Improving Root Separation Bounds

Aaron Herman

Department of Mathematics
North Carolina State University
Raleigh, NC 27695 USA

Hoon Hong

Department of Mathematics
North Carolina State University
Raleigh, NC 27695 USA

Elias Tsigaridas

INRIA, Paris-Rocquencourt Center, POLSYS *Project*
Sorbonne Universités, UPMC Univ Paris 06, CNRS, INRIA, LIP6 UMR 7606
Paris, France

Abstract

Let f be a polynomial (or polynomial system) with all simple roots. The root separation of f is the minimum of the pair-wise distances between the complex roots. A root separation bound is a lower bound on the root separation. Finding a root separation bound is a fundamental problem, arising in numerous disciplines. We present two new root separation bounds: one univariate bound, and one multivariate bound. The new bounds improve on the old bounds in two ways:

(1) The new bounds are usually *significantly bigger* (hence better) than the previous bounds.

(2) The new bounds *scale correctly*, unlike the previous bounds.

Crucially, the new bounds are *not* harder to compute than the previous bounds.

Email addresses: aherman@ncsu.edu (Aaron Herman), hong@ncsu.edu (Hoon Hong), elias.tsigaridas@inria.fr (Elias Tsigaridas).

1. Introduction

In this paper we present improved root separation bounds. A root separation bound is a lower bound on the distances between the roots of a polynomial (or polynomial system). First, we introduce the notion of the root separation by an example.

Example 1. Let us consider the case of a single polynomial in a single variable. Let $f(x) = x^4 - 60x^3 + 1000x^2 - 8000x$. The roots of $f(x)$ are plotted in Figure 1. The lengths of the red line segments are the distances between the roots of $f(x)$. The root separation is the smallest of these distances. The root separation of $f(x)$ is $\sqrt{200}$, so any number less than or equal to $\sqrt{200}$ is a root separation bound. □

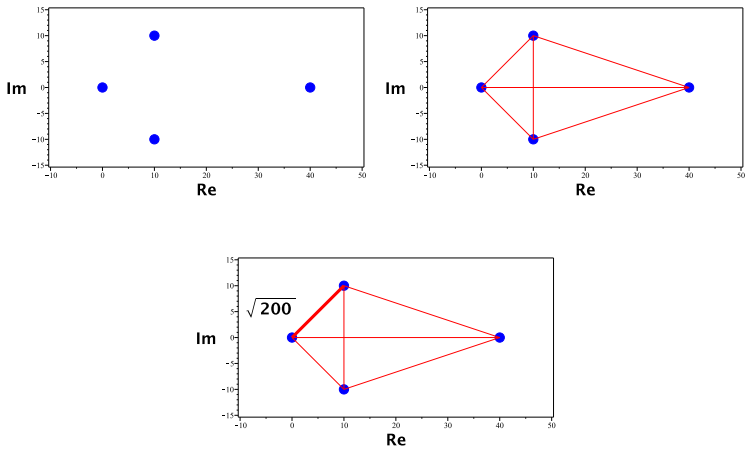

Fig. 1. Roots of $f(x)$ (top left), distances between roots (top right), minimum separation highlighted (bottom center)

Root separation bounds are fundamental tools in algorithmic mathematics, with numerous applications [11, 8, 3, 17, 19, 18]. As a consequence, there has been intensive effort in finding such bounds [12, 14, 16, 15, 19, 7, 4, 2], resulting in many important bounds. Unfortunately, it is well known that current bounds are very pessimistic. Furthermore, we have found another issue with current bounds. If the roots of a polynomial are doubled, the root separation is obviously doubled. Hence we naturally expect that a root separation bound would double if the roots are doubled. This does not happen: for the polynomial in the above example, the well known Mahler-Mignotte bound [12, 14] becomes *smaller* when the roots are doubled. If the roots are tripled, the Mahler-Mignotte bound is *even smaller*. In other words, the Mahler-Mignotte bound does not scale correctly; the bound is not compatible with the geometry of the roots. To the best of the authors' knowledge, the same observation holds for all *efficiently computable* root separation bounds [1]. We

[1] There do exist many bounds in the literature which scale correctly but are not efficiently computable. For example, the root separation bound which simply returns the exact root separation scales correctly with the roots. Other examples include the Mahler bound with the Mahler measure in the denominator (Theorem 2 of [12]) and several bounds due to Mignotte [15], all of which depend on the magnitudes of the roots.

elaborate further on this phenomena in the next section.

This discussion leads us to the following challenge: find new root separation bounds such that

(1) the new bounds are (almost always) less pessimistic than previous bounds,

(2) the new bounds scale correctly, and

(3) the new bounds can be computed as efficiently as previous bounds.

The main contribution of this paper is to provide two new bounds which meet the challenge: one univariate root separation bound, and one multivariate root separation bound. We found the new bounds by transforming known bounds into new bounds which meet the challenge. In the univariate case, we transform the celebrated bound of Mahler and Mignotte [12, 14]. In the multivariate case, we transform the DMM bound due to Emiris, Mourrain, and Tsigaridas [7]. Experimental evidence indicates that the improvement is usually very large, especially when the magnitudes of the roots are different from 1.

The structure of this paper is as follows. In Section 2 we elaborate on the challenge discussed above. In Section 3 we present the main contributions of this paper: one univariate root separation bound and one multivariate root separation bound, both of which meet the challenge. In Section 4 we derive the two new bounds. In Section 5 we discuss the experimental performance of the new bounds.

2. Challenge

In order to motivate our search for new root separation bounds, we recall the celebrated Mahler-Mignotte root separation bound [12, 14].

$$B_{MM}(f) \;\; = \frac{\sqrt{3|discr(f)|}}{d^{d/2+1} \;\; ||f||_2^{d-1}}$$

where $discr(f)$ is the discriminant [2] of f and d is the degree of f. Let us apply the Mahler-Mignotte bound to an example.

Example 2. Let $f(x) = x^4 - 60x^3 + 1000x^2 - 8000x$. As we saw in Example 1, the root separation of f is $\sqrt{200}$ (≈ 14.14). How does the Mahler-Mignotte bound perform on this polynomial? We have

$$|discr(f)| = 2.56 \times 10^{16}, \quad ||f||_2 = 8.06 \times 10^3$$

and the degree of f is 4. Combining these pieces, we have

$$B_{MM}(\, f(x)\,) \;=\; 8.26 \times 10^{-6}.$$

This bound is *significantly* smaller than the root separation of f. Actually it is smaller by several orders of magnitude!

[2] Recall that the discriminant can be calculated via the resultant: $discr(f) = (-1)^{\frac{1}{2}(d)(d-1)} \frac{1}{a_d} res(f, f')$, where a_d is the leading coefficient of f.

Now we consider the polynomial $f(x/2)$. Obviously, the root separation of $f(x/2)$ is twice the root separation of f. Hence we naturally expect that the Mahler-Mignotte bound of $f(x/2)$ is twice the Mahler-Mignotte bound of f. Let us see what happens:

$$B_{MM}(\, f(x/2)\,) \;=\; 1.05 \times 10^{-6}$$

It is not twice the Mahler-Mignotte bound of f. In fact, it is even *smaller* than the Mahler-Mignotte bound of f! This is very surprising. Maybe this is a peculiarity of our choice of 2. We will try scaling by a different number.

$$B_{MM}(\, f(x/3)\,) \;=\; 3.12 \times 10^{-7}$$

What happened? The Mahler-Mignotte bound of $f(x/3)$ is *even smaller* than the Mahler-Mignotte bound of $f(x/2)$. It appears that the Mahler-Mignotte bound is *decreasing* as we *increase* the distance between the roots. Can this be true? Lets calculate $B_{MM}(\, f(x/s)\,)$ for many different values of s and see. In Figure 2 we plot $B_{MM}(\, f(x/s)\,)$.

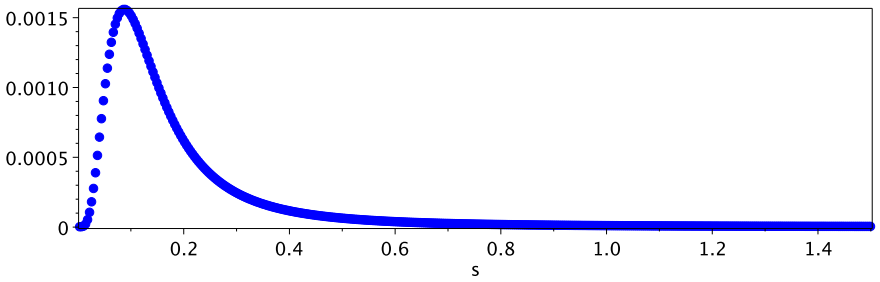

Fig. 2. $B_{MM}(\, f(x/s)\,)$

Unfortunately, our suspicions are correct. Look at $s = 1$, where $B_{MM}(\, f(x/1)\,)$ is simply the Mahler-Mignotte bound of f. To the right of $s = 1$, the function $B_{MM}(\, f(x/s)\,)$ is decreasing. In fact, the Mahler-Mignotte bound is approaching *zero* as the root separation increases. The situation is equally strange to the left of the Mahler-Mignotte bound of f. When we decrease s, we see that until s reaches a value around .18, the Mahler-Mignotte bound is increasing. In other words, the Mahler-Mignotte bound is *increasing* when the root separation is *decreasing*. This is very surprising and also undesirable. □

Let us summarize the observations from the above example.

(1) The Mahler-Mignotte bound is very pessimistic (several magnitudes smaller than the root separation).
(2) The Mahler-Mignotte bound does not scale correctly ("covariantly") with the roots of f.

We have also observed similar phenomena for other efficiently computable root separation bounds. Thus we have a challenge.

Challenge. Find a function $B : (\mathbb{C}\,[x_1, \ldots, x_n])^n \to \mathbb{R}_+$ such that

(1) $B(F)$ is a root separation bound.
(2) $B(F)$ is almost always larger (hence less pessimistic) than known root separation bounds.
(3) $B(F)$ scales covariantly.
(4) $B(F)$ can be computed as efficiently as previous bounds.

3. Main Results

The main results of this paper are two new root separation bounds: one univariate root separation bound and one multivariate root separation bound. The two new bounds meet the challenge posed in the previous section. In this section we will precisely state the main results of the paper. We use the following notation.

Notation 1.

$$
\begin{aligned}
f &= \textstyle\sum_{i=0}^{d} a_i x^i = a_d \prod_{i=1}^{d}(x-\alpha_i) \in \mathbb{C}[x] \\
\mathbb{F}_n &= \{F \in (\mathbb{C}[x_1,\ldots,x_n])^n : \ F \text{ has finitely many (at least two) solutions,} \\
&\qquad\qquad \text{and all solutions are simple.}\} \\
F &= (f_1,\ldots,f_n) \in \mathbb{F}_n \\
\Delta(F) &= \min_{\substack{\beta_1\neq\beta_2\in\mathbb{C}^n \\ F(\beta_1)=F(\beta_2)=0}} ||\beta_1-\beta_2||_2 \\
discr(f) &= a_d^{2d-2}\textstyle\prod_{i\neq j}(\alpha_i-\alpha_j) \\
E(f) &= Support(f) \\
d_i &= \deg(f_i) \\
D &= d_1\cdots d_n \\
M_i &= \textstyle\prod_{j\neq i} d_j
\end{aligned}
$$

Definition 1. A function $B : \mathbb{F}_n \to \mathbb{R}_+$ is a *root separation bound* if $B(F) \leq \Delta(F)$ for all $F \in \mathbb{F}_n$.

We begin by recalling two root separation bounds: a univariate bound due to Mahler and Mignotte [12, 14] [3]

$$B_{MM,k}(f) = \frac{\sqrt{|discr(f)|}}{||f||_k^{d-1}} P_k(d)$$

where

$$P_k(d) = \frac{\sqrt{3}}{d^{d/2+1}(d+1)^{(\frac{1}{2}-\frac{1}{k})(d-1)}}$$

and a multivariate bound due to Emiris, Mourrain, and Tsigaridas known as the Davenport-Mahler-Mignotte bound (or DMM bound) [7] [4]

$$B_{DMM}(F) = \frac{\sqrt{|discr(T_{f_0})|}}{\left(\prod_{i=1}^{n}||f_i||^{M_i}\right)^{D-1}} P(d_1,\ldots,d_n,n)$$

where

$$P(d_1,\ldots,d_n,n) = \frac{\sqrt{3}}{D^{D/2+1}\cdot n^{1/2} W \cdot \left(\sqrt{D+1}(n+1)^D W^D \prod_{i=1}^{n}\binom{d_i+n}{d_i}^{M_i}\right)^{D-1}}$$

[3] The Mahler-Mignotte bound is usually presented with the 2-norm (as in the previous section) or the ∞-norm. It can easily be extended to arbitrary k-norm ($k \geq 2$).

[4] We present a slight modification of the bound from [7]. See Lemma 8 for details.

$$T_{f_0} = \text{the resultant of } (f_0, f_1, \ldots, f_n) \text{ which eliminates } \{x_1, \ldots, x_n\}$$
$$f_0 = \text{a separating element in the set } \left\{u - x_1 - ix_2 - \cdots - i^{n-1}x_n : 0 \leq i \leq (n-1)\binom{D}{2}\right\}$$
$$W = \left((n-1)\binom{D}{2}\right)^{n-1}.$$

We are now ready to present the main contributions of this paper: a new univariate root separation bound, and a new multivariate root separation bound.

Definition 2 (New Univariate Bound)**.** Let $k \geq 2$. Define

$$B_{New,k}(f) = \frac{\sqrt{|discr(f)|}}{H_k^{d-1}} P_k(d)$$

where

$$H_k = \frac{\left\| \sum_{i=0}^{d} \tilde{s}_k^{d-i} a_i \cdot x^i \right\|_k}{\tilde{s_k}^{\frac{d}{2} - \frac{1}{d-1}}}$$
$$\tilde{s}_k = \max_{h(q)<0} \min_{h(p)>0} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}}$$
$$h(i) = \frac{d}{2} - i + \frac{1}{d-1}.$$

Theorem 1 (New Univariate Bound)**.** Let $k \geq 2$. Then
(1) $B_{New,k}$ is a root separation bound.
(2) If $k = \infty$, then $B_{New,k} \geq B_{MM,k}$ (when $k < \infty$, see the discussion in the following remark).
(3) $B_{New,k}$ scales covariantly.
(4) $\tilde{s}_k$ can be computed in $\mathcal{O}(d)$ algebraic operations [5] and comparisons using Algorithm 4.

Example 3. Let $f = x^4 - 60x^3 + 1000x^2 - 8000x$. Recall that the root separation of f is approximately 14.14. We have

$$B_{MM,\infty} = 7.56 \times 10^{-7}$$
$$\begin{aligned}
\tilde{s}_\infty &= \max_{q \in \{3,4\}} \min_{p \in \{1,2\}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}} \\
&= \max \left\{ \min \left\{ \left(\frac{|60|}{|8000|} \right)^{\frac{1}{3-1}}, \left(\frac{|60|}{|1000|} \right)^{\frac{1}{3-2}} \right\}, \min \left\{ \left(\frac{|1|}{|8000|} \right)^{\frac{1}{4-1}}, \left(\frac{|1|}{|1000|} \right)^{\frac{1}{4-2}} \right\} \right\} \\
&= \max \left\{ 6.00 \times 10^{-2}, 3.16 \times 10^{-2} \right\} \\
&= 6.00 \times 10^{-2}
\end{aligned}$$
$$H_\infty = \frac{||x^4 - 3.60x^3 + 3.60x^2 - 1.73x||_\infty}{(6.00 \times 10^{-2})^{\frac{4}{2} - \frac{1}{4-1}}}$$

[5] Arithmetic and radicals.

$$= \frac{3.60}{(6.00 \times 10^{-2})^{\frac{5}{3}}}$$
$$= 3.91 \times 10^{2}$$
$$B_{New,\infty} = 6.45 \times 10^{-3}.$$

Note that $B_{New,\infty}$ is a root separation bound for f, and is significantly larger than $B_{MM,\infty}$. To demonstrate the covariance, we plot the function $B_{New,\infty}(\, f(x/s)\,)$ in Figure 3. □

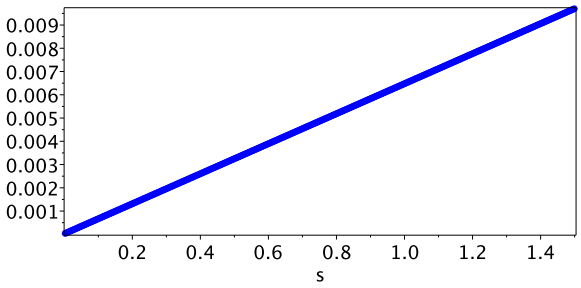

Fig. 3. Scaling covariance of $B_{New,\infty}$

Remark 1. Experimental evidence (presented in Section 5) indicates that $B_{New,k}$ is almost always larger than $B_{MM,k}$ for finite k. For example, with the same polynomial as in the preceding examples, we have

$$B_{New,2}(f) \;=\; 2.02 \times 10^{-2} \;\gg\; B_{MM,2}(f) \;=\; 8.26 \times 10^{-6}.$$

Furthermore, $B_{New,k}$ is almost always larger for *smaller* k, as the same example illustrates:

$$B_{New,2}(f) \;=\; 2.02 \times 10^{-2} \;>\; B_{New,\infty}(f) \;=\; 6.45 \times 10^{-3}.$$

In Section 5 we will provide theoretical justification for this observation.

However, it is not true that for finite k the new bound $B_{New,k}$ is *always* larger than $B_{MM,k}$ (*unlike* the case when $k = \infty$). As we will see later in the derivation of the bound, this is because the bound includes a certain approximation which becomes tighter as k increases. We can construct examples with $B_{New,k}(g) < B_{MM,k}(g)$. For example, let $k = 2$ and

$$g = x^4 - 3.844x^3 + 4.105x^2 - 2.104x.$$

Then [6]

$$B_{New,2}(f) = 1.29 \times 10^{-3}$$
$$B_{MM,2}(f) = 1.32 \times 10^{-3}.$$

Remark 2. For square-free integer polynomials, the discriminant has a lower bound of 1. Hence in practice the discriminant is almost always replaced by 1. In this case, part (4) of Theorem 1 implies that $B_{New,k}$ can be computed in $\mathcal{O}(d)$ algebraic operations and comparisons. Note that removing the discriminant sacrifices the scaling covariance. When the coefficients are not rational it is difficult to obtain a lower bound on the discriminant.

[6] A curious reader may wonder how this example was constructed. In the notation of Section 4.2, $g \approx f^{[s_2^*]}$, where f is the same polynomial we considered in the previous examples.

Remark 3. It is possible to replace $\tilde{s}_k$ by a number which is computed without using radicals if we allow ourselves to compute to arbitrary accuracy the real root of a polynomial. [7]

Definition 3 (New Multivariate Bound)**.** Define

$$B_{New}(F) = \frac{\sqrt{|discr(T_{f_0})|}}{H^{D-1}} P(d_1, \ldots, d_n, n)$$

where P, T_{f_0} and f_0 are from the definition of B_{DMM} and

$$H = \min_{s>0} R(s)$$
$$R(s) = \frac{\prod_{i=1}^{n} || \sum_{e \in E(f_i)} s^{d_i - |e|} |a_e| \, ||_\infty^{M_i}}{s^{\frac{D}{2} - \frac{1}{D-1}}}.$$

Theorem 2 (New Multivariate Bound)**.** We have

(1) B_{New} is a root separation bound.
(2) $B_{New} \geq B_{DMM}$.
(3) B_{New} scales covariantly.
(4) The minimizer of $R(s)$ can be computed in $\mathcal{O}(n \cdot m + n \cdot d)$ algebraic operations and comparisons using

$$FindMinimizer\left(F, (M_1, \ldots, M_n), \frac{D}{2} - \frac{1}{D-1}\right) \quad \text{(Algorithm 6)}$$

where

$$m = \#\text{ monomials of } F$$
$$d = \sum_{i=1}^{n} d_i.$$

Example 4. Let $F = (f_1, f_2)$, where

$$f_1 = x_1^2 + x_2^2 - 100$$
$$f_2 = x_2^2 - x_1^2 - 25.$$

It is simple to verify that the root separation of F is $\sqrt{150}$ (≈ 12.2). It is also simple to verify that

$$f_0 = u - x_1 - x_2$$

is a separating element in the set

$$\{u - x_1 - ix_2 : 0 \leq i \leq 6\}.$$

We compute

$$T_{f_0} = 4u^2 - 800u^2 + 2500$$
$$\sqrt{|discr(T_{f_0})|} = 2.40 \times 10^8$$

[7] The polynomial is obtained by replacing t with s^k in the definition of $Q_k(t)$ in Lemma 3.

$$
\begin{aligned}
P &= \sqrt{30}/48348866242924385372681011200 \\
||f_1||_\infty &= 100 \\
||f_2||_\infty &= 25 \\
||f_1||_\infty^2 ||f_2||_\infty^2 &= 6.25 \times 10^6.
\end{aligned}
$$

Hence

$$
B_{DMM}(F) \;=\; \frac{2.40 \times 10^8}{(6.25 \times 10^6)^4} \cdot \frac{\sqrt{3}}{48348866242924385372681011200} \;\approx\; 1.11 \times 10^{-40}.
$$

Now we compute H. We compute

$$
\begin{aligned}
s^* &= FindMinimizer(F, (2,2), \frac{4}{2} - \frac{1}{4-1}) \\
&= 1.00 \times 10^{-1}.
\end{aligned}
$$

Hence

$$
\begin{aligned}
H &= R(s^*) \\
&= 4.64 \times 10^1.
\end{aligned}
$$

Hence

$$
B_{New}(F) = \frac{2.40 \times 10^8}{(4.64 \times 10^1)^4} \cdot \frac{\sqrt{3}}{48348866242924385372681011200} \approx 2.71 \times 10^{-25}.
$$

Note that this number is still quite pessimistic; however, the new bound is significantly larger than $B_{DMM}(F)$. To demonstrate the covariance, we plot the function $B_{New}(\, F(x_1/s, x_2/s)\,)$ in Figure 4. $\square$

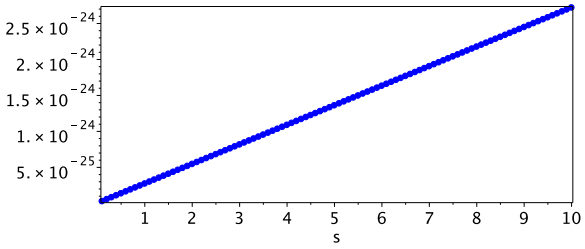

Fig. 4. Scaling covariance of B_{New}

Remark 4. Note that B_{New} is only defined for the $\infty-$norm. It turns out that generalizing the result to arbitrary norms is more difficult than in the univariate case.

Remark 5. For $F \in \mathbb{F}_n$ with integer coefficients, T_{f_0} is a square-free integer polynomial; in this case $discr(T_{f_0})$ has a lower bound of 1. Hence in practice the discriminant is almost always replaced by 1. In this case, part (4) of Theorem 2 implies that B_{New} can be computed in $\mathcal{O}\,(n \cdot m + n \cdot d)$ algebraic operations and comparisons. As with the new univariate bound, removing the discriminant sacrifices the scaling covariance. When the coefficients are not rational it is difficult to obtain a lower bound on the discriminant.

4. Derivation

4.1. Overall framework

In this section, we present the framework we use to derive the two new bounds. To make the presentation as general as possible, the framework will be derived for square polynomial systems. Of course, the univariate case is included by considering a square polynomial system which contains only one polynomial. We use the following notation.

Notation 2.
- $F^{[s]} = (f_1^{[s]}, \ldots, f_n^{[s]})$ where $f_i^{[s]} = s^{d_i} f_i(x_1/s, \ldots, x_n/s)$.

Note that in the above notation we scale the roots of F using a slight modification of the scaling operation in the introduction. Since the only difference between the two scaling operations are the leading coefficients, the two operations are equivalent. We use this scaling operation for later convenience.

In Propositions 1-3 we will incrementally develop the framework used to meet the challenge stated at the beginning of this paper.

Proposition 1 (Scaled Bound)**.** Let $B : \mathbb{F}_n \to \mathbb{R}_+$ be a root separation bound and $s \in \mathbb{R}_+$. Let

$$B^* : F \mapsto \frac{B(F^{[s]})}{s}.$$

Then

(1) B^* is a root separation bound.

We will illustrate the result by a simple example, since the proof is simple.

Example 5. Let $f(x) = x^4 - 60x^3 + 1000x^2 - 8000x$. We have

$$B_{MM,2}(\, f^{[2]}\,) \;=\; B_{MM,2}(\, 2^4 f(x/2)\,) \;=\; 1.05 \times 10^{-6}. \tag{1}$$

Since $B_{MM,2}$ is a root separation bound, it follows that

$$B_{MM,2}(\, f^{[2]}\,) \;\leq\; \Delta(\, f^{[2]}\,) \;=\; 2\,\Delta(f).$$

Rearranging yields

$$\frac{B_{MM,2}(\, f^{[2]}\,)}{2} \leq \Delta(f). \tag{2}$$

Combining (1) and (2) we have

$$\frac{1.05 \times 10^{-6}}{2} \;=\; 5.25 \times 10^{-7} \;\leq\; \Delta(f).$$

Note that $5.25 \times 10^{-7} \leq B_{MM,2}(f)$. So 2 was not a good choice for s.

How should we choose s? In Figure 5 we plot the function $B_{MM,2}(\, f^{[s]}\,)/s$. Clearly, we should choose s so that the function is maximized. We see that for $s \approx .06$, the new bound is approximately 2.00×10^{-2}. This new bound is significantly larger than $B_{MM,2}(f) = 8.26 \times 10^{-6}$. $\square$

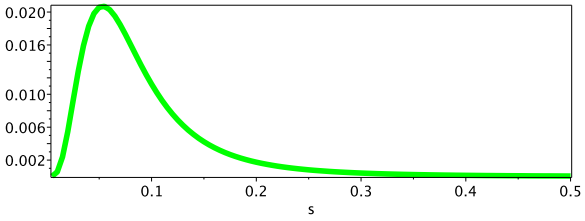

Fig. 5. Scaled bound for $B_{MM,2}$ and f.

Proposition 2 (Covariant Bound)**.** Let $B : \mathbb{F}_n \to \mathbb{R}_+$ be a root separation bound and $\sigma : \mathbb{F}_n \to \mathbb{R}_+$. Let

$$B^* : F \mapsto \frac{B(F^{[\sigma(F)]})}{\sigma(F)}.$$

If $\forall F \in \mathbb{F}_n$ and $\forall \gamma > 0$ we have

$$\sigma(F^{[\gamma]}) = \frac{1}{\gamma}\sigma(F)$$

then

(1) B^* is a root separation bound.
(2) B^* scales covariantly

Proof. The first property follows from Proposition 1.

We will now prove the second property. Let $F \in \mathbb{F}_n$ and $\gamma > 0$. By definition

$$B^*(F^{[\gamma)]}) \;=\; \frac{B\left(\left(F^{[\gamma]}\right)^{[\sigma(F^{[\gamma)]})]}\right)}{\sigma(F^{[\gamma)]})}.$$

Since $\sigma(F^{[\gamma]}) = \frac{1}{\gamma}\sigma(F)$, we have

$$\left(F^{[\gamma]}\right)^{[\sigma(F^{[\gamma]})]} = F^{[\gamma\sigma(F^{[\gamma]})]} = F^{[\gamma\cdot\frac{1}{\gamma}\cdot\sigma(F)]} = F^{[\sigma(F)]}. \tag{3}$$

Hence

$$\begin{aligned} B^*(F^{[\gamma]}) \;&=\; \frac{B\left(\left(F^{[\gamma]}\right)^{[\sigma(F^{[\gamma)]})]}\right)}{\sigma(F^{[\gamma)]})} \;=\; \frac{B(F^{[\sigma(F)]})}{\sigma(F^{[\gamma)]})} && \text{from (3)} \\ &=\; \frac{B(F^{[\sigma(F)]})}{\frac{1}{\gamma}\sigma(F)} \;=\; \gamma\frac{B(F^{[\sigma(F)]})}{\sigma(F)} \;=\; \gamma B^*(F). \end{aligned}$$

We have proved that B^* scales covariantly. □

Proposition 3 (Optimal Bound From Known Bound)**.** Let $B : \mathbb{F}_n \to \mathbb{R}_+$ be a root separation bound. Let

$$B^* : F \mapsto \max_{s>0} \frac{B(F^{[s]})}{s}.$$

Then

(1) B^* is a root separation bound.
(2) B^* scales covariantly

(3) $B^*(F) \geq B(F)$

Proof. The first property follows from Proposition 1.

To prove the second property, we will perform a rewrite and then make use of Proposition 2. Let

$$\sigma(f) = \arg\max_{s>0} \frac{B(F^{[s]})}{s}.$$

Clearly

$$B^* : F \mapsto \frac{B(F^{[\sigma(F)]})}{\sigma(F)}.$$

Let $F \in \mathbb{F}_n$ and $\gamma > 0$. We will show that $\sigma(F^{[\gamma]}) = \frac{1}{\gamma}\sigma(F)$. We have

$$\begin{aligned}
\sigma(F^{[\gamma]}) &= \arg\max_{s>0} \frac{B\left(\left(F^{[\gamma]}\right)^{[s]}\right)}{s} \\
&= \arg\max_{s>0} \frac{B\left(F^{[\gamma s]}\right)}{s} \\
&= \arg\max_{s>0} \frac{1}{\gamma} \frac{B\left(F^{[\gamma s]}\right)}{s} && \text{since } \gamma > 0 \\
&= \arg\max_{s>0} \frac{B\left(F^{[\gamma s]}\right)}{s\gamma} \\
&= \frac{1}{\gamma} \arg\max_{s>0} \frac{B\left(F^{[s]}\right)}{s} \\
&= \frac{1}{\gamma}\sigma(f).
\end{aligned}$$

Hence by Proposition 2, B^* scales covariantly.

We will now prove the third property. We have

$$B^*(F) = \max_{s>0} \frac{B(F^{[s]})}{s} \geq \frac{B(F^{(1)})}{1} = \frac{B(F)}{1} = B(F).$$

We have proved the Proposition. □

Let us summarize the framework built up in this section. We have seen that for a given root separation bound B

$$\max_{s>0} \frac{B(F^{[s]})}{s}$$

meets the challenge *if the maximum can be computed efficiently*. If the maximum cannot be computed efficiently, we can *approximate* the maximum. We can then use Proposition 2 to guarantee that the new bound is scaling covariant.

4.2. Derivation of New Univariate Bound

In this section we derive the new univariate bound. We will find a tight approximation $\tilde{s}_k$ of

$$s_k^* = \arg\max_{s>0} \frac{B_{MM,k}(\, f^{[s]}\,)}{s}.$$

We will then use Proposition 2 and a result due to Melhorn and Ray [13] to show that the bound

$$B_{New,k} = \frac{B_{MM,k}(\, f^{[\tilde{s}_k]}\,)}{\tilde{s}_k}$$

meets the challenge.

We will use our first few Lemmas to find a simplified expression for s_k^*. Our eventual goal is to find an expression for s_k^* that we can use to tightly approximate s_k^*. We will take advantage of the following easily verifiable identities:

Lemma 1. Let $g : \mathbb{R}_+ \to \mathbb{R}_+$, and $c > 0$. Then
(1) $\arg\max_{s>0}\ g(s) = \arg\max_{s>0}\ c \cdot g(s)$
(2) $\arg\max_{s>0}\ g(s) = \arg\max_{s>0}\ (g(s))^c$
(3) $\arg\max_{s>0}\ g(s) = (\arg\min_{s>0} g(s))^{-1}$.

As our first simplification step, we will find an expression for s_k^* which does not include the discriminant or $P_k(d)$.

Lemma 2. Let $f \in \mathbb{C}[x]$. Then

$$s_k^* = \arg\min_{s>0} R_k(s)$$

where

$$R_k(s) = \frac{||f^{[s]}||_k}{s^{\frac{d}{2}-\frac{1}{d-1}}}.$$

Proof. To prove the claim, we will expand the expression for

$$\frac{B_{MM,k}(\, f^{[s]}\,)}{s}$$

then simplify this expression with the identities of Lemma 1. We have

$$B_{MM,k}(\, f^{[s]}\,) = \frac{\sqrt{|discr(f^{[s]})|}}{||f^{[s]}||_k^{d-1}} P_k(d). \tag{4}$$

Since

$$f^{[s]} = s^d\, f(x/s) = s^d\, a_d \prod_{i=1}^{d}(x/s - \alpha_i) = a_d \prod_{i=1}^{d}(x - s\alpha_i)$$

we have

$$\begin{aligned} discr(f^{[s]}) &= a_d^{2d-2} \prod_{i\neq j}(s\alpha_i - s\alpha_j) \\ &= a_d^{2d-2} s^{d(d-1)} \prod_{i\neq j}(\alpha_i - \alpha_j) \end{aligned}$$

$$= s^{d(d-1)} discr(f). \tag{5}$$

Hence

$$\begin{aligned}
\frac{B_{MM,k}(f^{[s]})}{s} &= \frac{1}{s} \frac{\sqrt{|discr(f^{[s]})|}}{||f^{[s]}||_k^{d-1}} P_k(d) \\
&= \frac{1}{s} \frac{\sqrt{|s^{d(d-1)} discr(f)|}}{||f^{[s]}||_k^{d-1}} P_k(d) && \text{from (5)} \\
&= \frac{s^{\frac{d(d-1)}{2}} \sqrt{|discr(f)|}}{s||f^{[s]}||_k^{d-1}} P_k(d) \\
&= \frac{s^{\frac{d(d-1)}{2}-1}}{||f^{[s]}||_k^{d-1}} \sqrt{|discr(f)|} P_k(d) \\
&= \left(\frac{s^{\frac{d}{2}-\frac{1}{d-1}}}{||f^{[s]}||_k} \right)^{d-1} \sqrt{|discr(f)|} P_k(d) \\
&= \left(\frac{1}{R_k(s)} \right)^{d-1} \sqrt{|discr(f)|} P_k(d). && (6)
\end{aligned}$$

Now we apply the identities from Lemma 1 to the expression in (6):

$$\begin{aligned}
\arg\max_{s>0} \frac{B_{MM,k}(\, f^{[s]}\,)}{s} &= \arg\max_{s>0} \left(\frac{1}{R_k(s)} \right)^{d-1} \sqrt{|discr(f)|} P_k(d) \\
&= \arg\max_{s>0} \left(\frac{1}{R_k(s)} \right)^{d-1} && \text{(Identity 1)} \\
&= \arg\max_{s>0} \left(\frac{1}{R_k(s)} \right) && \text{(Identity 2)} \\
&= \arg\min_{s>0} R_k(s). && \text{(Identity 3)}
\end{aligned}$$

We have proved the Lemma. □

We will now find an even simpler expression for $\tilde{s}_k^*$ which depends only on the unique positive root of a polynomial with a particularly nice structure.

Lemma 3. Let $k \geq 2$. Then

$$s_k^* = (t^*)^{\frac{1}{k}}$$

where t^* is the unique positive root of

$$Q_k(t) = \sum_{i=0}^{d} h(i) \, |a_i|^k \cdot t^{d-i}$$

and $h(i) = \frac{d}{2} - i + \frac{1}{d-1}$.

Proof. For later convenience, we first rewrite $R_k(s)$. We will show that

$$R_k(s) = \tilde{R}_k(s)^{\frac{1}{k}}$$

where

$$\tilde{R}_k(s)^{\frac{1}{k}} = \sum_{i=0}^{d} (s^k)^{h(i)} \left|a_i\right|^k$$

Consider the following repeated rewriting:

$$\begin{aligned}
R_k(s) &= \frac{|| f^{[s]} ||_k}{s^{\frac{d}{2}-\frac{1}{d-1}}} \\
&= \frac{\left(\sum_{i=0}^{d} \left|s^{d-i}a_i\right|^k\right)^{\frac{1}{k}}}{s^{\frac{d}{2}-\frac{1}{d-1}}} \\
&= \frac{\left(\sum_{i=0}^{d} s^{kd-ki}\left|a_i\right|^k\right)^{\frac{1}{k}}}{s^{\frac{d}{2}-\frac{1}{d-1}}} \\
&= \left(\frac{\sum_{i=0}^{d} s^{kd-ki}\left|a_i\right|^k}{s^{\frac{kd}{2}-\frac{k}{d-1}}}\right)^{\frac{1}{k}} \\
&= \left(\sum_{i=0}^{d} s^{kd-ki-\left(\frac{kd}{2}-\frac{k}{d-1}\right)}\left|a_i\right|^k\right)^{\frac{1}{k}} \\
&= \left(\sum_{i=0}^{d} s^{\frac{kd}{2}-ki+\frac{k}{d-1}}\left|a_i\right|^k\right)^{\frac{1}{k}} \\
&= \left(\sum_{i=0}^{d} (s^k)^{\frac{d}{2}-i+\frac{1}{d-1}}\left|a_i\right|^k\right)^{\frac{1}{k}} \\
&= \left(\sum_{i=0}^{d} (s^k)^{h(i)}\left|a_i\right|^k\right)^{\frac{1}{k}} \qquad \text{since } h(i) = \frac{d}{2} - i + \frac{1}{d-1} \\
&= \tilde{R}_k(s)^{\frac{1}{k}} \qquad (7)
\end{aligned}$$

Combining Lemma 1 and (7), we have

$$s_k^* = \arg\min_{s>0} R_k(s) = \arg\min_{s>0} \tilde{R}_k(s). \qquad (8)$$

Hence from Calculus, we have

$$\tilde{R}_k'(s_k^*) = 0. \qquad (9)$$

Note that

$$\tilde{R}_k'(s) = \sum_{i=0}^{d} s^{kh(i)-1} \cdot kh(i)\left|a_i\right|^k.$$

Define the polynomial

$$Q_k(t) = \sum_{i=0}^{d} h(i)\left|a_i\right|^k \cdot t^{d-i}.$$

We have

$$ks^{\frac{-kd}{2}-\frac{k}{d-1}-1}Q_k(s^k) = s^{-\frac{kd}{2}-\frac{k}{d-1}-1}\sum_{i=0}^{d} kh(i)\left|a_i\right|^k \cdot (s^k)^{d-i}$$

$$
\begin{aligned}
&= s^{-\frac{kd}{2}-\frac{k}{d-1}-1} \sum_{i=0}^{d} kh(i)\,|a_i|^k \cdot (s^k)^{d-i} \\
&= \sum_{i=0}^{d} s^{kd-ki-\frac{kd}{2}-\frac{k}{d-1}-1} \cdot kh(i)\,|a_i|^k \\
&= \sum_{i=0}^{d} s^{\frac{kd}{2}-ki-\frac{k}{d-1}-1} \cdot kh(i)\,|a_i|^k \\
&= \sum_{i=0}^{d} s^{k(\frac{d}{2}-i-\frac{1}{d-1})-1} \cdot kh(i)\,|a_i|^k \\
&= \sum_{i=0}^{d} s^{kh(i)-1} \cdot kh(i)\,|a_i|^k \\
&= \tilde{R}_k'(s).
\end{aligned}
$$

Hence

$$\tilde{R}_k'(s) = 0 \iff Q_k(s^k) = 0 \quad \forall s > 0. \tag{10}$$

Note that $Q_k(t)$ has a single sign change [8], since $h(i)$ is strictly decreasing with i. By Descartes Rule of Signs, $Q_k(t)$ has a single positive root t^*. Combining (8), (9), and (10), we have

$$s_k^* = (t^*)^{\frac{1}{k}}.$$

We have proved the Lemma. □

Since Q_k is a polynomial with a single sign change, we can derive a tight approximation of its single positive root with the following result.

Theorem 3 (Herman, Hong, 2015 [10])**.** Let $f = \sum_{i=0}^{m} c_i x^{e_i}$ have a single sign change, and x^* be the unique positive root of f. Then

$$L \le x^* \le U$$

where

$$
\begin{aligned}
L &= \frac{1}{2}\,\mathcal{H}(f) \\
U &= 2\,\mathcal{H}(f) \\
\mathcal{H}(f) &= \max_{\substack{q \\ c_q<0}} \min_{\substack{p \\ c_p>0 \\ e_p>e_q}} \left(\frac{|c_q|}{|c_p|}\right)^{\frac{1}{e_p-e_q}}.
\end{aligned}
$$

We will now combine Theorem 3 and the definition of Q_k to approximate s_k^*.

Lemma 4. Let $k \ge 2$. Then

$$\left(\frac{1}{2}\right)^{\frac{1}{k}} (\mathcal{H}(Q_k))^{\frac{1}{k}} \le s_k^* \le 2^{\frac{1}{k}} (\mathcal{H}(Q_k))^{\frac{1}{k}}.$$

[8] The number of sign changes of a real polynomial is the number of times the signs of the the coefficients change from positive to negative, when the coefficients are ordered by degree.

Proof. From Lemma 3, we have

$$s_k^* = (t^*)^{\frac{1}{k}} \tag{11}$$

where t^* is the unique positive root of $Q_k(t)$. Since $Q_k(t)$ has single sign change, we can apply Theorem 3. We have

$$L \leq t^* \leq U \tag{12}$$

where

$$L = \frac{1}{2}\mathcal{H}(Q_k)$$
$$U = 2\mathcal{H}(Q_k).$$

Combining (11) and (12), we have

$$(L)^{\frac{1}{k}} \leq s_k^* \leq (U)^{\frac{1}{k}}.$$

Equivalently

$$\left(\frac{1}{2}\right)^{\frac{1}{k}} (\mathcal{H}(Q_k))^{\frac{1}{k}} \leq s_k^* \leq 2^{\frac{1}{k}} (\mathcal{H}(Q_k))^{\frac{1}{k}}.$$

We have proved the Lemma. □

Recall the definition of $\tilde{s}_k$ from Section 3:

$$\tilde{s}_k = \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}}$$

We will use the next two Lemmas to show that $\tilde{s}_k$ tightly approximates s_k^*. We split the Lemmas up for the sake of clarity.

Lemma 5. Let $k \geq 2$. We have

$$\tilde{s}_k = (\mathcal{H}(Q_k))^{\frac{1}{k}}$$

and

$$\lim_{k\to\infty} \tilde{s}_k = \mathcal{H}(G)$$

where

$$G = \sum_{\substack{p \\ h(p)>0}} \frac{1}{|a_p|} s^{d-p} - \sum_{\substack{q \\ h(q)<0}} \frac{1}{|a_q|} s^{d-q}.$$

Proof. We have

$$\begin{aligned}
(\mathcal{H}(Q_k))^{\frac{1}{k}} &= \left(\max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\frac{|h(q)|\,|a_q|^k}{|h(p)|\,|a_p|^k} \right)^{\frac{1}{(d-p)-(d-q)}} \right)^{\frac{1}{k}} \\
&= \left(\max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\frac{|h(q)|\,|a_q|^k}{|h(p)|\,|a_p|^k} \right)^{\frac{1}{q-p}} \right)^{\frac{1}{k}}
\end{aligned}$$

$$
\begin{aligned}
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\frac{|h(q)|\,|a_q|^k}{|h(p)|\,|a_p|^k} \right)^{\frac{1}{k(q-p)}} \\
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}} \\
&= \tilde{s}_k.
\end{aligned}
$$

We now consider the limit. We have

$$
\begin{aligned}
\lim_{k\to\infty} \tilde{s}_k &= \lim_{k\to\infty} \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}} \\
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}}.
\end{aligned}
$$

We also have

$$
\begin{aligned}
\mathcal{H}(G) &= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0 \\ (d-p)>(d-q)}} \left(\frac{\frac{1}{|a_q|}}{\frac{1}{|a_p|}} \right)^{\frac{1}{(d-p)-(d-q)}} \\
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0 \\ (d-p)>(d-q)}} \left(\frac{\frac{1}{|a_q|}}{\frac{1}{|a_p|}} \right)^{\frac{1}{q-p}} \\
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0 \\ (d-p)>(d-q)}} \left(\frac{|a_p|}{|a_q|} \right)^{\frac{1}{q-p}} \\
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0 \\ p<q}} \left(\frac{|a_p|}{|a_q|} \right)^{\frac{1}{q-p}} \\
&= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\frac{|a_p|}{|a_q|} \right)^{\frac{1}{q-p}} && \text{since } h(i) \text{ is strictly decreasing with } i \\
&= \lim_{k\to\infty} \tilde{s}_k.
\end{aligned}
$$

□

Lemma 6. Let $k \geq 2$ and

$$
s_k^* = \arg\max_{s>0} \frac{B_{MM,k}(\, f^{[s]}\,)}{s}.
$$

Then

$$
\left(\frac{1}{2}\right)^{\frac{1}{k}} \tilde{s}_k \;\leq\; s_k^* \;\leq\; 2^{\frac{1}{k}} \tilde{s}_k
$$

where

$$\tilde{s}_k = \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}}$$
$$h(i) = \frac{d}{2} - i + \frac{1}{d-1}.$$

Proof. Let $k \geq 2$ and

$$s_k^* = \arg\max_{s>0} \frac{B_{MM,k}(\, f^{[s]}\,)}{s}.$$

Combining Lemmas 2, 4 and 5, we have

$$\left(\frac{1}{2}\right)^{\frac{1}{k}} \tilde{s}_k \;\leq\; s_k^* \;\leq\; 2^{\frac{1}{k}} \tilde{s}_k.$$

We have proved the Lemma. □

We are now ready to define the new bound. In Lemma 6, we showed that $\tilde{s}_k$ is a tight approximation of s_k^*. As k increases, the approximation becomes tighter. Thus we choose to approximate the bound

$$\max_{s>0} \frac{B_{MM,k}(\, f^{[s]}\,)}{s}$$

with the bound

$$B_{New,k}(f) \;=\; \frac{B_{MM,k}(\, f^{[\tilde{s}_k]}\,)}{\tilde{s}_k} \;=\; \frac{\sqrt{|discr(f)|}}{H_k^{d-1}} P_k(d). \tag{13}$$

Before proving Theorem 1, we present an algorithm for computing $\tilde{s}_k$. We combine Lemma 5 and an ingenious algorithm due to Melhorn and Ray [13] to compute $\mathcal{H}(Q)$ in $\mathcal{O}(d)$ algebraic operations and comparisons. We formally state their complexity results in the Lemma below.

Lemma 7 (Melhorn, Ray, 2010 [13])**.** Let $g \in \mathbb{R}[x]$ with m non-zero coefficients. Then $\mathcal{H}(g)$ can be computed in $\mathcal{O}(m)$ algebraic operations and comparisons with the algorithm *Compute*$\mathcal{H}$ (Algorithm 3).

Algorithm 1: *LowerHullUpdate*

Input : $\mathcal{L}$ = a list of points which form a lower hull, sorted from left to right. $\mathcal{P}$ = a point to the left of L. $\mathcal{T}$ = a point in L. l = a line.
Output: $(\mathcal{L}', \mathcal{T}', l')$ where $\mathcal{L}'$ = the lower hull of $\mathcal{P} \cup \mathcal{L}$. $\mathcal{T}' = \mathcal{T}$ if $\mathcal{T} \in \mathcal{L}'$. Otherwise $\mathcal{T} = \mathcal{P}$. $l' = l$ if $\mathcal{T} \in \mathcal{L}'$. Otherwise l =the line from $\mathcal{P}$ to $(0, \infty)$.

```
1  begin
2      L' ← (P, L) [a];
3      T' ← T;
4      l' ← l;
5      P1, P2, P3 ← the first 3 elements of L';
6      while size(L') > 2 and S_{P1,P2} > S_{P2,P3} // A right hand turn is made
           on the path P1 → P2 → P3
7      do
8          Remove P2 from L';
9          if P2 = T then
10             T' ← P;
11             l' ←the line from P to (0, ∞);
12         P1, P2, P3 ← the first 3 elements of L;
13 end
```

[a] This notation should be read as "The list with $\mathcal{P}$ as its first element and the elements of $\mathcal{L}$ as the remaining elements".

Algorithm 2: *TangentPoint*

Input : $\mathcal{L}$ = a list of points which form a lower hull, sorted from left to right. $\mathcal{P}$ = a point to the left of $\mathcal{L}$. $\mathcal{T}$ = a point in $\mathcal{L}$.
Output: $\mathcal{T}'$: The tangent point of $\mathcal{P}$ and the points to the right of $\mathcal{T}$ in $\mathcal{L}$.

```
1  begin
2      T' ← T;
3      if T' is not the rightmost point in L then
4          Y ← the point to the right of T' in L;
5          while T' is not the rightmost point in L and S_{P,T'} > S_{P,Y} // The
               slope of the line from P to T' is greater than the slope
               of the line from P to Y
6          do
7              T' ← Y;
8  end
```

Algorithm 3: *Compute*$\mathcal{H}$

Input : $f = \sum_{i=0}^{d} a_i x^i \in \mathbb{R}[x]$
Output: $\mathcal{H}(f)$

begin
  $\mathcal{T} \leftarrow (d, |a_d|)$;
  $\mathcal{L} \leftarrow$ an empty list;
  $\mathcal{L} \leftarrow (\mathcal{T}, \mathcal{L})$;
  $l \leftarrow LineThrough(\ \mathcal{T},\ (0, \infty)\ )$;
  $\mathcal{H} \leftarrow -\infty$;
  for *i from* $d-1$ *to* 0 *by* -1 *do* **do**
    $\mathcal{P} \leftarrow (i, |a_i|)$;
    if a_i *is positive* **then**
      $(\mathcal{L}, \mathcal{T}, l) \leftarrow LowerHullUpdate(\mathcal{L}, \mathcal{P}, \mathcal{T}, l)$;
    else
      if $\mathcal{S}_{\mathcal{P},l[2]} < \mathcal{S}_{l[1],l[2]}$ `// ` $\mathcal{P}$ `lies below` l
      then
        $\mathcal{T} \leftarrow TangentPoint(L, \mathcal{P}, \mathcal{T})$;
        $l \leftarrow LineThrough(\ \mathcal{P},\ \mathcal{T})$;
        $\mathcal{H} \leftarrow \max\{\mathcal{H}, \mathcal{S}_{l[1],l[2]}\}$;
end

Algorithm 4: *Compute*$\tilde{s}$

Input : $f = \sum_{i=0}^{d} a_i x^i \in \mathbb{C}[x]$
  $k \geq 2$
Output: $\tilde{s}_k$

begin
  if k *is finite* **then**
    $Q \leftarrow \sum_{i=0}^{d} h(i)\,|a_i|^k \cdot t^{d-i}$;
    $\tilde{s} \leftarrow Compute\mathcal{H}(Q)^{\frac{1}{k}}$
  else
    $Q \leftarrow \sum\limits_{\substack{p \\ h(p)>0}} \frac{1}{|a_p|} s^{d-p} - \sum\limits_{\substack{q \\ h(q)<0}} \frac{1}{|a_q|} s^{d-q}$;
    $\tilde{s} \leftarrow Compute\mathcal{H}(Q)$
end

We slightly modify their algorithm to avoid logarithm computations:

- We represent points $(i, -\log(|a_i|))$ with the pair $(i, |a_i|)$.
- For points $\mathcal{P}_1$ and $\mathcal{P}_2$ represented by $(p_1, |a_{p_1}|)$ and $(p_2, |a_{p_2}|)$ respectively, let

$$\mathcal{S}_{\mathcal{P}_1,\mathcal{P}_2} = \left(\frac{|a_{p_2}|}{|a_{p_1}|}\right)^{\frac{1}{p_1 - p_2}}.$$

- For points $\mathcal{P}_1$ and $\mathcal{P}_2$ represented by $(p_1, |a_{p_1}|)$ and $(p_2, |a_{p_2}|)$ respectively, the line

from $\mathcal{P}_1$ to $\mathcal{P}_2$ is represented by

$$\left((p_1, |a_{p_1}|), (p_2, |a_{p_2}|) \right).$$

Remark 6. In [13], the point $\mathcal{T}$ and line l are not reset when $\mathcal{T}$ is removed from $\mathcal{L}$ (as we do in Algorithm 1). This appears to be a minor oversight which we correct here.

Proof of Theorem 1. We prove the claims of the Theorem one by one.

(1) Combine (13) and Proposition 1.

(2) From Lemma 6, we have

$$\tilde{s}_\infty = s^*_\infty.$$

Hence

$$\begin{aligned} B_{New,\infty}(f) &= \frac{B_{MM,\infty}(\, f^{(\tilde{s}_\infty)}\,)}{\tilde{s}_\infty} \\ &= \frac{B_{MM,\infty}(\, f^{(s^*_\infty)}\,)}{s^*_\infty} \\ &= \arg\max_{s>0} \frac{B_{MM,\infty}(\, f^{[s]}\,)}{s}. \end{aligned}$$

Hence by Proposition 3, $B_{New,\infty}(f) \geq B_{MM,\infty}(f)$ for all f.

(3) Let $\gamma > 0$. We have

$$\begin{aligned} \tilde{s}_k(f^{[\gamma]}) &= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{\gamma^{d-q}\,|a_q|}{\gamma^{d-p}\,|a_p|} \right)^{\frac{1}{(q-p)}} \\ &= \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \frac{1}{\gamma^{q-p}} \right)^{\frac{1}{(q-p)}} \\ &= \frac{1}{\gamma} \max_{\substack{q \\ h(q)<0}} \min_{\substack{p \\ h(p)>0}} \left(\left(\frac{|h(q)|}{|h(p)|} \right)^{\frac{1}{k}} \frac{|a_q|}{|a_p|} \right)^{\frac{1}{(q-p)}} \\ &= \frac{1}{\gamma} \tilde{s}_k. \end{aligned}$$

Hence by Proposition 2, $B_{New,k}$ scales covariantly.

(4) Combine Lemma 5, Lemma 7, and Algorithm 4.

We have completed the proof of Theorem 1. □

4.3. Derivation of New Multivariate Bound

In this section, we derive the new multivariate bound. We first briefly discuss the bound B_{DMM} presented in Section 3.

Lemma 8. Let $F \in \mathbb{F}_n$. Then $\Delta(F) \geq B_{DMM}(F)$.

Proof. To prove the result, we follow a proof almost identical to that in [7]. Instead of using the sparse resultant, we will use the multivariate resultant. Let f_0 be a separating

element and T_{f_0} be the resultant of F and f_0 which eliminates $x_1, \ldots, x_n$. We use the same coefficient bounds as in [7] to show that

$$||T_{f_0}||_\infty \leq \prod_{i=1}^{n} ||f_i||_\infty^{M_i} C^D (n+1)^D \prod_{i=1}^{n} \binom{n+d_i}{d_i}^{M_i} \tag{14}$$

From Equation (16) in [7] we have

$$\Delta(F) \geq \frac{\Delta(T_{f_0})}{n^{1/2} \cdot C}$$

Hence

$$\Delta(F) \geq \frac{B_{MM,\infty}(T_{f_0})}{n^{1/2} \cdot C} \tag{15}$$

Combining (14) and (15), we have

$$\Delta(F) \geq B_{DMM}(F)$$

□

For the remainder of this section, let $F \in \mathbb{F}_n$ be fixed, and f_0 a fixed separating element of F. Similar to the previous section, we will begin by deriving a simplified expression for

$$s^* = \arg\max_{s>0} \frac{B_{DMM}(F^{[s]})}{s}.$$

We first need to understand the effect that root scaling has on the discriminant of T_{f_0}. We make use of the following result from the proof of Proposition 5.8 of [5].

Lemma 9. Let F be zero-dimensional, have no solutions at infinity, and have no singular solutions. Let

$$f_0 = u + r_1 x_1 + \cdots + r_n x_n$$

and T_{f_0} be the resultant of (F, f_0) which eliminates $(x_1, \ldots, x_n)$. Then

$$T_{f_0} = C \prod_{\alpha \in V(F)} f_0(\alpha)$$

where

$$\begin{aligned} C &= Res(\widehat{F}) \\ \widehat{F} &= (\widehat{f_1}, \ldots, \widehat{f_n}) \\ \widehat{f_i} &= \sum_{\substack{e \in E(f_i) \\ |e| = d_i}} a_e \mathbf{x}^e. \end{aligned}$$

Lemma 10. Let $s > 0$. Let $T_{f_0}^{[s]}$ be the resultant of $F^{[s]}$ and f_0. Then

$$discr(T_{f_0}^{[s]}) = s^{D(D-1)} discr(T_{f_0}).$$

Proof. To prove the claim, we will first show that the leading coefficients of T_{f_0} and $T_{f_0}^{[s]}$ are the same. Then we will use the definition of the discriminant to complete the proof.

Let C be the leading coefficient of T_{f_0} and $C_{f_0}^{[s]}$ the leading coefficient of $T^{[s]}$. From Lemma 9, we have

$$C = Res(\widehat{F}) \text{ and } C^{[s]} = Res(\widehat{F}^{[s]}) \tag{16}$$

Note that

$$\begin{aligned}
\widehat{f_i^{[s]}} &= s^{d_i} f_i(\widehat{x_1/s, \ldots, x_n/s}) \\
&= s^{d_i} \sum_{|e|=d_i} a_e \left(\mathbf{x}^{[s]}\right)^e \\
&= s^{d_i} \sum_{|e|=d_i} a_e \left(\frac{x_1}{s}\right)^{e_1} \cdots \left(\frac{x_n}{s}\right)^{e_n} \\
&= s^{d_i} \sum_{|e|=d_i} \left(\frac{1}{s}\right)^{e_1+\cdots+e_n} a_e x_1^{e_1} \cdots x_n^{e_n} \\
&= s^{d_i} \sum_{|e|=d_i} \left(\frac{1}{s}\right)^{e_1+\cdots+e_n} a_e \mathbf{x}^e \\
&= s^{d_i} \sum_{|e|=d_i} \left(\frac{1}{s}\right)^{d_i} a_e \mathbf{x}^e \\
&= s^{d_i} \left(\frac{1}{s}\right)^{d_i} \sum_{|e|=d_i} a_e \mathbf{x}^e \\
&= \sum_{|e|=d_i} a_e \mathbf{x}^e \\
&= \widehat{f}_i.
\end{aligned}$$

Hence

$$\widehat{F} = \widehat{F^{[s]}}. \tag{17}$$

Combining (16) and (17), we have

$$C = C^{[s]}. \tag{18}$$

Note that the roots T_{f_0} are

$$\{r_1\gamma_{i,1} + \cdots + r_n\gamma_{i,n}\}_{i=1}^{D}$$

and the roots of $T_{f_0}^{[s]}$ are

$$\{s \cdot (r_1\gamma_{i,1} + \cdots + r_n\gamma_{i,n})\}_{i=1}^{D}.$$

We will now expand the discriminant of $T_{f_0}^{[s]}$. We have

$$\begin{aligned}
discr(T_{f_0}^{[s]}) &= \left(C^{[s]}\right)^{D(D-1)} \prod_{i\neq j} (s \cdot (r_1\gamma_{i,1} + \cdots + r_n\gamma_{i,n}) - s \cdot (r_1\gamma_{j,1} + \cdots + r_n\gamma_{j,n})) \\
&= C^{D(D-1)} \prod_{i\neq j} (s \cdot (r_1\gamma_{i,1} + \cdots + r_n\gamma_{i,n}) - s \cdot (r_1\gamma_{j,1} + \cdots + r_n\gamma_{j,n})) \qquad \text{from (18)} \\
&= s^{D(D-1)} C^{D(D-1)} \prod_{i\neq j} ((r_1\gamma_{i,1} + \cdots + r_n\gamma_{i,n}) - (r_1\gamma_{j,1} + \cdots + r_n\gamma_{j,n}))
\end{aligned}$$

$$= s^{D(D-1)} discr(T_{f_0}).$$

□

Now that we know the effect root scaling has on the discriminant of T_{f_0}, we can study the effect of root scaling on B_{DMM}. We will follow a similar procedure to the derivation of the univariate bound. First we will find an expression for the scaled bound which does not depend on the discriminant of T_{f_0} or $P(d_1, \ldots, d_n, n)$.

Lemma 11. Let $s > 0$. Then

$$\frac{B_{DMM}(F^{[s]})}{s} = \frac{\sqrt{|discr(T_{f_0})|}}{R(s)^{D-1}} P(d_1, \ldots, d_n, n)$$

where

$$R(s) = \frac{\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{M_i}}{s^{\frac{D}{2} - \frac{1}{D-1}}}.$$

Proof. We have

$$\begin{aligned}
\frac{B_{DMM}(F^{[s]})}{s} &= \frac{1}{s} \frac{\sqrt{|discr(T_{f_0}^{[s]})|}}{\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{M_i(D-1)}} P(d_1, \ldots, d_n, n) \\
&= \frac{1}{s} \frac{s^{\frac{D(D-1)}{2}} \sqrt{|discr(T_{f_0})|}}{\left(\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{M_i}\right)^{D-1}} P(d_1, \ldots, d_n, n) \qquad \text{from Lemma 10} \\
&= \frac{\sqrt{|discr(T_{f_0})|}}{\left(\frac{\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{M_i}}{s^{\frac{D}{2} - \frac{1}{D-1}}}\right)^{D-1}} P(d_1, \ldots, d_n, n) \\
&= \frac{\sqrt{|discr(T_{f_0})|}}{R(s)^{D-1}} P(d_1, \ldots, d_n, n).
\end{aligned}$$

□

Next, we find a simplified expression for s^*. As in the univariate case, our eventual goal is to find an expression for s^* which leads to an efficient computation of s^*.

Lemma 12. We have

$$s^* = \arg\min_{s>0} R(s).$$

Proof. To prove the claim, we will again make use of the identities in Lemma 1 . We have

$$\begin{aligned}
\arg\max_{s>0} \frac{B_{DMM}(F^{[s]})}{s} &= \arg\max_{s>0} \frac{\sqrt{|discr(T)|}}{R(s)^{D-1}} P(d_1, \ldots, d_n, n) \qquad \text{from Lemma 11} \\
&= \arg\max_{s>0} \frac{1}{R(s)^{D-1}} \qquad \text{(Identity 1)} \\
&= \arg\max_{s>0} \frac{1}{R(s)} \qquad \text{(Identity 2)} \\
&= \arg\min_{s>0} R(s) \qquad \text{(Identity 3)}
\end{aligned}$$

□

We will now consider the computation of $\arg\min_{s>0} R(s)$. For the sake of generality, we will study all functions of the form

$$R(s) = \frac{\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{U_i}}{s^V}$$

where $U_1, \ldots, U_n, V \in \mathbb{R}_{>0}$. Let $s^* = \arg\min_{s>0} R(s)$. We will show that s^* can be computed in $\mathcal{O}(\, n \cdot m + n \cdot d\,)$ algebraic operations and comparisons [9]. Our overall strategy will be to transform the problem into a new problem which is stated in terms of *linear* functions. More precisely, we will show that $\log(R(s))$ can be viewed as the *upper envelope* of a set of linear functions. We will make use of a technique for efficiently computing upper envelopes known as the *Convex Hull Trick* to compute s^* efficiently.

Lemma 13. Let $t = \log(s)$. We have

$$\log(R(s)) = \sum_{i=1}^{n} U_i \cdot \max_{e \in E(f_i)} (\, (d_i - |e|) \cdot t + \log(|a_e|)\,) - V \cdot t.$$

Proof. We have

$$\begin{aligned} \log(R(s)) &= \log\left(\frac{\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{U_i}}{s^V}\right) \\ &= \sum_{i=1}^{n} U_i \cdot \log(||f_i^{[s]}||_\infty) - V \cdot \log(s). \end{aligned} \tag{19}$$

Note that

$$\begin{aligned} \log(||f_i^{[s]}||) &= \log\left(\max_{e \in E(f_i)} s^{d_i - |e|}|a_e|\right) \\ &= \max_{e \in E(f_i)} \left(\log\left(s^{d_i-|e|}|a_e|\right)\right) \\ &= \max_{e \in E(f_i)} (\, (d_i - |e|) \cdot \log(s) + \log(|a_e|)\,) \\ &= \max_{e \in E(f_i)} (\, (d_i - |e|) \cdot t + \log(|a_e|)\,). \end{aligned} \tag{20}$$

Combining (19) and (20), we have

$$\log(R(s)) = \sum_{i=1}^{n} U_i \cdot \max_{e \in E(f_i)} (\, (d_i - |e|) \cdot t + \log(|a_e|)\,) - V \cdot t.$$

□

Since the sum of upper envelopes is an upper envelope, $\log(R(s))$ is an upper envelope. The upper envelope of a set of linear functions $l_i(t) = \beta_i \cdot t + \xi_i$ on $t > 0$ is represented by an ordered sequence $(l_{i_1}, 0), (l_{i_2}, t_{i_1,i_2}), \ldots, (l_{i_r}, t_{i_{r-1},i_r})$ such that

[9] Recall that $m = \#$ monomials of F and $d = \sum_{i=1}^{n} d_i$.

$$\max_i \; l_i(t) = \begin{cases} l_{i_1}(t) & -\infty \quad \leq t \leq t_{i_1,i_2} \\ l_{i_2}(t) & t_{i_1,i_2} \quad \leq t \leq t_{i_2,i_3} \\ \quad \vdots & \\ l_{i_r}(t) & t_{i_{r-1},i_r} \leq t \leq \infty \end{cases}$$

Given such a representation, finding the t which minimizes the upper envelope is trivial: we simply find the corner point t where the slopes of the lines in the upper envelope switch from negative to positive. In fact, this representation contains more information than is necessary to find the minimizer. We need only store the slopes of functions which lie on the upper envelope, as well as the corner points.

Hence we have the following initial strategy. For $i = 1, \ldots, n$, we compute the upper envelope representation of

$$\max_{e \in E(f_i)} \left(\, (d_i - |e|) \cdot t + \log(|a_e|) \, \right). \tag{21}$$

The most efficient algorithm for computing upper envelope representations of linear functions is known as the Convex Hull Trick. It is not clear who deserves credit for this trick; it appears to be folklore, not published in the literature. See [1] for a concise summary. We can combine the upper envelope representations to find the representation of

$$\log(R(s)) = \sum_{i=1}^{n} U_i \cdot \max_{e \in E(f_i)} \left(\, (d_i - |e|) \cdot t + \log(|a_e|) \, \right) - V \cdot t.$$

We then read off the minimizer t^* of $\log(R(s))$ and return

$$s^* = e^{t^*}.$$

We will now discuss improvements to the above strategy. Note that in the above strategy we must take logarithms. Recall that the current goal is to present an algorithm which produces the minimizer in

$$\mathcal{O}(n \cdot m + n \cdot d)$$

algebraic operations and comparisons. It turns out that it is a relatively trivial matter to modify the Convex Hull Trick algorithm to avoid logarithm computations for the current application. In the Convex Hull Trick algorithm, we compare corner points t_{i_1,i_2} and t_{i_3,i_4}. In our case, the corner points for the upper envelope of (21) are the points where

$$(d_i - |e_1|) \cdot t + \log(|a_{e_1}|) = (d_i - |e_2|) \cdot t + \log(|a_{e_2}|).$$

The above equality holds if and only if

$$t = \frac{\log(|a_{e_1}|) - \log(|a_{e_2}|)}{|e_1| - |e_2|} = \log\left(\left(\frac{|a_{e_1}|}{|a_{e_2}|} \right)^{\frac{1}{|e_1| - |e_2|}} \right).$$

Clearly,

$$\log\left(\left(\frac{|a_{e_1}|}{|a_{e_2}|} \right)^{\frac{1}{|e_1| - |e_2|}} \right) \leq \log\left(\left(\frac{|a_{e_3}|}{|a_{e_4}|} \right)^{\frac{1}{|e_3| - |e_4|}} \right) \iff \left(\frac{|a_{e_1}|}{|a_{e_2}|} \right)^{\frac{1}{|e_1| - |e_2|}} \leq \left(\frac{|a_{e_3}|}{|a_{e_4}|} \right)^{\frac{1}{|e_3| - |e_4|}}.$$

We can use this equivalence to perform all of the necessary comparisons in the Convex Hull Trick algorithm without computing any logarithms.

It is also possible to speed up the computation of the upper envelope representations by making use of the following Lemma.

Lemma 14. Let $s > 0$. Then

$$||f^{[s]}||_\infty = \max_{0 \le k \le \deg(f)} s^{d-k} \cdot b_k$$

where

$$\begin{aligned} d &= \deg(f) \\ b_k &= \max_{\substack{e \in E(f) \\ |e|=k}} |a_e|. \end{aligned}$$

Proof. Note that

$$f^{[s]} = s^d f(x_1/s, \ldots, x_n/s) = s^d \cdot \sum_{e \in E(f)} a_e \left(\frac{x_1}{s}\right)^{e_1} \left(\frac{x_2}{s}\right)^{e_2} \cdots \left(\frac{x_1}{s}\right)^{e_n} = \sum_{e \in E(f)} s^{d-|e|} a_e \cdot x^e.$$

Hence

$$\begin{aligned} ||f^{[s]}||_\infty &= \max_{e \in E(f)} s^{d-|e|} |a_e| \\ &= \max_{0 \le k \le d} \left\{ \max_{\substack{e \in E(f) \\ |e|=k}} s^{d-|e|} |a_e| \right\} \\ &= \max_{0 \le k \le d} \left\{ \max_{\substack{e \in E(f) \\ |e|=k}} s^{d-k} |a_e| \right\} \\ &= \max_{0 \le k \le d} s^{d-k} \left\{ \max_{\substack{e \in E(f) \\ |e|=k}} |a_e| \right\} \\ &= \max_{0 \le k \le d} s^{d-k} \cdot b_k. \end{aligned}$$

□

We are now ready to present *FindMinimizer* (Algorithm 6). For each f_i, we first find the coefficient of largest magnitude for each total degree (motivated by Lemma 14). We then use the sub-algorithm *UpperEnvelopeSlopes* (Algorithm 5) to compute the slopes of the lines which lie on the upper envelope of $\log(||f_i^{[s]}||_\infty)$, as well as the points s_{e_i,e_j} such that $t_{e_i,e_j} = \log(s_{e_i,e_j})$ is a corner point of the upper envelope. *UpperEnvelopeSlopes* is a straightforward modification of the Convex Hull Trick algorithm. Once the upper envelope slopes are computed for each $\log(||f_i^{[s]}||_\infty)$, we search for the smallest s such that the slope of $\log(R)$ is positive for $t > \log(s)$.

We are now ready to discuss the complexity of *FindMinimizer*.

Lemma 15. Let $U_1, \ldots, U_n, V \in \mathbb{R}_{>0}$ and

$$R(s) = \frac{\prod_{i=1}^n ||f_i^{[s]}||_\infty^{U_i}}{s^V}.$$

Algorithm 5: *UpperEnvelopeSlopes*

Input : $L = [l_1, \ldots, l_r]$ where
$l_i(t) = \beta_i \cdot t + \log(\xi_i)$
$\xi_i > 0$ for all i
$0 \le \beta_1 < \beta_2 < \ldots \beta_r$
$l_i(t)$ is represented by (β_i, ξ_i)

Output: M: an ordered list $[(\beta_{i_1}, 0), (\beta_{i_2}, s_{i_1,i_2}), \ldots, (\beta_{i_r}, s_{i_{r-1},i_r})]$ such that

$$\max_i \; l_i(t) = \begin{cases} \beta_{i_1} \cdot t + \log(\xi_{i_1}) & -\infty \le t \le t_{i_1,i_2} \\ \beta_{i_2} \cdot t + \log(\xi_{i_2}) & t_{i_1,i_2} \le t \le t_{i_2,i_3} \\ ; \;\; \vdots & \\ \beta_{i_r} \cdot t + \log(\xi_{i_r}) & t_{i_{r-1},i_r} \le t \le \infty \end{cases}$$

where $t_{i_j,i_k} = \log(s_{i_j,i_k})$.

1 **begin**
  // L will store the indices of the linear functions which lie on the upper envelope in the order in which they appear. We construct L using a slight modification of the Convex Hull Trick algorithm.
2 $L \leftarrow [1]$;
3 **for** i *from* 2 *to* r **do**
4 Append i to L;
5 **while** $size(L) > 2$ *and* $\left(\frac{\xi_{L[size(L)-1]}}{\xi_{L[size(L)]}}\right)^{\frac{1}{\beta_{L[size(L)]} - \beta_{L[size(L)-1]}}} < \left(\frac{\xi_{L[size(L)-2]}}{\xi_{L[size(L)-1]}}\right)^{\frac{1}{\beta_{L[size(L)-1]} - \beta_{L[size(L)-2]}}}$ **do**
6 Remove $L[size(L) - 1]$ from L;
7 $M \leftarrow [(\beta_{L[1]}, 0)]$;
8 **for** i *from* 2 *to* $size(L)$ **do**
9 Append $\left(\beta_{L[i]}, \left(\frac{\xi_{L[i-1]}}{\xi_{L[i]}}\right)^{\frac{1}{\beta_{L[i]} - \beta_{L[i-1]}}}\right)$ to M;
10 **end**

Then

$$\arg\min_{s>0} R(s)$$

can be computed in $\mathcal{O}(\, n \cdot m + n \cdot d)$ algebraic operations and comparisons, where

$$m = \#\text{ monomials of } F$$
$$d = \sum_{i=1}^{n} d_i.$$

Proof. We consider the total time spent on each line of $FindMinimizer$.

In Line 3, we compute

$$L_i \leftarrow [((d_i - k), \;\; 0), \quad k = 0, \ldots, d_i]$$

Algorithm 6: *FindMinimizer*

Input : F, U, V

Output: $s^* = \arg\min_{s>0} \frac{\prod_{i=1}^{n} ||f_i^{[s]}||_\infty^{U_i}}{s^V}$

1 **begin**
2 **for** i *from* 1 *to* n *do* **do**
3 $L_i \leftarrow [((d_i - k),\ 0), \quad k = 0, \ldots, d_i]$ `// Lines are represented by` (`slope`, $e^{intercept}$);
4 **for** $e \in E(f_i)$ **do**
5 **if** $L_i[|e|][2] < |a_e|$ **then**
6 $L_i[|e|][2] = |a_e|$ `// Find the largest magnitude coefficient for each degree (Lemma 14)`;
7 $Z_i \leftarrow UpperEnvelopeSlopes(L_i)$;
8 $M \leftarrow$ the list of triples (β, i, s), sorted in ascending order with respect to s, where (β, s) is an element of Z_i;
  `// Search for the first` s `where` $\log(R)$ `has positive slope after` $\log(s)$:
9 $C \leftarrow [0, \ldots, 0]$`//` $C[i]$ `stores the slope of` $\log(||f_i^{[s]}||_\infty)$;
10 **for** m *in* M **do**
11 $C[m[2]] = m[1]$`// Update the slope for` $\log(||f_i^{[s]}||_\infty)$;
12 $\alpha \leftarrow U_1 \cdot C[1] + \cdots + U_n \cdot C[n] - V$`// Calculate the slope of` $\log(R)$ `for` t `immediately after` $\log(s) = \log(m[3])$;
13 **if** $\alpha > 0$ **then**
14 **return** $m[3]$;
15 **end**

which requires a total of $\mathcal{O}(\sum_{i=0}^{n} d_i)$ algebraic operations.

In Lines 5 and 6 we check and potentially update the entry $L_i[|e|][2]$. This is done for every $e \in E(f_i)$. Since the computation of $|e|$ requires $\mathcal{O}(n)$ algebraic operations, the number of algebraic operations in lines $5-6$ is $\mathcal{O}(n \cdot \sum_{i=1}^{n} \#E(f_i)) = \mathcal{O}(n \cdot m)$.

In Line 7 we compute

$$Z_i \leftarrow UpperEnvelopeSlopes(L_i).$$

It is straightforward to see that *UpperEnvelopeSlopes* requires $\mathcal{O}(r)$ algebraic operations and comparisons when r linear functions are input. Since L_i has $\mathcal{O}(d_i)$ elements, line 7 requires $\mathcal{O}(d_i)$ algebraic operations and comparisons. Hence the total amount of work performed in Line 9 is $\mathcal{O}(\sum_{i=1}^{n} d_i)$.

In Line 8 we compute

$$M \leftarrow \text{the list of triples } (\beta, i, s), \text{ sorted in ascending order with respect to } s \text{ where } (\beta, s) \text{ is an element of } Z_i.$$

Note that every list Z_i is already sorted in ascending order with respect to s, and Z_i has $\mathcal{O}(d_i)$ elements. Hence constructing M requires $\mathcal{O}(n \cdot \sum_{i=1}^{n} d_i)$ algebraic operations and comparisons.

Line 9 can clearly be computed in a constant number of algebraic operations.

In the remainder of the algorithm, we potentially loop over all $\mathcal{O}(\sum_{i=1}^{n} d_i)$ elements of M. Lines 11 and 13 both require a constant number of algebraic operations and comparisons. Line 12 requires $\mathcal{O}(n)$ algebraic operations. Hence the total number of algebraic operations and comparisons performed in lines $10-14$ is $\mathcal{O}(n \cdot \sum_{i=1}^{n} d_i)$.

Combining all of the above, the total number of algebraic operations and comparisons required to compute $FindMinimizer(F, U, V)$ is

$$\mathcal{O}\left(n \cdot \sum_{i=1}^{n} \#E(f_i) + n \cdot \sum_{i=1}^{n} d_i\right) = \mathcal{O}\left(n \cdot m + n \cdot d\right).$$

□

We are now ready to prove Theorem 2.

Proof of Theorem 2. Note that

$$\begin{aligned} B_{New}(F) &= \frac{\sqrt{|discr(T_{f_0})|}}{R(s^*)^{D-1}} P(d_1, \ldots, d_n, n) \\ &= \frac{\sqrt{|discr(T_{f_0})|}}{\left(\arg\min_{s>0} R(s)\right)^{D-1}} P(d_1, \ldots, d_n, n) \\ &= \max_{s>0} \frac{B_{DMM}(F^{[s]})}{s}. \qquad \text{from Lemma 11} \end{aligned}$$

Hence parts 1, 2 and 3 of the Theorem follow immediately from Proposition 3. The fourth part follows from Lemmas 11 and 15. □

5. Performance

In this Section, we discuss the experimental performance of the new bounds. We first repeat the observation of Remark 1: experimental evidence indicates that $B_{New,k}$ is almost always *larger* for *smaller* k. This is unsurprising once we consider the derivation strategy in the previous section. Let $k_1 \leq k_2$. We have

$$B_{New,k_1} \approx \max_{s>0} \frac{B_{MM,k_1}\left(f^{[s]}\right)}{s}, \qquad B_{New,k_2} \approx \max_{s>0} \frac{B_{MM,k_2}\left(f^{[s]}\right)}{s}$$

and

$$\max_{s>0} \frac{B_{MM,k_1}\left(f^{[s]}\right)}{s} \geq \frac{B_{MM,k_1}\left(f^{(s^*_{k_2})}\right)}{s^*_{k_2}} \geq \frac{B_{MM,k_2}\left(f^{(s^*_{k_2})}\right)}{s^*_{k_2}} = \max_{s>0} \frac{B_{MM,k_2}\left(f^{[s]}\right)}{s}$$

where the third inequality holds due to known inequalities on polynomial norms.

We have also observed that the improvement is usually very large for the new bounds, especially when the magnitudes of the roots are different from 1. To generate data points, we generated 100 random monic polynomials (or square Pham polynomial systems) with fixed degree and height (defined below) and calculated the average value of the improvement:

$$\frac{B_{New,k}(f)}{B_{MM,k}(f)} = \left(\frac{||f_k||}{H_k}\right)^{d-1} \qquad \text{and} \qquad \frac{B_{New}(F)}{B_{DMM}(F)} = \left(\frac{\prod_{i=1}^{n} ||f_i||^{M_i}}{H}\right)^{D-1}.$$

Note that the improvement is independent of the discriminant for both new bounds. This observation allowed us to avoid many expensive computations when performing experiments (in particular, no resultants need be computed in the multivariate case).

We will measure the size of the coefficients of a monic univariate polynomial with the following expression:

$$||f||_B = \max_{0 \le i \le d-1} \frac{|a_i|^{\frac{1}{d-i}}}{\binom{d}{i}}$$

We will call this the B-Height (B for "binomial"). We can naturally extend the B-Height to Pham polynomials (see [9] for a precise definition) of degree d with the following expression

$$||f||_B = \max_{e \in Support(\text{trailing polynomial of } f)} \frac{|a_e|^{\frac{1}{d-|e|}}}{\binom{d}{e}}.$$

It is well known that both height definitions above are linearly related to the size of the roots. To generate a polynomial (or polynomial system) with the height r_n/r_d, we uniformly generated an integer c in the range $(-r_n, r_d)$ for every trailing coefficient. The corresponding integer for one coefficient was randomly chosen to be fixed at r_n. We then set

$$|a_e| = \left(\frac{r_n}{r_d}\right)^{d-|e|} \binom{d}{e}$$

and defined $f_i = x_i^d +$ trailing polynomial.

In the top plot of Figure 6, we plot the log of the average improvement of $B_{New,2}$ for 100 monic polynomials of degree 4 and given B-Height. We see similar plots both for other degrees and other choices of the norm ($B_{New,k}$ with $k \neq 2$). In the bottom plot of Figure 6, we plot the log of the average improvement of B_{New} for 100 Pham systems with $n = 3$ and the degree of every polynomial 3. We see similar plots both for other degrees and other choices of n. As we can see from Figure 6, the improvement increases as the magnitude of the roots becomes much different from 1.

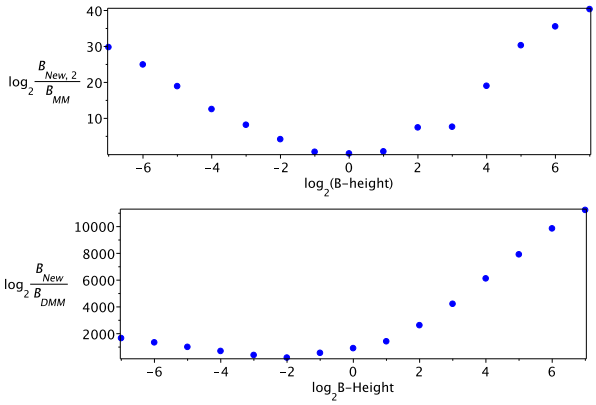

Fig. 6. Improvement and Binomial Height

We will also study the experimental performance of the new bounds on a special class of polynomials known as Mignotte polynomials. A Mignotte polynomial is defined as

$$Mig(d, h) = x^d - 2(hx - 1)^2.$$

It is well known that Mignotte polynomials have very small root separation (approximately h^{-d}). In Figure 7, we plot the logarithm of the exact root separation, $B_{New,2}$, and $B_{MM,2}$ of the improvement of $B_{New,2}$ for certain values of h and d. In the top plot, we fix h to be 10 and vary the degree. In the bottom plot, we fix the degree to be 4 and vary h. We can see from the plots that the new bound is consistently a tighter lower bound on the root separation. Furthermore, in the top plot we see that as the degree increases the improvement of the new bound over $B_{MM,2}$ increases.

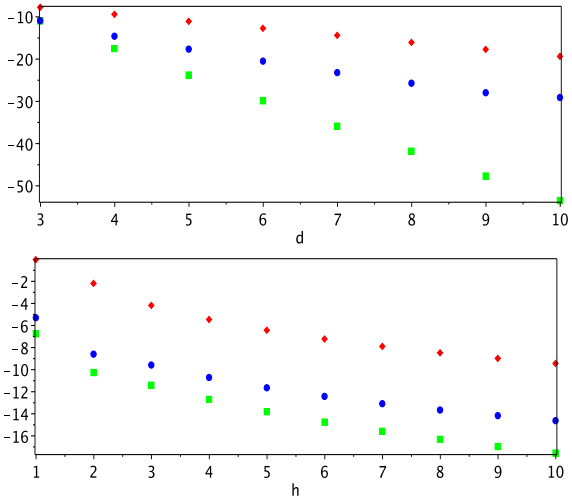

Fig. 7. Log plots of exact separation bound (red diamonds), $B_{New,2}$ (blue circles) and $B_{MM,2}$ (green boxes) for Mignotte polynomials.

6. Conclusion

In this paper we presented two improved root separation bounds. The new bounds improve on the previous bounds in two ways:

(1) The new bounds are usually *significantly bigger* (hence better) than the previous bounds.
(2) The new bounds *scale correctly*, unlike the previous bounds.

Crucially, the improved bounds are not harder to compute than the previous bounds. The improved bounds meet an important challenge facing researchers in our field (Section 2). To the best of the authors' knowledge, the new improved bounds are significantly bigger than *all* known (efficiently computable) root separation bounds and are the only known (efficiently computable) root separation bounds which scale correctly.

Of course, there remains plenty of room for improvement in the new bounds. In particular, the new multivariate bound presented in this paper is still quite pessimistic. One possible strategy for improving the new bound is to modify the bound to account for sparsity. This strategy does present a challenge: the derivation in this paper requires an understanding of the scaling behavior of the leading coefficient of T_{f_0} (see Lemmas 9 and 10). If T_{f_0} is calculated by computing the *sparse* resultant instead of the univariate resultant, the formula for the leading coefficient is much more challenging to work with (see [6]).

7. Acknowledgements

The authors would like to thank Carlos D'Andrea for his insights regarding the multivariate resultant. Hoon Hong acknowledges the partial support from the grant US NSF 1319632. Elias Tsigaridas is partially supported by GeoLMI (ANR 2011 BS03 011 06), HPAC (ANR ANR-11-BS02-013), and an FP7 Marie Curie Career Integration Grant.

References

[1] Convex hull trick. `http://wcipeg.com/wiki/Convex_hull_trick`.

[2] Y. Bugeaud and M. Mignotte. On the Distance Between Roots of Integer Polynomials. *Proc. Edinburgh Math. Soc. 47 (3)*, pages 553–556, 2004.

[3] C. Burnikel, S. Funke, K. Melhorn, S. Schirra, and S. Schmitt. A Separation Bound for Real Algebraic Expressions. *Lecture Notes in Computer Science*, pages 254–265, 2001.

[4] G. Collins and E. Horowitz. The Minimum Root Separation of a Polynomial. *Mathematics of Computation, Volume 28, Number 126*, pages 589 -597, 1974.

[5] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Springer, 2nd edition, 2005.

[6] C. D'Andrea and M. Sombra. A Poisson Formula for the Sparse Resultant. *Proc. London Math. Soc. (3) 110*, pages 932964 2015.

[7] I. Emiris, B. Mourrain, and E. Tsigaridas. The DMM Bound: Multivariate (Aggregate) Separation Bounds. *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 243–250, 2010.

[8] I. Emiris and E. Tsigaridas. Comparing Real Algebraic Numbers of Small Degree. *Lecture Notes in Computer Science Volume 3221*, 2004.

[9] L. Gonzalez-Vega and Neila Gonzalez-Campos. Simultaneous Elimination by using Several Tools from Real Algebraic Geometry. *Journal of Symbolic Computation, Volume 28*, pages 89-103, 1999.

[10] A. Herman and H. Hong. Quality of positive root bounds. *Journal of Symbolic Computation, Volume 74*, pages 592-602, 2015.

[11] C. Li, S. Pion, and C.Yap. Recent Progress in Exact Geometric Computation. *Journal of Logic and Algebraic Programming, Volume 64*, pages 85-111, 2004.

[12] K. Mahler. An Inequality for the Discriminant of a Polynomial. *The Michigan Mathematical Journal,11 (iss. 3),* page 257, 1964.

[13] K. Melhorn and S. Ray. Faster Algorithms for Computing Hong's Bound on Absolute Positivity. *Journal of Symbolic Computation, Volume 45*, pages 677-683, 2010.

[14] M. Mignotte. An Inequality About Factors of Polynomials. *Mathematics of Computation*, 28(128), pages 1153-1157, 1974.

[15] M. Mignotte. On the Distance Between the Roots of a Polynomial. *Applicable Algebra in Engineering, Communication, and Computing, Volume 6*, pages 327-332, 1995.

[16] S. Rump. Polynomial Minimum Root Separation. *Mathematics of Computation, Volume 33, Number 145*, pages 327-336, 1979.

[17] C. Schultz and R. Moller. Quantifier Elimination over Real Closed Fields in the Context of Applied Description Logics. *Univ., Bibl. des Fachbereichs Informatik.*, 2005.

[18] A. Strzebonksi and E. Tsigaridas. Univariate Real Root Isolation in an Extension Field and Applications. hal-01248390, 2011.

[19] E. Tsigaridas and I. Emiris. On the Complexity of Real Root Isolation Using Continued Fractions. *Theor. Comput. Sci., 392*, pages 158–173, 2008.