



Stable Divisorial Gonality is in NP

Hans L. Bodlaender¹ · Marieke van der Wegen¹  · Tom C. van der Zanden²

Accepted: 12 November 2020 / Published online: 7 December 2020
© The Author(s) 2020

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 et al., 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 total number of subdivisions needed for minimum stable divisorial gonality of a graph with m edges is bounded by $m^{O(mn)}$.

Keywords Computational complexity · Graphs · Gonality

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.

Hans L. Bodlaender was partially supported by the Networks project, founded by the Dutch Organization for Scientific Research NWO, while partly associated to the Department of Mathematics and Computer Science at Eindhoven University of Technology.

This work was done while Tom C. van der Zanden is associated with the Department of Information and Computing Sciences of Utrecht University.

An extended abstract is published in *SOFSEM 2019: Theory and Practice of Computer Science* [1].

✉ Marieke van der Wegen
M.vanderWegen@uu.nl

¹ Department of Information and Computing Sciences, Universiteit Utrecht, Princetonplein 5, 3584 CC Utrecht, The Netherlands

² Department of Data Analytics and Digitalisation, Maastricht University, Maastricht, The Netherlands

[2]). Baker and Norine introduced divisor theory on graphs in [3]. The notion of divisorial gonality was introduced by Baker [4], under the name gonality. As there are several different notions of gonality in use (cf. [4–6]), we add the term *divisorial*, following [5]. See [6, 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 number of chips. When we *fire* a vertex $v \in V$, we move a chip from vertex v over each edge with v as endpoint. Vertex v has its number of chips decreased by the degree of v , and each neighbour u of v has its number of chips increased by the number of edges from v to u . 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 firing moves resulting in at least one chip on v and a non-negative number of chips on the other vertices. See Section 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 [7]. The stable divisorial gonality of a graph is at most the divisorial gonality, but this inequality can be strict, see for example [8, Figure 2].

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 in [9] 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 a partial certificate that describes some divisors. Checking whether those divisors are equivalent is non-trivial, but can be done by solving an integer linear program. Membership in NP follows by adding a certificate for the derived ILP instance to the partial certificate that describes the divisors. As a corollary, we have that the total number of subdivisions needed for minimum stable divisorial gonality of a graph with m edges is at most $m^{O(mn)}$.

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. [8] showed that deciding whether a graph has (stable) divisorial gonality at most 2 can be done in $O(n \log n + m)$ time. From [10] and [11], 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 in [9].

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 (see the end of this section). For a graph G , we will denote by n its number of vertices and by m the number of edges.

A *divisor* D is a vector in $\mathbb{Z}^{V(G)}$. We can consider a divisor D as a function $D: V(G) \rightarrow \mathbb{Z}$, which is an assignment of chips to vertices, each vertex v having $D(v)$ chips. The *degree* of a divisor is the total number of chips on the graph: $\text{deg}(D) = \sum_{v \in V} D(v)$. The support of D is $\text{supp}(D) = \{v \in V \mid D(v) \neq 0\}$. We call a divisor *effective* if $D(v) \geq 0$ for all vertices v .

We will define an equivalence relation on the set of divisors. The Laplacian matrix \mathcal{L} is defined as follows:

$$\mathcal{L}_{uv} = \begin{cases} \text{deg}(v) & \text{if } u = v, \\ - \text{the number of edges from } u \text{ to } v & \text{otherwise.} \end{cases}$$

Two divisors D and D' are *equivalent* if and only if there exists a vector $f \in \mathbb{Z}^{V(G)}$ such that $D' - D = \mathcal{L}f$. We can consider f as a function $f: V(G) \rightarrow \mathbb{Z}$, we will use the terms vector and function interchangeably.

An equivalent definition can be given using a notion of firing vertices. When we *fire* a vertex v , we move chips from v to its neighbours along its incident edges: starting from a divisor D , we obtain a new divisor, where $D(v)$ is decreased by the degree of v and for all neighbours u of v the value $D(u)$ is increased by the number of edges from v to u . Two divisors D and D' are equivalent if D can be transformed into D' by firing vertices. This is indeed equivalent to the definition above: f describes how often every vertex is fired.

A divisor D *reaches* a vertex v if it is equivalent to an effective divisor D' with $D'(v) \geq 1$.

The *divisorial gonality* $\text{dgon}(G)$ of a graph G is the minimum number k such that there exists a divisor D with degree k that reaches all vertices of G . Since D is equivalent to an effective divisor, we can assume D to be effective.

A subdivision of a graph G is a graph H obtained from G by applying the following operation a nonnegative number of times: 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 exist a subdivision H of G and a divisor D on H with degree k that reaches all vertices of H . That is, the stable divisorial gonality of a graph G is the minimum divisorial gonality over all subdivisions of G .

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. But since the stable divisorial gonality does not change when adding a vertex with degree one, we only consider subdivisions.

As mentioned before, we remove selfloops of a graph, as that will not change the stable divisorial gonality. Now that we have seen the definition of stable divisorial gonality we can give a short argument why this holds. Notice that there will never be chips fired along a selfloop, so removing a selfloop will not change the divisorial gonality of a graph. A loop can be subdivided into a cycle however, but this will not decrease the divisorial gonality of the graph. Suppose that G has a loop at vertex v , and that H is the refinement of G at which the stable divisorial gonality is attained by a divisor D . If the selfloop is subdivided, let $D' \sim D$ be a divisor with as few chips on the subdivisions of the loop as possible, that is, with at most 1 chip in total on the vertices that are added to the selfloop. Notice that D' reaches all vertices of H . If we remove the subdivided selfloop from H , firing the same vertices will yield the same result, with possibly some more chips on vertex v . Hence, D' still reaches all vertices. We conclude that the stable divisorial gonality does not increase by removing the selfloops of G .

3 Equivalence

In this section we will discuss a condition for two divisors to be equivalent. This section is based on a result in [7, Section 4], but given its relevance for this paper, we will explain it here.

Recall that two divisors are equivalent if there exists a vector $f \in \mathbb{Z}^{V(G)}$ such that $D' - D = \mathcal{L}f$. We can consider f as a function $V(G) \rightarrow \mathbb{Z}$; f describes for every vertex the number of times this vertex is fired. In this section we will show that we can find an equivalent definition using a function $g: E(G) \rightarrow \mathbb{Z}$, which describes for every edge how many chips are moved along that edge.

For an edge uv , we will need to know whether $g(uv)$ describes the number of chips that moves from u to v , or the other way around. For this, we need an orientation of the edges.

Let $G = (V, E)$ be a graph. Assign to every edge e of G an arbitrary orientation $\sigma(e)$. For every cycle C , and every edge $e \in E$ let

$$\sigma_C(e) = \begin{cases} 1 & \text{if } e \text{ is traversed in } C \text{ in the forward direction,} \\ -1 & \text{if } e \text{ is traversed in } C \text{ in the backward direction,} \\ 0 & \text{if } e \text{ is not contained in } C. \end{cases}$$

Analogously, we define $\sigma_P(e)$ for every path P . If a path P consists of a single edge uv , we write $\sigma_{uv}(e)$ for $\sigma_P(e)$. So,

$$\sigma_{uv}(e) = \begin{cases} 1 & \text{if } e = uv \text{ is oriented from } u \text{ to } v, \\ -1 & \text{if } e = uv \text{ is oriented from } v \text{ to } u, \\ 0 & \text{if } e \neq uv. \end{cases}$$

The following lemma is known as Kirchhoff’s potential law, see [12, pages 40, 297]. We include a proof, so that this paper is more self-contained.

Lemma 3.1 *For a function $g : E \rightarrow \mathbb{Z}$, there exists a function $f : V \rightarrow \mathbb{Z}$ such that for every edge uv , oriented from u to v , it holds that $g(uv) = f(u) - f(v)$ if and only if for every fundamental cycle C :*

$$\sum_{e \in C} g(e)\sigma_C(e) = 0.$$

Proof We define a function f as follows: Pick a vertex $v \in V$, and set $f(v) = a$ for some $a \in \mathbb{Z}$. For $u \in V$, let P be a path from v to u and set

$$f(u) = a - \sum_{e \in P} g(e)\sigma_P(e).$$

Notice that for f holds that $g(uv) = f(u) - f(v)$. Moreover, every function f for which $g(uv) = f(u) - f(v)$ holds, is of this form.

This function f is well-defined if and only if for all paths P and Q from v to u , it holds that

$$a - \sum_{e \in P} g(e)\sigma_P(e) = a - \sum_{e \in Q} g(e)\sigma_Q(e).$$

Hence, if and only if

$$\sum_{e \in P} g(e)\sigma_P(e) + \sum_{e \in Q} g(e) \cdot -\sigma_Q(e) = 0.$$

Since P followed by Q is reversed order forms a cycle, f is well-defined if and only if for every cycle C :

$$\sum_{e \in C} g(e)\sigma_C(e) = 0. \tag{1}$$

Since every cycle is the sum of some fundamental cycles, we conclude that f is well-defined if and only if (1) holds for every fundamental cycle C . □

Now we can provide an equivalent definition for two divisors to be equivalent.

Lemma 3.2 *Let H be a subdivision of G . Let the orientation of the edges of H be induced by the orientation of the edges of G . For every edge $e \in E(G)$, let $l(e)$ be the number of edges that e is subdivided into in H . Let D, D' be two divisors on G with degree k . Then D and D' are equivalent on H if and only if there exists a function $g : E(G) \rightarrow \{-k, -k + 1, \dots, k - 1, k\}$ such that*

- for every $v \in V(G)$:

$$(D' - D)(v) = \sum_{u \sim v \text{ in } G} g(uv)\sigma_{vu}(uv), \tag{2}$$

- for every fundamental cycle C :

$$\sum_{e \in C} g(e)l(e)\sigma_C(e) = 0. \tag{3}$$

Proof Suppose that D and D' are equivalent in H . From the definition in Section 2, we know that there is a function $f': V(H) \rightarrow \mathbb{Z}$, such that $\mathcal{L}f' = D' - D$. This function f' describes how often every vertex v is fired to transform D into D' .

We now define the function g' , this will describe how many chips move over edge e in the forward direction. Let $e \in E(H)$ and suppose that edge e is directed from u to v , set $g'(e) = f(u) - f(v)$. Notice that $-k \leq g'(e) \leq k$, since there can move at most k chips along each edge. By Lemma 3.1, it follows that for every fundamental cycle C it holds that

$$\sum_{e \in C} g'(e)\sigma_C(e) = 0. \tag{4}$$

Let $e = uv$ be an edge of G , that is subdivided in e_1, e_2, \dots, e_r edges in H . Let $w \in V' \setminus V$ be a vertex that is added to this edge, and let e_i and e_{i+1} be its incident edges. Since D and D' are divisors on G , we see that $D'(w) = D(w) = 0$. It follows that $g'(e_i) = g'(e_{i+1})$. So for every edge e of G , that is subdivided in e_1, e_2, \dots, e_r edges in H , it holds that $g'(e_1) = g'(e_2) = \dots = g'(e_r)$. Define the function $g: E(G) \rightarrow \mathbb{Z}$ by $g(e) = g'(e_1)$. Notice that $g(uv) = (f'(u) - f'(v))/l(e)$.

By the definition of g and the cycle condition (4) for g' , it follows that for every fundamental cycle C it holds that

$$\sum_{e \in C} g(e)l(e)\sigma_C(e) = 0.$$

Moreover, since $\mathcal{L}f' = D' - D$, for every vertex $v \in V(G)$ we have that

$$\begin{aligned} (D' - D)(v) &= \mathcal{L}f'(v) \\ &= \deg(v)f'(v) - \sum_{u \sim v \text{ in } H} f'(u) \\ &= \sum_{u \sim v \text{ in } H} (f'(v) - f'(u)) \\ &= \sum_{u \sim v \text{ in } H} g'(uv)\sigma_{vu}(uv) \\ &= \sum_{u \sim v \text{ in } G} g(uv)\sigma_{vu}(uv). \end{aligned}$$

For the other direction, suppose that there exists a function $g: E(G) \rightarrow \mathbb{Z}$, that satisfies the conditions (2) and (3). By Claim 3.1 it follows that there exists a function $f: V(G) \rightarrow \mathbb{Z}$ such that for every edge $uv \in E(G)$, oriented from u to v , it holds that $g(uv)l(uv) = f(u) - f(v)$.

Define the function $f': V(H) \rightarrow \mathbb{Z}$ as follows. For $v \in V(G)$, set $f'(v) = f(v)$. For every vertex edge $e = uv \in E(G)$, oriented from u to v , let w_1, w_2, \dots, w_r be the vertices that are added to e in H , in order from u to v . Set $f'(w_i) = f(u) - ig(uv)$.

It remains to check that $\mathcal{L}f' = D' - D$. For w_i we see that

$$\begin{aligned} \mathcal{L}f'(w_i) &= 2f(w_i) - f(w_{i-1}) - f(w_{i+1}) \\ &= 0 \\ &= (D' - D)(w_i). \end{aligned}$$

For $v \in V(G)$, we see that

$$\begin{aligned} \mathcal{L}f'(v) &= \sum_{u \sim v \text{ in } H} (f'(v) - f'(u)) \\ &= \sum_{u \sim v \text{ in } G} g(uv)\sigma_{vu}(uv). \end{aligned}$$

By (2), it follows that $\mathcal{L}f'(v) = (D' - D)(v)$. We conclude that D' and D are equivalent in H . □

4 A 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 an effective 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. However, we can include a polynomial subdivision of G . We will choose a subdivision that allows us to include D as well.

Lemma 4.1 *Let G be a graph with n vertices and stable divisorial gonality at most k . There exists a subdivision G' of G , such that*

1. *The number of vertices of G' is at most $n + k(n + 1)$, i.e., G' is obtained from G by adding at most $k(n + 1)$ subdivision vertices.*
2. *There is a subdivision H of G' , such that there is an effective divisor D on H with k chips that reaches all vertices and has only chips on vertices of G' . Moreover, for all vertices $v \in V(G)$, there is an effective divisor D_v on H with $D \sim D_v$ with at least one chip on v and only chips on vertices of G' .*

Proof Since G has stable divisorial gonality at most k , there exists a subdivision H and an effective divisor D on H with k chips that reaches all vertices. For every vertex $v \in V(G)$, let D_v be an effective divisor on H such that $D_v \sim D$ and that has a chip on vertex v . Now define $U \subseteq V(H)$ as

$$U = \text{supp}(D) \cup \bigcup_{v \in V(G)} \{\text{supp}(D_v)\}.$$

Let G' be the subdivision of G that contains all the vertices of U . Notice that H is a subdivision of G' , and that the divisors D and D_v only have chips on G' .

Notice that U contains at most $(n + 1)k$ vertices, since we take the union over $n + 1$ divisors, which each have k chips. So the number of vertices of G' is at most $n + k(n + 1)$. \square

Remark 4.2 Notice that the divisor D could be chosen to be equal to one of the divisors D_v . In that case, U would contain at most kn vertices, and the number of vertices of G' would be improved to $n + kn$.

This subdivision G' is the first ingredient of our polynomial certificate. The second ingredient is the divisor D , and the third ingredient all divisors D_v .

We now need to extend the certificate so that we can check in polynomial time whether there is a subdivision H of G' such that for all v it holds that $D \sim D_v$ in H . To this end, we will check whether the conditions (2) and (3) in Lemma 3.2 are satisfied. For this, we pick an orientation of the edges of G' , and add this to the certificate. Moreover, for every $v \in V(G)$, we include a function $g_v: E(G') \rightarrow \{-k, -k + 1, \dots, k - 1, k\}$ in the certificate. Now we only need the lengths $l(e)$ of the edges to be able to check the conditions.

This brings us back to the original problem: we do not know whether the number of subdivision is polynomial, hence, we do not know whether the lengths $l(e)$ are polynomial. But we do know that $D \sim D_v$ in H for all v if and only if there exist lengths $l(e)$ such that the conditions on fundamental cycles are satisfied. We make an integer linear program \mathcal{I} of these equations. So, for every vertex $v \in V(G)$ and fundamental cycle C in G' , we have an equation

$$\sum_{e \in C} g_v(e)l(e)\sigma_C(e) = 0.$$

Moreover, we have an inequality $l(e) \geq 1$ for every edge e . This program has a solution if and only if there is a subdivision H of G' such that all divisors D_v are equivalent to D in H . Since integer linear programming is in NP, we know that if there is a solution to \mathcal{I} , then there is a certificate \mathcal{D} for the ILP instance that is polynomial in the size of \mathcal{I} . Since \mathcal{I} is polynomial in the size of G' , this certificate is polynomial in the size of G' as well. In order to obtain a certificate for the STABLE DIVISORIAL GONALITY problem, we include this certificate for \mathcal{I} in the certificate for STABLE DIVISORIAL GONALITY.

To conclude, a certificate for the STABLE DIVISORIAL GONALITY problem consists of

- a subdivision G' of G ,
- an effective divisor D on G' ,
- an effective divisor D_v on G' with $D_v(v) \geq 1$, for every $v \in V(G)$,
- an orientation σ of the edges of G' ,
- a function g_v for every $v \in V(G)$, and
- a certificate for the integer linear program \mathcal{I} .

We can check the correctness of the certificate by checking whether the integer linear program \mathcal{I} is satisfied and whether for every $v \in V(G)$ and $w \in V(G')$ the equation

$$(D_v - D)(w) = \sum_{u \sim w} g_v(uw)\sigma_{wu}(uw)$$

holds. This can be done in polynomial time, since there is a polynomial number of equations to check.

5 Correctness

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

Lemma 5.1 *Let G be a graph with $\text{sdgon}(G) \leq k$. There exists a certificate \mathcal{C} .*

Proof Let n be the number of vertices of G and let m be the number of edges. By Lemma 4.1, there exists a subdivision G' , and effective divisors D and D_v on G' for every $v \in V(G)$, such that there is a subdivision H of G' such that $D \sim D_v$. For every edge $e \in E(G')$, let $l(e)$ be the number of edges that e is subdivided into in H .

Choose an arbitrary orientation σ of the edges of G' . Since $D \sim D_v$ for every vertex $v \in V(G)$, by Lemma 3.2 there exist functions $g_v: E(G') \rightarrow \mathbb{Z}$ such that for every $w \in V(G')$ holds $(D_v - D)(w) = \sum_{u \sim w} g_v(uw)\sigma_{wu}(uw)$ and for every fundamental cycle C holds $\sum_{e \in C} g_v(e)l(e)\sigma_C(e) = 0$. Now consider the integer linear program \mathcal{I} with a variable l_e for every edge $e \in E(G')$, and inequalities $l_e \geq 1$ and for every fundamental cycle C of G' equations $\sum_{e \in C} g_v(e)l_e\sigma_C(e) = 0$ for all $v \in V(G)$. Notice that the number of equations is polynomial, since there are $m - n + 1$ fundamental cycles. We know that $l(e)$ is a solution to this integer linear program. As ILP's have certificates with polynomially many bits (see e.g., [13]), there exists a polynomial certificate for this program. This yields a polynomial certificate for STABLE DIVISORIAL GONALITY. \square

We illustrate this proof with an example.

Example 5.2 Consider the graph in Fig. 1a. Consider the subdivision in Fig. 1b and the divisor D with 5 chips on u . This divisor is equivalent to the divisors D_v and D_w in Fig. 1a and f. The graph G' in the certificate is the graph in Fig. 1e.

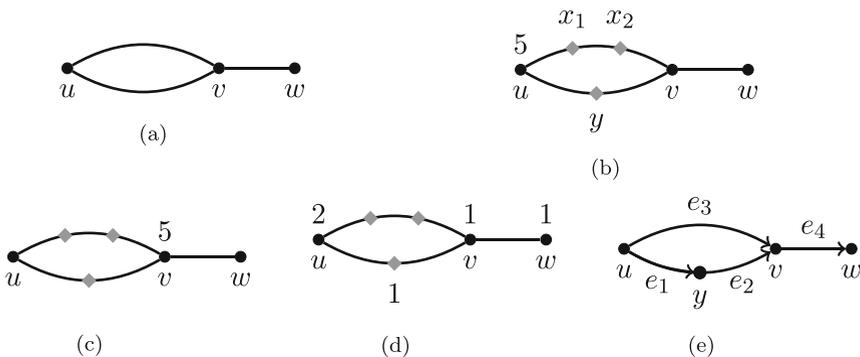


Fig. 1 (a) A graph G . (b) A subdivision of G and divisor $D = D_u$. (c) A divisor D_v . (d) A divisor D_w . (e) The graph G' .

Orient e_1 from u to y , e_2 from y to v , e_3 from u to v and e_4 from v to w . The corresponding functions are:

$$g_u(e_1) = 0, \quad g_u(e_2) = 0, \quad g_u(e_3) = 0, \quad g_u(e_4) = 0$$

$$g_v(e_1) = 3, \quad g_v(e_2) = 3, \quad g_v(e_3) = 2, \quad g_v(e_4) = 0$$

$$g_w(e_1) = 2, \quad g_w(e_2) = 1, \quad g_w(e_3) = 1, \quad g_w(e_4) = 1$$

The only cycle in G' is e_1, e_2, e_3 , so the integer linear program \mathcal{I} is:

$$\begin{aligned} l_{e_1} &\geq 1, & l_{e_2} &\geq 1 \\ l_{e_3} &\geq 1, & l_{e_4} &\geq 1 \\ 0 \cdot l_{e_1} + 0 \cdot l_{e_2} + 0 \cdot l_{e_3} &= 0 \\ 3 \cdot l_{e_1} + 3 \cdot l_{e_2} + -2 \cdot l_{e_3} &= 0 \\ 2 \cdot l_{e_1} + 1 \cdot l_{e_2} + -1 \cdot l_{e_3} &= 0 \end{aligned}$$

A solution to this program is $l_{e_1} = l_{e_2} = l_{e_4} = 1$ and $l_{e_3} = 3$. This completes the certificate.

Lemma 5.3 *Let G be a graph and k a natural number. If there exists a certificate \mathcal{C} , then $\text{sdgon}(G) \leq k$.*

Proof Suppose we are given a certificate. Let H be a the subdivision of G' where every edge e is subdivided into $l(e)$ edges. We know that all the equations $(D_v - D)(w) = \sum_{u \sim w} g_v(uw)\sigma_{wu}(uw)$ and $\sum_{e \in \mathcal{C}} g_v(e)l(e)\sigma_{\mathcal{C}}(e) = 0$ hold. By Lemma 3.2 it follows that $D \sim D_v$ in H for all vertices $v \in V(G)$. So D reaches all vertices of G .

Let $w \in V(H)$ be a vertex which is added to an edge uv of G . Either a chip is fired along e to go from D_u to D_v , or there is a divisor $D_{uv} \sim D$ with a chip on u and a chip on v . In both cases we see that D reaches w . So D reaches all vertices of H . We conclude that $\text{sdgon}(G) \leq k$. □

By Lemmas 5.1 and 5.3, 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 5.4 STABLE DIVISORIAL GONALITY *belongs to the class NP.*

Combined with the NP-hardness of STABLE DIVISORIAL GONALITY by Gijswijt et al. [9], this yields the following result.

Theorem 5.5 STABLE DIVISORIAL GONALITY *is NP-complete.*

6 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 6.1 (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 6.2 *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 H is obtained from G by subdividing at most $m^{O(mn)}$ times.*

Proof Let n' be the number of vertices in G' and m' the number of edges in G' . We have $n' \leq n + k(n + 1)$, and $m' \leq m + k(n + 1)$, with n the number of vertices of G and m the number of edges of G .

By Lemma 5.1, 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' . If we have an upper bound on the number of subdivisions per edge needed to obtain H from G' , say α , then $(k(n + 1) + 1)\alpha + k(n + 1)$ is an upper bound on the number of subdivisions per edge to obtain H from G . Applying Theorem 6.1 to the ILP gives such a bound, as described below.

The ILP has at most m' variables of the form l_e . The number of 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 $2m' = O(m + kn)$ variables. For every edge of G' we have an equation. In addition, for every vertex $v \in V(G)$ and every fundamental cycle of G' , we have an equation. So, the total number of equations is $m' + n(m - n + 1)$. Note that $m' + n(m - n + 1) = O(mn)$; as G is connected, $n \leq m + 1$ and $k \leq n$. Also, note that all values in matrix A and vector b are between $-k$ and k , i.e., we can set $a = k$ in the application of Theorem 6.1. So, by Theorem 6.1, we obtain that if there is a solution to the ILP, then there is one where all variables are set to values at most

$$O(m + kn) \cdot O(kmn)^{O(mn)} = m^{O(mn)}.$$

We know that there is at least one certificate with a solution, so we can bound the number of subdivisions of G' by $m^{O(mn)}$. This gives that the number of subdivisions per edge of G is at most

$$(k(n + 1) + 1)m^{O(mn)} + k(n + 1) = O(m^2)m^{O(mn)} + O(m^2) = m^{O(mn)}.$$

This results in $m \cdot m^{O(mn)} = m^{O(mn)}$ subdivisions in total. \square

7 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 et al. [9], 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?

Acknowledgments We thank Gunther Cornelissen and Nils Donselaar for helpful discussions. We thank an anonymous referee for the suggestion of a shorter and more elegant proof of the main result.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bodlaender, H.L., van der Wegen, M., van der Zanden, T.C.: Stable divisorial gonality is in NP [extended abstract]. In: Catania, B., Kráľovič, R., Nawrocki, J., Pighizzini, G. (eds.) SOFSEM 2019: Theory and practice of computer science, *SOFSEM 2019*, vol. 11376 of Springer Lecture Notes in Computer Science, pp. 81–93 (2019). Extended abstract of arXiv:1808.06921
2. Corry, S., Perkinson, D.: Divisors and sandpiles: An introduction to chip-firing. American Mathematical Society (2018)
3. 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>
4. 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>
5. Caporaso, L.: Gonality of algebraic curves and graphs. In: Algebraic and Complex Geometry, pp. 77–108. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-05404-9_4
6. 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>
7. Van Dobben de Bruyn, J., Gijswijt, D.: Treewidth is a lower bound on graph gonality. arXiv:1407.7055v2 (2014)

8. Bodewes, J.M., Bodlaender, H.L., Cornelissen, G., van der Wegen, M.: Recognizing hyperelliptic graphs in polynomial time. *Theor. Comput. Sci.* **815**, 121–146 (2020). <https://doi.org/10.1016/j.tcs.2020.02.013>
9. Gijswijt, D., Smit, H., van der Wegen, M.: Computing graph gonality is hard. *Discret. Appl. Math.* **287**, 134–149 (2020). <https://doi.org/10.1016/j.dam.2020.08.013>
10. Van Dobben de Bruyn, J.: Reduced divisors and gonality in finite graphs. Bachelor thesis, Leiden University. <https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/bachvandobbendebruyn.pdf> (2012)
11. Baker, M., Shokrieh, F.: Chip-firing games, potential theory on graphs, and spanning trees. *J. Comb. Theory Ser.* **120**(1), 164–182 (2013). <https://doi.org/10.1016/j.jcta.2012.07.011>
12. Bollobas, B.: *Modern graph theory*. Springer. <https://doi.org/10.1007/978-1-4612-0619-4> (1998)
13. Papadimitriou, C.H.: On the complexity of integer programming. *J. ACM* **28**(4), 765–768 (1981). <https://doi.org/10.1145/322276.322287>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.