rule (R) that imposes that the subgraph induced by the taken vertices is connected during the whole game. The most famous cases, introduced by P. Winkler in his famous book *Mathematical Puzzles* [6], are the ones when the graph is a path and a cycle and the problems are entitled COINS IN A ROW and SHARING A PIZZA.

In the case of a cycle P. Winkler shows that if the number of vertices (slices of the pizza) is even Alice can always get at least half of the weight. In contrast, and quite surprisingly, when the number of vertices is odd, the maximum fraction of the total weight that Alice can always grab is only $4/9$ [1, 4]. In [2], we studied in details the case of a path (under the name SHARING A LINEAR PIZZA). During the presentation of our results at FUN 2022, Tomasz Idziaszek informed us that he has obtained in 2012 a linear-time algorithm to determine the gain of each player ([3]).

In [2], we introduced the d -degenerate rule (D_d) yielding to the **d -degenerate game**.

(D_d): At each move, a player can only pick a **d -removable** vertex, which is a vertex of degree at most d in the remaining graph.

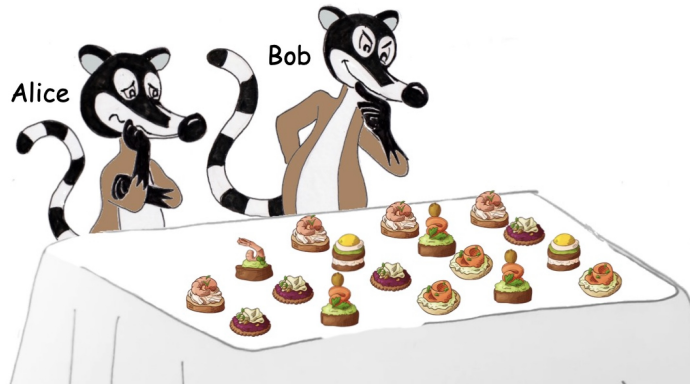
Of course, with such a rule, the graph can be entirely shared (i.e. all the vertices can be removed) only if at each step there is a removable vertex. This is the case if and only if the graph is d -degenerate. Recall that a graph is **d -degenerate** if each of its subgraphs has a vertex of degree at most d . The 1-degenerate connected graphs are the trees. In particular, the 1-degenerate game on a path is exactly COINS IN A ROW.

In [2], Hungry Coatis focused on a particularly important case (at least for them), called SHARING A PISSALADIÈRE. “Pissaladière” is marvelous dish of higher importance to the authors, originally a specialty of the French city of Nice (under the name “pissaladeria”) and of Liguria (under the name “piscialandrea”). The classical pissaladière is presented as a rectangular grid where the slices are small rectangles. Its traditional topping consists of caramelised (almost pureed) onions, anchovies and delicious small black olives of Nice, of which everyone is fond. The politeness dictates that players take turns picking a slice which is adjacent to at most two other slices. Hence, in the first move Alice can only pick one of the four corners of the pissaladière, and in the second move Bob can only pick one of the three remaining corners or one of the two slices adjacent to the corner removed by Alice. Grids are 2-degenerate graphs and the 2-degenerate game on grids is exactly SHARING A PISSALADIÈRE.

In this paper, we introduce and study the **directed grabbing game**, which is a directed variation played on a weighted acyclic digraph D . There are two players Alice a.k.a. A and Bob a.k.a. B . Alice plays first; then they play alternately. At a given odd (resp. even) move, Alice (resp. Bob) chooses a **sink**, that is a vertex of out-degree 0, grabs the weight (olives) on it and then removes the vertex. The aim of each player is to grab a maximum weight, i.e. a maximum number of olives.

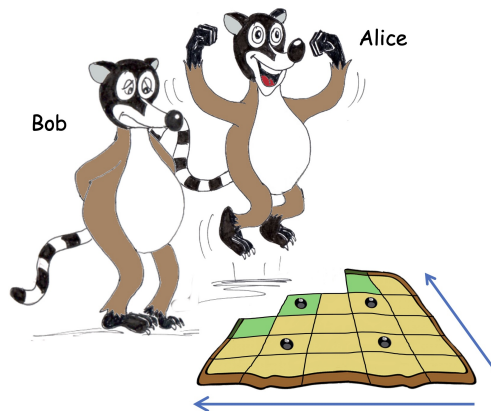
This game is inspired by the behaviour that guests are expected to adopt during a reception or cocktail party. On such occasions, hors d’œuvres are arranged on trays of various shapes, and it is considered quite rude to take a piece that is behind another. Thus, by representing each hors d’œuvre as a vertex and drawing an arc from one to another if the first is located behind the second, we obtain an acyclic digraph in which only the sinks can be chosen. Moreover, since hors d’œuvres are generally not of equal quality, we can assign each a weight corresponding to the pleasure derived from eating it. If two people are gathered around the same tray, politeness also dictates that they take turns. But of course, beneath this veneer of courtesy, each person will try to maximize their own pleasure – that is, to take hors d’œuvres/vertices such that the sum of the weights is as large as possible.

A classic setting appears when the hors d'oeuvres are arranged on slightly spaced parallel lines, such that politeness allows one to take the first hors d'oeuvre from each line. It corresponds to playing on a digraph which is the union of disjoint directed paths (see Figure 1).



■ **Figure 1** Two hungry coatis, Alice and Bob, are facing hors d'oeuvres arranged in five parallel lines of four. Their favorite ones are those topped with an olive, so their goal is to eat as many of them as possible. Three hors d'oeuvres have already been removed (two from the first line and one from the second line). It is Bob's turn. He could remove the hors d'oeuvre at the front of any row, which are the ones topped with shrimp. However, his optimal strategy is to remove the one with the shrimp tail pointing upward, and then remove the hors d'oeuvre immediately behind the one Alice has just taken. By doing so, he will collect all the olives.

Another popular case is pissaladière where there are olives on each piece and each player wants to eat as many olives as possible. The traditional shape of pissaladières is rectangular, and at receptions, they are often presented on a table along a wall, with one corner of the rectangle pointing toward the guests. The guests must then begin eating the pissaladières starting from that corner. In such a case, the digraph D is a directed grid (see Figure 2).



■ **Figure 2** Now our two hungry coatis are facing a pissaladière with 25 parts arranged in a 5×5 grid. Four parts have already been removed and it is Alice's turn to play. Alice can get any of the three parts in green and she is happy because the one in the middle has an olive. Furthermore, thanks to Theorem 11, she has a strategy to get all the remaining olives.

1.1 Our results

We first deal with the case of general weights.

In Section 2 we prove the intuitive result stating that, if a vertex x is a sink with maximum weight, then there is an optimal play starting with x .

In Section 3, we consider the first setting of hors d'oeuvre. We give an algorithm that, given a weighted digraph D of order n which is the union of q disjoint directed paths, computes an optimal play in $O(n \log q)$ time.

Next, we consider the “pissaladière case” where the digraph D is a directed grid $\vec{G}_{p \times q}$ (see precise definition in Section 4). We establish two opposite behaviours depending on whether the grid has an odd or an even number of vertices (i.e. depending on whether $n = pq$ is odd or even). Let **Content** be Alice when pq is odd and Bob when pq is even, and let **Defeated** be Bob when pq is odd and Alice when pq is even. We establish several results showing that Content has a strategic advantage on a directed grid.

First, in Section 4, we present some strategies which enable Content to remove some large sets of vertices (or order $n/2$) in directed grids. Then, in Section 5, using the previously established strategies, we determine which player removes a given vertex. In particular, we show in Theorem 15 that if the vertex is not in the border (i.e. the union of the first row and first column), then Content removes it. Finally, in Section 6, we restrict our study to the case where w is a $\{0, 1\}$ -valued function. This corresponds to the case of a pissaladière where there is zero or one olive on each piece. In particular, we give bounds on the minimum number of olives that a player collects over all possible games with k olives on $\vec{G}_{p \times q}$. We show in Proposition 19 that Defeated is not guaranteed to collect any olive unless the number of olives is at least half the size of the grid. In contrast, we prove in Theorem 22 that Content can always collect at least one third of the olives minus a small constant and even more when the number of olives becomes large compared to the size of the grid.

We conclude the paper in Section 7, by giving some open problems.

1.2 Definitions

Let (D, w) be a weighted acyclic digraph. A **configuration** at time t is the configuration obtained after t moves. So the configuration at time 0 is the initial digraph. Move t is the move played by a player between time $t - 1$ and time t . It is the removal by the player of a sink (vertex of out-degree 0) in the configuration at time $t - 1$. So the first move of Alice is move 1 between time 0 and time 1.

A **play** σ is a sequence of moves made alternately by Alice and Bob. This is an **acyclic ordering** of D , that is an ordering $(v_1, v_2, \dots, v_n)$ of the vertices such that there is no arc $v_i v_j$ with $i < j$.

Observe that if the digraph has a hamiltonian dipath $v_n v_{n-1} \dots v_1$ (v_1 being the sink), then its unique acyclic ordering (and thus play) is $(v_1, v_2, \dots, v_n)$, so Alice grabs the weight of odd-indexed vertices and Bob those of even-indexed vertices.

The (odd) moves done by Alice in a play will be called the **play of A** or **A-play** and denoted by σ^A , while the (even moves) done by Bob will be called the **play of B** or **B-play** and denoted by σ^B . Thus a play σ is the alternate sequence of elements of σ^A and σ^B in order. We write $\sigma = \sigma^A \oplus \sigma^B$.

We denote by $\text{col}^A(D, w, \sigma)$ (resp. $\text{col}^B(D, w, \sigma)$) the weight collected by Alice (resp. Bob) when these players remove the vertices according to the play σ . The **value** of a play σ in (D, w) is $\text{val}(D, w, \sigma) = \text{col}^A(D, w, \sigma) - \text{col}^B(D, w, \sigma)$.

An **optimal play** is a sequence of moves made alternately by Alice and Bob when they both play optimally, that is according to the classical MiniMax recursive algorithm on the game tree. More precisely Alice (resp. Bob) has a h -strategy if Alice (resp. Bob) grabs at least a weight h whatever Bob (resp. Alice) plays. The **optimal gain** of Alice (resp. Bob), denoted by $\text{gain}^A(D, w)$ (resp. $\text{gain}^B(D, w)$) is the maximum h such that Alice (resp. Bob) has an h -strategy. This strategy will be called **optimal**. The **optimal value** of (D, w) , denoted by $\text{opt}(D, w)$, is the value of an optimal play, which is equal to $\text{gain}^A(D, w) - \text{gain}^B(D, w)$.

2 Removing a sink with maximum weight

The **distance** between two weight functions w and w' of D is $\text{dist}(w, w') = \sum_{v \in V(D)} |w(v) - w'(v)|$.

► **Lemma 1.** *Let D be an acyclic digraph and w and w' be two weight functions at distance d . Then $|\text{opt}(D, w) - \text{opt}(D, w')| \leq d$.*

Proof. Let σ and σ' be optimal plays for (D, w) and (D, w') respectively. We have $\text{val}(D, w, \sigma) = \text{opt}(D, w)$ and $\text{val}(D, w', \sigma') = \text{opt}(D, w')$.

$$\begin{aligned} \text{opt}(D, w) &\geq \text{val}(D, w, \sigma') \\ &= \text{col}^A(D, w, \sigma') - \text{col}^B(D, w, \sigma') \\ &\geq \text{col}^A(D, w', \sigma') - \text{col}^B(D, w', \sigma') - \sum_{v \in V(D)} |w(v) - w'(v)| \\ &= \text{val}(D, w', \sigma') - d \\ &= \text{opt}(D, w') - d \end{aligned}$$

Similarly we get $\text{opt}(D, w') \geq \text{opt}(D, w) - d$. ◀

A **maximum-weight** vertex in (D, w) is a vertex x such that $w(x) \geq w(v)$ for all $v \in V(D)$.

► **Lemma 2.** *Let (D, w) be a weighted acyclic digraph. If x is a sink and maximum-weight vertex, then there is an optimal play starting with x .*

Proof. We prove the result by induction on the number of vertices of D , the result holding trivially for $|V(D)| = 1$.

Assume now that $|V(D)| \geq 2$, and that there is a vertex x with out-degree 0 with maximum weight. Let $\sigma = (v_1, v_2, \dots, v_n)$ be an optimal play for (D, w) . If $x = v_1$, then we have the result. Henceforth, we may assume that $x \neq v_1$. Now, the induction hypothesis applied to $(D - v_1, w)$ (in which x is a sink and a maximum-weight vertex) implies that there is an optimal play for $(D - v_1, w)$ starting with x . Thus, we may assume that $x = v_2$. Let $D^* = D - \{v_1, v_2\} = D - \{v_1, x\}$. We have $\text{opt}(D, w) = \text{val}(D, w, \sigma) = w(v_1) - w(x) + \text{opt}(D^*, w)$.

Let $\max_x$ be the maximal value of a play starting with x and optimal afterwards. $\max_x = w(x) - \text{opt}(D - x, w)$. Let w' be the weight function defined on $D - x$ by $w'(u) = w(x)$ if $u = v_1$ and $w'(u) = w(u)$ otherwise. Observe that in $(D - x, w')$, v_1 is a sink and a maximum-weight vertex. By the induction hypothesis, there is an optimal play starting with v_1 . So $\text{opt}(D - x, w') = w'(v_1) - \text{opt}(D^*, w') = w(x) - \text{opt}(D^*, w)$. Thus, by Lemma 1, $\text{opt}(D - x, w) \leq \text{opt}(D - x, w') + w(x) - w(v_1) = 2w(x) - w(v_1) - \text{opt}(D^*, w)$. Hence $\max_x \geq w(v_1) - w(x) + \text{opt}(D^*, w) = \text{opt}(D, w)$. Therefore, there is an optimal play starting with x . ◀

3 Union of disjoint directed paths

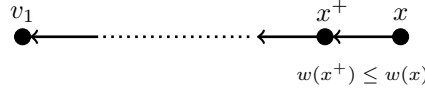
Let D be the union of disjoint directed paths and v a vertex of D . If $d^+(v) = 1$, then the unique out-neighbour of v in D is denoted by v^+ . If $d^-(v) = 1$, then the unique in-neighbour of v in D is denoted by v^- .

The aim of Propositions 3 and 5 is to give reductions which relate the value of (D, w) to the value of a weighted digraph obtained by deleting two vertices of D (and adjusting some weights). Repeating these reductions, we will end to compute the value of a set of disjoint directed paths where the weights on each directed path are strictly increasing and where the value can be computed in $O(n)$ time.

► **Proposition 3** (First reduction). *Let D be a union of disjoint directed paths, and w a weight function on $V(D)$.*

If $d^-(x) = 0$ and $d^+(x) = 1$, and $w(x) \geq w(x^+)$, then

$$\text{val}(D, w) = \text{val}(D - \{x, x^+\}, w) + (-1)^{|V(D)|} (w(x^+) - w(x)).$$



■ **Figure 3** The situation of Proposition 3.

Proof. Assume $|V(D)|$ is even.

Bob can play the following strategy. Bob never removes vertex x^+ . This is possible because $|V(D)|$ is even and x^+ has to be removed before x . If Alice removes a vertex in $D - \{x, x^+\}$, then Bob plays according to an optimal strategy on $(D - \{x, x^+\}, w)$. If Alice removes x^+ , then Bob removes x . Doing so Bob collects at least $\text{gain}^B(D - \{x, x^+\}, w) + w(x)$ so

$$\text{gain}^B(D, w) \geq \text{gain}^B(D - \{x, x^+\}, w) + w(x) \quad (1)$$

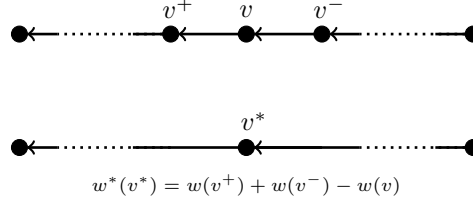
Alice can play the following strategy. If Bob removes x^+ , then Alice removes x . Otherwise she removes if possible a vertex of $D - \{x, x^+\}$; but as $|V(D)|$ is even, at the end Alice is forced to remove x^+ . Doing so, Alice collects either $\text{gain}^A(D - \{x, x^+\}, w) + w(x)$ if Bob removes x^+ , or $\text{gain}^A(D - \{x, x^+\}, w) + w(x^+)$ if Alice removes x^+ at the end. Since $w(x^+) \leq w(x)$, we get

$$\text{gain}^A(D, w) \geq \text{gain}^A(D - \{x, x^+\}, w) + w(x^+) \quad (2)$$

Finally, as $\text{gain}^A(D, w) + \text{gain}^B(D, w) = \text{gain}^A(D - \{x, x^+\}, w) + \text{gain}^B(D - \{x, x^+\}, w) + w(x) + w(x^+)$, Inequalities (1) and (2) imply $\text{gain}^A(D, w) = \text{gain}^A(D - \{x, x^+\}, w) + w(x^+)$ and $\text{gain}^B(D, w) = \text{gain}^B(D - \{x, x^+\}, w) + w(x)$ and so $\text{val}(D, w) = \text{val}(D - \{x, x^+\}, w) + w(x^+) - w(x)$.

When $|V(D)|$ is odd, the proof is similar with the role of Alice and Bob interchanged. ◀

► **Definition 4** (v -contract of (D, w)). *If $d^-(v) = d^+(v) = 1$, then the v -contract of (D, w) is the weighted digraph (D^*, w^*) where D^* is the digraph obtained from D by contracting the vertices v^-, v, v^+ into a single vertex v^* and w^* is defined by $w^*(v^*) = w(v^+) + w(v^-) - w(v)$ and $w^*(u) = w(u)$ for every $u \in V(D^* - v^*)$. See Figure 4.*



■ **Figure 4** A directed path (top) and its v -contract (bottom).

► **Proposition 5** (Second reduction). *Let D be the union of disjoint directed paths, and w weight function on $V(D)$. Suppose that there is one directed path $P = (v_p, v_{p-1}, \dots, v_2, v_1)$ of D and an index i , $2 \leq i \leq p-1$, such that is $w(v_p) < w(v_{p-1}) < \dots < w(v_{i+1}) < w(v_i)$ and $w(v_i) \geq w(v_{i-1})$. Then $\text{val}(D, w) = \text{val}(D^*, w^*)$ where (D^*, w^*) is the v_i -contract of (D, w) .*

Proof. We first prove that $\text{val}(D, w) \leq \text{val}(D^*, w^*)$.

To do so we construct simultaneously move by move two plays, one in (D, w) and one in (D^*, w^*) . In (D, w) , Alice plays an optimal strategy resulting in a play denoted by σ_{opt}^A and in (D^*, w^*) Bob plays an optimal strategy resulting in a play denoted by σ_{opt}^{*B} . In (D^*, w^*) , we construct an A -play σ^{*A} (not necessarily optimal) for Alice which follows (mimics) σ_{opt}^A and in (D, w) we construct a B -play σ^B (not necessarily optimal) for Bob which follows σ_{opt}^{*B} . In particular, if Alice removes, according to σ_{opt}^A , a vertex v not in $\{v_p, v_{p-1}, \dots, v_{i+1}, v_i, v_{i-1}\}$, we impose that Alice removes the same vertex v in the simultaneous move in (D^*, w^*) . Similarly, if Bob removes, according to σ_{opt}^{*B} , a vertex v not in $\{v_p, v_{p-1}, \dots, v_{i+2}, v_{i+1}, v_i^*\}$, we impose that Bob removes the same vertex v in the simultaneous move in (D, w) .

We consider the first move m when either Bob removes vertex v_i^* in (D^*, w^*) with B -play σ_{opt}^{*B} (Case 1) or Alice removes vertex v_{i-1} in (D, w) with A -play σ_{opt}^A (Case 2). Note that in the moves strictly before m , Alice has been imposed to remove the same vertices in (D^*, w^*) and (D, w) because σ^{*A} follows σ_{opt}^A , and Bob has been imposed to remove the same vertices in (D, w) and (D^*, w^*) because σ^B follows σ_{opt}^{*B} .

- Case 1: At move m (even), Bob removes vertex v_i^* with B -play σ_{opt}^{*B} in (D^*, w^*) and v_{i-1} has not been removed in (D, w) by Alice with strategy σ_{opt}^A . In that case, the vertices $v_1, \dots, v_{i-2}$ have been all removed in (D, w) and (D^*, w^*) and we impose Bob to remove vertex v_{i-1} within σ^B at move m .

Let μ be the first move after m when Alice removes a vertex in P with A -play σ_{opt}^A in (D, w) . In the odd moves strictly before μ (if any), Alice removes a vertex v not in P with A -play σ_{opt}^A in (D, w) , and we impose that Alice removes the same vertex v in the simultaneous move in (D^*, w^*) . In the even moves strictly before μ , if Bob removes a vertex v in D^* not in P with B -play σ_{opt}^{*B} , we impose that Bob removes the same vertex v in the simultaneous move in (D, w) . If Bob removes a vertex v_k ($i+2 \leq k \leq p$) in (D^*, w^*) with B -play σ_{opt}^{*B} , we impose that, in the simultaneous move in (D, w) , Bob removes the vertex v_{k-2} which is the first available. We note that move μ exists because, as with the above conditions, Bob will not remove v_{p-1} before Alice removes a vertex of P . We also note that μ is odd and $m+1 \leq \mu \leq n-2$.

Let v_j ($i \leq j \leq p-1$) be the vertex removed by Alice at move μ . Then we impose that Bob removes with B -play σ^B in move $\mu+1$ vertex v_{j+1} . Then, for the end of the plays, we impose that Alice removes with A -play σ^{*A} in (D^*, w^*) at move $\mu+2h$ ($h \geq 0$) the same vertex as the one removed by Alice in (D, w) at move $\mu+2+2h$ with A -play σ_{opt}^A . We impose that Bob removes in (D, w) at move $\mu+3+2h$ ($h \geq 0$) the same vertex as the one removed by Bob with B -play σ_{opt}^{*B} in (D^*, w^*) at move $\mu+1+2h$.

In summary in this Case 1, Alice has removed in (D, w) (according to $\sigma = \sigma_{opt}^A \oplus \sigma^B$) the same vertices as in (D^*, w^*) (according to $\sigma^* = \sigma^{*A} \oplus \sigma_{opt}^{*B}$) plus v_j . As $w(v_j) \leq w(v_i)$ for $j \geq i$, we have:

$$\text{col}^A(D, w, \sigma) = \text{col}^A(D^*, w^*, \sigma^*) + w(v_j) \leq \text{col}^A(D^*, w^*, \sigma^*) + w(v_i).$$

- Case 2: At move m (odd), Alice removes vertex v_{i-1} with A -play σ_{opt}^A in (D, w) and v_i^* has not been removed by Bob with B -play σ_{opt}^{*B} in (D^*, w^*) . Then we impose that Bob removes vertex v_i within σ^B at move $m + 1$.

We also impose that in all the moves after $m + 1$, if Alice removes a vertex $v \neq v_{i+1}$ with A -play σ_{opt}^A at a move $m + 2 + 2h$, then Alice removes the same vertex v in move $m + 2h$ in (D^*, w^*) with A -play σ^{*A} . We also impose that, if Bob removes vertex $v \neq v_i^*$ at move $m + 1 + 2h$ with B -play σ_{opt}^{*B} in (D^*, w^*) , then Bob removes the same vertex in (D, w) at move $m + 3 + h$ with B -play σ^B .

We now distinguish two sub-cases.

- Sub-case 2.1: Alice removes vertex v_{i+1} with σ_{opt}^A at some odd move $\mu > m$. Bob cannot have removed vertex v_i^* in (D^*, w^*) with σ_{opt}^{*B} in a move before $\mu - 3$ otherwise Bob would have removed v_{i+1} in a move before $\mu - 1$ and so v_{i+1} cannot be removed by Alice at move μ . Then we impose that Alice removes vertex v_i^* at move $\mu - 2$ in (D^*, w^*) with σ^{*A} . Therefore Bob cannot remove v_i^* .
- Sub-case 2.2: Bob removes vertex v_i^* with σ_{opt}^{*B} at some even move $\nu > m$. Alice cannot have removed v_{i+1} in (D, w) at a move before $\nu + 1$, otherwise Alice would have removed v_i^* in (D^*, w^*) before move $\nu - 1$. Then we impose that Bob removes vertex v_{i+1} at move $\nu + 2$ in (D, w) with σ^B and so Alice will never remove v_{i+1} .

In Sub-case 2.1, Alice has removed the same vertices in $D - \{v_{i-1}, v_i, v_{i+1}\}$ as in D^* . Furthermore Alice has removed v_{i-1} and v_{i+1} in D while Alice has removed v_i^* in D^* with weight $w^*(v_i^*) = w(v_{i-1}) + w(v_{i+1}) - w(v_i)$. Thus:

$$\text{col}^A(D, w, \sigma) \leq \text{col}^A(D^*, w^*, \sigma^*) + w(v_i).$$

In Sub-case 2.2, Alice has removed the same vertices in D as in D^* plus v_{i-1} . Thus:

$$\text{col}^A(D, w, \sigma) \leq \text{col}^A(D^*, w^*, \sigma^*) + w(v_{i-1}) \leq \text{col}^A(D^*, w^*, \sigma^*) + w(v_i).$$

Recall that $\text{val}(D, w) = \text{gain}^A(D, w) - \text{gain}^B(D, w)$ and $\sum_{v \in D} w(v) = \text{gain}^A(D, w) + \text{gain}^B(D, w)$. Thus $\text{val}(D, w) = 2 \text{gain}^A(D, w) - \sum_{v \in D} w(v)$. Similarly, $\text{val}(D^*, w^*) = 2 \text{gain}^A(D^*, w^*) - \sum_{v \in D^*} w^*(v)$. Moreover, by construction, $\sum_{v \in D} w(v) = \sum_{v \in D^*} w^*(v) - 2w(v_i)$.

σ_{opt}^A is a play obtained by playing an optimal strategy and σ^B is not necessarily optimal, thus $\text{gain}^A(D, w) \leq \text{col}^A(D, w, \sigma)$. As in both above subcases $\text{col}^A(D, w, \sigma) \leq \text{col}^A(D^*, w^*, \sigma^*) + w(v_i)$, we get:

$$\text{val}(D, w) \leq 2 \text{col}^A(D, w, \sigma) - \sum_{v \in D} w(v) \leq 2 \text{col}^A(D^*, w^*, \sigma^*) - \sum_{v \in D^*} w^*(v).$$

Now σ_{opt}^{*B} is a B -play obtained by playing an optimal strategy and σ^{*A} is not necessarily optimal. Thus $\text{col}^A(D^*, w^*, \sigma^*) \leq \text{gain}^A(D^*, w^*)$ and so

$$\text{val}(D, w) \leq 2 \text{gain}^A(D^*, w^*) - \sum_{v \in D^*} w^*(v) = \text{val}(D^*, w^*).$$

One can prove $\text{val}(D^*, w^*) \leq \text{val}(D, w)$ in a similar way. More precisely, Alice plays in (D^*, w^*) an optimal strategy resulting in a A -play σ_{opt}^{*A} and Bob plays in (D, w) an optimal strategy resulting in a σ_{opt}^B . We construct in (D, w) an A -play σ^A for Alice that follows σ_{opt}^{*A} , and in (D^*, w^*) a B -play σ^{*B} for Bob that follows σ_{opt}^B . The proof is the same as above by interchanging the roles of Alice and Bob. ◀

► **Corollary 6.** *Given a weighted digraph D of order n which is the union of q disjoint directed paths an optimal play can be computed in $O(n \log q)$ time.*

Proof. The algorithm proceeds as follows. It first reduces each component of D using the first and second reduction. Note that this can be done in linear time in n since the reduction applies (if any) around the smallest index i such that $w(v_i) \geq w(v_{i-1})$.

After such a reduction, we obtain a union of disjoint directed paths such that the weight function is strictly increasing on each directed path. In particular, at each turn the maximum-weight vertex will be a sink and thus removable. Therefore, by an easy induction with Lemma 2, there is an optimal play in which each player removes the maximum-weight vertex at each step. Hence, it suffices to sort the vertices in increasing order of their weight to get an optimal play. This corresponds to merging the q sorted lists into a single sorted list, which can be done in $O(n \log q)$ time using heaps. ◀

4 Strategies to collect some fixed set of vertices in grids

► **Definition 7.** *The directed grid $\vec{G}_{p \times q}$ is the cartesian product of two directed paths. (See Figure 5). More precisely, its vertex and arc set are*

$$V(\vec{G}_{p \times q}) = [p] \times [q], \text{ and}$$

$$A(\vec{G}_{p \times q}) = \{((i+1, j), (i, j)) \mid i \in [p-1], j \in [q]\} \cup \{((i, j+1), (i, j)) \mid i \in [p], j \in [q-1]\}.$$

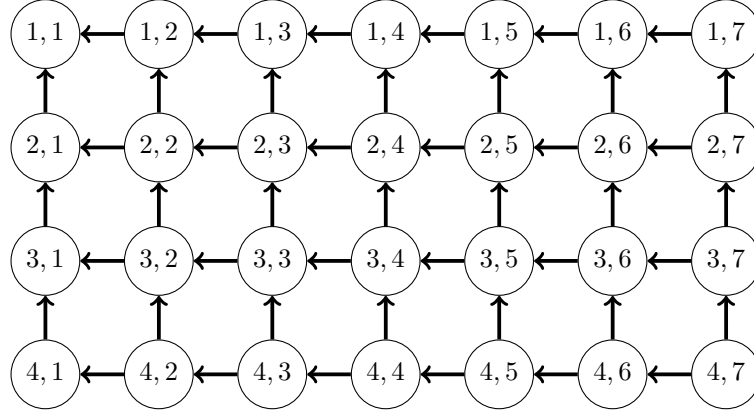
It has p rows $R_i = \{(i, j) \mid j \in [q]\}$, $1 \leq i \leq p$, and q columns, $C_j = \{(i, j) \mid i \in [p]\}$, $1 \leq j \leq q$. Hence the vertex in row R_i and column C_j is (i, j) . Observe that in $\vec{G}_{p \times q}$, vertex $(1, 1)$ is the unique sink, the vertices of $(R_1 \cup C_1) \setminus \{(1, 1)\}$ have out-degree 1, and all others vertices have out-degree 2.

Let D be an acyclic digraph and let $S \subseteq V(D)$ be a set of vertices. We say that a player X has an S -strategy if X has a strategy to remove all vertices of S . A trivial example is that Alice has an $\{s\}$ -strategy for every sink s of D , because Alice can remove s in her first move.

In this section, we describe some strategies for Bob and Alice to remove some of the following sets of vertices of the grid.

$$\begin{aligned} \mathbb{R}_{\text{odd}} &:= \text{set of vertices in odd rows} \\ \mathbb{R}_{\text{even}} &:= \text{set of vertices in even rows} \\ \mathbb{C}_{\text{odd}} &:= \text{set of vertices in odd columns} \\ \mathbb{C}_{\text{even}} &:= \text{set of vertices in even columns} \\ \mathbb{R}_{\text{odd}}^- &:= \mathbb{R}_{\text{odd}} \setminus R_1 \text{ (vertices in odd rows except the first one)} \\ \mathbb{C}_{\text{odd}}^- &:= \mathbb{C}_{\text{odd}} \setminus C_1 \text{ (vertices in odd columns except the first one)} \\ \mathbb{R}_{\text{odd}}^0 &:= \mathbb{R}_{\text{odd}}^- \cup (R_1 \cap \mathbb{C}_{\text{even}}) \\ \mathbb{R}_{\text{odd}}^1 &:= \mathbb{R}_{\text{odd}}^- \cup (R_1 \cap \mathbb{C}_{\text{odd}}) \\ \mathbb{C}_{\text{odd}}^0 &:= \mathbb{C}_{\text{odd}}^- \cup (C_1 \cap \mathbb{R}_{\text{even}}) \\ \mathbb{C}_{\text{odd}}^1 &:= \mathbb{C}_{\text{odd}}^- \cup (C_1 \cap \mathbb{R}_{\text{odd}}) \end{aligned}$$

We start with two very simple strategies for Bob, which are symmetric to each other.



■ Figure 5 Grid $\vec{G}_{4 \times 7}$.

► **Theorem 8.**

- (i) If q is even, then Bob has a $\mathbb{C}_{\text{even}}$ -strategy. In other words, Bob can remove all the vertices of the even columns.
- (ii) If p is even, then Bob has an $\mathbb{R}_{\text{even}}$ -strategy. In other words, Bob can remove all the vertices of the even rows.

Proof.

- (i) When q is even, Bob can apply the following strategy.

If Alice removes $(i, 2j - 1)$ for some $i \in [p]$ and $j \in [q/2]$, then Bob removes $(i, 2j)$.

Observe that Bob can apply this strategy all play long. Indeed when it is Alice's turn to play, only vertices in odd-columns are removable. Doing so Bob remove all vertices of $\mathbb{C}_{\text{even}}$.

- (ii) Symmetrically, when p is even, Bob can apply the following strategy and remove all vertices of $\mathbb{R}_{\text{even}}$.

If Alice removes $(2i - 1, j)$ for some $i \in [p/2]$ and $j \in [q]$, then Bob removes $(2i, j)$. ◀

When p and q have different parities, Bob can apply similar strategies, either for odd columns if p is even after some bootstrap in the first column, or for odd rows if q is even after some bootstrap in the first row. These two strategies are symmetric to each other.

► **Theorem 9.**

- (i) If p is even and q is odd, then Bob has a $\mathbb{C}_{\text{odd}}^0$ -strategy. In other words, Bob can remove the vertices of the first column in even rows and all the vertices of the other odd columns.
- (ii) If p is odd and q is even, then Bob has an $\mathbb{R}_{\text{odd}}^0$ -strategy. In other words, Bob can remove the vertices of the first row in even columns and all the vertices of the other odd rows.

Proof.

- (i) Assume p is even and q is odd. Bob can apply the following strategy.

If Alice removes a vertex $(2i - 1, 1)$ for some $i \in [p/2]$, then Bob removes $(2i, 1)$.

If Alice removes a vertex $(i, 2j)$ for some $j \in [(q - 1)/2]$, then Bob removes the vertex $(i, 2j + 1)$.

When it is Alice's turn to play, two cases may occur. If the first column has not been fully removed, then the removable vertices are one odd-row vertex in the first column and some vertices in even columns. If the first column has been removed and perhaps some pairs of columns (in an even number of moves as p is even), then only vertices in even columns are removable. In both cases, Alice can never remove a vertex in $\mathbb{C}_{\text{odd}}^0$.

- (ii) Assume p is odd and q is even. The proof of (ii) is symmetric to (i). An $\mathbb{R}_{\text{odd}}^0$ -strategy for Bob is the following.

If Alice removes a vertex $(1, 2j - 1)$ for some $j \in [q/2]$, then Bob removes $(1, 2j)$.

If Alice removes a vertex $(2i, j)$ for some $i \in [(p-1)/2]$, then Bob removes the vertex $(2i + 1, j)$. ◀

We now give some strategies for Alice when pq is odd.

► **Theorem 10.**

- (i) *If pq is odd, then Alice has a $\mathbb{C}_{\text{odd}}^1$ -strategy. In other words, Alice can remove the vertices of column C_1 in odd rows and all the vertices of the other odd columns.*
- (ii) *If pq is odd, then Alice has an $\mathbb{R}_{\text{odd}}^1$ -strategy. In other words, Alice can remove the vertices of row R_1 in odd columns and all the vertices of the other odd rows.*

Proof.

- (i) Assume p and q are both odd. Alice can apply the following strategy.

If Bob removes a vertex $(2i, 1)$ for some $i \in [(p-1)/2]$, then Alice removes $(2i + 1, 1)$.

If Bob removes a vertex $(i, 2j)$ for some $j \in [(q-1)/2]$, then Alice removes $(i, 2j + 1)$.

When it is Bob's turn to play, two cases may occur. If the first column has not been fully removed, then the removable vertices are one even-row vertex in the first column and some vertices in even columns (indeed as q is odd Alice could have applied the second rule). If the first column has been removed and perhaps some pairs of others consecutive even-odd columns (in an odd number of moves as p is odd), then only vertices in even columns are removable. In both cases, Bob can never remove a vertex in $\mathbb{C}_{\text{odd}}^1$.

- (ii) The proof is symmetric to (i). An $\mathbb{R}_{\text{odd}}^1$ -strategy for Alice is the following.

If Bob removes a vertex $(1, 2j)$ for some $j \in [(q-1)/2]$, then Alice removes $(1, 2j + 1)$.

If Bob removes a vertex $(2i, j)$ for some $i \in [(p-1)/2]$, then Alice removes $(2i + 1, j)$. ◀

We now describe more elaborate strategies which will enable Alice or Bob to remove vertices not removed by the simple strategies above. Let us specify some definitions and notations.

A configuration of the play on the $p \times q$ grid at time t is completely determined by the numbers r_i^t of vertices that have not been removed from rows R_i . Said otherwise, vertex (i, j) has been removed for $1 \leq j \leq q - r_i^t$ and vertex (i, j) has not been removed for $q + 1 - r_i^t \leq j \leq q$. Clearly, at any time t , we have $0 \leq r_i^t \leq r_{i+1}^t \leq q$. At the beginning of the game (time 0), we have $r_i^0 = q$ for all $i \in [p]$. At time t , we have $\sum_{i=1}^p r_i^t = pq - t$. This sum is even when either t is odd and pq is odd, or t is even and pq is even.

We note that a vertex is removable at move $t + 1$ if it is of the form $(i, q + 1 - r_i^t)$ with $r_i^t > r_{i-1}^t$. Furthermore if $r_i^t > r_{i-1}^t + 1$, then after the removal of $(i, q + 1 - r_i^t)$ at move $t + 1$, we have $r_i^{t+1} = r_i^t - 1 > r_{i-1}^t$, so the vertex to the right $(i, q + 2 - r_i^t)$ becomes removable.

- **Theorem 11.** *If pq is odd, then Alice has an $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$ -strategy. In other words, Alice can remove all the vertices in both even rows and even columns.*

Proof. A configuration on the $p \times q$ grid at time t odd is *A-good* if there exists a non-negative i_0^t , shortly denoted by i_0 , such that

1. $r_{2i-1}^t = r_{2i}^t$ for $i \leq i_0$,
2. $r_{2i_0+1}^t$ is even, and
3. r_i^t is odd for $2i_0 + 2 \leq i \leq p$.

Observe that if the configuration at time t odd is *A-good*, no vertex of $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$ is removable by Bob at move $t + 1$. Indeed, recall that a removable vertex is of the form $(i, q + 1 - r_i^t)$ with $r_i^t > r_{i-1}^t$. For $i \leq 2i_0 + 1$, there are only removable vertices in odd rows, and so not in $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$. For $i > 2i_0 + 1$, r_i^t is odd, so a removable vertex in row i is in an odd column and so not in $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$.

Hence, it suffices for Alice to leave an *A-good* configuration after each of her moves to collect all vertices of $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$. We shall now prove by induction on t , that Alice has a strategy to do so. In her first move, Alice necessarily removes vertex $(1, 1)$. So $r_1^1 = q - 1$ (which is even) and $r_i^1 = q$ (which is odd) for all $2 \leq i \leq p$. Thus, the configuration at time 1 is *A-good* with $i_0 = 0$. This proves the base case of the induction. The inductive case is the following claim.

▷ **Claim 12.** Suppose that the configuration at time t odd is *A-good*. Then, whatever vertex Bob removes at move $t + 1$, Alice can always remove a vertex so that the configuration at time $t + 2$ is also *A-good*.

Proof of the claim.

- Suppose Bob removes vertex $(i_1, q + 1 - r_{i_1}^t)$ in a row $i_1 < 2i_0 + 1$ (at move $t + 1$).
Hence i_1 is necessarily odd. Then Alice removes the vertex below $(i_1 + 1, q + 1 - r_{i_1+1}^t)$ which is now removable as $r_{i_1+1}^{t+1} = r_{i_1+1}^t = r_{i_1}^t = r_{i_1}^{t+1} + 1$. Consequently, $r_{i_1+1}^{t+2} = r_{i_1+1}^t - 1 = r_{i_1}^t - 1 = r_{i_1}^{t+2}$, and $r_i^{t+2} = r_i^t$ for all $i \notin \{i_1, i_1 + 1\}$. Hence, since the configuration at time t was *A-good*, so the configuration at time $t + 2$ is also *A-good* with $i_0^{t+2} = i_0$.
- Suppose Bob removes vertex $(2i_0 + 1, q + 1 - r_{2i_0+1}^t)$. Necessarily, $r_{2i_0+1}^t > r_{2i_0}^t$.
If $r_{2i_0+1}^t > r_{2i_0}^t + 1$, Alice removes the vertex on the right $(2i_0 + 1, q + 2 - r_{2i_0+1}^t)$. Consequently, $r_{2i_0+1}^{t+2} = r_{2i_0+1}^t - 2$ (which is even) and $r_i^{t+2} = r_i^t$ for all $i \notin \{2i_0 + 1\}$. Hence, since the configuration at time t was *A-good*, so is the configuration at time $t + 2$ with $i_0^{t+2} = i_0$.
If $r_{2i_0+1}^t = r_{2i_0}^t + 1$, we note that $r_{2i_0-1}^t = r_{2i_0}^t = r_{2i_0+1}^t - 1$ is odd. Let i_3 be the smallest value such that $r_{2i_3+1}^t = r_{2i_3+2}^t = r_{2i_0+1}^t - 1$. Then $(2i_3 + 1, q + 1 - r_{2i_3+1}^t)$ is removable and Alice removes it. Then $r_{2i_3+1}^{t+2} = r_{2i_3+1}^t - 1$ is even. Furthermore, for all $2i_3 + 1 < i \leq 2i_0 + 1$, r_i^{t+2} is odd. So the configuration at time $t + 2$ is *A-good* with $i_0^{t+2} = i_3$.
- Suppose Bob removes vertex $(2i_0 + 2, q + 1 - r_{2i_0+2}^t)$. That implies that $r_{2i_0+2}^t > r_{2i_0+1}^t$.
If $r_{2i_0+2}^t > r_{2i_0+1}^t + 1$, then Alice removes the vertex on the right $(2i_0 + 2, q + 2 - r_{2i_0+2}^t)$. This results in an *A-good* configuration with $i_0^{t+2} = i_0$ because $r_{2i_0+2}^{t+2} = r_{2i_0+2}^t - 2$ is odd, and $r_i^{t+2} = r_i^t$ for all $i \neq 2i_0 + 2$.
If $r_{2i_0+2}^t = r_{2i_0+1}^t + 1$, then Alice removes the vertex below $(2i_0 + 3, q + 1 - r_{2i_0+3}^t)$. This results in an *A-good* configuration with $i_0^{t+2} = i_0 + 1$ because $r_{2i_0+3}^{t+2} = r_{2i_0+3}^t - 1$ is even, $r_{2i_0+1}^{t+2} = r_{2i_0+2}^{t+2} = r_{2i_0+1}^t$, and $r_i^{t+2} = r_i^t$ for all $i \notin \{2i_0 + 2, 2i_0 + 3\}$.
- Suppose Bob removes a vertex $(i, q + 1 - r_i^t)$ with $i > 2i_0 + 2$. Note that $r_i^t > r_{i-1}^t + 1$ (as both values are odd) and so Alice removes vertex $(i, q + 2 - r_i^t)$. This results in an *A-good* configuration with $i_0^{t+2} = i_0$ because $r_i^{t+2} = r_i^t - 2$ is odd, and $r_i^{t+2} = r_i^t$ for all $i \neq 2i_0 + 2$. ◁

The claim is proved and therefore also the theorem. ◀

When pq is even using similar ideas, we will show that Bob has also a strategy to remove vertices not removed by Theorems 8 and 9.

► **Theorem 13.**

- (a) *If p is even and q is odd, then Bob has an $\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$ -strategy. In other words, Bob can remove all the vertices that are in both odd rows and even columns.*
- (b) *If p is odd and q is even, then Bob has an $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{odd}}$ -strategy. In other words, Bob can remove all the vertices that are in both even rows and odd columns.*
- (c) *If p and q are even, then Bob has an $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (\mathbb{R}_{\text{even}} \cap C_1)$ -strategy. In other words, Bob can remove all the vertices that are either in both odd rows and odd columns except the first one or in the first column and even rows.*
- (d) *If p and q are even, Bob has an $(\mathbb{R}_{\text{odd}}^- \cap \mathbb{C}_{\text{odd}}) \cup (R_1 \cap \mathbb{C}_{\text{even}})$ -strategy. In other words, Bob can remove all the vertices that are either in both odd columns and odd rows except the first one or in the first row and even columns.*

Proof. By symmetry, it is sufficient to prove the proposition for p even, that is for cases (a) and (c). We want to show that, when q is odd (resp. even), Bob has an $\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$ -strategy (resp. $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (R_{\text{even}} \cap C_1)$ -strategy).

A configuration on the $p \times q$ grid (with p even) at time t even is **B -good** if there exists i_0^t , shortly denoted by i_0 , and i_1^t , shortly denoted by i_1 , such that $i_1 \geq i_0 \geq 0$, $i_0 \leq p/2$ and the following hold:

1. if $i_0 > 0$, then $r_1^t = 0$;
2. if $i_0 > 1$, then $r_{2i}^t = r_{2i+1}^t$ for $1 \leq i < i_0$;
3. if $i_0 > 0$, then $r_{2i_0}^t$ is even;
4. r_i^t is odd for $2i_0 + 1 \leq i \leq 2i_1$.
5. $r_i^t = q$ for $2i_1 < i \leq p$.

Observe that if the configuration at time t even is B -good and if q is odd (resp even), no vertex of $\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$ (resp. $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (R_{\text{even}} \cap C_1)$) is removable by Alice at move $t + 1$. Indeed, if there exists a removable vertex in row $i \leq 2i_0$, then it is in an even row and not in the first column, and so not in $\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$ neither in $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (R_{\text{even}} \cap C_1)$. If q is odd and the removable vertex is in row $i > 2i_0$, it is in an odd column, and so not in $\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$. If q is even and $2i_0 + 1 \leq i \leq 2i_1$, the removable vertex is in an even column, and so not in $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (R_{\text{even}} \cap C_1)$. If q is even $2i_1 < i \leq p$, the removable vertex is $(2i_1 + 1, 1)$, and so it is not in $(R_{\text{even}} \cap C_1)$.

Hence, it suffices for Bob to leave a B -good configuration after each of his moves to collect all vertices of $\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$ (resp. $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (R_{\text{even}} \cap C_1)$) if q is odd (resp. even). We shall now prove by induction on t , that Bob has a strategy to do so. At time 0, $r_i^0 = q$ for all $i \in [p]$, so the configuration is B -good with $i_0 = i_1 = 0$. This proves the base case of the induction. The inductive case is the following claim.

▷ **Claim 14.** Suppose that the configuration at time t even is B -good. Then whatever vertex Alice removes at move $t + 1$, Bob can always remove a vertex so that the configuration at time $t + 2$ is B -good.

Proof of the claim. The proof is very similar to the proof of Claim 12. We just indicate how Bob replies to the possible moves of Alice.

- Suppose Alice removes a vertex in a row $< 2i_0$. It must be in an even row due to condition 2, so the removed vertex is $(2i, q + 1 - r_{2i}^t)$ for some $i < i_0$. ($r_{2i}^t > 0$). Then Bob removes the vertex below $(2i + 1, q + 1 - r_{2i+1}^t)$ which is removable as $r_{2i+1}^t = r_{2i}^t$.

- Suppose Alice removes a vertex in row $2i_0$ namely $(2i_0, q+1-r_{2i_0}^t)$ (with $r_{2i_0}^t \geq 2$). This implies that $r_{2i_0}^t > r_{2i_0-1}^t$.
 If $r_{2i_0}^t > r_{2i_0-1}^t + 1$, then Bob removes the vertex on the right $(2i_0, q+2-r_{2i_0}^t)$.
 If $r_{2i_0}^t = r_{2i_0-1}^t + 1$, we note that $r_{2i_0-2}^t = r_{2i_0-1}^t = r_{2i_0}^t - 1 = r_{2i_0}^{t+1}$ is odd. Let i_3 be the smallest index such that $r_{2i_3}^t = r_{2i_3+1}^t = r_{2i_0}^t - 1$. Then $(2i_3, q+1-r_{2i_3}^t)$ is removable and Bob removes it. (B -good configuration with $i_0^{t+2} = i_3$).
 - If $i_1 > i_0$ suppose Alice removes vertex $(i, q+1-r_i^t)$ with $2i_0+1 < i \leq 2i_1$.
 If $r_i^t > r_{i-1}^t + 1$ then Bob removes the vertex on the right $(i, q+2-r_i^t+1)$.
 If $r_i^t = r_{i-1}^t + 1$ which happens only if $i = 2i_0+1$, then Bob removes the vertex $(2i_0+2, q+1-r_{2i_0+2}^t)$ (note that for parity reason, $2i_0+2 \leq 2i_1$ and so $r_{2i_0+2}^t$ is odd) (B -good configuration with $i_0^{t+2} = i_0+1$).
 - Suppose Alice removes the vertex $(2i_1+1, 1)$.
 Suppose q is odd. If $q > r_{2i_1}^t + 1$, then Bob removes the vertex on the right $(2i_1+1, 2)$.
 If $q = r_{2i_1}^t + 1$ which implies $i_1 = i_0$ Bob removes the vertex $(2i_1+2, 1)$. Suppose q is even. Then Bob removes the vertex $(2i_1+2, 1)$. $\triangleleft$
- The claim is proved and therefore also the theorem. $\blacktriangleleft$

5 Weight on a unique vertex

In this section, we study the case where all the weight is located at a unique vertex, which will be denoted by o . In other words $w(o) > 0$ and $w(v) = 0$ for all $v \neq o$. In this case, the objective of both players is to remove the vertex o . Hence, we say that a player **wins** if he removes o .

The strategies defined in the preceding section enable us to easily determine the winner in the case where o is neither in the first row nor in the first column.

► **Theorem 15.** *If $o = (i, j)$ with $i > 1$ and $j > 1$, then Alice wins if pq is odd and Bob wins if pq is even.*

Proof. Assume pq is odd. If i (resp. j) is odd, then o is in $\mathbb{R}_{\text{odd}}^-$ (resp. $\mathbb{C}_{\text{odd}}^-$), so Alice can remove o by Theorem 10. If i and j are even, then $o \in \mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$, so Alice can remove o by Theorem 11.

Assume pq is even. By symmetry, we can suppose that p is even. If i is even, then $o \in \mathbb{R}_{\text{even}}$, so Bob can remove o by Theorem 8. Henceforth, we may assume i is odd. We distinguish two cases depending on the parity of q .

Suppose q is even. If j is even, then $o \in \mathbb{C}_{\text{even}}$, so Bob can remove o by Theorem 8. If j is odd, then $o \in \mathbb{R}_{\text{odd}}^- \cap \mathbb{C}_{\text{odd}}^-$, so Bob can remove o by Theorem 13.

Suppose q is odd. If j is odd, then $o \in \mathbb{C}_{\text{odd}}^-$, so Bob can remove o by Theorem 9. If j is even, then $o \in \mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{even}}$, so Bob can remove o by Theorem 13. $\blacktriangleleft$

► **Theorem 16.** *If $o = (1, 1)$, then Alice wins.*

If $o = (1, j)$, with $j > 1$, then Alice wins if and only if p is odd and j is odd.

If $o = (i, 1)$, $i > 1$, then Alice wins if and only if q is odd and i is odd.

Proof. If $o = (1, 1)$, then Alice takes it at the first move and so wins.

Suppose that $o = (1, j)$ with $j > 1$. If a player removes vertex $(1, j-1)$, then the other will grab o . So the aim of each player is to avoid removing $(1, j-1)$. A player can remove safely any “safe” vertex (i, j') with $1 \leq j' \leq j-2$. The number of such safe vertices is $s = (j-2)p$. If s is odd, which happens when p is odd and j odd, then Bob is obliged to remove $(1, j-1)$ and Alice wins. Otherwise if s is even, then Alice is obliged to remove $(1, j-1)$ and Bob wins.

The proof is similar when the olive $o = (i, 1)$ with $i > 1$. Here the number of safe vertices is $q(i - 2)$. So Alice wins when this number is odd, that is when q is odd and i is odd, and Bob wins otherwise. ◀

6 Minimum number of olives that can be collected by a player

In this section, we consider the case where the weight function w is a $\{0, 1\}$ -valued function. In such a case, w is completely determined by the set $O = \{v \mid w(v) = 1\}$. This set is then called the **olive set** and its elements the **olives**. The cardinality of O is often denoted by k . Let D be an acyclic graph on n vertices and let $O \subseteq V(D)$ be an olive set of cardinality k . Then for $X \in \{A, B\}$, we denote by $\text{gain}^X(D, O)$ the number of olives that X collects in the directed grabbing game, if both players play optimally. Note that $\text{gain}^A(D, O) + \text{gain}^B(D, O) = k$.

We denote by $\text{sec}^X(D, k)$ the number of olives that X ($X \in \{A, B\}$) is sure to collect. In other words, $\text{sec}^X(D, k)$ is the minimum number of olives that X collects over all possible games with k olives:

$$\text{sec}^X(D, k) = \min\{\text{gain}^X(D, O) \mid O \subseteq V(D) \text{ and } |O| = k\}.$$

By definition, $\text{sec}^X(D, k) \geq 0$. Moreover, when the set O contains the first k vertices of an acyclic ordering of D , then by Lemma 2, there is an optimal play in which the olives are taken on the first k moves. Thus $\text{gain}^A(D, O) = \lceil k/2 \rceil$ and $\text{gain}^B(D, O) = \lfloor k/2 \rfloor$, and so $\text{sec}^A(D, k) \leq \lceil k/2 \rceil$ and $\text{sec}^B(D, k) \leq \lfloor k/2 \rfloor$. Finally, Alice collects at most $\lceil |V(D)|/2 \rceil$ vertices and Bob collects at most $\lfloor |V(D)|/2 \rfloor$ vertices. So $\text{sec}^B(D, k) \geq k - \lceil |V(D)|/2 \rceil$ and $\text{sec}^A(D, k) \geq k - \lfloor |V(D)|/2 \rfloor$. Thus

$$\max\{0, k - \lfloor |V(D)|/2 \rfloor\} \leq \text{sec}^A(D, k) \leq \lceil k/2 \rceil, \quad (3)$$

$$\max\{0, k - \lceil |V(D)|/2 \rceil\} \leq \text{sec}^B(D, k) \leq \lfloor k/2 \rfloor. \quad (4)$$

The existence of S -strategies yields some bounds on $\text{sec}^A(D, k)$ and $\text{sec}^B(D, k)$.

► **Lemma 17.** *Let D be an acyclic digraph and let $S \subseteq V(D)$. If $\bar{X}$ has an S -strategy, then $\text{sec}^X(D, k) = 0$ for all $k \leq |S|$ and $\text{sec}^X(D, k) \leq k - |S|$ for all $k \geq |S|$.*

Proof. Assume first $k \leq |S|$. Take a subset O of S of size k . Applying his S -strategy, $\bar{X}$ can remove all the vertices of S and in particular all the olives. Hence $\text{gain}^X(D, O) = 0$, and thus $\text{sec}^X(D, k) = 0$.

Assume now $k \geq |S|$. Take a set O of size k which contains S . Applying his S -strategy, $\bar{X}$ can remove all the vertices of S and thus collects at least $|S|$ olives. Hence $\text{gain}^X(D, O) \leq k - |S|$, and thus $\text{sec}^X(D, k) \leq k - |S|$. ◀

► **Lemma 18.** *Let D be an acyclic digraph, s be a positive integer and $S_r \subseteq V(D)$ for all $r \in [s]$. If X has an S_r -strategy for all $r \in [s]$, then, for every $k \leq |V(D)|$, we have*

$$\text{sec}^X(D, k) \geq \frac{1}{s} \left(k - |V(D)| + \left| \bigcup_{r \in [s]} S_r \right| \right).$$

Proof. Set $S = \bigcup_{r \in [s]} S_r$. Let O be a set of k olives on D . Set $O_S = O \cap S$. Observe that $|O_S| \geq |O| - (|V(D)| \setminus |S|) \geq k - |V(D)| + |S|$. Now, if X has an S_r -strategy for all $r \in [s]$, then applying the most profitable of those strategies, X can collect at least $|O_S|/s$ olives. Hence $\text{gain}^X(D, O) \geq |O_S|/s \geq \frac{1}{s}(k - |V(D)| + |S|)$. Since this is true for any set O of k olives, we have the result. ◀

► **Proposition 19.**

- (i) If pq is even, then $\sec^A(\vec{G}_{p \times q}, k) = \max\{0, k - pq/2\}$.
- (ii) If pq is odd, then $\sec^B(\vec{G}_{p \times q}, k) = \max\{0, k - (pq + 1)/2\}$.

Proof.

- (i) If pq is even, then either p is even or q is even. By symmetry, we may assume q is even. By Theorem 8, Bob has a $\mathbb{C}_{\text{even}}$ -strategy. Thus, as $|\mathbb{C}_{\text{even}}| = pq/2$, Lemma 17 yields $\sec^A(\vec{G}_{p \times q}, k) \leq \max\{0, k - pq/2\}$. Moreover, by Equation (3), we have $\sec^A(\vec{G}_{p \times q}, k) \geq \max\{0, k - pq/2\}$. Thus $\sec^A(\vec{G}_{p \times q}, k) = \max\{0, k - pq/2\}$.
- (ii) By Theorem 10, Alice has a $\mathbb{C}_{\text{odd}}^1$ -strategy. Thus, as $|\mathbb{C}_{\text{odd}}^1| = (pq+1)/2$, Lemma 17 yields $\sec^B(\vec{G}_{p \times q}, k) \leq \max\{0, k - (pq+1)/2\}$. Now, by Equation (4), we have $\sec^B(\vec{G}_{p \times q}, k) \geq \max\{0, k - (pq + 1)/2\}$. Thus $\sec^B(\vec{G}_{p \times q}, k) = \max\{0, k - (pq + 1)/2\}$. ◀

This proposition shows that when pq is even (resp. odd), then Alice (resp. Bob) is not guaranteed to get any olive unless the number of olives is bigger than half the vertices, so that Bob (resp. Alice) cannot collect all of them. Hence, the main challenge is to determine $\sec^A(\vec{G}_{p \times q}, k)$ when pq is odd and $\sec^B(\vec{G}_{p \times q}, k)$ when pq is even.

When k is small compared to p and q , then those parameters are equal to 0 because the opponent has strategies to remove some small set of vertices on which all the olives can be.

Let $S_1 = \{(1, 2j) \mid 1 \leq j \leq (q-1)/2\} \cup \{(2i, 1) \mid 2 \leq i \leq (p-1)/2\}$. It is the set of vertices that are either in both the first row and an even column, or both the first column and an even row distinct from R_2 . Note that $|S_1| = (p+q-4)/2$.

► **Proposition 20.**

- (i) If pq is odd, then Bob has an S_1 -strategy.
- (ii) If p is even and q is odd, then Alice has a $C_1 \cap \mathbb{R}_{\text{odd}}$ -strategy.
- (iii) If p is odd and q is even, then Alice has an $R_1 \cap \mathbb{C}_{\text{odd}}$ -strategy.

Proof.

- (i) At the first move, Alice removes $(1, 1)$ and at the second move Bob grabs the olive on $(1, 2)$.

We use the terminology defined before Theorem 11. A configuration on the $p \times q$ grid at an even time t is **B -good** if the following conditions hold:

- a. $r_1^t = 0$ or r_1^t is odd ;
- b. either (a) $r_2^t = q$ or (b) there exists an i_0 ($1 \leq i_0 \leq (p-1)/2$) such that $r_i^t < q$ for $i \leq 2i_0$ and $r_i^t = q$ for $i \geq 2i_0 + 1$ or (c) $r_i^t < q$ for all i .

Observe that, if the configuration at an even time t is B -good, then no vertex of S_1 is removable by Alice at move $t+1$. Indeed, if there exists a removable vertex in R_1 , it is in an odd column and so not in S_1 . If the removable vertex is in C_1 , then it is either in row R_2 or in an odd row, and so not in S_1 . Hence, it suffices for Bob to leave a B -good configuration after each of his moves to collect all vertices of S_1 . We shall now prove by induction on t , that Bob has a strategy to do so. At time 2, $r_1 = q-2$ is odd and $r_2 = q$. This proves the base case of the induction. Suppose that the configuration at an even time t is B -good. Then we claim that whatever vertex Alice removes at move $t+1$, Bob can always remove a vertex so that the configuration at time $t+2$ is B -good. Assume Alice removes a vertex in R_1 . This vertex is either of the form $(1, 2j_0 + 1)$ ($1 \leq j_0 < (q-1)/2$) in which case Bob removes $(1, 2j_0 + 2)$, or $(1, q)$ in which case Bob removes the removable vertex in R_2 . In both cases, we are still in a B -good configuration.

Assume Alice removes a vertex v in C_1 . If we are in case (a), then $v = (2, 1)$, and Bob removes vertex $(2, 2)$. If we are in case (b) then, either $v = (2i_0 + 1, 1)$ ($2 \leq i_0 < (p-1)/2$) and Bob removes vertex $(2i_0 + 2, 1)$, or $v = (p, 1)$ and Bob removes the removable vertex in C_2 . In all cases, we are in a B -good configuration.

Assume Alice removes a vertex (i, j) with both $i \neq 1$ and $j \neq 1$. If both $(1, q)$ and $(p, 1)$ have been removed, then Bob removes any removable vertex. So w.l.o.g. suppose $(1, q)$ has not been removed and let $(1, 2j_0)$ be the last vertex removed in R_1 . The set of removable vertices at time t is included in an even set as $2j_0$ is even. So, as the number of vertices removed at time t is even and Alice was able to remove a vertex, there exists at time $t + 1$ at least one removable vertex (i', j') with $i' \neq 1$ and $j' \neq 1$. Then Bob removes this vertex and leaves a B -good configuration.

- (ii) Suppose p is even and q is odd. Alice grabs $(1, 1)$ at the first move. Then, as q is odd, Bob will be obliged to play first in $(2, 1)$ (it suffices to Alice to play in row 1 till Bob plays in $(2, 1)$). Then Alice grabs the olive $(3, 1)$. And so on, due to parity, Alice never plays in $(2i, 1)$ till Bob is forced to play there and Alice grabs $(2i + 1, 1)$. In summary, Alice grabs all the olives of $C_1 \cap \mathbb{R}_{\text{odd}}$.
- (iii) is symmetric to (ii). ◀

► **Corollary 21.**

- (i) If pq is odd, then $\text{sec}^A(\vec{G}_{p \times q}, k) = 0$ for all $k \leq (p + q - 4)/2$.
- (ii) If p is odd and q is even, then $\text{sec}^B(\vec{G}_{p \times q}, k) = 0$ for all $k \leq q/2$.
- (iii) If p is even and q is odd, then $\text{sec}^B(\vec{G}_{p \times q}, k) = 0$ for all $k \leq p/2$.
- (iv) If p is even and q is even, then $\text{sec}^B(\vec{G}_{p \times q}, 1) = 0$ and $\text{sec}^B(\vec{G}_{p \times q}, k) > 0$ for all $k \geq 2$.

Proof. Together with Lemma 17, Proposition 20 immediately yields the items (i), (ii), and (iii).

(iv) follows from the fact that, when p and q are both even, Bob can grab any olive except $(1, 1)$ by Theorems 15 and 16. ◀

The strategies proved in Section 4 and Lemma 18 yield the following bounds on $\text{sec}^A(\vec{G}_{p \times q}, k)$ when pq is odd and $\text{sec}^B(\vec{G}_{p \times q}, k)$ when pq is even.

► **Theorem 22.**

- (i) If pq is odd, then $\text{sec}^A(\vec{G}_{p \times q}, k) \geq \max \left\{ 0, \frac{k}{3} - \frac{p+q-2}{6}, \frac{k}{2} - \frac{pq+p+q-3}{8}, k - \frac{pq}{2} \right\}$.
- (ii) If p is odd and q is even, then $\text{sec}^B(\vec{G}_{p \times q}, k) \geq \max \left\{ 0, \frac{k}{3} - \frac{q}{6}, \frac{k}{2} - \frac{pq+q}{8}, k - \frac{pq}{2} \right\}$.
- (iii) If p is even and q is odd, then $\text{sec}^B(\vec{G}_{p \times q}, k) \geq \max \left\{ 0, \frac{k}{3} - \frac{p}{6}, \frac{k}{2} - \frac{pq+p}{8}, k - \frac{pq}{2} \right\}$.
- (iv) If p is even and q is even, then $\text{sec}^B(\vec{G}_{p \times q}, k) \geq \max \left\{ 0, \frac{k-1}{4}, \frac{k}{3} - \min \left\{ \frac{p}{6}, \frac{q}{6} \right\}, \frac{k}{2} - \frac{pq}{8}, k - \frac{pq}{2} \right\}$.

Proof. By Equations (3) (resp. (4)), we have $\text{sec}^A(\vec{G}_{p \times q}, k)$ (resp. $\text{sec}^B(\vec{G}_{p \times q}, k)$) is at least $\max\{0, k - pq/2\}$ when pq is odd (resp. even). Let us prove the other lower bounds.

- (i) Assume pq is odd. Then Alice has a $\mathbb{C}_{\text{odd}}^1$ -strategy, an $\mathbb{R}_{\text{odd}}^1$ -strategy (Theorem 10), and an $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$ -strategy (Theorem 11). Thus, Lemma 18 yields $\text{sec}^A(\vec{G}_{p \times q}, k) \geq k/2 - (pq + p + q - 3)/4$ when applied to $\mathbb{C}_{\text{odd}}^1$ and $\mathbb{R}_{\text{odd}}^1$ and $\text{sec}^A(\vec{G}_{p \times q}, k) \geq k/3 - (p + q - 2)/6$, when applied to $\mathbb{C}_{\text{odd}}^1$, $\mathbb{R}_{\text{odd}}^1$ and $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$.
- (ii) Assume p is odd and q is even. Then Bob has a $\mathbb{C}_{\text{even}}$ -strategy, (Theorem 8), an $\mathbb{R}_{\text{odd}}^0$ -strategy (Theorem 9), and an $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{odd}}$ -strategy (Theorem 13). Thus, Lemma 18 yields $\text{sec}^B(\vec{G}_{p \times q}, k) \geq k/2 - (p + 1)q/8$ when applied to $\mathbb{C}_{\text{even}}$ and $\mathbb{R}_{\text{odd}}^0$ and $\text{sec}^B(\vec{G}_{p \times q}, k) \geq k/3 - (p + q - 2)/6$, when applied to $\mathbb{C}_{\text{odd}}^1$, $\mathbb{R}_{\text{odd}}^1$ and $\mathbb{R}_{\text{even}} \cap \mathbb{C}_{\text{even}}$.

(iii) is symmetric to (ii).

(iv) Assume p and q are even. Then Bob has a $\mathbb{C}_{\text{even}}$ -strategy, an $\mathbb{R}_{\text{even}}$ -strategy by Theorem 8, and an $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (\mathbb{R}_{\text{even}} \cap C_1)$ -strategy and an $(\mathbb{R}_{\text{odd}}^- \cap \mathbb{C}_{\text{odd}}) \cup (R_1 \cap \mathbb{C}_{\text{even}})$ -strategy by Theorem 13(c). Hence, Lemma 18 applied firstly to the four sets $\mathbb{C}_{\text{even}}$, $\mathbb{R}_{\text{even}}$, $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (\mathbb{R}_{\text{even}} \cap C_1)$, $(\mathbb{R}_{\text{odd}}^- \cap \mathbb{C}_{\text{odd}}) \cup (R_1 \cap \mathbb{C}_{\text{even}})$ yields $\sec^B(\vec{G}_{p \times q}, k) \geq (k-1)/4$. Moreover, Lemma 18 applied firstly to the three sets $\mathbb{C}_{\text{even}}$, $\mathbb{R}_{\text{even}}$, $(\mathbb{R}_{\text{odd}} \cap \mathbb{C}_{\text{odd}}^-) \cup (\mathbb{R}_{\text{even}} \cap C_1)$, and secondly to the three sets $\mathbb{C}_{\text{even}}$, $\mathbb{R}_{\text{even}}$, $(\mathbb{R}_{\text{odd}}^- \cap \mathbb{C}_{\text{odd}}) \cup (R_1 \cap \mathbb{C}_{\text{even}})$, yields $\sec^B(\vec{G}_{p \times q}, k) \geq \max\{0, \frac{k}{3} - \min\{\frac{p}{6}, \frac{q}{6}\}, k - \frac{pq}{2}\}$. Finally, Lemma 18 applied to $\mathbb{C}_{\text{even}}$ and $\mathbb{R}_{\text{even}}$ yields $\sec^B(\vec{G}_{p \times q}, k) \geq k/2 - pq/8$. $\blacktriangleleft$

When $k > (p+q-4)/2$, the lower bound of Theorem 22 (i) is positive and so $\sec^A(\vec{G}_{p \times q}, k) > 0$. This shows that the bound $(p+q-4)/2$ in Corollary 21 (i) is tight.

When p is odd, q even and $k > q/2$, the lower bound is positive and so $\sec^B(\vec{G}_{p \times q}, k) > 0$. This shows that the bound $q/2$ in Corollary 21 (ii) is tight. Similarly, the bound $p/2$ in Corollary 21 (iii) is tight.

7 Further research

7.1 Digraph covered by q directed paths

Let D be an acyclic digraph covered by q directed paths. At each step, at most q vertices are sinks (at most one per path). Therefore, $O(n^q)$ subdigraphs may appear during the Directed Grabbing game, and value of the game can be computed by dynamic programming in $O(n^q)$ using the formula:

$$\text{val}(D, w) = \max\{w(s) - \text{val}(D - s, w) \mid s \text{ sink of } D\}.$$

Hence a natural question is whether faster and ideally linear-time algorithms exist?

► **Problem 23.** *Does there exist an algorithm that given a digraph covered by q directed paths computes an optimal play in $O(qn)$ time, or at least $o(n^q)$ time?*

The answer is affirmative when $q = 2$. One can easily construct a linear-time algorithm that computes an optimal play for the Directed Grabbing game on a given weighted acyclic digraph which is covered by two directed paths. The question is open for larger values of q .

7.2 Improving bounds on $\sec^A$ and $\sec^B$

Recall that **Content**, which we denote by C , is Alice when pq is odd and Bob otherwise. In Theorem 22, we gave lower bounds on $\sec^C(\vec{G}_{p \times q}, k)$. Those bounds are piecewise linear with successive slopes 0, $1/3$, $1/2$, and 1. It would be nice to investigate whether they are optimal or not. In particular, establishing upper bounds on $\sec^C(\vec{G}_{p \times q}, k)$ that are significantly lower than the trivial $\lceil k/2 \rceil$ or $\lfloor k/2 \rfloor$ would be nice.

Studying carefully the $(2 \times q)$ -grid, we are able to prove $\sec^B(\vec{G}_{2 \times q}, k) = \lceil \frac{k-1}{3} \rceil$ for all $k \leq 3q/2$. We can then derive that, when p is even, $\sec^B(\vec{G}_{p \times q}, k) \leq \lceil \frac{k-1}{3} \rceil$ for all $k \leq 3q/2$. This needs to be completed.

7.3 Average gain

In this paper, we only considered the minimum gains of both players ($\sec^A$ and $\sec^B$) but other parameters could also be studied. One natural candidate is the average gain, denoted by $\overline{\text{gain}}^X(D, k)$, and defined as the expected number of olives that a player X collects in the directed grabbing game when averaging over all possible sets of k olives in D .

$$\overline{\text{gain}}^X(D, k) = \frac{1}{\binom{n}{k}} \sum_{O \subseteq V(D), |O|=k} \text{gain}^X(D, O)$$

By definition $\text{sec}^X(D, k) \leq \overline{\text{gain}}^X(D, k)$ and $\overline{\text{gain}}^A(D, k) + \overline{\text{gain}}^B(D, k) = k$.

Assume $pq \gg k$. Using the strategies of Section 4, Content can remove two fixed sets of at least $pq/2$ vertices of $\vec{G}_{p \times q}$, which in average contain $k/2$ olives each. Using probabilistic arguments, we are able to show that $\overline{\text{gain}}^C(\vec{G}_{p \times q}, k) \geq k/2 + \Omega(\sqrt{k})$. Thus Content collects a bit more than half of the olives in average (when the number of olives is small compared to the size of the grid). A natural question is whether Content collects in average a fraction strictly larger than one half of the olives.

► **Problem 24.** *Does there exists a positive constant $\epsilon > 0$ such that $\overline{\text{gain}}^C(\vec{G}_{p \times q}, k) \geq k/2 + \epsilon k$ when p and q are sufficiently large?*

It might even be possible that $\overline{\text{gain}}^C(\vec{G}_{p \times q}, k) = k - o(1)$ as it is the case when $k = 1$ by Theorem 15. (Here $o(1)$ must be understood as a function tending to 0 when p and q tend to infinity.)

References

- 1 Josef Cibulka, Jan Kyncl, Viola Mészáros, Rudolf Stolar, and Pavel Valtr. Solution of Peter Winkler's pizza problem. In Jirí Fiala, Jan Kratochvíl, and Mirka Miller, editors, *International Workshop on Combinatorial Algorithms - IWOCA*, volume 5874 of *Lecture Notes in Computer Science*, pages 356–367. Springer, 2009. doi:10.1007/978-3-642-10217-2_35.
- 2 Hungry Coatis. Grabbing Olives on Linear Pizzas and Pissaladières. In *The Eleventh International Conference on Fun with Algorithms (FUN 2022)*, Island of Favignana, Italy, May 2022. doi:10.4230/LIPIcs.FUN.2022.12.
- 3 Tomasz Idziaszek. An optimal algorithm for calculating the profit in the coins in a row game. <https://www.mimuw.edu.pl/~idziaszek/termity/termity.pdf>. (An overview of his result was published in "Looking for a Challenge" book at the university of Warsaw)., 2012.
- 4 Kolja Knauer, Piotr Micek, and Torsten Ueckerdt. How to eat $4/9$ of a pizza. *Discrete Mathematics*, 311(16):1635–1645, 2011. doi:10.1016/j.disc.2011.03.015.
- 5 Peter Winkler. *Mathematical Puzzles: a connoisseur's collection*. CRC Press, 2004.
- 6 Peter Winkler. *Mathematical Puzzles*. A K Peters CRC Press, 2020.