Stable Divisorial Gonality is in NP

Hans L. Bodlaender[1,2], Marieke van der Wegen[1(✉)], and Tom C. van der Zanden[1]

[1] Department of Information and Computing Sciences, Universiteit Utrecht, Princetonplein 5, 3584 CC Utrecht, The Netherlands
M.vanderWegen@uu.nl

[2] Department of Mathematics and Computer Science, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands

Abstract. Divisorial gonality and stable divisorial gonality are graph parameters, which have an origin in algebraic geometry. Divisorial gonality of a connected graph G can be defined with help of a chip firing game on G. The stable divisorial gonality of G is the minimum divisorial gonality over all subdivisions of edges of G.

In this paper we prove that deciding whether a given connected graph has stable divisorial gonality at most a given integer k belongs to the class NP. Combined with the result that (stable) divisorial gonality is NP-hard by Gijswijt, we obtain that stable divisorial gonality is NP-complete. The proof consists of a partial certificate that can be verified by solving an Integer Linear Programming instance. As a corollary, we have that the number of subdivisions needed for minimum stable divisorial gonality of a graph with n vertices is bounded by $2^{p(n)}$ for a polynomial p.

1 Introduction

The notions of the divisorial gonality and stable divisorial gonality of a graph find their origin in algebraic geometry and are related to the abelian sandpile model (cf. [8]). The notion of divisorial gonality was introduced by Baker and Norine [1,2], under the name gonality. As there are several different notions of gonality in use (cf. [1,6,7]), we add the term *divisorial*, following [6]. See [7, Appendix A] for an overview of the different notions.

Divisorial gonality and stable divisorial gonality have definitions in terms of a chip firing game. In this chip firing game, played on a connected multigraph $G = (V, E)$, each vertex has a non-negative number of chips. When we *fire* a set of vertices $S \subseteq V$, we move from each vertex $v \in S$ one chip over each edge with v as endpoint. Each vertex v in S has its number of chips decreased by the number of edges from v to a neighbour not in S, and each vertex v not in S has

H. L. Bodlaender—This work was supported by the NETWORKS project, funded by the Netherlands Organization for Scientific Research NWO under project no. 024.002.003.

B. Catania et al. (Eds.): SOFSEM 2019, LNCS 11376, pp. 81–93, 2019.
https://doi.org/10.1007/978-3-030-10801-4_8

its number of chips increased by the number of edges from v to a neighbour in S. Such a firing move is only allowed when after the move, each vertex still has a nonnegative number of chips. The *divisorial gonality* of a connected graph G can be defined as the minimum number of chips in an initial assignment of chips (called *divisor*) such that for each vertex $v \in V$, there is a sequence of allowed firing moves resulting in at least one chip on v. Interestingly, this number equals the number for a *monotone* variant, where we require that each set that is fired has the previously fired set as a subset. See Sect. 2 for precise definitions.

A variant of divisorial gonality is *stable divisorial gonality*. The stable divisorial gonality of a graph is the minimum of the divisorial gonality over all subdivisions of a graph; we can subdivide the edges of the graph any nonnegative number of times. (In the application in algebraic geometry, the notion of *refinement* is used. Here, we can subdivide edges but also add new degree-one vertices to the graph in a refinement, but as this never decreases the number of chips needed, we can ignore the possibility of adding leaves. Thus, we use subdivisions instead of refinements).

It is known that treewidth is a lower bound for stable divisorial gonality [10]. The stable divisorial gonality of a graph is at most the divisorial gonality, but this inequality can be strict, see for example [4, Fig. 1].

In this paper, we study the complexity of computing the stable divisorial gonality of graphs: i.e., we look at the complexity of the Stable Divisorial Gonality problem: given an undirected graph $G = (V, E)$ and an integer k, decide whether the stable divisorial gonality of G is at most k. It was shown by Gijswijt [11] that divisorial gonality is NP-complete. The same reduction gives that stable divisorial gonality is NP-hard. However, membership of stable divisorial gonality in NP is not trivial: it is unknown how many subdivisions are needed to obtain a subdivision with minimum divisorial gonality. In particular, it is open whether a polynomial number of edge subdivisions are sufficient.

In this paper, we show that stable divisorial gonality belongs to the class NP. We use the following proof technique, which we think is interesting in its own right: we give partial certificates that describe only some aspects of a firing sequence. Checking if a partial certificate indeed corresponds to a solution is non-trivial, but can be done by solving an integer linear program. Membership in NP follows by adding to the partial certificate, that describes aspects of the firing sequence, a certificate for the derived ILP instance. As a corollary, we have that the number of subdivisions needed for minimum stable divisorial gonality of a graph with n vertices is at most $2^{p(n)}$ for a polynomial p.

We finish this introduction by giving an overview of the few previously known results on the algorithmic complexity of (stable) divisorial gonality. Bodewes et al. [4] showed that deciding whether a graph has stable divisorial gonality at most 2, and whether it has divisorial gonality at most 2 can be done in $O(n \log n + m)$ time. From [9] and [3], it follows that divisorial gonality belongs to the class XP, i.e. there is an algorithm that decides in time $O(n^{f(k)})$ whether $\text{dgon}(G) \leq k$. It is open whether stable divisorial gonality is in XP. NP-hardness of the notions was shown by Gijswijt [11].

2 Preliminaries

In this paper, we assume that each graph is a connected undirected multigraph, i.e., we allow parallel edges. In the algebraic number theoretic application of (stable) divisorial gonality, graphs can also have selfloops (edges with both endpoints at the same vertex), but as the (stable) divisorial gonality of graph does not change when we remove selfloops, we assume that there are no selfloops.

A *divisor* D is a function $D\colon V(G) \to \mathbb{Z}$. We can think of a divisor as an assignment of chips, each vertex v has $D(v)$ chips. The *degree* of a divisor is the total number of chips on the graph: $\deg(D) = \sum_{v \in V} D(v)$. We call a divisor *effective* if $D(v) \geq 0$ for all vertices v. Let D be an effective divisor and A a set of vertices. We call A *valid*, if for all vertices $v \in A$ it holds that $D(v)$ is at least the number of edges from v to a vertex outside A. When we *fire* a set A, we obtain a new divisor: for every vertex $v \in A$, the value of $D(v)$ is decreased by the number of edges from v to vertices outside A and for every vertex $v \notin A$, the value $D(v)$ is increased by the number of edges from v to A. We are only allowed to fire valid sets, so that the divisor obtained is again effective.

Two divisors D and D' are called *equivalent*, if there is an increasing sequence of sets $A_1 \subseteq A_2 \subseteq \ldots \subseteq A_k \subseteq V$ such that for every i the set A_i is valid after we fired $A_1, A_2, \ldots, A_{i-1}$ starting from D, and firing $A_1, A_2, \ldots, A_k$ yields D'. We write $D \sim D'$ to denote that two divisors are equivalent. For two equivalent divisors D and D', the difference $D' - D$ is called *transformation* and the sequence $A_1, A_2, \ldots, A_k$ is called a *level set decomposition* of this transformation. A divisor D *reaches* a vertex v if it is equivalent to a divisor D' with $D'(v) \geq 1$.

A subdivision of a graph G is a graph H obtained from G by applying a nonnegative number of times the following operation: take an edge between two vertices v and w and replace this edge by two edges to a new vertex x.

The *stable divisorial gonality* $\text{sdgon}(G)$ of a graph G is the minimum number k such that there exists a subdivision H of G and a divisor on H with degree k that reaches all vertices of H.

There are several equivalent definitions, which we omit here. If we do not require that the sequence of firing sets is increasing, i.e., we omit the requirements $A_i \subseteq A_{i+1}$, then we still have the same graph parameter (see [9]). The notion of a firing set can be replaced by an algebraic operation (see [2]); instead of subdivisions, we can use *refinements* where we allow that we add subdivisions and trees, i.e., we can repeatedly add new vertices of degree one. The definition we use here is most intuitive and useful for our proofs.

3 A (Partial) Certificate

Assume that we are given a yes-instance (G, k) of the problem. Without loss of generality, we assume that $k \leq n$. There exists a subdivision H and a divisor D on H with k chips that reaches all vertices. We do not know whether the number of subdivisions in H is polynomial in the size of the graph, i.e. in the number of vertices and edges of the graph, so we cannot include H in a polynomial certificate for this instance. But the chips in D can be placed on added vertices of H, so we cannot include D in our certificate either. We will prove that when we subdivide every edge once, we can assume that there is a divisor D' that reaches all vertices and has all chips on vertices of this new graph, and hence we can include D' in a polynomial certificate.

Definition 1. *Let G be a graph. Let G_1 denote the graph obtained by subdividing every edge of G once.*

Lemma 2. *Let G be a graph. The stable divisorial gonality of G is at most k if and only if there is a subdivision H of G_1 and a divisor D on H such that*

- *D has at most k chips, i.e. has degree at most k,*
- *D reaches all vertices of H,*
- *D has only chips on vertices of G_1.*

Proof. Suppose that there exists a subdivision H of G_1 and a divisor with the desired properties. Then it is clear that the stable divisorial gonality of G is at most k, since H is a subdivision of G as well.

Suppose that G has stable divisorial gonality at most k. Then there is a subdivision H of G and a divisor D on H with degree at most k that reaches all vertices. If not every edge of G is subdivided in H, then subdivide every edge of H to obtain H_1. Consider the divisor D on H_1. By [12, Corollary 3.4] D reaches all vertices of H_1.

Let $e = uv$ be an edge of G, and let $a_1, a_2, \ldots, a_r$ be the vertices that are added to e in H_1. Suppose that D assigns more than one chip to those added vertices, say it assigns one chip to a_i and one to a_j with $i \leq j$. Then we can fire sets $\{a_h \mid i \leq h \leq j\}$, $\{a_h \mid i-1 \leq h \leq j+1\}$, ... until at least one of the chips lies on u or v. Hence, D is equivalent to a divisor which has one chip less on added vertices. Repeat this procedure until there is for every edge of G at most one chip assigned to the vertices added to that edge. The divisor obtained in this way is equivalent to D, so it reaches all vertices of H_1 and has at most k chips. Thus we have obtained a divisor with the desired properties. □

Now a certificate can contain the graph G_1 and the divisor D as in Lemma 2. From now on we assume D to have chips on vertices of G_1 only. A divisor D as in Lemma 2 reaches all vertices, so for every vertex $w \in V(G_1)$ there is a divisor $D_w \sim D$ with a chip on w and a level set decomposition $A_1, A_2, \ldots, A_r$ of the transformation $D_w - D$. Again we do not know whether r is polynomial in the size of G, so we cannot include this level set decomposition in the certificate. However, we can define some of the sets to be 'relevant', and include all relevant sets in the certificate.

Definition 3. *Let G be a graph and H a subdivision of G. Let D be a divisor on H and $A_1, A_2, \ldots, A_r$ a level set decomposition of a transformation $D' - D$. Let $D_0, D_1, \ldots, D_r$ be the associated sequence of divisors. We call A_i* relevant *if any of the following holds:*

– *A_i moves a chip from a vertex of G, i.e. there is a vertex v of G such that $D_i(v) - D_{i-1}(v) < 0$, or*
– *A_i moves a chip to a vertex of G, i.e. there is a vertex v of G such that $D_i(v) - D_{i-1}(v) > 0$, or*
– *there is a vertex of G such that A_i is the first level set that contains this element, i.e. $(A_i \backslash A_{i-1}) \cap V(G)$ is not empty.*

Lemma 4. *Let G be a graph and H a subdivision of G. Let D be a divisor on H with k chips and $A_1, A_2, \dots, A_r$ a level set decomposition of a transformation $D' - D$. Let $D_0, D_1, \dots, D_r$ be the associated sequence of divisors. Then there are at most $2kn + n$ relevant level sets.*

Proof. Each chip can reach each vertex at most once and can depart at most once from each vertex. So, there are at most kn sets A_i that fulfil the first condition of Definition 3 and at most kn sets that fulfil the second condition. Clearly, the number of sets A_i that fulfil the third condition is upper bounded by the number of vertices of G. □

This lemma shows that the number of relevant sets in a level set decomposition is polynomial, since $k \leq n$. However, the number of elements of each of these sets can still be exponential, so we cannot include those sets in a polynomial certificate. Instead, for a relevant set A_i, we will include $A_i \cap V(G_1)$ in our certificate. Moreover, for each relevant set, we will describe which chips move from/to a vertex of G_1 by firing A_i. When chip j is moved from a vertex v along edge e, we include a tuple $(v, j, -1, e)$, and when a chip j is moved towards a vertex v along edge e, we include a tuple $(v, j, +1, e)$.

Now, a *partial certificate* $\mathcal{C}$ consists of

– a divisor D on G_1 with k chips, where the chips are labelled $1, 2, \dots, k$,
– for every vertex $w \in V(G_1)$, a series of pairs $(A_{w,1}, M_{w,1})$, $(A_{w,2}, M_{w,2})$, …, (A_{w,a_w}, M_{w,a_w}) for some integer a_w, such that
  - $A_{w,1} \subseteq A_{w,2} \subseteq \dots \subseteq A_{w,a_w} \subseteq V(G_1)$,
  - $M_{w,i} = \{(v, j, \sigma, e) \mid v \in V(G_1), 1 \leq j \leq k, \sigma \in \{-1, +1\}, e \in E(G_1)\}$.

This partial certificate should satisfy a lot of conditions, which are implicit in the intuitive explanation of this partial certificate. We list the intuition behind these conditions below and give the formal definition between brackets.

Incidence requirement. The edge along which a chip is fired is incident to the vertex from/to which it is fired. (For every $M_{w,i}$ and every tuple $(v, j, \sigma, e) \in M_{w,i}$, it holds that e is incident to v.)

Departure requirement. If a chip leaves a vertex, then this vertex is fired and its neighbour is not. (For every $M_{w,i}$ and $(v, j, -1, uv) \in M_{w,i}$, it holds that $v \in A_{w,i}$ and $u \notin A_{w,i}$.)

Arrival requirement. If a chip arrives at a vertex, then this vertex is not fired and its neighbour is. (For every $M_{w,i}$ and $(v, j, +1, uv) \in M_{w,i}$, it holds that $v \notin A_{w,i}$ and $u \in A_{w,i}$.)

Unique departure per edge requirement. For every vertex at most one chip leaves along each edge. (For every $M_{w,i}$ and $(v, j_1, -1, e), (v, j_2, -1, e) \in M_{w,i}$, it holds that $j_1 = j_2$.)

Unique arrival per edge requirement. For every vertex at most one chip arrives along each edge. (For every $M_{w,i}$ and $(v, j_1, +1, e), (v, j_2, +1, e) \in M_{w,i}$, it holds that $j_1 = j_2$.)

Unique departure per chip requirement. A chip can leave a vertex along at most one edge. (For every $M_{w,i}$ and $(v_1, j, -1, e_1), (v_2, j, -1, e_2) \in M_{w,i}$, it holds that $v_1 = v_2$ and $e_1 = e_2$.)

Unique arrival per chip requirement. A chip can arrive at a vertex along at most one edge. (For every $M_{w,i}$ and $(v_1, j, +1, e_1), (v_2, j, +1, e_2) \in M_{w,i}$, it holds that $v_1 = v_2$ and $e_1 = e_2$.)

Immediate arrival requirement. If a chip leaves a vertex v and arrives at another vertex u at the same time, then the chip is fired along the edge uv. (For every $M_{w,i}$ and $(v_1, j, -1, e_1), (v_2, j, +1, e_2) \in M_{w,i}$, it holds that $e_1 = e_2 = v_1 v_2$.)

Departure location requirement. If a chip leaves a vertex, then this chip was on this vertex, that is, either the last movement of this chip was to this vertex, or it was assigned to this vertex by D and did not move. (For every $M_{w,i}$ and $(v, j, -1, e) \in M_{w,i}$, the following holds. Let $i' < i$ be the greatest index such that there is a tuple $(u, j, \sigma, e') \in M_{w,i'}$, if it exists. Then there is a tuple $(v, j, +1, e') \in M_{w,i'}$ for some e'. If no such index i' exists, then D assigns j to v.)

Arrival location requirement. If a chip arrives at a vertex, then this chip was moving along an edge to this vertex, that is, either this chip just left the other end of the edge, or it left before and did not yet arrive. (For every $M_{w,i}$ and $(v, j, +1, e) \in M_{w,i}$, either $(u, j, -1, e) \in M_{w,i}$ where $u \neq v$, or the following holds. Let $i' < i$ be the greatest index such that there is a tuple $(u, j, \sigma, e') \in M_{w,i'}$. There is a tuple $(u, j, -1, e) \in M_{w,i'}$ with $u \neq v$ and $(v, j, +1, e) \notin M_{w,i'}$.)

Outgoing edges requirement. A chip is fired along each outgoing edge, that is, for each outgoing edge uv either a new chip leaves u or there is a chip that left u already and did not yet arrive at v. (For every $A_{w,i}$ and for every edge uv such that $u \in A_{w,i}$, $v \notin A_{w,i}$, the following holds. Either $(u, j, -1, uv) \in M_{w,i}$ for some j, or there is a $1 \leq j \leq k$ and an $i' < i$ such that $(u, j, -1, uv) \in M_{w,i'}$ and $(v, j, +1, uv) \notin M_{w,i''}$ for all $i' \leq i'' < i$.)

Previous departure requirement. If a chip leaves a vertex v along some edge e, and v was in the previous firing set as well, then a chip left v along e when the previous set was fired. (For every $A_{w,i}$ and $M_{w,i}$, the following holds. If $v \in A_{w,i}$, $v \in A_{w,i+1}$ and $(v, j, -1, e) \in M_{w,i+1}$ for some j and e, then $(v, j', -1, e) \in M_{w,i}$ for some $j' \neq j$.)

Next arrival requirement. If a chip arrives at a vertex v along some edge e, and v is not in the next firing set as well, then a chip will arrive at v along e when the next set is fired. (For every $A_{w,i}$ and $M_{w,i}$, the following holds. If $v \notin A_{w,i}$, $v \notin A_{w,i+1}$ and $(v, j, +1, e) \in M_{w,i}$ for some j and e, then $(v, j', +1, e) \in M_{w,i+1}$ for some $j' \neq j$.)

Reach all vertices requirement. For all vertices w, at the end of the sequence $A_{w,1}, \ldots, A_{w,a_w}$, there is a chip on w. (For every vertex w, either there is a $1 \leq j \leq k$ and an i such that $(w, j, +1, e) \in M_{w,i}$ for some e and $(w, j, -1, e') \notin M_{w,i'}$ for all $i' \geq i$, or there is a $1 \leq j \leq k$ that D assigns to w and $(w, j, -1, e) \notin M_{w,i}$ for all i.)

Now for a given graph G, and such a partial certificate $\mathcal{C}$, we want to decide whether there is a subdivision of G_1 such that for every vertex $w \in V(G_1)$ there is a divisor $D_w \sim D$ with a chip on w such that the sets $A_{w,1}, \ldots, A_{w,a_w}$ are the relevant sets of the level set decomposition of the transformation $D_w - D$. To decide this, we will construct an integer linear program $\mathcal{I}_{\mathcal{C}}$, such that this program has a solution if and only if there is such a subdivision of G_1. Since integer linear programming is in NP, we know that if there is a solution to $\mathcal{I}_{\mathcal{C}}$, then there is a polynomial certificate $\mathcal{D}$ for the ILP instance. In order to obtain a certificate for the Stable Divisorial Gonality problem, we add the certificate for the ILP instance to the partial certificate, as defined above. Thus, a certificate for the Stable Divisorial Gonality problem is then of the form $(\mathcal{C}, \mathcal{D})$.

For the integer linear program $\mathcal{I}_{\mathcal{C}}$, we introduce some variables. For every vertex $w \in V(G_1)$ and every $1 \leq i < a_w$, we define a variable $t_{w,i}$. This variable represents the number of sets that is fired between $A_{w,i}$ and $A_{w,i+1}$, including $A_{w,i}$ and excluding $A_{w,i+1}$. For every edge e of G_1, we define a variable l_e, which represents the length of e, i.e. the number of edges that e is subdivided into. Now we construct $\mathcal{I}_{\mathcal{C}}$:

- For every edge $e \in E(G_1)$, include the inequality $l_e \geq 1$. (Every edge has length at least one.)
- For every vertex $w \in V(G_1)$ and $1 \leq i < a_w$, include the inequality $t_{w,i} \geq 1$. (The set $A_{w,i}$ is fired, so $t_{w,i} \geq 1$.)
- For every edge $e = uv$ of G_1 such that there is a set $M_{w,i}$ with $(v, j, -1, e)$, $(u, j, +1, e) \in M_{w,i}$ for some j, include $l_e = 1$ in $\mathcal{I}_{\mathcal{C}}$. (If a chip arrives immediately after it is fired, then the edge has length one.)
- For every vertex $w \in V(G_1)$ and $1 \leq i < a_w$ such that there are v, j_1, j_2, e such that $(v, j_1, -1, e) \in M_{w,i}$ and $(v, j_2, -1, e) \in M_{w,i+1}$, include $t_{w,i} = 1$ in $\mathcal{I}_{\mathcal{C}}$. (If there is a set A that is fired between $A_{w,i}$ and $A_{w,i+1}$, then $A_{w,i} \subseteq A \subseteq A_{w,i+1}$. It follows that A fires a chip from v along e as well. But then A is a relevant set. We conclude that $t_{w,i} = 1$.)
- For every vertex $w \in V(G_1)$ and $1 \leq i \leq a_w$ such that there are v, j, e such that $(v, j, +1, e) \in M_{w,i}$, include $t_{w,i} = 1$ in $\mathcal{I}_{\mathcal{C}}$. (Notice that the set fired after $A_{w,i}$ either contains v or causes a chip to arrive at v, so this set is relevant.)
- For every vertex w and edge $e = uv$ of G_1, let i_0 be the smallest index such that $(v, j, -1, e) \in M_{w,i_0}$ for some j, i_1 the greatest index such that $(v, j, -1, e) \in M_{w,i_1}$ for some j, i_2 the smallest index such that $(u, j, +1, e) \in M_{w,i_2}$ for some j, and i_3 the greatest index such that $(u, j, +1, e) \in M_{w,i_3}$ for some j. Include the following inequalities in $\mathcal{I}_{\mathcal{C}}$:

$$(i_1 - i_0 + 1)l_e - (i_1 - i_0) + (i_3 - i_2) \geq \sum_{i=i_0}^{i_3} t_{w,i} \tag{1}$$

$$(i_3 - i_2 + 1)l_e + (i_1 - i_0) - (i_3 - i_2) \leq \sum_{i=i_0}^{i_3} t_{w,i}. \tag{2}$$

(There are $i_1 - i_0 + 1$ chips that left v along edge e, and $i_3 - i_2 + 1$ chips that arrived at u along e. There are $\sum_{i=i_0}^{i_3} t_{w,i}$ sets fired since the first chip left until the last chip arrives, and every of these sets causes one chip to move one step. The chips that arrived at u took l_e steps, the chips that did not arrive took at least one and at most $l_e - 1$ steps. This yields the inequalities.)

Now a certificate for the stable divisorial gonality problem is a pair $(\mathcal{C}, \mathcal{D})$. Here, the partial certificate $\mathcal{C}$ contains a divisor D on G_1 with labelled chips and for every vertex $w \in V(G_1)$ a series of pairs $(A_{w,1}, M_{w,1}), (A_{w,2}, M_{w,2}), \ldots, (A_{w,a_w}, M_{w,a_w})$, and satisfies all requirements above. And $\mathcal{D}$ is a certificate of the integer linear program $\mathcal{I}_{\mathcal{C}}$.

4 Correctness

It remains to prove that there exists a certificate $(\mathcal{C}, \mathcal{D})$ if and only if $\text{sdgon}(G) \leq k$.

Lemma 5. *Let G be a graph with* $\text{sdgon}(G) \leq k$*. There exists a certificate $(\mathcal{C}, \mathcal{D})$.*

Proof. By Lemma 2 we know that there is a subdivision H of G_1 and a divisor D with k chips, all on vertices of G_1, that reaches all vertices. Choose a labeling of the chips and let D be the divisor in $\mathcal{C}$.

For every vertex $w \in V(G_1)$, there is a divisor $D_w \sim D$ with a chip on w and a level set decomposition $A_{w,1}, \ldots, A_{w,a_w}$. Let $A_{w,i_1}, \ldots, A_{w,i_{b_w}}$ be the subsequence consisting of all relevant sets. Let $B_{w,1} = A_{w,i_1} \cap V(G_1), \ldots, B_{w,b_w} = A_{w,i_{b_w}} \cap V(G_1)$.

Fire the sets $A_{w,1}, \ldots, A_{w,a_w}$ in order. For every i_j, set $M_{w,j} = \emptyset$. When firing the set A_{w,i_j}, check for every chip h whether it arrives at a vertex v of G_1 or leaves a vertex v of G_1. If so, add the tuple (v, h, σ, e) to $M_{w,j}$, where $\sigma = +1$ if h arrives at v and $\sigma = -1$ if h leaves v, and e is the edge of G_1 along which h moves.

The divisor D together with the sequences $(B_{w,i}, M_{w,i})$, for every vertex $w \in V(G_1)$, is the partial certificate $\mathcal{C}$. Notice that by definition $\mathcal{C}$ satisfies all conditions: Incidence requirement, Departure requirement, Arrival requirement, Unique departure per edge, Unique arrival per edge, Unique departure per chip, Unique arrival per chip, Immediate arrival, Departure location, Arrival location, Outgoing edges requirement, Previous departure, Next arrival and Reach all vertices.

For every edge e of G_1, define l_e as the number of edges that e is subdivided into in H. For every vertex w of G_1 and $1 \leq j \leq b_w - 1$, define $t_{w,i}$ as the number of sets between $A_{w,i+1}$ and $A_{w,i}$, including $A_{w,i}$ and excluding $A_{w,i+1}$. Notice that this is a solution to the integer linear program $\mathcal{I}_\mathcal{C}$. So this is a certificate for this program, write $\mathcal{D}$ for this certificate. Now $(\mathcal{C}, \mathcal{D})$ is a certificate for (G, k). □

We illustrate our proof with an example.

Fig. 1. (a) A graph G (b) A subdivision of G and divisor

Example 6. Consider the graph in Fig. 1a. Consider the subdivision in Fig. 1b and the divisor D with 7 chips on u. This divisor reaches v, for example by firing the following sets:

$$\begin{aligned} &\{u\}, \{u\}, \{u\}, \{u, y_1\}, \{u, x_1, y_1\}, \{u, x_1, y_1\}, \\ &\{u, x_1, y_1, y_2\}, \{u, x_1, y_1, y_2\}, \{u, x_1, x_2, y_1, y_2\}, \\ &\{u, x_1, x_2, y_1, y_2, y_3\}, \{u, x_1, x_2, y_1, y_2, y_3\}, \{u, x_1, x_2, y_1, y_2, y_3\}. \end{aligned}$$

We describe the corresponding partial certificate $(\mathcal{C}, \mathcal{D})$. The divisor D will be included in $\mathcal{C}$. Notice that there are 8 relevant sets. We obtain the following series of pairs, after labelling the chips $1, 2, \ldots, 7$:

$$\begin{aligned} A_{v,1} &= \{u\}, \quad M_{v,1} = \{(u, 1, -1, e_1), (u, 2, -1, e_2)\} \\ A_{v,2} &= \{u\}, \quad M_{v,2} = \{(u, 3, -1, e_1), (u, 4, -1, e_2)\} \\ A_{v,3} &= \{u\}, \quad M_{v,3} = \{(u, 5, -1, e_1), (u, 6, -1, e_2)\} \\ A_{v,4} &= \{u\}, \quad M_{v,4} = \{(u, 7, -1, e_1)\} \\ A_{v,5} &= \{u\}, \quad M_{v,5} = \{(v, 1, 1, e_1)\} \\ A_{v,6} &= \{u\}, \quad M_{v,6} = \{(v, 3, 1, e_1), (v, 2, 1, e_2)\} \\ A_{v,7} &= \{u\}, \quad M_{v,7} = \{(v, 5, 1, e_1), (v, 4, 1, e_2)\} \\ A_{v,8} &= \{u\}, \quad M_{v,8} = \{(v, 7, 1, e_1), (v, 6, 1, e_2)\} \end{aligned}$$

This gives the partial certificate $\mathcal{C}$. The partial certificate $\mathcal{D}$ consists of a solution to the integer linear program $\mathcal{I}_{\mathcal{C}}$. Here, the corresponding program is:

$$\begin{aligned}
&l_{e_1} \geq 1 && && 4l_{e_1} \leq \sum_{i=0}^{8} t_{v,i}\\
&l_{e_2} \geq 1 && && \\
&t_{v,i} \geq 1 && \text{for } i \in \{1,2,\ldots,8\} && 3l_{e_2} - 1 \geq \sum_{i=0}^{8} t_{v,i}\\
&t_{v,i} = 1 && \text{for } i \in \{1,2,3\} && \\
&t_{v,i} = 1 && \text{for } i \in \{5,6,7,8\} && 3l_{e_2} + 1 \leq \sum_{i=0}^{8} t_{v,i}\\
&4l_{e_1} \geq \sum_{i=0}^{8} t_{v,i} && &&
\end{aligned}$$

We can simplify this to:

$$\begin{aligned}
&l_{e_1} \geq 1 && && 4l_{e_1} \geq t_{v,4} + 7\\
&l_{e_2} \geq 1 && && 4l_{e_1} \leq t_{v,4} + 7\\
&t_{v,i} = 1 && \text{for } i \in \{1,2,3,5,6,7,8\} && 3l_{e_2} \geq t_{v,4} + 7\\
&t_{v,4} \geq 1 && && 3l_{e_2} \leq t_{v,4} + 7
\end{aligned}$$

We see that $l_{e_1} = 3$, $l_{e_2} = 4$, $t_{v,4} = 5$ and $t_{v,i} = 1$ for $i \neq 4$ is a solution to this program, let this solution be the certificate $\mathcal{D}$.

Lemma 7. *Let G be a graph and k a natural number. If there exists a certificate $(\mathcal{C}, \mathcal{D})$, then* $\text{sdgon}(G) \leq k$.

The idea of the proof of this lemma is as follows. Suppose we are given a certificate. Subdivide every edge of G_1 in l_e edges. Make $t_{w,i}$ copies of set $A_{w,i}$. For every edge $e = uv$ we distribute the added vertices over the copies of $A_{w,i}$ such that as many chips depart from u along e as described by the tuples and as many chips arrive at v along e as described by the tuples. Using the conditions that our certificate satisfies, we can prove that all chips are moved as described by the tuples in the sets $M_{w,i}$. We illustrate this idea in the following example. For all details see [5, Lemma 4.3].

Example 8. Again consider the graph in Fig. 1a and the certificate in Example 6. Since $l_{e_1} = 3$, we subdivide e_1 with two vertices x_1 and x_2 and since $l_{e_2} = 4$, we subdivide e_2 with three vertices y_1, y_2 and y_3.

We make 5 copies of set $A_{w,4}$, since $t_{w,4} = 5$. The first set that fires a chip along e_1 is $A_{v,1}$ and the last such set is $A_{v,8}$, in total there are 12 sets that fire a chip along e_1. When we fire the first four sets, a chip departs from u along e_1, so we will not add x_1 and x_2 to the first four sets. When we fire the last four sets, a chip arrives at v along e_1, so we add x_1 and x_2 to the last four sets. We add x_1 to the middle four sets, so that the chips move from x_1 to x_2. This yields:

$$\begin{aligned}
A_{v,1} = \{u\}, A_{v,2} = \{u\}, A_{v,3} = \{u\}, A_{v,4,1} = \{u\}, A_{v,4,2} = \{u, x_1\},\\
A_{v,4,3} = \{u, x_1\}, A_{v,4,4} = \{u, x_1\}, A_{v,4,5} = \{u, x_1\}, A_{v,5} = \{u, x_1, x_2\},\\
A_{v,6} = \{u, x_1, x_2\}, A_{v,7} = \{u, x_1, x_2\}, A_{v,8} = \{u, x_1, x_2\}.
\end{aligned}$$

Analogously for e_2, we add the vertices y_1, y_2 and y_3:

$$\{u\}, \{u\}, \{u\}, \{u, y_1\}, \{u, x_1, y_1\}, \{u, x_1, y_1\},$$
$$\{u, x_1, y_1, y_2\}, \{u, x_1, y_1, y_2\}, \{u, x_1, x_2, y_1, y_2\},$$
$$\{u, x_1, x_2, y_1, y_2, y_3\}, \{u, x_1, x_2, y_1, y_2, y_3\}, \{u, x_1, x_2, y_1, y_2, y_3\}.$$

We see that we obtained the same subdivision and firing sets as we started with in Example 6.

As ILP's have certificates with polynomially many bits (see e.g., [13]), and the partial certificate is of polynomial size (see also Lemma 4), we have that, using Lemmas 5 and 7, the problem whether a given graph has divisorial gonality at most a given integer k has a polynomial certificate, which gives our main result.

Theorem 9. Stable Divisorial Gonality *belongs to the class NP.*

Combined with the NP-hardness of Stable Divisorial Gonality by Gijswijt [11], this yields the following result.

Theorem 10. Stable Divisorial Gonality *is NP-complete.*

5 A Bound on Subdivisions

In this section, we give as corollary of our main result a bound on the number of subdivisions needed. We use the following result by Papadimitriou [13].

Theorem 11 (Papadimitriou [13]). *Let A be an $m \times n$ matrix, and b be a vector of length m, such that each value in A and b is an integer in the interval $[-a, +a]$. If $Ax = b$ has a solution with all values being positive integers, then $Ax = b$ has a solution with all values positive integers that are at most $n(ma)^{2m+1}$.*

Corollary 12. *Let G be a graph with stable divisorial gonality k. There is a graph H, that is a subdivision of G, with the divisorial gonality of H equal to the stable divisorial gonality of G, and each edge in H is obtained by subdividing an edge from G at most $m^{O(km^2)}$ times.*

Proof. By Lemma 5, we know that there is a certificate whose corresponding ILP has a solution. The values l_e in this solution give the number of subdivisions of edges in G_1. If we have an upper bound on the number of subdivisions per edge needed to obtain H from G_1, say α, then $2\alpha+1$ is an upper bound on the number of subdivisions per edge to obtain H from G. Applying Theorem 11 to the ILP gives such a bound, as described below.

The ILP has at most $n' \cdot (2kn' + n')$ variables of the form $t_{w,i}$, by Lemma 4, and m' variables of the form l_e, with n' the number of vertices in G_1 and m' the number of edges in G_1. We have $n' = n + m$, and $m' = 2m$, with n' the number of vertices of G and m the number of edges of G.

The number of equations and inequalities in the ILP is linear in the number of variables. An inequality can be replaced by an equation by adding one variable. This gives a total of $O(kn'^2 + m')$ variables and $O(kn'^2 + m')$ equations. Note that $O(kn'^2 + m') = O(km^2)$; as G is connected, $n \leq m - 1$. Also, note that all values in matrix A and vector b are -1, 0, or 1, i.e., we can set $a = 1$ in the application of Theorem 11. So, by Theorem 11, we obtain that if there is a solution to the ILP, then there is one where all variables are set to values at most

$$O(kn'^2 + m') \cdot O(kn'^2 + m')^{O(kn'^2+m')} = O(km^2) \cdot O(km^2)^{O(km^2)} = m^{O(km^2)}.$$

Denoting by k the stable divisorial gonality of G, we know there is at least one certificate with a solution, so we can bound the number of subdivisions in G_1 by $m^{O(km^2)}$, which gives our result. □

6 Conclusion

In this paper, we showed that the problem to decide whether the stable divisorial gonality of a given graph is at most a given number k belongs to the class NP. Together with the NP-hardness result of Gijswijt [11], this shows that the problem is NP-complete. We think our proof technique is interesting: we give a certificate that describes some of the essential aspects of the firing sequences; whether there is a subdivision of the graph for which this certificate describes the firing sequences and thus gives the subdivision that reaches the optimal divisorial gonality can be expressed in an integer linear program. Membership in NP then follows by adding the certificate of the ILP to the certificate for the essential aspects.

As a byproduct of our work, we obtained an upper bound on the number of subdivisions needed to reach a subdivision of G whose divisorial gonality gives the stable divisorial gonality of G. Our upper bound still is very high, namely exponential in a polynomial of the size of the graph. An interesting open problem is whether this bound on the number of needed subdivisions can be replaced by a polynomial in the size of the graph. Such a result would give an alternative (and probably easier) proof of membership in NP: first guess a subdivision, and then guess the firing sequences.

There are several open problems related to the complexity of computing the (stable) divisorial gonality of graphs. Are these problems *fixed parameter tractable*, i.e., can they be solved in $O(f(k)n^c)$ time for constant c and some function f that depends only on k? Or can they be proven to be $W[1]$-hard, or even, is there a constant c, such that deciding if (stable) divisorial gonality of a given graph G is at most c is already NP-complete? Also, how well can we approximate the divisorial gonality or stable divisorial gonality of a graph?

Acknowledgements. We thank Gunther Cornelissen and Nils Donselaar for helpful discussions.

References

1. Baker, M.: Specialization of linear systems from curves to graphs. Algebra Number Theory **2**(6), 613–653 (2008). https://doi.org/10.2140/ant.2008.2.613
2. Baker, M., Norine, S.: Riemann-Roch and Abel-Jacobi theory on a finite graph. Adv. Math. **215**(2), 766–788 (2007). https://doi.org/10.1016/j.aim.2007.04.012
3. Baker, M., Shokrieh, F.: Chip-firing games, potential theory on graphs, and spanning trees. J. Comb. Theory Ser. A **120**(1), 164–182 (2013). https://doi.org/10.1016/j.jcta.2012.07.011
4. Bodewes, J.M., Bodlaender, H.L., Cornelissen, G., van der Wegen, M.: Recognizing hyperelliptic graphs in polynomial time. In: Brandstädt, A., Köhler, E., Meer, K. (eds.) Graph-Theoretic Concepts in Computer Science, pp. 52–64 (2018). (extended abstract of http://arxiv.org/abs/1706.05670)
5. Bodlaender, H.L., van der Wegen, M., van der Zanden, T.C.: Stable divisorial gonality is in NP (2018). http://arxiv.org/abs/1808.06921
6. Caporaso, L.: Gonality of algebraic curves and graphs. In: Frühbis-Krüger, A., Kloosterman, R., Schütt, M. (eds.) Algebraic and Complex Geometry, vol. 71, pp. 77–108. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05404-9_4
7. Cornelissen, G., Kato, F., Kool, J.: A combinatorial Li-Yau inequality and rational points on curves. Math. Ann. **361**(1–2), 211–258 (2015). https://doi.org/10.1007/s00208-014-1067-x
8. Corry, S., Perkinson, D.: Divisors and Sandpiles: An Introduction to Chip-Firing. American Mathematical Society, Providence (2018)
9. van Dobben de Bruyn, J.: Reduced divisors and gonality in finite graphs. Bachelor thesis, Leiden University (2012). https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/bachvandobbendebruyn.pdf
10. van Dobben de Bruyn, J., Gijswijt, D.: Treewidth is a lower bound on graph gonality (2014). http://arxiv.org/abs/1407.7055v2
11. Gijswijt, D.: Computing divisorial gonality is hard (2015). http://arxiv.org/abs/1504.06713
12. Hladký, J., Král', D., Norine, S.: Rank of divisors on tropical curves. J. Comb. Theory Ser. A **120**(7), 1521–1538 (2013). https://doi.org/10.1016/j.jcta.2013.05.002
13. Papadimitriou, C.H.: On the complexity of integer programming. J. ACM **28**(4), 765–768 (1981). https://doi.org/10.1145/322276.322287