

THE COMPLEXITY OF GEOMETRIC SCALING

ANTOINE DEZA

McMaster University, Hamilton, Ontario, Canada
e-mail: deza@mcmaster.ca

SEBASTIAN POKUTTA

Zuse Institute Berlin, Germany
e-mail: pokutta@zib.de

LIONEL POURNIN

Université Paris 13, Villetaneuse, France
e-mail: lionel.pournin@univ-paris13.fr

ABSTRACT

Geometric scaling, introduced by Schulz and Weismantel in 2002, solves the integer optimization problem $\max\{c \cdot x : x \in P \cap \mathbb{Z}^n\}$ by means of primal augmentations, where $P \subset \mathbb{R}^n$ is a polytope. We restrict ourselves to the important case when P is a 0/1-polytope. Schulz and Weismantel showed that no more than $O(n \log n \|c\|_\infty)$ calls to an augmentation oracle are required. This upper bound can be improved to $O(n \log \|c\|_\infty)$ using the early-stopping policy proposed in 2018 by Le Bodic, Pavelka, Pfetsch, and Pokutta. Considering both the maximum ratio augmentation variant of the method as well as its approximate version, we show that these upper bounds are essentially tight by maximizing over a n -dimensional simplex with vectors c such that $\|c\|_\infty$ is either n or 2^n .

1. Introduction

The computational performance of linear optimization algorithms is closely related to the geometric properties of the feasible region. The combinatorial properties can also play an important role, in particular for integer optimization algorithms. Starting with the Klee–Minty cubes [8] exhibiting an exponential number of simplex pivots, worst-case constructions have helped providing

a deeper understanding of how the structural properties of the input affect the performance of linear optimization. Recent examples include the construction of Allamigeon, Benchimol, Gaubert, and Joswig [1, 2] for which the primal-dual log-barrier interior point method performs an exponential number of iterations, and thus is not strongly polynomial. In a similar spirit, a lower bound on the number of simplex pivots required in the worst case to perform linear optimization on a lattice polytope has been recently established in [5, 6]. In turn, a pre-processing and scaling algorithm has been proposed by Del Pia and Michini [4] to construct simplex paths that are short relative to these lower bounds.

We focus on geometric scaling, an oracle based method introduced in [10] for integer optimization on 0/1-polytopes. Other classes of oracle based optimization methods are studied in [4, 7, 12]. No worst-case instances have been given for geometric scaling to the best of our knowledge. In contrast, a tight lower bound has been provided by Le Bodic, Pavelka, Pfetsch, and Pokutta [3] for bit scaling methods [11]. A 0/1-polytope P is the convex hull of a subset of the vertex set of the unit n -dimensional hypercube $[0, 1]^n$. Given a vector c in \mathbb{Z}^n , we are interested in the following optimization problem:

$$\max\{c \cdot x : x \in v(P)\},$$

where $v(P)$ denotes the vertex set of P .

In order to solve that problem, geometric scaling methods perform a sequence of steps that can be of two kinds: starting from a vertex \tilde{x} of P , an *augmentation step* returns a point that belongs to a well-defined subset $\mathcal{S}_P(\mu, \tilde{x})$ of the vertices of P such that $c \cdot x$ is greater than $c \cdot \tilde{x}$. The size of $\mathcal{S}_P(\mu, \tilde{x})$ is controlled by a parameter μ . Roughly, the larger μ , the smaller that subset is. When μ is very large, $\mathcal{S}_P(\mu, \tilde{x})$ may be empty and in that case, a *halving step* divides μ by 2 in order to enlarge $\mathcal{S}_P(\mu, \tilde{x})$. There are several variants of geometric scaling depending on which oracle is used to pick x within $\mathcal{S}_P(\mu, \tilde{x})$ and we will focus on two of them, *maximum-ratio augmentation* (MRA) based geometric scaling and *feasibility* based geometric scaling. We show the following.

THEOREM 1.1: *The maximum-ratio augmentation variant of geometric scaling can require $n + \log n \|c\|_\infty + 1$ steps to maximize $c \cdot x$ over P and the feasibility based variant can require $n/3 + \log n \|c\|_\infty + 1$ steps.*

Refined upper bounds on the complexity of geometric scaling will also be given using the early stopping policy from [3]. We will further highlight how

the chosen oracle contributes to the complexity of geometric scaling by studying the complexity of a variant of feasibility based geometric scaling where, instead of dividing μ by 2, halving steps divide μ by a positive parameter α .

We recall how geometric scaling works and describe its two variants in Section 2. We refer the reader to [3, 9, 10] for more comprehensive expositions. In Section 3, we show that maximum-ratio augmentation based geometric scaling can require $n + \log n \|c\|_\infty + 1$ steps and in Section 4 that feasibility based geometric scaling can require $n/3 + \log n \|c\|_\infty + 1$ steps. In Section 5, we highlighting the tradeoff between the chosen amount of scaling and the accuracy of the feasibility oracle used in the implementation by studying a generalization of feasibility based geometric scaling where halving steps divide μ by an arbitrary positive number α . Finally, we discuss upper bounds on the complexity of feasibility based geometric scaling in Section 6 and show that these upper bounds are largely dependent on the performance of the oracle.

2. Geometric scaling

In this section, we recall the setup and some key properties of the geometric scaling algorithm described in [3]. All the variants of geometric scaling are based on the general framework described by Algorithm 1. Given an initial vertex \tilde{x} of a 0/1-polytope P , this algorithm uses a certain oracle \mathcal{O} in order to return (in Line 3) another vertex x of P within the set

$$\mathcal{S}_P(\mu, \tilde{x}) = \left\{ x \in v(P) : c \cdot (x - \tilde{x}) > \mu \|x - \tilde{x}\|_1 \right\}.$$

It should be noted that $\mathcal{S}_P(\mu, \tilde{x})$ is a subset of the vertices x of P such that $c \cdot x$ is greater than $c \cdot \tilde{x}$. The extent of $\mathcal{S}_P(\mu, \tilde{x})$ is controlled by the parameter μ : the smaller μ is, the larger $\mathcal{S}_P(\mu, \tilde{x})$ gets and when μ is small enough then $\mathcal{S}_P(\mu, \tilde{x})$ is made up of all the vertices x of P such that $c \cdot x$ is greater than $c \cdot \tilde{x}$. If the oracle finds a point in $\mathcal{S}_P(\mu, \tilde{x})$, then \tilde{x} is replaced by this point (in Line 7) and the procedure repeats. This is referred to as an *augmenting step*. If however $\mathcal{S}_P(\mu, \tilde{x})$ is empty, it may either mean that μ is too large and prevents the algorithm to access to desirable vertices of P or that \tilde{x} is already optimal. In that case, the algorithm performs a *halving step*: it divides μ by 2 (in Line 5) and repeats. This goes on until μ is small enough to guarantee that $\mathcal{S}_P(\mu, \tilde{x})$ being empty implies the optimality of \tilde{x} . We refer the reader to [3] for a proof that the stopping criterion in Line 9 of Algorithm 1 implies optimality.

Algorithm 1: Geometric scaling

Input: a 0/1-polytope P contained in \mathbb{R}^n ,
 a vector c in \mathbb{Z}^n ,
 a vertex x^0 of P , and
 a number μ_0 greater than $\|c\|_\infty$.

Output: A vertex x^* of P that maximizes $c \cdot x$.

```

1  $\mu \leftarrow \mu_0, \tilde{x} \leftarrow x^0$ 
2 repeat
3   compute  $x$  in  $\mathcal{S}_P(\mu, \tilde{x})$  according to an oracle  $\mathcal{O}$ 
4   if  $\mathcal{S}_P(\mu, \tilde{x})$  is empty then
5      $\mu \leftarrow \mu/2$  (halving step)
6   else
7      $\tilde{x} \leftarrow x$  (augmenting step)
8   end
9 until  $\mu < 1/n$ ;
10 Return  $\tilde{x}$ 

```

In the remainder of the article, we will refer to a series of consecutive augmentation steps performed with same the value of μ as a *scaling phase*, and to a series of consecutive halving steps as an *halving phase*.

Let us turn our attention to the oracle \mathcal{O} used in Line 3 of Algorithm 1, which allows for several variants of that algorithm. In the following, we are especially interested in two variants. In the first variant, *maximum-ratio augmentation* (or for short MRA) based geometric scaling, the oracle \mathcal{O} in Line 3 of Algorithm 1 returns a point x in $\mathcal{S}_P(\mu, \tilde{x})$ such that the ratio

$$\frac{c \cdot (x - \tilde{x})}{\|x - \tilde{x}\|_1}$$

is maximal. In the second variant, *feasibility* based geometric scaling, the oracle \mathcal{O} in Line 3 outputs any feasible point x in $\mathcal{S}_P(\mu, \tilde{x})$.

The following remarks about geometric scaling hold for both the variants of Algorithm 1 that we consider here; for details we refer the interested reader to [3]. In particular, the combination of these two remarks provides a slightly differentiated picture on the complexity we study here.

Remark 2.1 ([3]): The sequence of points x^1, x^2, \dots, x^k generated by geometric scaling is monotone with respect to the vector c :

$$c \cdot x^1 < c \cdot x^2 < \dots$$

Note that this is very different from bit scaling, another augmentation-based optimization approach for 0/1-polytopes introduced in [11], where points can be revisited in successive scaling phases and the sequence of generated points is not strictly increasing with respect to the original objective c . This fact also impacts the structure of our lower bounds: for bit scaling it was shown in [3] that the number of required augmenting steps can depend on $\log \|c\|_\infty$ by making bit scaling revisit points. It will not be possible to do the same here and, in contrast to the bounds obtained for bit scaling, we will only be able to show that the total number of steps (the sum of the number of augmenting steps and the number of halving steps) depends on $\log \|c\|_\infty$. Our bounds for the number of required augmenting steps do not exceed n .

Remark 2.2 ([3]): Consider the value of μ taken before a halving step is performed. Either μ is equal to μ_0 and then, by definition this is a lower bound or μ arose from a previous halving step. In that halving iteration, before the actual halving, we had for some iterate \tilde{x} :

$$\max_{y \in v(P)} c \cdot (y - \tilde{x}) \leq \mu \|y - \tilde{x}\|_1 \leq \mu n.$$

The worst-case complexity in the number of total steps for geometric scaling on 0/1-polytopes is $O(n \log n \|c\|_\infty)$. The above two remarks allow to improve the worst-case complexity of geometric scaling slightly in the case of 0/1-polytopes as shown in [3]. Observe that geometric scaling requires $O(n \log \|c\|_\infty)$ iterations until $\mu \leq 1/2$. According to Remark 2.2, we know that

$$\max_{y \in v(P)} c \cdot (y - \tilde{x}) \leq \mu \|y - \tilde{x}\|_1 \leq 2\mu n \leq n$$

and by Remark 2.1, we know that each augmentation improves $c \cdot x$ by at least 1, so that the total number of iterations can be bounded as

$$O(n \log \|c\|_\infty + n) = O(n \log \|c\|_\infty)$$

iterations; we assume here that one would simply stop the algorithm after (at most) n additional steps and does not continue performing unnecessary halving steps as we are guaranteed to be optimal. In the following, we will refer to these

improved bounds as *early stopping* bounds. With this we obtain the following upper bounds that we compare against.

PROPOSITION 2.3 ([3]): *Given a 0/1-polytope P of dimension at most n and a vector c from \mathbb{Z}^n , geometric scaling solves*

$$\max_{x \in v(P)} c \cdot x$$

in no more than $O(n \log n \|c\|_\infty)$ steps using either variant of Algorithm 1 and no more than $O(n \log \|c\|_\infty)$ steps via early stopping.

In light of the above discussion, it follows from Proposition 2.3 that using the early stopping variants reduces the number of required halving steps, and thus the lower bounds, by the n term under log.

3. Worst-case instances for geometric scaling via MRA

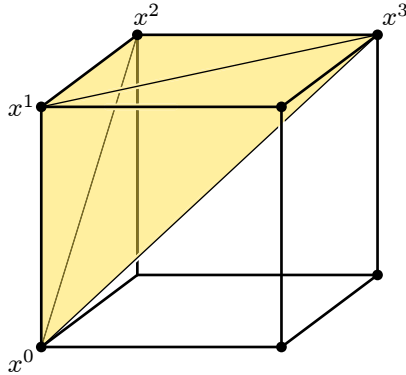


Figure 1. The simplex S when $n = 3$.

For any integer i such that $0 \leq i \leq n$, denote by x^i the point in \mathbb{R}^n whose last i coordinates are equal to 1 and whose other coordinates are equal to 0. Note that x^0 is the origin of \mathbb{R}^n . This point will be our initial vertex for MRA based geometric scaling. Recall that, with this variant of Algorithm 1, the point x computed in Line 3 is a point such that

$$c \cdot (x - \tilde{x}) / \|x - \tilde{x}\|_1$$

is maximal. Consider the n -dimensional simplex S , illustrated in Figure 1 in the special case when $n = 3$, whose vertices are the points x^0 to x^n . Further consider the vector c whose i th coordinate is i :

$$c = (1, 2, \dots, n).$$

In the remainder of the section, S and c are fixed as above, and we study how MRA based geometric scaling behaves when when P is equal to S .

LEMMA 3.1: *If, during the execution of MRA based geometric scaling on the simplex S , the point \tilde{x} is equal to x^i , then \tilde{x} is set to x^{i+1} by the next augmentation step, regardless of the value of μ .*

Proof. Let us compute the value of

$$(1) \quad \frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1}$$

where $j \neq i$. If j is less than i , then $x^j - x^i$ has no positive coordinate and at least one negative coordinate. As a consequence, $c \cdot (x^j - x^i)$ is negative, as well as the ratio (1). If j is greater than i , then

$$\frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} = \frac{1}{j - i} \sum_{k=i}^{j-1} c_{d-k},$$

where c_{d-k} is the $(d - k)$ th coordinate of c . As $c_{d-i} > c_{n-k}$ when $k > i$,

$$\frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} \leq c_{n-i},$$

with equality if and only if $j = i + 1$. In other words, when $j > i + 1$

$$\frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} < \frac{c \cdot (x^{i+1} - x^i)}{\|x^{i+1} - x^i\|_1}.$$

Therefore, if at the beginning of a step during the execution of MRA based geometric scaling, \tilde{x} is equal to x^i where $i < n$, then x will be set to x^{i+1} in Line 3, and the next augmentation will set \tilde{x} to x^{i+1} as announced. ■

THEOREM 3.2: *Starting at the origin of \mathbb{R}^n , MRA based geometric scaling requires n augmentation steps and $\log n \|c\|_\infty + 1$ halving steps in order to maximize $c \cdot x$ over S . With the early stopping policy, the number of required halving steps decreases to $\log \|c\|_\infty + 1$.*

Proof. Note that the optimal solution of the problem is x^n . According to Lemma 3.1, the algorithm performs n augmenting steps to reach x^n from x^0 . As a consequence, it suffices to observe that this algorithm performs at least $\log n \|c\|_\infty + 1$ halving steps in order to scale μ down to less than $1/n$. ■

4. Worst-case instances for feasibility based geometric scaling

Let us now consider feasibility based geometric scaling, the variant of Algorithm 1 that uses the feasibility based oracle. In that variant, the point x computed in Line 3 of Algorithm 1 can be any vertex of P that satisfies

$$c \cdot (x - \tilde{x}) > \mu \|x - \tilde{x}\|_1.$$

In particular, x is possibly not a maximizer of the ratio

$$\frac{c \cdot (x - \tilde{x})}{\mu \|x - \tilde{x}\|_1}.$$

We show that feasibility based geometric scaling can require

$$n/3 + \log n \|c\|_\infty + 1$$

steps to reach optimality. In order to do that, we will use the same simplex S as in Section 3, with vertices x^0 to x^n but a different vector c whose coordinates are exponential. More precisely, c is the vector whose i th coordinate is 2^i :

$$c = (2, 4, \dots, 2^n).$$

Note that, as in Section 3, we will start the algorithm at vertex x^0 .

LEMMA 4.1: *Assume that, at the start of a step during the execution of feasibility based geometric scaling on S , \tilde{x} is equal to x^i . If, in addition,*

$$\mu < c_{n-i} \leq 2\mu$$

then the step ends with an augmentation that sets \tilde{x} to x^{i+1} , x^{i+2} , or x^{i+3} .

Proof. We proceed as in the proof of Lemma 3.1 by computing

$$(2) \quad \frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1}$$

when $j \neq i$. If $j < i$, this ratio is negative because $x^j - x^i$ has at least one negative coordinate and none of its coordinates is positive. In particular, the

next augmentation cannot set \tilde{x} to x^j . Now assume that $j > i$. In this case,

$$\begin{aligned} \frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} &= \frac{1}{j-i} \sum_{k=i}^{j-1} c_{n-k}, \\ &= \frac{1}{j-i} \left(\sum_{k=i}^n 2^{n-k} - \sum_{k=j}^n 2^{n-k} \right), \\ &= \frac{2^{n-i+1} - 2^{n-j+1}}{j-i}, \\ &= 2^n \frac{2^{1-i} - 2^{1-j}}{j-i}. \end{aligned}$$

If in addition $\mu < c_{n-i} \leq 2\mu$, then

$$2^i \mu < 2^n \leq 2^{i+1} \mu.$$

As a consequence,

$$2 \frac{1 - 2^{i-j}}{j-i} \mu < \frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} \leq 4 \frac{1 - 2^{i-j}}{j-i} \mu.$$

As the ratio $(1 - 2^{-t})/t$ is less than $1/4$ when t belongs to $[4, +\infty[$, the step cannot end with an augmentation that sets \tilde{x} to x^j where $j \geq i + 4$. Now observe that this ratio is equal to $1/2$ when t is equal to 1. Hence,

$$\frac{c \cdot (x^{i+1} - x^i)}{\|x^{i+1} - x^i\|_1} > \mu.$$

This proves that the step will end by an augmentation that sets \tilde{x} to one of the vertices x^{i+1} , x^{i+2} , or x^{i+3} , as desired. ■

THEOREM 4.2: *Starting at the origin of \mathbb{R}^n , feasibility based geometric scaling requires $n/3$ augmentation steps and $\log n \|c\|_\infty + 1$ halving steps in order to maximize $c \cdot x$ over S . With the early stopping policy, the number of required halving steps decreases to $\log \|c\|_\infty + 1$.*

Proof. Observe again that the algorithm performs at least $\log n \|c\|_\infty + 1$ halving steps. Theorem 4.2 then follows from Lemma 4.1 and from the observation that, after a halving step where \tilde{x} is equal to x^i , either c_{n-i} is less than μ (in which case the next step is also a halving step) or satisfies $\mu < c_{n-i} \leq 2\mu$. ■

5. The tradeoff between scaling and oracle accuracy

In this section, we consider a generalization of feasibility based geometric scaling where, in Line 5 of Algorithm 1, μ is divided by α instead of by 2. This modified algorithm will be referred to as *generalized feasibility based geometric scaling*. Note that feasibility based geometric scaling is recovered simply by setting $\alpha = 2$. While μ is no longer halved, we still refer to this operation as a halving step. The parameter α controls the amount of both augmenting and halving steps performed by the algorithm. If α is close to 1, then only a small region is made feasible after each halving step. In this case, the feasibility oracle in Line 3 of Algorithm 1 has few choices for feasible solutions and its ability to find the best possible feasible point is not important. If, on the contrary α is large, then many new points will be feasible after each halving step. In fact, for large enough values of α , the algorithm will be completely *descaled* as all the vertices x of the polytope such that $c \cdot x$ is greater than $c \cdot \tilde{x}$ will be made feasible after the first halving step. In this case, the number of steps required to reach an optimal solution is completely determined by the ability of the feasibility oracle (called in Line 3 in Algorithm 1) to reach optimality. In other words, α also controls whether the complexity of the procedure is mainly due to the augmenting steps or to the accuracy of the feasibility oracle.

It turns out that α also explains the gap between the lower bounds provided by Theorems 3.2 and 4.2 on the complexity of geometric scaling. In particular, we will show how the term $n/3$ in the latter lower bound depends on α .

We consider, again, the same simplex S as in Sections 3 and 4 but use an objective vector whose i th coordinate is $\lceil \alpha \rceil^i$:

$$c = (\lceil \alpha \rceil, \lceil \alpha \rceil^2, \dots, \lceil \alpha \rceil^n).$$

LEMMA 5.1: *Assume that, at the start of some step during the execution of generalized feasibility based geometric scaling, \tilde{x} is equal to x^i . If in addition,*

$$\mu < c_{n-i} \leq \alpha \mu,$$

then that step ends with an augmentation that sets \tilde{x} to x^j where $j > i$ and

$$(3) \quad \alpha \lceil \alpha \rceil \frac{1 - \lceil \alpha \rceil^{i-j}}{j - i} > 1.$$

Proof. Let us compute the ratio

$$(4) \quad \frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1}$$

when $j \neq i$. As in the proof of Lemma 4.1, this ratio is negative when $j < i$. In that case, the next augmentation will not set \tilde{x} to x^j . If, on the contrary, $j > i$ then the same calculation as in the proof of Lemma 4.1 yields

$$\frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} = [\alpha]^n \frac{[\alpha]^{1-i} - [\alpha]^{1-j}}{j - i}.$$

Now assume that $\mu < c_{n-i} \leq \alpha\mu$. In that case,

$$[\alpha]^i \mu < [\alpha]^n \leq \alpha [\alpha]^i \mu,$$

and it immediately follows that

$$[\alpha] \frac{1 - [\alpha]^{i-j}}{j - i} \mu < \frac{c \cdot (x^j - x^i)}{\|x^j - x^i\|_1} \leq \alpha [\alpha] \frac{1 - [\alpha]^{i-j}}{j - i} \mu.$$

First observe that, when $j = i + 1$, the first inequality is

$$([\alpha] - 1)\mu < \frac{c \cdot (x^{i+1} - x^i)}{\|x^{i+1} - x^i\|_1}.$$

As $\alpha > 1$, it follows that the step will end by an augmentation. Moreover that augmentation can set \tilde{x} to x^{i+1} . Finally, if the augmentation sets \tilde{x} to x^j , then j must satisfy (3) by the second inequality. ■

Now denote by ω_α the number of integers t such that

$$\alpha [\alpha] \frac{1 - [\alpha]^{-t}}{t} > 1.$$

As already noted in the proof of Lemma 5.1, that inequality is always satisfied when $t = 1$ because $\alpha > 1$, and thus $\omega_\alpha \geq 1$. One can check that the first few values of ω_α are $\omega_\alpha = 1$ when

$$1 < \alpha \leq \frac{4}{3},$$

$\omega_\alpha = 2$ when

$$\frac{4}{3} < \alpha \leq \frac{12}{7},$$

and $\omega_\alpha = 3$ when

$$\frac{12}{7} < \alpha \leq 2.$$

Then, ω_α jumps to 6 when

$$2 < \alpha \leq \frac{729}{364}$$

because $\lceil \alpha \rceil$ is no longer equal to 2, but to 3. Further note that ω_α grows like α^2 when α goes to infinity.

THEOREM 5.2: *Starting at the origin of \mathbb{R}^n , generalized feasibility based geometric scaling requires n/ω_α augmentation steps and $\log n\|c\|_\infty + 1$ halving steps to maximize $c \cdot x$ over S . With the early stopping policy, the number of required halving steps decreases to $\log \|c\|_\infty + 1$.*

Proof. Recall that generalized feasibility based geometric scaling is identical to feasibility based geometric scaling, except that μ is divided by α in Line 5 of Algorithm 1. Therefore, it still performs $\log n\|c\|_\infty + 1$ halving steps. The theorem is then a consequence of Lemma 5.1. Indeed, as $\lceil \alpha \rceil \geq \alpha$, after a halving step where \tilde{x} is equal to x^i , either c_{n-i} is less than μ (in which case the next step is also an halving step) or satisfies $\mu < c_{n-i} \leq \alpha\mu$ (in which case the next step is an augmenting step) and in the latter case, it follows from Lemma 5.1 that at most ω_α vertices of S are feasible. ■

Note that Theorem 4.2 is the special case of Theorem 5.2 obtained when $\alpha = 2$. Indeed, in this case, ω_α is equal to 3 and, therefore at most three new vertices are made feasible after each halving step. However, choosing $\alpha = 4/3$ (or, in fact, any α satisfying $1 < \alpha \leq 4/3$) provides Corollary 5.3 because in that case, ω_α is only equal to 1. More precisely, just as MRA based geometric scaling requires n augmentation steps with the vector

$$c = (1, 2, \dots, n),$$

generalized feasibility based geometric scaling requires n augmentation steps in order to maximize $c \cdot x$ over S when α is equal to $4/3$ and

$$c = \left(\left\lceil \frac{4}{3} \right\rceil, \left\lceil \frac{4}{3} \right\rceil^2, \dots, \left\lceil \frac{4}{3} \right\rceil^n \right) = (2, 4, 8, \dots, 2^n).$$

COROLLARY 5.3: *If α is equal to $4/3$ and*

$$c = (2, 4, 8, \dots, 2^n),$$

then, starting at the origin of \mathbb{R}^n , generalized feasibility based geometric scaling requires n augmentation steps and $\log n \|c\|_\infty + 1$ halving steps to maximize $c \cdot x$ over S . With early stopping, only $\log \|c\|_\infty + 1$ halving steps are required.

6. A remark on upper bounds

It is shown in [3] that the number of augmentation and halving steps performed by feasibility based geometric scaling is always at most $O(n \log \|c\|_\infty)$. This bound relies on a result from [10] whereby the algorithm performs at most $O(n)$ augmentations between two consecutive halving steps. However, recall that with feasibility based geometric scaling, the oracle called at Line 3 in Algorithm 1 can pick any vertex x of P in $\mathcal{S}_P(\mu, \tilde{x})$. We show that in fact, the oracle can always pick x such that at most one augmentation is performed between any two consecutive halving steps.

LEMMA 6.1: *If at the beginning of a step during the execution of feasibility based geometric scaling, the set $\mathcal{S}_P(\mu, \tilde{x})$ is non-empty, then $\mathcal{S}_P(\mu, \tilde{x})$ contains a point x such that $\mathcal{S}_P(\mu, x)$ is empty.*

Proof. Assume that $\mathcal{S}_P(\mu, \tilde{x})$ is non-empty at the beginning of a step during the execution of feasibility based geometric scaling. It suffices to show that for any point x in $\mathcal{S}_P(\mu, \tilde{x})$ the set $\mathcal{S}_P(\mu, x)$ is contained in $\mathcal{S}_P(\mu, \tilde{x})$. Indeed, this implies that, if $\mathcal{S}_P(\mu, x)$ is non-empty, any of the points it contains could have been picked by the oracle instead of x . Since $\mathcal{S}_P(\mu, \tilde{x})$ is non-empty and $c \cdot y$ is greater than $c \cdot x$ for any point y in $\mathcal{S}_P(\mu, x)$, this shows that the oracle can always pick x in such a way that $\mathcal{S}_P(\mu, x)$ is empty.

For any point x in $\mathcal{S}_P(\mu, \tilde{x})$,

$$c \cdot (x - \tilde{x}) > \mu \|x - \tilde{x}\|_1$$

and for any point y in $\mathcal{S}_P(\mu, x)$,

$$c \cdot (y - x) > \mu \|y - x\|_1.$$

Summing these two equalities yields

$$(5) \quad c \cdot (y - \tilde{x}) > \mu (\|y - x\|_1 + \|x - \tilde{x}\|_1).$$

However, by the triangle inequality, the right-hand side of (5) is at least $\mu \|y - \tilde{x}\|_1$ and as a consequence, y belongs to $\mathcal{S}_P(\mu, \tilde{x})$, as desired. ■

Now recall that any variant of geometric scaling performs at most $\log \|c\|_\infty + 1$ halving steps. Hence, we get the following from Lemma 6.1.

THEOREM 6.2: *There always is an execution of feasibility based geometric scaling that performs at most $2 \log \|c\|_\infty + 2$ augmentation and halving steps.*

The gap between this bound and the $O(n \log \|c\|_\infty)$ bound from [3] illustrates the critical role of the oracle for geometric scaling algorithms.

Acknowledgments. We thank the 2021 HIM program Discrete Optimization during which part of this work was developed.

References

- [1] Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig, *Log-barrier interior point methods are not strongly polynomial*, SIAM Journal on Applied Algebra and Geometry **2** (2018), no. 1, 140–178.
- [2] Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig, *What tropical geometry tells us about the complexity of linear programming*, SIAM Review **63** (2021), no. 1, 123–164.
- [3] Pierre Le Bodic, Jeffrey W. Pavelka, Marc E. Pfetsch, and Sebastian Pokutta, *Solving MIPs via scaling-based augmentation*, Discrete Optimization **27** (2018), 1–25.
- [4] Alberto Del Pia and Carla Michini, *Short simplex paths in lattice polytopes*, Discrete & Computational Geometry **67** (2018), no. 2, 503–524.
- [5] Antoine Deza and Lionel Pournin, *Primitive point packing*, Mathematika **68** (2022), 979–1007.
- [6] Antoine Deza, Lionel Pournin and Noriyoshi Sukegawa, *The diameter of lattice zonotopes*, Proceedings of the American Mathematical Society **148** (2020), no. 8, 3507–3516.
- [7] András Frank and Éva Tardos, *An application of simultaneous diophantine approximation in combinatorial optimization*, Combinatorica **7** (1987), no. 1, 49–65.
- [8] Victor Klee and George J. Minty, *How good is the simplex algorithm?*, Inequalities III (Oved Shisha, ed.), Academic Press, New York, 1972, pp. 159–175.
- [9] Sebastian Pokutta, *Restarting algorithms: sometimes there is free lunch*, Lecture Notes in Computer Science **12296** (2020), 22–38.
- [10] Andreas S. Schulz and Robert Weismantel, *The complexity of generic primal algorithms for solving general integer programs*, Mathematics of Operations Research **27** (2002), no. 4, 681–692.
- [11] Andreas S. Schulz, Robert Weismantel and Günter M. Ziegler, *0/1-integer programming: Optimization and augmentation are equivalent*, European Symposium on Algorithms '95, Springer, 1995, pp. 473–483.
- [12] Abdelouahab Zaghroui, François Soumis and Issmail El Hallaoui, *Integral simplex using decomposition for the set partitioning problem*, Operations Research **62** (2014), no. 2, 435–449.