Sendra, J. R. & Winkler, S.M., 2016, "A heuristic and evolutionary algorithm to optimize the coefficients of curve parametrizations", Journal of Computational and Applied Math., vol. 305, pp. 18-35

Available at https://doi.org/10.1016/j.cam.2016.03.020

**The final journal version of this paper appears in Sendra J.R., Winkler S.
"A Heuristic and Evolutionary Algorithm to Optimize the Coefficients of Curve Parametrizations."
Journal of Computational and Applied Mathematics 305 (2016) 18--35, and it is available at
doi:10.1016/j.cam.2016.03.020
Se also corrigendum at doi:10.1016/j.cam.2016.04.028**

A Heuristic and Evolutionary Algorithm to Optimize the Coefficients of Curve Parametrizations *

J. Rafael Sendra
Research Group ASYNACS
Dpto. de Fisica y Matemáticas, Universidad de Alcalá
E-28871 Madrid, Spain

Stephan M. Winkler
University of Applied Sciences Upper Austria,
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, A-4232 Hagenberg, Austria

August 20, 2016

Abstract

Parametric representations may have unnecessarily huge integer coefficients. This can be a computational problem in practical applications. In this paper we present an evolutionary algorithm that reduces the maximum length of the coefficients for a proper curve parametrization with integer coefficients. This method is tested with different families of parametrizations, and as we show the results are very satisfactory in terms of achievable quality and runtime consumption. According to our knowledge, this is the first algorithmic approach to this problem.

Keywords: Rational curve, parametrization height, heuristic algorithm, evolutionary algorithm, arithmetic optimality.

1 Introduction

Rational algebraic curves and surfaces are basic tools in computer graphics, CAD/CAM, and surface/geometric modeling (see e.g. [16]) and they are applicable in many mathematical areas as, for instance, diophantine equations (see e.g. introduction in [34]) or symbolic solutions of algebraic differential equations (see e.g. [10],

*Rafael.Sendra@uah.es, Stephan.Winkler@fh-hagenberg.at

[18], [19]). One of the main advantages of these geometric objects is that they can be represented parametrically by means of rational functions. Nevertheless, since rational algebraic sets admit infinitely many different rational parametrizations, the power of their applicability varies depending on which parametrization is chosen.

In order to improve the applicability of the parametric representations, several authors have addressed the problem of transforming a given parametrization into a new parametrization satisfying certain required optimality criterium as, for instance, the injectivity or the surjectivity of the induced rational map, the degree (over the ground field) of the field of parametrization, or the height, i.e., the maximal absolute value of the integer coefficients of the parametrization (see e.g. [34]). In this paper we put the main focus on curves; readers interested in the analysis of surfaces may consult e.g. [3], [4], [5], [6], [9], [25], [28], [29], and [30]. Injective parametrizations are usually called proper parametrizations and, as a consequence of Lüroth's theorem, any rational curve can always be parametrized properly. Algorithmically, the problem was solved in [26]. Similarly, any rational curve can always be parametrized by means of a surjective parametrization (see [2], [9], or [27] for algorithmic approaches), although one may need to introduce complex numbers in the description of the parametrization. Concerning the field of parametrization, Hilbert and Hurtwitz [11] stated that one can always parametrize over a field algebraic extension of degree at most two over the ground field. Algorithms to achieve such parametrizations can be found in [12], [22], and [32]. Nevertheless, computing parametrizations with optimal height is, up to our knowledge, a fully open problem; in [20] one can see an example of applicability of our results. We refer to this last question as the arithmetic optimality problem, and in this paper we explain our approach to solve it by means of evolutionary computation.

Our proposal in this paper is to approach the problem by means of evolutionary computation improving the initial results and ideas given in [31]. Evolution strategies have been developed since the 1960s as explained and analyzed in detail in, e.g., [23], [24], and [7]. Since then, evolutionary computation has been applied to solve problems in many different areas (see, e.g., [14] [15]). Nevertheless, this does not mean that any optimization problem can be directly approached via evolutionary techniques; a pre-analysis of a suitable search strategy is required, since otherwise the evolutionary search may turn out to be a blind random process.

Let us briefly describe why our evolutionary strategy is suitable for the arithmetic optimality problem. We look for different changes of parameters such that the input proper parametrization is transformed into a better one; i.e., here, one with smaller height. We know that all possible changes of parameters are of the form

$$\frac{at+b}{ct+d} \text{ with } a,b,c,d \in \mathbb{Z} \text{ and } ad-cb \neq 0. \tag{1}$$

We visualize the space of solution candidates in 3D in so-called fitness landscapes [21]: In the x-axis we set the numerator (namely the pair (a,b)), in the y-axis we set the

denominator (i.e. the pair (c, d)) and the color represents the height of the resulting parametrization; red is small height (i.e. good result) and blue is big height (i.e. bad result). Such a direct description of the search space is not suited for the evolutionary algorithm since it is not smooth (see Fig. 5 (left)) and small variations (for instance, due to mutations) will produce big changes of quality in the answer. Instead, in Section 3, we prove that one can associate to each numerator (or denominator, resp.) a quantity that partially indicates the quality of the final answer (see eq. (27)). Thus, before generating the x-axis and the y-axis of the fitness landscape, we order the numerators as well as the denominators according to this partial quality. This produces fitness landscapes (see Fig. 5 (right)) which are well suited for our purposes. In this situation, in each region of the search space we look for improvement by optionally using strict offspring selection.

Based on these ideas, in this paper we present an evolutionary algorithm for dealing with the arithmetic optimality problem with a very satisfactory performance. The paper is structured as follows: In Section 2 we formally state the problem we aim to solve in the paper. Section 3 has 4 subsections: In Subsection 3.1 we analyze the search space, in Subsection 3.2 we study the notion of partial and complete quality, Subsection 3.3 is devoted to the evolutionary strategy, and in Subsection 3.4 we present our solution approach in detail. In the last section we empirically analyze the algorithm's performance: we use it for optimizing coefficients of parametrizations associated with a random family of conics; for a random family of parametrizations; as well as of three different families of space curves where the height of the applied Möbius transformation varies, and in all cases the results are very satisfactory. We also demonstrate the algorithm's performance used with varied algorithmic parameter settings, the details of the test results are given in tables that are shown in the paper's appendix.

2 Problem statement

In this section, we introduce the notation and terminology that will be used throughout the paper, and we state the problem we will deal with.

Let $\mathbf{P}(t)$ be a proper rational parametrization (with integer coefficients) of a curve $\mathbf{C}$ in the r-dimensional space $\mathbb{R}^r$. We recall that proper means that the parametrization is injective almost everywhere, i.e., almost all points in $\mathbf{C}$ are reached by exactly one parameter value via $\mathbf{P}(t)$ (see e.g. [34]). Let us **assume** that $\mathbf{P}(t)$ is expressed as

$$\mathbf{P}(t) = \left(\frac{p_1(t)}{q(t)}, \ldots, \frac{p_r(t)}{q(t)} \right) \tag{2}$$

such that

1. p_i, q are polynomials with integer coefficients such that $\gcd(p_1, \ldots, p_r, q) = 1$,

2. no component of $\mathbf{P}(t)$ is constant.

It is clear that assumption (1) is always reachable. Below, we will see that condition (2) does not imply either any loss of generality.

We recall that the height of a polynomial $f(t) = a_0 + \cdots + a_m t^m \in \mathbb{R}[t]$ is defined as $\mathrm{H}(f) = \max\{|a_0|, \ldots, |a_m|\}$. We define the height of the parametrization $\mathbf{P}(t)$ as

$$\mathrm{H}(\mathbf{P}(t)) = \max\{\mathrm{H}(p_1), \ldots, \mathrm{H}(p_r), \mathrm{H}(q)\}. \tag{3}$$

In this situation, the arithmetic optimality problem can be stated as follows:

> Problem Statement (General Version). Given $\mathbf{P}(t)$ as above, determine a rational change of parameter $\phi(t)$ such that $\mathbf{Q}(t) := \mathbf{P}(\phi(t))$ is proper, has integer coefficients, and $\mathrm{H}(\mathbf{Q}(t))$ is minimal.

We observe that if any component (say the first) of $\mathbf{P}(t)$ is a constant $c \in \mathbb{R}$, also for any other parametrization of $\mathbf{C}$ the first component equals c. Therefore, assumption (2) does not imply loss of generality.

Since $\mathbf{P}$ is proper, we have $\phi(t) = \mathbf{P}^{-1}(\mathbf{Q}(t))$. Moreover, since elimination theory techniques do not extend the ground field, $\mathbf{P}^{-1}$ has integer coefficients, and hence $\phi(t) \in \mathbb{Z}(t)$. Note that the properness condition is fundamental for the previous statement: for instance, $\mathbf{P}(t) = (\frac{1}{2}t^2, \frac{1}{2}t^2)$ is a non-proper parametrization of the line $y = x$, and $\mathbf{P}(\sqrt{2}t) = (t^2, t^2)$.

On the other hand, since $\mathbf{Q}(t)$ should also be proper, $\phi(t)$ has to be a Möbius transformation, i.e., a rational function of the form $(at + b)/(ct + d)$ with $ad - bc \neq 0$. Therefore, we reach the following equivalent formulation of the problem:

> Problem Statement (Equivalent Version). Given $\mathbf{P}(t)$ as above, determine $a, b, c, d \in \mathbb{Z}$, with $ad - bc \neq 0$, such that
>
> $$\mathrm{H}\left(\mathbf{P}\left(\frac{at + b}{ct + d}\right)\right) \tag{4}$$
>
> is minimal.

We will refer to the set $\{(a, b, c, d) \in \mathbb{Z}^4 \mid ad - bc \neq 0\}$ as either the solution space or the space of solutions of the here addressed problem, all the solutions of the problem belong to this set.

Let us discuss the minimality condition. The problem, as stated, asks for $\mathbf{Q}(t)$ such that any other proper parametrization of $\mathbf{C}$ has height greater than or equal to $\mathrm{H}(\mathbf{Q}(t)))$. However, according to our knowledge, determining the value of this minimum is, in general, still an open problem. Alternatively, one may consider the search for upper bounds on the minimum. Results in that direction can be found in the field of Diophantine geometry (see e.g. [8], [13]). For instance, the local Eisenstein

theorem for polynomials (see Corollary 11.5.16. pp. 370 in [8]) gives upper bounds for the case of local parametrizations around a regular point. Similarly, bounds on the height of rational points on the curves can be found (e.g. in [8], [13]), and these bounds could be translated into lower bounds for our problem. An upper bound can be found in [17], but it is too big. One can also derive a lower bound: Let $f(x_1, x_2)$ be the defining polynomial of the plane curve parametrically given by

$$\left(\frac{p_1(t)}{q(t)}, \frac{p_2(t)}{q(t)}\right). \tag{5}$$

Note that f is the implicit equation of the projection of $\mathbf{C}$ onto the first two coordinates. Therefore, if $r = 2$, f is the defining polynomial of $\mathbf{C}$. We assume that f is primitive, i.e. the gcd of the coefficients of f is 1. By Theorem 8 in [33], the primitive part of the resultant w.r.t. t of the polynomials $A(x_1, t) = q(t)x_1 - p_1(t)$ and $B(x_2, t) = q(t)x_2 - p_2(t)$ is f. Thus, using the definition of determinant as well as the degree bounds for f given in Theorem 5 in [33], one deduces that

$$\mathrm{H}(\mathbf{P}(t)) \geq \left\lceil \sqrt[n]{\frac{\mathrm{H}(f)}{n!}} \right\rceil. \tag{6}$$

where $n = \max\{\deg(p_1), \ldots, \deg(p_r), \deg(q)\}$. Of course, this bound is trivial if $\mathrm{H}(f) \leq n!$. Also, one could consider bounding $\max\{|a|, |b|, |c|, |d|\}$ for $(at+b)/(ct+d)$ providing the minimum. Again, up to our knowledge, it is an open question.

However, the application in practice of the above theoretical results is far from being effective. So, since the aim of this paper is more practical than theoretical, we will transform the statement of the problem into a more tractable form. For this purpose, we will limit the size of the space of solutions, and we will use a quantity to control the quality of the output. We will refer to this new subset of the solution space as a search space, i.e. the set of solutions that are potentially checked by the here proposed algorithm.

Let us start by considering bounds on the size. For $N \in \mathbb{N}$, we denote by $\mathbb{Z}(N)$ the set

$$\mathbb{Z}(N) = \{-N, \ldots, 0, \ldots, N\} \subset \mathbb{Z} \tag{7}$$

and by $\mathcal{M}(N)$ the set

$$\mathcal{M}(N) = \{(a, b, c, d) \in \mathbb{Z}(N)^4 \mid \gcd(a, b, c, d) = 1, ad - bc \neq 0\}. \tag{8}$$

Then, for a fixed $N \in \mathbb{N}$, we take $\mathcal{M}(N)$ as the search space. In other words, we ask for $(a, b, c, d) \in \mathcal{M}(N)$ such that

$$\mathrm{H}\left(\mathbf{P}\left(\frac{at+b}{ct+d}\right)\right)$$

is minimal. Since $\mathcal{M}(N)$ is a finite set, a direct approach would be to try all possible changes of parameter $(at+b)/(ct+d)$, with $(a,b,c,d) \in \mathcal{M}(N)$, and check which one has the best height. However, if N is not small, the cardinality of $\mathcal{M}(N)$ is huge and the direct method is unfeasible in practice. Alternatively, one may try a probabilistic approach. However, the next example shows that this is not a good idea.

Example 2.1. Let $\mathbf{P}(t)$ be the parametrization

$$\mathbf{P}(t) = \left(\frac{-54\,t^2 + 18\,t - 6}{9\,t^2+1}, \frac{2}{9\,t^2+1} \right). \tag{9}$$

Let us denote by $\#(X)$ the cardinality of a set X. Then, $\#(\mathcal{M}(5)) = 13216$ and the number of elements in $\mathcal{M}(5)$, providing a better parametrization than $\mathbf{P}(t)$, is 176; hence the probability of choosing a good candidate is 0.0133. In this case, the best choice is $(-1,-1,-3,0)$ yielding the parametrization

$$\mathbf{Q}(t) := \mathbf{P}\left(\frac{-t-1}{-3t} \right) = \left(-6\,\frac{t^2+t+1}{2\,t^2+2\,t+1}, 2\,\frac{t^2}{2\,t^2+2\,t+1} \right). \tag{10}$$

Note that $\mathrm{H}(\mathbf{Q}(t)) = 6$ while $\mathrm{H}(\mathbf{P}(t)) = 54$. In addition, the number of elements in $\mathcal{M}(5)$ providing height 6 is only 16. □

Our approach, presented in this paper, is to use evolutionary algorithms to provide the answer. Now, we are ready to state the problem we deal with in this paper.

Problem Statement (Practical version). Given $\mathbf{P}(t)$ as above, $N \in \mathbb{N}$, and $\theta \in \mathbb{R}^+$ determine $(a,b,c,d) \in \mathcal{M}(N)$, such that

$$\mathrm{H}\left(\mathbf{P}\left(\frac{at+b}{ct+d} \right) \right) \leq \theta \tag{11}$$

We finish this section with an example that illustrates these ideas. In order to measure the improvement we will compute the ratio of the β-length of the heights, where $\beta \geq 2$ is a fixed natural number (see [35], Section 2.1). We subtract 1 from this quantity so that the improvement is 0 if both heights are equal. If the input parametrization is $\mathbf{P}$ and the output $\mathbf{Q}$ we will measure the improvement as

$$\mathrm{Imp}_\beta(\mathbf{P},\mathbf{Q}) = \frac{\lfloor \log_\beta \mathrm{H}(\mathbf{P}) \rfloor + 1}{\lfloor \log_\beta \mathrm{H}(\mathbf{Q}) \rfloor + 1} - 1. \tag{12}$$

If $\mathrm{H}(\mathbf{P}) = \mathrm{H}(\mathbf{Q})$ then the improvement is 0, and it increases when the $\mathrm{H}(\mathbf{Q}) < \mathrm{H}(\mathbf{P})$. In the experiments we will take $\beta = 10$, and hence we will measure the ratio of the number of decimal digits of the heights of $\mathbf{P}$ and $\mathbf{Q}$.

Example 2.2. The example consists in the following experiment. We take a parametrization

$$\mathbf{P}(t) = \left(\frac{-3\,t^2 - 9}{t^2 + 1}, \frac{t^2 + 2\,t + 1}{t^2 + 1} \right) \tag{13}$$

of a conic $\mathbf{C}$ of equation $x^2 + 9\,y^2 + 12\,x - 18\,y + 36 = 0$. Note that $\mathrm{H}(\mathbf{P}(t)) = 9$. Now, we generate rational points $P_i := \mathbf{P}(12^i)$, $i \in \mathbb{N}$, on $\mathbf{C}$ and, using each of them, we compute a new proper parametrization $\mathbf{P}_i(t)$ of $\mathbf{C}$; see [34] for the parametrization algorithm. We get

$$\begin{aligned}
\mathbf{P}_0(t) &= \left(\tfrac{-54\,t^2+18\,t-6}{9\,t^2+1}, \tfrac{2}{9\,t^2+1} \right) \\
\mathbf{P}_1(t) &= \left(\tfrac{-3969\,t^2+432\,t-1299}{1305\,t^2+145}, \tfrac{1089\,t^2+858\,t+169}{1305\,t^2+145} \right) \\
\mathbf{P}_2(t) &= \left(\tfrac{-559953\,t^2+5184\,t-186627}{186633\,t^2+20737}, \tfrac{184041\,t^2+124410\,t+21025}{186633\,t^2+20737} \right) \\
\mathbf{P}_3(t) &= \left(\tfrac{-80621649\,t^2+62208\,t-26873859}{26873865\,t^2+2985985}, \tfrac{26842761\,t^2+17915898\,t+2989441}{26873865\,t^2+2985985} \right) \\
\mathbf{P}_4(t) &= \left(\tfrac{-11609505873t^2+746496\,t-3869835267}{3869835273\,t^2+429981697}, \tfrac{3869462025\,t^2+2579890170\,t+430023169}{3869835273\,t^2+429981697} \right)
\end{aligned}$$

Now, the question is whether we can improve the coefficients in the new parametrizations $\mathbf{P}_i(t)$. For this purpose, we apply the algorithm, presented in this paper, to each $\mathbf{P}_i(t)$. Since the aim of the example is to illustrate the problem, at this place of the paper, we will apply the algorithm as a *black-box* ; for details, see Section 3. In the application of the algorithm we will control the size of the search space by $N = 2 \cdot \mathrm{H}(\mathbf{P}_i(t))^2$, and $\theta = 10^3$; see the problem statement above. Let $\mathbf{Q}_i(t)$ denote the output. We get

$$\begin{aligned}
\mathbf{Q}_0 &= \mathbf{P}_0 \\
\mathbf{Q}_1 &= \left(\tfrac{-6(t^2+t+1)}{t^2+1}, \tfrac{2t^2}{t^2+1} \right) \\
\mathbf{Q}_2 &= \left(\tfrac{-6(t^2+t+1)}{t^2+2\,t+2}, \tfrac{2}{t^2+2\,t+2} \right) \\
\mathbf{Q}_3 &= \left(-\tfrac{228\,t^2+558\,t+441}{58\,t^2+174\,t+145}, \tfrac{100\,t^2+260\,t+169}{29(2\,t^2+6\,t+5)} \right) \\
\mathbf{Q}_4 &= \left(\tfrac{-6(t^2+t+1)}{t^2+1}, \tfrac{2}{t^2+1} \right)
\end{aligned}$$

In Table 1 we show the results of the experiments. For all $i > 0$ we get significantly better heights, this can be seen in the right column of Table 1. Furthermore, in three cases we get norm 6, which is even smaller than the height of $\mathbf{P}$.

3 Solution approach

In this section, we develop our solution approach to the problem stated in Section 2. One may distinguish two levels when approaching the solution: first, we study how

i	(a,b,c,d)	$\mathrm{H}(\mathbf{P}_i)$	$\mathrm{H}(\mathbf{Q}_i)$	$\mathrm{Impr}_{10}(\mathbf{P}_i,\mathbf{Q}_i)$
0	(1,0,0,1)	54	54	0
1	(-11, 13, -39, -33)	3969	6	3
2	(-145,-143,429,-435)	559953	6	5
3	(417, 715, 537, 180)	80621649	558	1.6
4	(-20737,41472,62205,6)	11609505873	6	10

Table 1: Results of the executions of the algorithm when applied to the parametrizations $\mathbf{P}_i$

to reduce the solution space into the search space, and then we solve the remaining problem with evolutionary search techniques. Throughout this section, we will use the same hypotheses and notation for $\mathbf{P}$ as in Section 2.

3.1 Search Space

Let $\mathcal{M}(N) \subset \mathbb{Z}^4$ be the search space introduced in Section 2, with N fixed; see (8) for the description of $\mathcal{M}(N)$. The idea now is to reduce the size of $\mathcal{M}(N)$ using heuristic reasonings. We can see the search spaces as sets Ω^2, with $\Omega \subset \mathbb{Z}(N)^2$; see (7) for the definition of $\mathbb{Z}(N)$. So reducing the size of Ω, the size of the search space also reduces. We will refer to Ω as the seed set of the search space. Taking into account this remark, we introduce the following notation. For $\Omega \subset \mathbb{Z}(N)^2$ we denote the search space associated to Ω as $\mathrm{Space}(\Omega)$ where:

$$\mathrm{Space}(\Omega) = \{(a,b,c,d) \in \Omega^2 \mid \gcd(a,b,c,d) = 1 \text{ and } ad - bc \neq 0\}. \tag{14}$$

Note that $\mathcal{M}(N) = \mathrm{Space}(\mathbb{Z}(N)^2)$ is the biggest search space for N fixed.

In order to reduce the search space, we consider the homogenization $\mathbf{P}^H(t,h)$ of $\mathbf{P}(t)$. Let $\mathbf{P}^H(t,h)$ be expressed as

$$\mathbf{P}^H(t,h) = (P_1(t,h), \ldots, P_r(t,h), Q(t,h)) \tag{15}$$

with $\gcd(P_1, \ldots, P_r, Q) = 1$ and where all polynomials P_i, Q have the same degree; i.e. $P_i(t,1) = p_i(t), Q(t,1) = q(t)$. Furthermore, for $(a,b,c,d) \in \mathrm{Space}(\Omega)$, let

$$\mathbf{Q}(t) = \mathbf{P}\left(\frac{at+b}{ct+d}\right) = \left(\frac{A_1(t)}{B(t)}, \ldots, \frac{A_r(t)}{B(t)}\right), \tag{16}$$

where $\gcd(A_1, \ldots, A_r, B) = 1$, and let

$$\mathbf{Q}^*(t) = (A_1(t), \ldots, A_r(t), B(t)). \tag{17}$$

Then, the following lemma holds.

Lemma 3.1.

1. $\mathbf{P}^H(b,d) = \mathbf{Q}^*(0)$.
2. $\mathbf{P}^H(a,c) = (\mathrm{LC}(A_1),\ldots,\mathrm{LC}(A_r),\mathrm{LC}(B))$, *where* LC *denotes the leading coefficient of the polynomial.*

Proof. It follows taking into account that $\mathbf{Q}^*(t) = \mathbf{P}^H(at+b, ct+d)$. □

For $(\alpha,\beta) \in \mathbb{Z}^2$, we introduce the quantity

$$\Delta(\alpha,\beta) := \frac{\|\mathbf{P}^H(\alpha,\beta)\|_\infty}{\gcd(|P_1(\alpha,\beta)|,\ldots,|P_r(\alpha,\beta)|,|Q(\alpha,\beta)|)} \tag{18}$$

where $\|\mathbf{P}^H(\alpha,\beta)\|_\infty = \max\{|P_1(\alpha,\beta)|,\ldots,|P_r(\alpha,\beta)|,|Q(\alpha,\beta)|\}$.

Theorem 3.2.

$$\mathrm{H}(\mathbf{Q}) \geq \max\{\|\mathbf{P}^H(a,c)\|_\infty, \|\mathbf{P}^H(b,d)\|_\infty\} \geq \max\{\Delta(a,c),\Delta(b,d)\}.$$

Proof. By construction, $\mathrm{H}(\mathbf{Q}) = \mathrm{H}(\mathbf{Q}^*)$ and, by Lemma 3.1, we have

$$\begin{aligned}\mathrm{H}(\mathbf{Q}^*) &\geq \|\mathbf{Q}^*(0)\|_\infty = \|\mathbf{P}^h(b,d)\|_\infty,\\ \mathrm{H}(\mathbf{Q}^*) &\geq \|(\mathrm{LC}(A_1),\ldots,\mathrm{LC}(A_r),\mathrm{LC}(B))\|_\infty = \|\mathbf{P}^h(a,c)\|_\infty.\end{aligned}$$

Hence, the first inequality in the statement holds. The second inequality follows taking into account that $\Delta(\alpha,\beta) \leq \|\mathbf{P}^H(\alpha,\beta)\|_\infty$. □

Our goal is to find $(a,b,c,d) \in \mathrm{Space}(\Omega)$ such that $\mathrm{H}(\mathbf{Q})$ is small. On the other hand, Theorem 3.2 provides a lower bound of the height. So, small values of the height will be reached only for small values of the lower bound. Based on this fact, we will use the following principle in our evolutionary algorithm.

$$[\mathsf{A}] : \textit{the smaller } \max\{\Delta(a,c),\Delta(b,d)\} \textit{ is, the smaller } \mathrm{H}(\mathbf{Q}) \textit{ could be} \tag{19}$$

Using [A] we may reduce the seed set as follows. Let $k \in \mathbb{R}^+$ ($k = 10^2$, e.g.), then we consider the seed set

$$\Omega(\mathbf{P},N,k) = \left\{(\alpha,\beta) \in \mathbb{Z}(N)^2 \mid k\cdot\Delta(\alpha,\beta) < \mathrm{H}(\mathbf{P})\right\}. \tag{20}$$

In practice, $\#(\mathbb{Z}(N)^2)$ can be huge, and the computation of $\Omega(\mathbf{P},N,K)$ may consume too much time. To deal with this problem, we introduce a subset $\Omega(\mathbf{P},N,K)$ of smaller cardinality (the prime seed set), and from it we easy compute a bigger set (the extended seed set) where the solution is searched. In general, the extended seed set will not include all elements in $\Omega(\mathbf{P},N,K)$ but will contain other candidates not considered in $\Omega(\mathbf{P},N,K)$. So, instead of working with the initial seed set $\Omega(\mathbf{P},N,K)$, which is

complicated to compute, we will work with the extended seed set - let us now develop these ideas:

We observe, since the polynomial expressions in Δ are homogenous in $\{\alpha, \beta\}$ and of the same degree, that

$$\Delta(\alpha, \beta) = \Delta(\rho\alpha, \rho\beta), \quad \text{for } \rho \neq 0. \tag{21}$$

Thus, when computing $\Omega(\mathbf{P}, N, k)$, we only need to find those coprime α, β satisfying $k \cdot \Delta(\alpha, \beta) < \mathrm{H}(\mathbf{P})$. This motivates the notion of prime seed set associated to $\Omega(\mathbf{P}, N, k)$:

$$\Omega_p(\mathbf{P}, N, k) = \{(\alpha, \beta) \in \Omega(\mathbf{P}, N, k) \mid \gcd(\alpha, \beta) = 1\} . \tag{22}$$

Let $\eta \cdot \mathcal{H}$ denote the set $\{i \cdot u \,|\, u \in \mathcal{H}, i \in \mathbb{N}, i \leq \eta\}$. Then, for $w \in \mathbb{N}$ we define the w-extended seed set as

$$\Omega_e(\mathbf{P}, N, k, w) = w \cdot \Omega_p(\mathbf{P}, N, k), \tag{23}$$

and we call w the amplitude. Note that we have the following inclusions

$$\Omega_e(\mathbf{P}, N, k, w) \supset \Omega_p(\mathbf{P}, N, k) \subset \Omega(\mathbf{P}, N, k). \tag{24}$$

In the following we illustrate these ideas with two examples. The first example is theoretical while the second correspond to a particular parametrization.

Example 3.1. Let us assume that $N = 4$, and (as shown in Fig. 1 left) we get:

$$\begin{aligned}\Omega(\mathbf{P}, 4, k) = \quad & \{(\pm1, \pm1), (\pm2, \pm2), (\pm3, \pm3), (\pm4, \pm4), (\pm1, 0), (\pm2, 0), (\pm3, 0), \\ & (\pm4, 0), (-1, 2), (1, -2), (\pm2, \pm4), (\pm4, \pm2)\}\end{aligned}$$

Then, the prime seed set is (as shown in Fig. 1, right):

$$\Omega_p(\mathbf{P}, 4, k) = \{(\pm1, \pm1), (\pm1, 0), (-1, 2), (1, -2)\}.$$

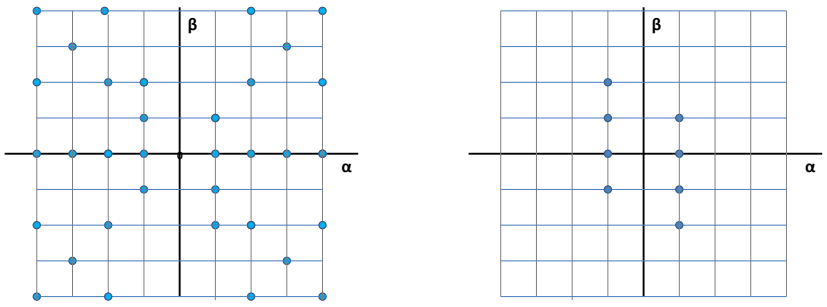

Figure 1: Illustration of seed sets: $\Omega(\mathbf{P}, 4, k)$ (left), $\Omega_p(\mathbf{P}, 4, k)$ (right)

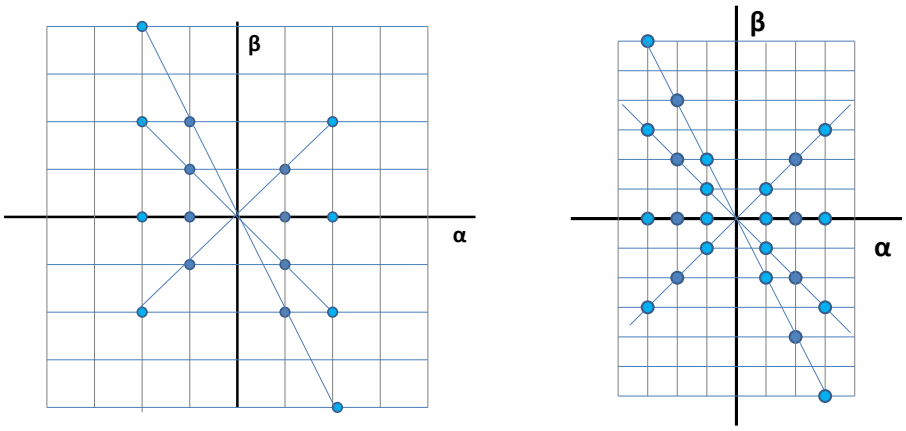

Figure 2: Illustration of seed sets: $\Omega_e(\mathbf{P}, 4, k, 2)$ (left), $\Omega_e(\mathbf{P}, 4, k, 3)$ (right)

Different instances of the extended seed set are (see Fig. 2)

$$\begin{aligned}\Omega_e(\mathbf{P}, 4, k, 2) = \; & \{((\pm 1, \pm 1), (\pm 2, \pm 2), (\pm 1, 0), (\pm 2, 0), (-1, 2), \\ & (-2, 4), (1, -2), (2, -4)\} \\ \Omega_e(\mathbf{P}, 4, k, 3) = \; & \{((\pm 1, \pm 1), (\pm 2, \pm 2), (\pm 3, \pm 3), (\pm 1, 0), (\pm 2, 0), \\ & (\pm 3, 0), (-1, 2), (-2, 4), (-3, 6), (1, -2), (2, -4), (3, -6)\}\end{aligned}$$

Observe that $\#(\Omega(\mathbf{P}, 4, k)) = 34$, $\#(\Omega_p(\mathbf{P}, 4, k)) = 8$, $\#(\Omega_e(\mathbf{P}, 4, k, 2) = 16$, and $\#(\Omega_e(\mathbf{P}, 4, k, 3) = 24$.

Example 3.2. We consider the parametrization

$$\mathbf{P}(t) = \left(\frac{-3969\, t^2 + 432\, t - 1299}{1305\, t^2 + 145}, \frac{1089\, t^2 + 858\, t + 169}{1305\, t^2 + 145} \right). \tag{25}$$

We observe that this parametrization is indeed the parametrization $\mathbf{P}_1(t)$ in Example 2.2. We get that $\#(\Omega(\mathbf{P}, 300, 10^{1.5})) = 1640$ and $\#(\Omega_p(\mathbf{P}, 300, 10^{1.5})) = 332$; note that $\#(\mathbb{Z}(300)^2) = 361201$ (see Fig. 3). In addition, we consider a smaller prime seed set and we extend it. We take $\Omega_p(\mathbf{P}, 50, 10^{1.5})$ and $\Omega_e(\mathbf{P}, 50, 10^{1.5}, 6)$. We get that $\#(\Omega_p(\mathbf{P}, 50, 10^{1.5})) = 76$ and $\#(\Omega_e(\mathbf{P}, 50, 10^{1.5}, 6)) = 1200$ (see Fig. 4). Observe that $\Omega(\mathbf{P}, 300, 10^{1.5}) \neq \Omega_e(\mathbf{P}, 50, 10^{1.5}, 6)$. Finally, we observe that the application of our algorithm with $\Omega(\mathbf{P}, 300, 10^{1.5})$ and $\Omega_e(\mathbf{P}, 50, 10^{1.5}, 6)$ provided the same result, namely $\mathbf{Q}_1(t)$ in Example 2.2.

Remark. The following idea was provided by an anonymous referee and can be applied to work with the prime seed instead of the extended seed: Let $(\mathcal{G}, \circ)$ denote the group of Möbius transformations with the composition as operation. Every $\Phi \in \mathcal{G}$ can be decomposed as

$$\Phi = \Phi_1 \circ \Phi_2$$

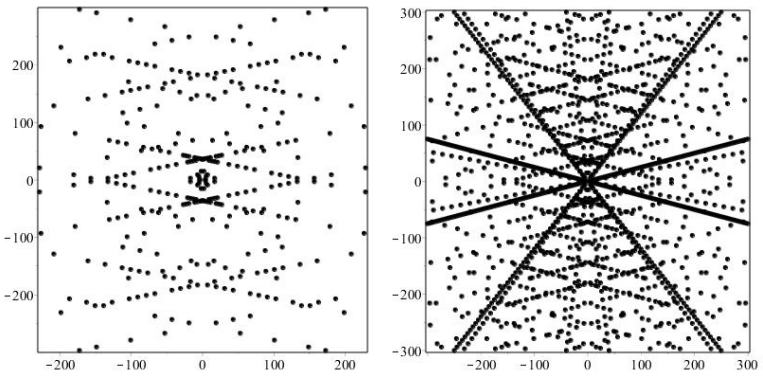

Figure 3: Seed set $\Omega_p(\mathbf{P}, 300, 10^{1.5})$ (left) and $\Omega(\mathbf{P}, 300, 10^{1.5})$ (right) of Example **??**

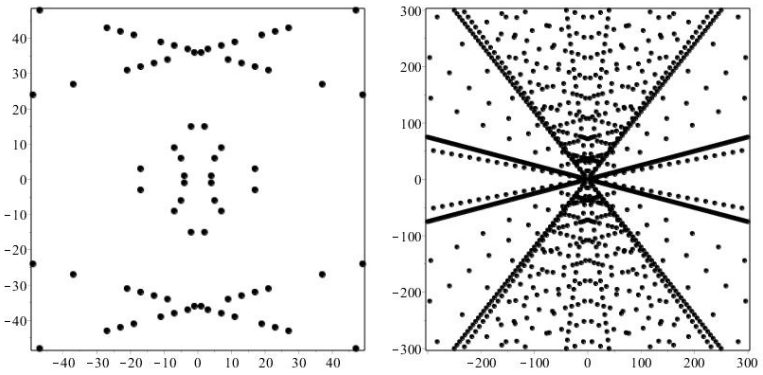

Figure 4: Seed set $\Omega_p(\mathbf{P}, 50, 10^{1.5})$ (left) and $\Omega_e(\mathbf{P}, 50, 10^{1.5}, 6)$ (right) of Example **??**

where $\Phi_1, \Phi_2 \in \mathcal{G}$ are of the form

$$\Phi_1(t) = \frac{e}{f}t, \;\; \Phi_2(t) = \frac{at+b}{ct+d}, \;\; \text{with} \;\; \gcd(a,b) = \gcd(c,d) = 1.$$

This implies that Φ_2 can be determined by combining solutions generated by the prime seed. For the computation of the Φ_1 it is only relevant that e, f are divisors of the gcd of the independent coefficients and of the leading coefficients of the polynomials involved in the parametrization, respectively.

3.2 Identification of Partial Solutions

In Subsection 3.1 we have introduced the notion of extended seed set $\Omega_e(\mathbf{P}, N, k, w)$, as well as the concept of search space $\text{Space}(\Omega_e(\mathbf{P}, N, k, w))$ associated to $\Omega_e(\mathbf{P}, N, k, w)$. We will refer to the elements in $\Omega_e(\mathbf{P}, N, k, w)$ as partial solution (candidate) and to the elements in $\text{Space}(\Omega_e(\mathbf{P}, N, k, w))$ as complete solution candidates; in the sequel, unless risk of ambiguity, we simplify the notation as Ω_e and $\text{Space}(\Omega_e)$ respectively.

The composition of complete solution candidates from partial solution candidates is defined as follows: Let $(\mathbf{o}_1, \mathbf{o}_2) \in \Omega_e \times \Omega_e$ with $\mathbf{o}_1 = (o_{1,1}, o_{1,2})$, and $\mathbf{o}_2 = (o_{2,1}, o_{2,2})$. Then, the associated complete solution candidate is $S_{\mathbf{o}_1,\mathbf{o}_2} := (o_{1,1}, o_{1,2}, o_{2,1}, o_{2,2})$. Conversely, every complete solution candidate $(a, b, c, d) \in \text{Space}(\Omega_e)$ can be seen as a combination of elements in Ω_e, namely $(a, b), (c, d) \in \Omega_e$.

We first generate a seed set Ω_e and then our goal is to find the best combinations of elements in Ω_e for composing the final complete solution of the given problem. In order to measure the quality of a complete solution candidate, we introduce the notion of complete quality. Given $\mathbf{s} := (a, b, c, d) \in \text{Space}(\Omega_e)$ we define the complete quality of $\mathbf{s}$ as

$$\text{Quality}_c(\mathbf{s}, \mathbf{P}) = \text{H}\left(\mathbf{P}\left(\frac{at+b}{ct+d}\right)\right). \tag{26}$$

Please note that good candidates correspond to low qualities.

3.3 Evolutionary Search for Optimal Combinations of Elements of the Search Space

The last challenge is to find optimal combinations of partial solution candidates $\mathbf{o}_1$ and $\mathbf{o}_2$ ($\mathbf{o}_1, \mathbf{o}_2 \in \Omega_e$). Depending on the input as well as on the parameters defining Ω_e, the size of Ω_e varies; there may be thousands of elements in Ω_e. Thus, exhaustive search is possible, but may lead to very high runtime consumption as even for small input examples millions of combinations have to be evaluated if all possible combinations are to be checked. Please note that the same phenomenon will happen if, instead of working with Ω_e, we work with Ω_p as described in Remark 3.1.

This is why we have developed a heuristic search method for the search of optimal combinations of partial solution candidates. This search strategy works in an iterative, evolutionary way and is based on the theory of evolution strategies (see [7]). The main strategy of our method is to use the extended seed to generate partial solutions, order according to their partial quality, and apply evolutionary search to find optimal complete solutions.[1]

Of course, as already mentioned, in order to find the best solution (or a solution among the best ones), we could try the direct approach by computing $\text{Quality}_c(\mathbf{s}, \mathbf{P})$ for all $\mathbf{s} \in \text{Space}(\Omega_e)$. However, although we have simplified the space of solutions by means of the notion of extended seed, the computation of the quality involves the composition of rational functions, the simplification of rational functions, and integer gcds computations of so many complete solution candidates that the direct approach is unfeasible in practice. Instead, we want to deduce good complete solution candidates from the partial solutions. More precisely, we apply evolutionary techniques to identify

[1] An alternative approach would be to use the prime seed and apply evolutionary search for finding optimal complete solutions - nevertheless, we do not investigate this approach here.

optimal combinations of partial solution candidates. For this purpose, we introduce the notion of partial quality. The concept is based on the principle [A] (see Subsection 3.1): Given $\mathbf{o} = (o_1, o_2) \in \Omega_e$ we define the partial quality of $\mathbf{o}$ as (see (15))

$$\text{Quality}_{\text{p}}(\mathbf{o}, \mathbf{P}) = \gcd(P_1(\mathbf{o}), \ldots, P_r(\mathbf{o}), Q(\mathbf{o})). \tag{27}$$

In order to motivate our evolutionary search strategy we consider an example:

Example 3.3. Let $\mathbf{P}(t)$ be the parametrization

$$\mathbf{P}(t) = \left(\frac{26182752640t^2 + 12512544t + 517752}{771199968t^2 + 45486}, \frac{8476943376t^2 + 1026528t + 500715}{771199968t^2 + 45486} \right).$$

For this example, setting $N = 1000$, $k = 10^{1.5}$, and $w = 1$, the set of partial solution candidates $\Omega_e(\mathbf{P}, N, k, w)$ consists of 348 elements. In order to analyze the viability of an evolutionary strategy we perform two experiments:

(i) For $\mathbf{o}_1, \mathbf{o}_2 \in \Omega_e$, we compute $\text{Quality}_{\text{c}}(\mathbf{o}_1, \mathbf{o}_2)$. Then, $(\mathbf{o}_1, \mathbf{o}_2, \text{Quality}_{\text{c}}(\mathbf{o}_1, \mathbf{o}_2))$ is represented in Figure 5.

(ii) We order the elements in Ω_e by the value of $\text{Quality}_{\text{c}}$. Let Ω_e^{ord} be the ordered set. Then, we proceed with the elements in Ω_e^{ord} as in (i).

In Figure 5 (left) we show the fitness landscape (see [21]) for (i), and in Figure 5 (right) we show the fitness landscape for (ii), where red indicates good quality and blue indicates bad quality. As we see in Figure 5 (right), ordering partial solution candidates using this fitness function (partial quality) leads to a search space with a much smoother fitness landscape, which makes the use of evolutionary search reasonable.

In Figure 6 we show the 3-dimensional visualization of the landscape. Here we see that in the upper-right area of the landscape there are many high-quality partial solutions, hence we start searching for optimal complete solutions in that area.

□

So, let $\ell = \#(\Omega_e)$, and assume that the elements of Ω_e are ordered as $\{\mathbf{o}_1, \ldots, \mathbf{o}_\ell\}$ where $\mathbf{o}_i \leq \mathbf{o}_j$ if $\text{Quality}_{\text{p}}(\mathbf{o}_i) \leq \text{Quality}_{\text{p}}(\mathbf{o}_j)$. First, we create an initial population of μ complete solution candidates $\mathbf{s}_{\mathbf{o}_i, \mathbf{o}_j}$: we take $i, j \in \{1, \ldots, \#(\Omega_e)\}$; 50% of the initial partial solution candidates are created randomly, 50% are created with $i, j \in \{\lceil \#(\Omega_e)/2 \rceil, \ldots, \#(\Omega_e)\}$, as we assume that there is a higher chance to find optimal solutions by combining partial solution candidates with rather high partial qualities.

Next, the evolutionary search process is started and executed generation-wise: In each generation we create λ new solution candidates ("offspring") by drawing a parent solution candidate of the current population randomly and mutating it randomly; mutating a solution candidate $\mathbf{s}_{\mathbf{o}_i, \mathbf{o}_j}$ here means that i and j are randomly increased or decreased by a offset $z \in \{1, \ldots, r\}$ where r defines the mutation radius.

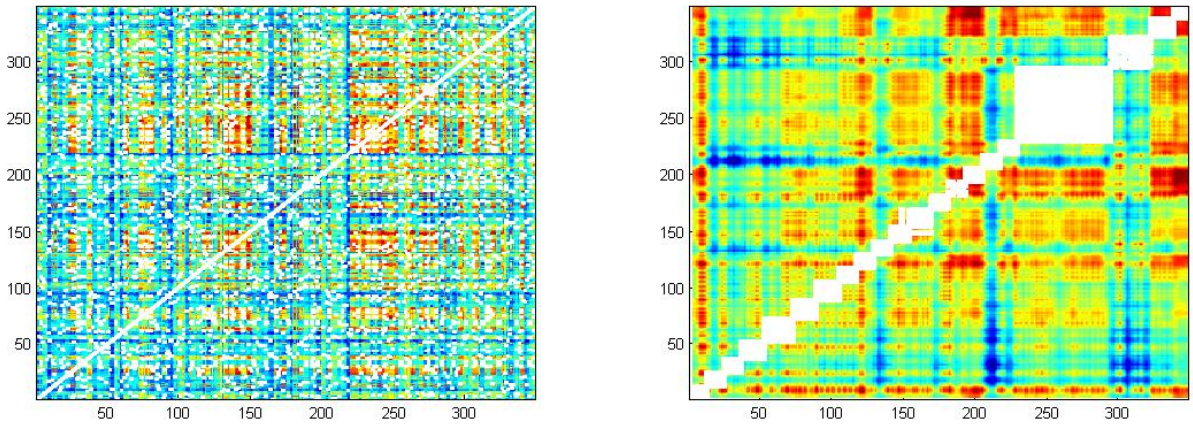

Figure 5: Fitness landscape for combinations of elements of Ω_e for the parametrization $\mathbf{P}(t)$ in Example 3.3; $N = 1000$, $k = 10^{1.5}$, $w = 1$. Here, the number of elements in Ω_e is 348. Each cell (x, y) represents the fitness of combination of $x \in \Omega_e$ (right $x \in \Omega_e^{\text{ord}}$ and $y \in \Omega_e$ (right $y \in \Omega_e^{\text{ord}}$), where red indicates good quality and blue indicates bad quality.

Each solution candidate $\mathbf{s}_{\mathbf{o}_i,\mathbf{o}_j}$, generated in this way, is evaluated using the complete quality function. From the so generated λ offspring candidates we select the μ best solution candidates. If *offspring selection* (see [1]) is applied, then we consider only those children that are better than their parents; if *elitism* (see [1]) is applied, then the previous generation's best individual is also copied to the next generation.

Additionally, we calculate the success ratio R as the ratio of mutants that are evaluated better than their respective parents; if R is smaller than 0.2, i.e., if less than 20% of the mutations lead to better solution candidates, then R is decreased (divided by a given factor ς, $\varsigma > 1.0$), and if R is bigger than 0.2, then it is increased (multiplied by ς).

This procedure is repeated until R is decreased to 0 or the maximum number of generations is reached or the number of consecutive unsuccessful generations is reached; a generation is here considered unsuccessful if no candidate is found that is better than the previous generation's best individual.

3.4 Composed method

Summarizing the algorithm defined in the previous sections we here define the overall solution approach. In the description of the algorithms, for a given list L, we denote by $L[i]$ the i-th element in the list.

- Algorithm 1 is the complete algorithm. It is represented as

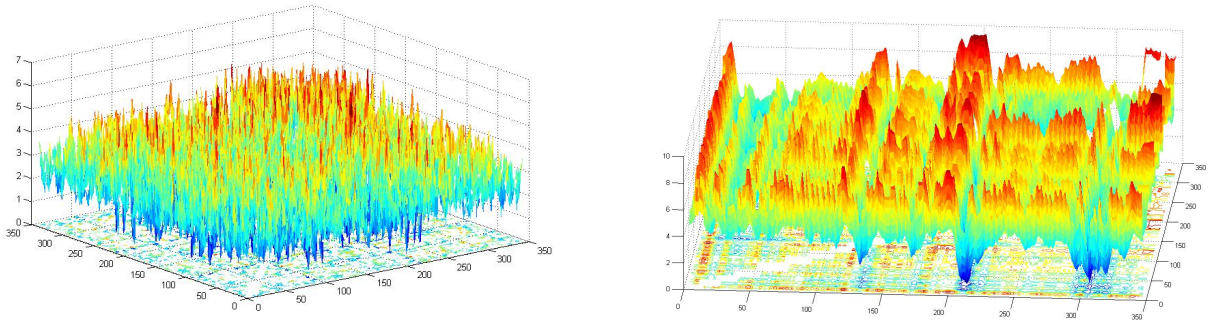

Figure 6: Fitness landscape for combinations of elements of Ω_e for the parametrization $\mathbf{P}(t)$ in Example 3.3; $N = 1000$, $k = 10^{1.5}$, $w = 1$. Here, the number of elements in Ω_e is 348. Each cell (x, y) represents the fitness of combination of $x \in \Omega_e$ (right $x \in \Omega_e^{\text{ord}}$ and $y \in \Omega_e$ (right $y \in \Omega_e^{\text{ord}}$).

$$\text{FindOptimalParametrization}(\mathbf{P}, N_0, w_0, N, \theta, k, \mu, \lambda, \text{ø}, \rho, \gamma, \gamma_{us}, \psi, \varepsilon, \varsigma).$$

The input parameters are:

- $\mathbf{P}$ is the input parametrization.
- $N_0 \in \mathbb{N}$ is the size of the prime seed set to be used in the algorithm.
- $w_0 \in \mathbb{N}$ is the initial size of the amplitude.
- $N \geq N_0$ is the size of the search space. In our experiments, based on the reasoning in Section 2, we take $N = 2\,\mathrm{H}(\mathbf{P})^2$.
- θ is the upper bound for the expected height.
- k is the controlling factor in the definition of Ω_e (see Eq. (20)).
- For the evolutionary search for the best combination of elements of the sorted set Ω_e we define the following parameters:
  - μ defines the number of individuals in the population.
  - λ defines the number of children created each generation using mutation.
  - ø defines whether strict offspring selection is applied. ø is either false or true.
  - ρ is the initial mutation radius.
  - γ defines the maximum number of generations.
  - γ_{us} defines the maximum number of unsuccessful consecutive generations; a generation is considered unsuccessful if no candidate is found that is better than the previous generation's best candidate.
  - ψ defines whether *plus* selection is applied. If *plus* selection is applied then the next generation's individuals are chosen from the joint pool of the current generation's children and parents.

    - ε defines whether elitism is applied. If elitism is applied, then the best individual of a generation is automatically included in the next generation's population.
    - ς is the factor for decreasing or increasing the mutation with depending on the ratio of successful mutations in a generation.

- Algorithm 2 defines the evaluation of a combination $(\mathbf{o}_1, \mathbf{o}_2)$ of partial solution candidates $\mathbf{o}_1, \mathbf{o}_2 \in \Omega_e$. It is represented as Evaluate$(\mathbf{o}_1, \mathbf{o}_2, \mathbf{P})$.

- Algorithm 3 defines the mutation of a solution candidate $(\mathbf{o}_i, \mathbf{o}_j)$ representing a combination of partial solution candidates. It is represented as Mutate$(\mathbf{o}_i, \mathbf{o}_j, R, \Omega_e^{\text{ord}})$. Ω_e^{ord} is the set Ω_e after ordering its elements by their partial quality as explained in Subsection 3.3. In this sense, $\mathbf{o}_k$ denotes the k-th element in Ω_e^{ord}. R is used in the normal distribution.

- Algorithm 4 is the main sub-algorithm and it is called with increasing values of the amplitude (w) in Algorithm 1. It is represented as

$$\text{OptimizeCoefficients}(\mathbf{P}, N, k, w, \mu, \lambda, \emptyset, \rho, \gamma, \gamma_{us}, \psi, \varepsilon, \varsigma)$$

Algorithm 1 FindOptimalParametrization$(\mathbf{P}, N_0, w_0, N, \theta, k, \mu, \lambda, \emptyset, \rho, \gamma, \gamma_{us}, \psi, \varepsilon, \varsigma)$

1: $w \leftarrow w_0$, $\hat{\theta} \leftarrow \text{H}(\mathbf{P})$, $\hat{\mathbf{P}} \leftarrow \mathbf{P}$, $i \leftarrow 1$.
2: **while** $(w \cdot N_0) < N \wedge \hat{\theta} > \theta$ **do**
3: $\quad (a, b, c, d) \leftarrow \text{OptimizeCoefficients}(\hat{\mathbf{P}}, (i+1) \cdot N_0, k, w, \mu, \lambda, \emptyset, \rho, \gamma, \gamma_{us}, \psi, \varepsilon, \varsigma)$.
4: $\quad \hat{\mathbf{P}} \leftarrow \hat{\mathbf{P}}\left(\frac{at+b}{ct+d}\right)$, $\hat{\theta} \leftarrow \text{H}(\hat{\mathbf{P}})$, $i \leftarrow i+1$ and $w \leftarrow i \cdot w$.
5: **end while**
6: **return** $\hat{\mathbf{P}}$

Algorithm 2 Evaluate$(\mathbf{o}_1, \mathbf{o}_2, \mathbf{P})$

1: compile complete solution from the partial solution candidates $\mathbf{o}_1 = (\alpha_1, \beta_1), \mathbf{o}_2 = (\alpha_2, \beta_2)$:

$$\mathbf{s} = (a, b, c, d) \leftarrow (\alpha_1, \beta_1, \alpha_2, \beta_2)$$

2: calculate quality of complete solution candidate:

$$q \leftarrow \text{Quality}_{\text{c}}(\mathbf{s}, \mathbf{P}) = \text{H}\left(\mathbf{P}\left(\frac{at+b}{ct+d}\right)\right)$$

3: **return** q

Algorithm 3 Mutate($\mathbf{o}_i, \mathbf{o}_j, R, \Omega_e^{\mathrm{ord}}$)

draw z_i and z_j randomly from the normal distribution $\mathcal{N}(0, R)$
$\hat{i} \leftarrow i + z_i, \hat{j} \leftarrow j + z_j$
$n \leftarrow \#(\Omega_e^{\mathrm{ord}})$
if $\hat{i} > n$ **then** $\hat{i} \leftarrow n - (\hat{i} - n)$
if $\hat{i} < 1$ **then** $\hat{i} \leftarrow 1 + (1 - \hat{i})$
if $\hat{j} > n$ **then** $\hat{j} \leftarrow n - (\hat{j} - n)$
if $\hat{j} < 1$ **then** $\hat{j} \leftarrow 1 + (1 - \hat{j})$
$\hat{\mathbf{o}} = (\mathbf{o}_{\hat{i}}, \mathbf{o}_{\hat{j}})$
return $(\hat{\mathbf{o}})$

Algorithm 4 OptimizeCoefficients($\mathbf{P}, N, k, w, \mu, \lambda, \emptyset, \rho, \gamma, \gamma_{us}, \psi, \varepsilon, \varsigma$)

calculate the prime seed set $\Omega_p = \{(\alpha, \beta) \in \mathbb{Z}^2 \mid k \cdot \Delta(\alpha, \beta) < \mathrm{H}(\mathbf{P}) \text{ and } \gcd(\alpha, \beta) = 1\}$ ▷ See Eq. (18) for the definition of Δ.
calculate the w-extended seed set as $\Omega_e = w \cdot \Omega_p$
sort elements in Ω_e according to their partial quality and store resulting sorted set in Ω_e^{ord} ▷ See Eq. (27) for the notion of partial quality.
$n \leftarrow \#(\Omega_e^{\mathrm{ord}})$
initialize population for search for best combination of elements of Ω_e^{ord}: $pop = \{\}$
for $i = 1$ to $\mu/2$ **do** $pop \leftarrow pop \cup (x, y)$ with $x, y \in \{1, \ldots, \lfloor n/2 \rfloor\}$
for $i = 1$ to $\mu/2$ **do** $pop \leftarrow pop \cup (x, y)$ with $x, y \in \{1, \ldots, n\}$
evaluate all individuals in pop using Algorithm 2: $\forall \mathbf{o} \in pop : quality(\mathbf{o}) = Evaluate(\mathbf{o}, \mathbf{P})$ ▷ Note that $quality(\mathbf{o}) = \mathrm{Quality_c}(\mathbf{o}, \mathbf{P})$
$R \leftarrow \rho$, $gen \leftarrow 1, gen_{us} \leftarrow 0$ ▷ gen_{us} stores the number of unsuccessful generations
while $gen < \gamma \wedge gen_{us} < \gamma_{us}$ **do**
    initialize next generation: $children \leftarrow \{\}$
    $sm \leftarrow 0$ ▷ sm stores the number of successful mutations
    for $i = 1$ to λ **do**
        select randomly chosen individual from population: $\mathbf{o} \leftarrow pop[j], j \in \{1, \ldots, \mu\}$.
        create offspring by mutation using Algorithm 3: $\hat{\mathbf{o}} \leftarrow Mutate(\mathbf{o}, R, \Omega_e^{\mathrm{ord}})$
        evaluate offspring: $quality(\hat{\mathbf{o}}) = Evaluate(\hat{\mathbf{o}}, \mathbf{P})$
        if $quality(\hat{\mathbf{o}}) < quality(\mathbf{o})$ **then** $sm \leftarrow sm + 1$
        if $quality(\hat{\mathbf{o}}) < quality(\mathbf{o}) \vee \emptyset = false$ **then**
            $children \leftarrow children \cup \{\hat{\mathbf{o}}\}$
        end if
    end for
    if $sm > \mu/5$ **then**
        $R \leftarrow R \cdot \varsigma$
    else if $sm < \mu/5$ **then**
        $R \leftarrow R/\varsigma$
    end if
    if ψ **then** $children \leftarrow children \cup pop$
    sort $children$ according to quality of elements
    if $quality(children[1]) > quality(pop[1])$ **then**
        $gen_{us} \leftarrow gen_{us} + 1$
    else
        $gen_{us} \leftarrow 0$
    end if
    select μ best children: $p\hat{o}p \leftarrow children_i : i \in \{1, \ldots, \mu\}$.
    if ε **then**
        $p\hat{o}p[\mu] \leftarrow pop[1]$
        sort $p\hat{o}p$ according to quality of elements
    end if
    generational replacement: $pop \leftarrow p\hat{o}p$
    $gen \leftarrow gen + 1$
end while
return $pop[1]$

4 Empirical Tests

In this section we analyze the performance of Algorithm 1. We summarize and discuss its performance for three different sets of problem instances. In the first tests (Subsection 4.1) the input given to the algorithm is the output of a parametrization algorithm executed to a family of 10 randomly chosen (implicitly given) conics. In the second part (Subsection 4.2) we use a collection of 20 randomly generated parametrizations, and we execute the algorithm with different settings of its parameters. In the third part (Subsection 4.3) we apply a non-random approach, we work with a family of parametrizations with low height and apply different Möbius transformations with different heights.

All computations were done on a Intel(R) Core(TM) i7-2630QM with 2.0 GHz and 16 GB RAM using Maple 18. The randomly generated parametrizations that are used in Subsections 4.1 and 4.2 can be found online[2]. The results of the experiments are summarized in the tables collected in the Appendix of this paper.

4.1 Implicit Experiment

In this experiment, we work with a family of parametrizations obtained from a family of implicitly given genus zero curves. More precisely, the data used here have been generated as follows:

- We take a family $\mathcal{F} = \{\mathbf{C}_i\}_{1\leq i\leq 10}$ of 10 conics generated randomly with integer coefficients in $\{-400,\ldots,400\}$. In order to guarantee the existence of rational parametrizations over $\mathbb{Q}$, we force each conic in $\mathcal{F}$ to pass through a rational point P; for simplicity, we take P as the origin. The generated polynomials for $\mathcal{F}$ are

$$\begin{aligned}&\{-3x^2-182xy+26y^2+380x-382y, 135x^2+395xy-289y^2+12x+112y,\\ &-207x^2+188xy+359y^2+198x+335y, 333x^2+246xy+85y^2-257x-214y,\\ &273x^2+359xy-308y^2-382x+47y, -365x^2-47xy+42y^2+98x+59y,\\ &107x^2-237xy-145y^2+226x+105y, -240x^2+238xy+232y^2-257x-133y,\\ &-374x^2+287xy-174y^2-239x+194y, -62x^2+83xy+163y^2-3x+314y\}\end{aligned}$$

- We generated the family of rational parametrizations $\{\mathbf{P}_i(t)\}_{1\leq i\leq 10}$ (see the web link[2]) by applying the parametrization command in Maple to each conic.

Test 4.1. We now apply our algorithm to each parametrization $\mathbf{P}_i(t)$ using the following parameter settings:

$$\begin{array}{llllll} N_0=100, & N=2\,\mathrm{H}(\mathbf{P})^2, & \theta=400, & k=10^2, & \mu=100, \\ \lambda=1000, & \emptyset=false, & \rho=50, & \gamma=100, & \gamma_{us}=20, \\ \psi=true, & \varepsilon=true, & \varsigma=0.9, & w_0=10 & \end{array} \tag{28}$$

[2] http://www3.uah.es/rsendra/data-in-paper-Sendra-StWinkler.pdf

Note that, since the height of the implicit equation of each conic is around 400, we take $\theta = 400$ which means that a parametrization height around 400 can be considered a satisfactory result. By $\mathbf{Q}_i(t)$ we denote the output of our algorithm applied to $\mathbf{P}_i(t)$. In Table 2 we show the heights of each input and output parametrization, the improvement (see definition in (12)), and the consumed runtime (in seconds). We observe that in no case the Maple parametrization algorithm returned an optimal (height) parametrization. Indeed, all heights produced by the Maple command are bigger than the implicit equation height. However, our algorithm improves the height in all cases. Moreover, in all but one, the output height is around the required expected height, namely ~ 400. We observe that the average runtime is 36 seconds.

4.2 Random Parametric Experiments

We consider a rational parametrization $\mathbf{R}(t)$, of a curve in $\mathbb{C}^r$, constructed as follows:

- the degree n of the parametrization is taken randomly among the natural numbers in $\{2, \ldots, 20\}$,
- the coefficients of the parametrization are taken randomly among the natural numbers in $\{-100, \ldots, 100\}$,
- the ambient space dimension r of the curve defined by $\mathbf{R}(t)$ is taken randomly among the natural numbers in $\{2, \ldots, 5\}$.

In addition we introduce a Möbius transformation $\Phi(t)$ whose coefficients are taken ramdonly among the natural numbers in $\{-100, \ldots, 100\}$. Finally we gets the input parametrization

$$\mathbf{P}(t) = \mathbf{R}(\Phi(t)).$$

We use 20 inputs $\{\mathbf{P}_1, \ldots, \mathbf{P}_{20}\}$ as described above.

Test 4.2. We applied our algorithm to each parametrization $\mathbf{P}_i$ using for its parameters the setting described in (28). By $\mathbf{Q}(t)$ we denote the output of our algorithm applied to $\mathbf{P}(t)$. In Table 3 we show the heights of each input and output parametrization, the improvement (see definition in (12)), and the consumed runtime (in seconds). We observe that in all cases the algorithm reached the expectable optimal height, namely ~ 100, and that the consumed runtime varied from 17.66 seconds to less than 4 minutes.

In the second part of this test series we applied the algorithm to the family of inputs $\{\mathbf{P}_1, \ldots, \mathbf{P}_{20}\}$ and used different settings of the parameters μ, λ, ψ, and ϕ. More precisely, we consider the settings:

- $\mu = 100, \lambda = 1000$; results are given in Table 4 in the appendix.
- $\mu = 50, \lambda = 500$; results are given in Table 5 in the appendix.
- $\mu = 10, \lambda = 100$; results are given in Table 6 in the appendix.

- $\mu = 10, \lambda = 50$; results are given in Table 7 in the appendix.

In Subsection 4.4, we analyze the obtained results.

4.3 Non-Random Experiments

In the second part (Tests 4.3, 4.4, 4.5), we use a different, non-random approach. We consider a family of parametrizations with low height and we apply different Möbius transformations with different heights. More precisely, we now consider the five parametrizations

$$\mathbf{R}_m(t) = \left(\frac{t^m + t^2 + 1}{\sum_{j=0}^{m} j t^j}, \frac{t^m + 2t + 1}{\sum_{j=0}^{m} j t^j}, \frac{t^m + 3t^4 + 1}{\sum_{j=0}^{m} j t^j} \right) \text{ with } 6 \leq m \leq 10. \tag{29}$$

Note that $\deg(\mathbf{R}_m(t)) = m$ and that $\mathrm{H}(\mathbf{R}_m) = m$. Hence, for these parametrizations, the degree can be seen as a potential optimal of the height.

We tested our algorithm after performing different changes of parameters. The algorithm was executed 5 times for each case, using the parameter values described in (28). We consider three different type of re-parametrizations:

$$\phi_1(t) = \frac{t - m^2}{t - m}, \ \phi_2(t) = \frac{t - 2^m}{2^m t - 1}, \ \phi_3(t) = \frac{t - 2^m}{t + 2^m}. \tag{30}$$

Observe that the first change will place the answer in integers of value $\mathcal{O}(m^2)$, and the second and the third will place the answer in integers of value $\mathcal{O}(2^m)$, where $m = 6, 7, 8, 9, 10$.

Test 4.3. We manipulate the parametrization $\mathbf{R}_m$ using the Möbius transformation

$$\phi_1(t) = \frac{t - m^2}{t - m} \tag{31}$$

to get the new parametrizations

$$\mathbf{P}_m(t) = \mathbf{R}_m(\phi_1(t)). \tag{32}$$

Observe that the $\deg(\mathbf{P}_m) = \deg_m(\mathbf{R}_m) = m$, but $\mathrm{H}(\mathbf{P}_m)$ is much bigger than $\mathrm{H}(\mathbf{R}_m) = m$. The idea is to apply the main algorithm (Algorithm 1) to $\mathbf{P}_m$ and test how much we can improve the height.

For each parametrization $\mathbf{P}_m$ we ran the algorithm 5 times with input parameters as given in (28). Table 8 shows the heights of the input parametrization $\mathbf{P}_m$ and of the parametrizations $\{\mathbf{Q}_m^j\}_{1 \leq j \leq 5}$ produced by our algorithm, as well as the means and the standard deviations of the achieved improvements (as defined in (12)). We observe that in all cases we get a significant improvement of the height. Moreover, in all cases, with the exception of $m = 7$, the potential optimum was found, namely the degree m.

For each m, we show the achieved transformation $\phi(t) = (at + b)/(ct + d)$ producing the simplified parametrization.

$$\begin{aligned} \mathbf{P}_6 &\mapsto \tfrac{36t+6}{t+1} \mapsto \mathbf{Q}_6^2 & \mathbf{P}_9 &\mapsto \tfrac{9t-81}{t-1} \mapsto \mathbf{Q}_9^3 \\ \mathbf{P}_7 &\mapsto \tfrac{-49t+56}{-t+2} \mapsto \mathbf{Q}_7^5 & \mathbf{P}_{10} &\mapsto \tfrac{10t+100}{t+1} \mapsto \mathbf{Q}_{10}^2 \\ \mathbf{P}_8 &\mapsto \tfrac{-64t+8}{-t+1} \mapsto \mathbf{Q}_8^3 \end{aligned}$$

Test 4.4. In this second test we use the Möbius transformation

$$\phi_2(t) = \frac{t - 2^m}{2^m t - 1} \tag{33}$$

to get the new parametrizations

$$\mathbf{P}_m(t) = \mathbf{R}_m(\phi_2(t)). \tag{34}$$

Again, observe that the $\deg(\mathbf{P}_m) = \deg_m(\mathbf{R}) = m$ but $\mathrm{H}(\mathbf{P}_m)$ is much bigger than $\mathrm{H}(\mathbf{R}_m) = m$. For each parametrization $\mathbf{P}_m$ we run our algorithm (as defined in Algorithm 1) 5 times, with input parameters given in (28). Table 9 shows the heights of the input parametrization $\mathbf{P}_m$ and of the 5 parametrizations $\{\mathbf{Q}_m^j\}_{1\le j\le 5}$ produced by our algorithm, as well as the means and the standard deviations of the achieved improvements (12). We observe that in all cases we get a significant improvement of the height. Moreover, in all cases, with the exception of $m = 8$, we get the potential optimum, namely the degree m. For each m, we show the achieved transformation $\phi_2(t)$ producing the simplified parametrization.

$$\begin{aligned} \mathbf{P}_6 &\mapsto \tfrac{t+64}{64t+1} \mapsto \mathbf{Q}_6^1 & \mathbf{P}_9 &\mapsto \tfrac{521t+1}{t+512} \mapsto \mathbf{Q}_9^1 \\ \mathbf{P}_7 &\mapsto \tfrac{128t-1}{t-128} \mapsto \mathbf{Q}_7^1 & \mathbf{P}_{10} &\mapsto \tfrac{1024t+1}{t+1024} \mapsto \mathbf{Q}_{10}^1 \\ \mathbf{P}_8 &\mapsto \tfrac{86t+171}{171t+86} \mapsto \mathbf{Q}_8^1 \end{aligned}$$

Test 4.5. In this test we used the Möbius transformation

$$\phi_3(t) = \frac{t - 2^m}{t + 2^m} \tag{35}$$

to get the new parametrizations

$$\mathbf{P}_m(t) = \mathbf{R}_m(\phi_3(t)). \tag{36}$$

Again, observe that the $\deg(\mathbf{P}_m) = \deg_m(\mathbf{R}) = m$ but $\mathrm{H}(\mathbf{P}_m)$ is much bigger than $\mathrm{H}(\mathbf{R}_m) = m$. For each parametrization $\mathbf{P}_m$ we run the algorithm 5 times with input parameters given in (28). Table 10 shows the hights of the input parametrization $\mathbf{P}_m$

and of the 5 parametrizations $\{\mathbf{Q}_m^j\}_{1\leq j\leq 5}$ produced by our algorithm as well as the means and the standard deviations of the achieved improvements (12). We observe that in all cases we get a significant improvement of the height. Moreover, in all cases we get the potential optimum, namely the degree m. For each m, we show the achieved transformation $\phi(t)$ producing the simplified parametrization.

$$\mathbf{P}_6 \mapsto \tfrac{-64t-64}{t-1} \mapsto \mathbf{Q}_6^1 \qquad \mathbf{P}_9 \mapsto \tfrac{-512(t-1)}{t+1} \mapsto \mathbf{Q}_9^3$$
$$\mathbf{P}_7 \mapsto \tfrac{-128t-128}{t-1} \mapsto \mathbf{Q}_7^1 \qquad \mathbf{P}_{10} \mapsto \tfrac{-1024(t+1)}{t-1} \mapsto \mathbf{Q}_{10}^1$$
$$\mathbf{P}_8 \mapsto \tfrac{256(t+1)}{t-1} \mapsto \mathbf{Q}_8^1$$

4.4 Discussion

In Test 4.1 we executed our algorithm on a family of conic parametrizations. The first remarkable fact is that the direct application of the parametrization algorithms do not provide optimal parametrizations (i.e., parametrizations with optimal height) and hence it shows a natural frame for the applicability of our algorithm. The reason for this behavior of the parametrization algorithms is that they require the determination of regular rational points [34]. This question can be reduced to the computation of one rational point on a conic, and the height of the output parametrization depends on the height of this rational point. Our second remark is that the execution of our algorithm improves the height in all cases. Moreover, in all but one, the output height is around the required expected height, namely ~ 400. We also observe that the average runtime is 36 seconds.

In Test 4.2 we executed our algorithm first for the setting of parameters described in (28). One can see in Table 3 that the performance of our algorithm is very satisfactory. In all cases, the algorithm reached the expectable optimal height, namely ~ 100. Moreover, the time of executions varied from 17.66 seconds to less than 4 minutes.

In the second part of Test 4.2 we analyze the performance of our algorithm with different settings of the evolutionary search for optimal solutions. We see in Table 4 that setting $\mu = 100$ and $\lambda = 1000$ yields best results; in all cases optimal solutions are found, regardless whether offspring selection is applied or not and regardless whether plus or comma selection is applied. In fact, using smaller populations tends to lead us to worse results: as we see in Table 5, some test runs did not find optimal solutions with $\mu = 50$ and $\lambda = 500$, and also the consumed runtime was not reduced significantly in all cases because it takes the algorithm more generations to converge (and we use convergence as a termination criterion via γ_{us}, the maximum number of unsuccessful consecutive generations). Lowering the population size even more leads to even worse results, as we see in Tables 6 and 7.

Concerning to the second collection of tests, those documented in Subsection 4.3, we also conclude that the execution of the algorithm is satisfactory. Indeed, we observe

that in all cases we get a significant improvement of the height. More precisely, in the worst case, the algorithm reduces

- (in Test 4.3) input parametrizations with coefficients with $\alpha \in \{11, 13, 16, 19, 22\}$ decimal digits to output parametrizations with coefficients with 4 decimal digits,
- (in Test 4.4) input parametrizations with coefficients with $\alpha \in \{12, 16, 21, 26, 32\}$ decimal digits to output parametrizations with coefficients with 4 decimal digits,
- (in Test 4.5) input parametrizations with coefficients with $\alpha \in \{23, 31, 40, 50, 81\}$ decimal digits to output parametrizations with coefficients with 4 decimal digits.

Moreover, in all cases, with the exception of $m = 7$ (in Test 4.3) and $m = 8$ (in Test 4.4), we get the potential optimum, namely the degree of the parameterization.

On the other hand, in Subsection 3.1 we have analyzed the search space for the problem. There we have seen that each search space Space(Ω) is generated by a seed subset $\Omega \subset \mathbb{Z}(N)^2$. The efficiency of our method is due to the adaptation and application of evolutionary techniques, and the reduction of the seed set Ω. In our tests, we have taken N as $2\,\mathrm{H}(\mathbf{P}_m(t))$. Therefore, the direct approach will use the search space Space($\mathbb{Z}(2\,\mathrm{H}(\mathbf{P}_m(t))^2)$, whose cardinality is $\mathcal{O}(\mathrm{H}(\mathbf{P}_m(t))^2)$. Analyzing Tables 8 and 9 we see how huge the height of the input parametrization is, and hence how big the search space is when using the direct approach. In Table 11 we provide the cardinality of the actual seed set used in each case.

5 Conclusion

Parametric curves admit infinitely many different parametrizations. Depending on the chosen one, the applicability of the parametrization varies. There are different criteria for choosing a parametrization. For most of them there exist algorithms and results to be used. Nevertheless, for the case of the arithmetic optimality this is not the case. In this paper we provide an evolutionary algorithm for approximately solving this problem. According to our knowledge, this is the first algorithmic approach to the problem of arithmetic optimality.

Acknowledgments

The authors thank Franz Winkler at the Research Institute for Symbolic Computation, Johannes Kepler University Linz, for his advice. A major part of this work was developed while S. M. Winkler was visiting J.R. Sendra at the Universidad de Alcalá in the frame of the project *Giner de los Rios*. J.R. Sendra is partially supported by *Ministerio de Economía y Competitividad*, and by the European Regional Development Fund (ERDF), under the Project MTM2014-54141-P. Stephan M. Winkler's research

on evolutionary algorithms is supported by the The Austrian Research Promotion Agency (FFG) within the *K-Projekt HOPL* as well as the Department for Medical and Bioinformatics at FH Upper Austria, Hagenberg.

References

[1] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham. *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications.* Chapman & Hall / CRC, 2009.

[2] C. Andradas, T. Recio. Plotting missing points and branches of real parametric curves. Appl Algebra Engrg Comm Comput 2007;18(12):10726.

[3] C. Andradas, T. Recio, J.R. Sendra, L.F. Tabera. On the simplication of the coefficients of a parametrization. Journal of Symbolic Computation vol. 44, pp. 192-210 (2009).

[4] C. Andradas, T. Recio, J.R. Sendra, L.F. Tabera, C. Villarino. Proper Real Reparametrization of Rational Ruled Surfaces. Computer Aided Geometric Design 28 (2011) 102113.

[5] C. Andradas, T. Recio, J.R. Sendra, L.F. Tabera, C. Villarino. Reparametrizing swung surfaces over the reals. Appl Algebra Engrg Comm Comput 2014;25:3965.

[6] C.L. Bajaj, A.V. Royappa. Finite representations of real parametric curves and surfaces. Internat J Comput Geom Appl 1995;5(3):31326.

[7] Hans-Georg Beyer. *The Theory of Evolution Strategies.* Springer, Berlin / Heidelberg / New York 1998.

[8] E. Bomberi, W. Gubler. Heights in Diophantine Geometry. *Cambridge University Press.* 2006.

[9] S.C. Chou, X.S. Gao. On the normal parametrization of curves and surfaces. Internat. J. Comput. Geom. Appl. 1991;1:12536.

[10] G. Grasegger. Radical solutions of first order autonomous algebraic ordinary differential equations, in: K. Nabeshima (Ed.), ISSAC 2014: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2014, pp. 217–223.

[11] D. Hilbert, A. Hurwitz. Über die Diophantischen Gleichungen vom Geschlecht Null. Acta math; 14, 217224 (1890)

[12] M. van Hoeij. Rational parametrization of curves using canonical divisors. Journal of Symbolic Computation; 23, 209227 (1997)

[13] S. Lange. Fundamentals of Diophantine Geometry *Springer-Verlag New York.* 1983

[14] T.-C. Lu, J.-C. Juang. A region-based quantum evolutionary algorithm (RQEA) for global numerical optimizationOriginal Research Article Journal of Computational and Applied Mathematics, Volume 239, 2013, Pages 1-11.

[15] J. McCall. Genetic algorithms for modelling and optimisationOriginal Research Article Journal of Computational and Applied Mathematics, Volume 184, Issue 1, 1 2005, Pages 205-222.

[16] D. Marsh. *Applied geometry for computer graphics and CAD.* 2nd ed. Springer; 2005.

[17] M. Narváez. On the height of a rational parametrization of a plane algebraic curve Proc. EACA2012 pp. 139-142 (2012). Servicios de publicaciones UAH I.S.B.N.:978-84-8138-770-4.

[18] L.X.C. Ngô, J.R. Sendra, F. Winkler. Birational Transformations Preserving Rational Solutions of Algebraic Ordinary Differential Equations. Journal of Computational and Applied Mathematics (2015), pp. 114-127.

[19] L.X.C. Ngô, F. Winkler. Rational general solutions of first order non-autonomous parametrizable ODEs. J. Symbolic Comput. 45 (12) (2010) 14261441.

[20] S. Orevkov Parametric equations of plane sextic curves with a maximal set of double points Journal of Algebra and Its Applications p Vol 14, No 9 (2015) DOl. 10 1142/50219498315400137.

[21] E. Pitzer, M. Affenzeller. A Comprehensive Survey on Fitness Landscape. *Recent Advances in Intelligent Engineering Systems*, Editors: Jnos Fodor, Ryszard Klempous, Carmen Paz Surez Araujo. Springer, 2011, pp. 161-191.

[22] T. Recio, J.R. Sendra, L. F. Tabera, C.Villarino. Generalizing circles over algebraic extensions. Mathematics of Comp utation vol. 79, num. 270, pp. 1067-1089 (2010).

[23] I. Rechenberg. Evolutionsstrategie. Friedrich Frommann Verlag, 1973.

[24] H.-P. Schwefel. Evolutionsstrategie und numerische Optimierung. Technische Universität Berlin, 1975.

[25] J. Schicho. Rational Parametrization of Surfaces. Journal of Symbolic Computation 26, 19 (1998)

[26] T. W. Sederberg. Improperly parametrized rational curves. Computer Aided Geometric Design; 3, 6775 (1986)

[27] J. R. Sendra. Normal parametrizations of algebraic plane curves. J Symbolic Comput 2002;33(6):86385.

[28] J.R. Sendra, D. Sevilla, C. Villarino C. Covering of surfaces parametrized without projective base points. In: Proceeding ISSAC14. ACM Press; 2014. p. 37580. ISBN: 978-1-4503-2501-1.

[29] J.R. Sendra, D. Sevilla, C. Villarino C. In: Schicho J, Weimann M, Gutierrez J, editors. Some results on the surjectivity of surface parametrizations. LNCS, vol. 8942. Springer; 2015. p. 192203.

[30] J.R. Sendra, D. Sevilla, C. Villarino C. Missing sets in rational parametrizations of surfaces of revolution. Computer-Aided Design 66 (2015) 55-61.

[31] J.R. Sendra, S. M. Winkler. Optimization of Coefficients of Lists of Polynomials by Evolutionary Algorithms. Annales Mathematicae et Informaticae 44 (2015) pp. 177-185

[32] J.R. Sendra, F. Winkler. Parametrization of Algebraic Curves over Optimal Field Extensions. Journal of Symbolic Computation; 23, 191207 (1997)

[33] J.R. Sendra, F. Winkler. Tracing Index of Rational Curve Parametrizations. Computer Aided Geometric Design vol. 18/8, pp. 771-795 (2001)

[34] J.R. Sendra, F. Winkler, S. Pérez-Díaz. Rational Algebraic Curves: A Computer Algebra Approach. *Springer-Verlag Heidelberg. In series Algorithms and Computation in Mathematics.* Volume 22. 2007

[35] F. Winkler. Polynomial Algorithms in Computer Algebra. *Springer-Verlag, Wien New York,* 1996.

Appendix: Tables of the tests in Section 4

	$H(\mathbf{P}_i)$	$H(\mathbf{Q}_i)$	Impr_{10}	Time
$\mathbf{P}_1(t)$	128440	380	1	15.99
$\mathbf{P}_2(t)$	47708	523	0.67	36.739
$\mathbf{P}_3(t)$	6179051637	74668	1	51.667
$\mathbf{P}_4(t)$	54355590	231	1.67	35.35
$\mathbf{P}_5(t)$	85453560192	473	2.67	57.533
$\mathbf{P}_6(t)$	63067620	243	1.67	41.746
$\mathbf{P}_7(t)$	10348505	343	1.67	34.445
$\mathbf{P}_8(t)$	3340800	532	1.3	32.994
$\mathbf{P}_9(t)$	122668260	342	2	33.914
$\mathbf{P}_1(t)$	7640136	320	1.3	17.769

Table 2: Results of Test 4.1 in Subsection 4.1.

In Tables 4, 5, 6, 7, we have emphasized with bold face style when the result has height bigger than expected. In addition, the processes were aborted when they took more than 900 second of CPU execution; this is expressed in the tables as > 900.

	$\mathbb{C}^r$	$\deg(\mathbf{P})$	$\mathrm{H}(\mathbf{P})$	$\mathrm{H}(\mathbf{Q})$	Impr_{10}	Time
$\mathbf{P}_1$	$r=4$	13	204736329239648870805142382813249 28	93	16.5	137.624
$\mathbf{P}_2$	$r=2$	14	14327349120521962049851258346394869760	100	11.6	79.951
$\mathbf{P}_3$	$r=5$	18	23835451069101336970437962951450836 1018181504	100	14.	148.076
$\mathbf{P}_4$	$r=5$	2	349874	99	2.	17.660
$\mathbf{P}_5$	$r=2$	9	2827199650699677729492	98	10.	27.768
$\mathbf{P}_6$	$r=2$	11	179415500145130976754852	99	11.	51.402
$\mathbf{P}_7$	$r=4$	3	99712956	99	3.	16.427
$\mathbf{P}_8$	$r=5$	11	114596398334900456757657 60	99	12.	95.036
$\mathbf{P}_9$	$r=2$	14	13732461167434626494051719	99	12.	76.425
$\mathbf{P}_{10}$	$r=5$	12	155812617201464671339552964608	94	14.	110.980
$\mathbf{P}_{11}$	$r=3$	15	97391256589092147972234745 60	95	13.	105.051
$\mathbf{P}_{12}$	$r=2$	13	10449902240826825838981688 4375	81	14.	63.758
$\mathbf{P}_{13}$	$r=3$	15	14468233071912452686951794965 7418797	98	17.	70.731
$\mathbf{P}_{14}$	$r=3$	19	13882371944970067330719963341 28305280	92	17.5	224.455
$\mathbf{P}_{15}$	$r=3$	6	217962317958120	96	6.5	115.956
$\mathbf{P}_{16}$	$r=2$	16	17387870899587518734596104415 8054400	89	17.	111.338
$\mathbf{P}_{17}$	$r=2$	7	12758128747585929	97	7.5	21.014
$\mathbf{P}_{18}$	$r=2$	19	33983437729266597841239819191 401076220	87	18.	79.373
$\mathbf{P}_{19}$	$r=5$	16	12771484560624808234894591668 95616	98	16.	139.434
$\mathbf{P}_{20}$	$r=2$	19	72041875367087233119811624862 8172305818123	97	20.	86.143

Table 3: Results of the first experiment of Test 4.2 in Subsection 4.2.

$\mu = 100$ $\lambda = 1000$	$\psi = T,\ \phi = T$ [H(**Q**), Time]	$\psi = T,\ \phi = F$ [H(**Q**), Time]	$\psi = F,\ \phi = T$ [H(**Q**), Time]	$\psi = F,\ \phi = F$ [H(**Q**), Time]
$\mathbf{P}_1$	[93, 137.624]	[93, 134.255]	[93, 130.682]	[93, 119.544]
$\mathbf{P}_2$	[100, 79.951]	[100, 80.247]	[100, 79.966]	[100, 84.756]
$\mathbf{P}_3$	[100, 148.076]	[100, 148.700]	[100, 151.196]	[100, 153.895]
$\mathbf{P}_4$	[99, 17.660]	[99, 17.675]	[99, 17.597]	[99, 19.594]
$\mathbf{P}_5$	[98, 27.768]	[98, 27.456]	[98, 28.283]	[98, 47.845]
$\mathbf{P}_6$	[99, 51.402]	[99, 50.825]	[99, 51.527]	[99, 52.931]
$\mathbf{P}_7$	[99, 16.427]	[99, 16.677]	[99, 16.567]	[99, 17.457]
$\mathbf{P}_8$	[99, 95.036]	[99, 94.786]	[99, 208.823]	[99, 100.105]
$\mathbf{P}_9$	[99, 76.425]	[99, 74.693]	[99, 77.127]	[99, 79.140]
$\mathbf{P}_{10}$	[94, 110.980]	[94, 108.296]	[94, 107.984]	[94, 108.249]
$\mathbf{P}_{11}$	[95, 105.051]	[95, 105.223]	[95, 153.412]	[95, 92.727]
$\mathbf{P}_{12}$	[81, 63.758]	[81, 61.059]	[81, 63.383]	[81, 67.393]
$\mathbf{P}_{13}$	[98, 70.731]	[98, 67.689]	[98, 71.043]	[98, 101.837]
$\mathbf{P}_{14}$	[92, 224.455]	[92, 199.338]	[92, 412.888]	[92, 213.721]
$\mathbf{P}_{15}$	[96, 115.956]	[96, 36.473]	[96, 107.703]	[96, 41.044]
$\mathbf{P}_{16}$	[89, 111.338]	[89, 104.427]	[89, 100.652]	[89, 104.255]
$\mathbf{P}_{17}$	[97, 21.014]	[97, 20.389]	[97, 21.278]	[97, 22.854]
$\mathbf{P}_{18}$	[87, 79.373]	[87, 79.935]	[87, 78.203]	[87, 80.076]
$\mathbf{P}_{19}$	[98, 139.434]	[98, 138.155]	[98, 142.071]	[98, 143.864]
$\mathbf{P}_{20}$	[97, 86.143]	[97, 86.658]	[97, 86.549]	[97, 89.514]

Table 4: Results for Test 4.2 achieved with $\mu = 100$, $\lambda = 1000$; $T = true, F = false$.

$\mu = 50$ $\lambda = 500$	$\psi = T,\ \phi = T$ [H(**Q**), Time]	$\psi = T,\ \phi = F$ [H(**Q**), Time]	$\psi = F,\ \phi = T$ [H(**Q**), Time]	$\psi = F,\ \phi = F$ [H(**Q**), Time]
$\mathbf{P}_1$	[93, 73.305]	[93, 75.723]	[93, 74.132]	[93, 75.442]
$\mathbf{P}_2$	[100, 41.808]	[100, 46.021]	[100, 44.756]	[100, 44.585]
$\mathbf{P}_3$	[100, 79.686]	[100, 80.371]	[100, 78.360]	[100, 77.501]
$\mathbf{P}_4$	[235, 9.236]	[99, 9.969]	[235, 9.423]	[235, 9.719]
$\mathbf{P}_5$	[98, 25.880]	[98, 27.799]	[98, 26.255]	[98, 26.582]
$\mathbf{P}_6$	[99, 28.455]	[99, 30.670]	[99, 29.312]	[99, 29.407]
$\mathbf{P}_7$	**[1318, 550.902]**	**[1318, 554.038]**	[99, 8.892]	**[608, 541.120]**
$\mathbf{P}_8$	[99, 53.696]	[99, 55.520]	[99, 55.194]	[99, 54.943]
$\mathbf{P}_9$	[99, 47.222]	[99, 50.436]	[99, 39.998]	[99, 39.952]
$\mathbf{P}_{10}$	[94, 60.793]	[94, 63.383]	[94, 41.169]	[94, 60.575]
$\mathbf{P}_{11}$	[95, 47.877]	[95, 51.043]	[95, 55.240]	[95, 83.882]
$\mathbf{P}_{12}$	[81, 35.755]	[81, 38.299]	[81, 35.053]	[81, 35.943]
$\mathbf{P}_{13}$	[98, 35.412]	[98, 38.922]	[98, 37.581]	[98, 37.128]
$\mathbf{P}_{14}$	[92, 122.165]	[92, 238.869]	[92, 102.851]	[92, 221.959]
$\mathbf{P}_{15}$	[96, 21.310]	[96, 23.212]	[96, 19.423]	[96, 22.511]
$\mathbf{P}_{16}$	[89, 54.320]	[89, 59.374]	[89, 54.491]	[89, 54.116]
$\mathbf{P}_{17}$	[97, 12.183]	[97, 12.496]	[97, 10.857]	[97, 12.028]
$\mathbf{P}_{18}$	[87, 40.420]	[87, 42.448]	[87, 37.534]	[87, 41.122]
$\mathbf{P}_{19}$	[98, 76.753]	[98, 71.323]	[98, 70.887]	[98, 77.516]
$\mathbf{P}_{20}$	[97, 43.586]	[97, 42.136]	[97, 39.234]	[97, 44.976]

Table 5: Results for Test 4.2 achieved with $\mu = 50$, $\lambda = 500$; $T = true, F = false$.

$\mu = 10$ $\lambda = 100$	$\psi = T,\ \phi = T$ [H(**Q**), Time]	$\psi = T,\ \phi = F$ [H(**Q**), Time]	$\psi = F,\ \phi = T$ [H(**Q**), Time]	$\psi = F,\ \phi = F$ [H(**Q**), Time]
$\mathbf{P}_1$	$>$ **900**	[93, 30.373]	[93, 30.608]	[93, 38.376]
$\mathbf{P}_2$	[100, 80.778]	[100, 18.361]	[100, 18.034]	[100, 26.271]
$\mathbf{P}_3$	[100, 176.952]	[100, 39.530]	[100, 23.556]	[100, 34.195]
$\mathbf{P}_4$	[99, 14.103]	[99, 3.526]	[99, 3.323]	[99, 4.977]
$\mathbf{P}_5$	[98, 52.261]	$>$ **900**	[98, 20.093]	$>$ **900**
$\mathbf{P}_6$	$>$ **900**	[99, 21.840]	[99, 12.558]	[99, 19.079]
$\mathbf{P}_7$	$>$ **900**	**[891, 760.692]**	**[891, 586.439]**	**[1318, 681.459]**
$\mathbf{P}_8$	[99, 119.528]	[99, 32.666]	[99, 23.057]	[99, 28.267]
$\mathbf{P}_9$	[99, 93.305]	[99, 26.364]	[99, 31.262]	[99, 20.312]
$\mathbf{P}_{10}$	[94, 127.640]	[94, 34.398]	[94, 23.915]	[94, 30.326]
$\mathbf{P}_{11}$	[95, 194.315]	[95, 54.428]	[95, 37.347]	$>$ **900**
$\mathbf{P}_{12}$	[81, 78.016]	[81, 21.373]	[81, 15.226]	[81, 19.016]
$\mathbf{P}_{13}$	[98, 65.895]	[98, 19.328]	[98, 12.121]	[98, 15.819]
$\mathbf{P}_{14}$	[92, 468.408]	$>$ **900**	[92, 60.248]	$>$ **900**
$\mathbf{P}_{15}$	[96, 43.650]	[96, 15.178]	[96, 9.063]	[96, 13.728]
$\mathbf{P}_{16}$	[89, 106.908]	[89, 30.873]	[89, 21.278]	[89, 27.269]
$\mathbf{P}_{17}$	[97, 19.126]	[97, 6.178]	[97, 4.010]	[97, 4.930]
$\mathbf{P}_{18}$	[87, 71.136]	[87, 18.299]	[87, 12.573]	[87, 16.832]
$\mathbf{P}_{19}$	$>$ **900**	[98, 43.181]	[98, 29.703]	[98, 37.097]
$\mathbf{P}_{20}$	[97, 23.666]	[97, 32.043]	[97, 22.058]	[97, 27.674]

Table 6: Results for Test 4.2 achieved with $\mu = 10$, $\lambda = 100$; $T = true, F = false$.

$\mu = 10$ $\lambda = 50$	$\psi = T,\ \phi = T$ [H(**Q**), Time]	$\psi = T,\ \phi = F$ [H(**Q**), Time]	$\psi = F,\ \phi = T$ [H(**Q**), Time]	$\psi = F,\ \phi = F$ [H(**Q**), Time]
$\mathbf{P}_1$	[93, 41.434]	$>$ **900**	[93, 82.946]	[93, 32.386]
$\mathbf{P}_2$	[100, 17.612]	[100, 22.636]	[100, 47.705]	[100, 18.892]
$\mathbf{P}_3$	[100, 20.780]	[100, 103.959]	[100, 52.790]	[100, 18.860]
$\mathbf{P}_4$	[99, 3.385]	**[262, 4.353]**	[99, 8.705]	[99, 3.370]
$\mathbf{P}_5$	[98, 22.105]	$>$ **900**	$>$ **900**	[98, 22.994]
$\mathbf{P}_6$	$>$ **900**	[99, 19.562]	[99, 49.796]	[99, 22.807]
$\mathbf{P}_7$	**[1318, 863.341]**	[99, 10.920]	[99, 10.125]	**[608, 812.687]**
$\mathbf{P}_8$	$>$ **900**	[99, 65.162]	[99, 79.935]	[99, 28.969]
$\mathbf{P}_9$	[99, 59.452]	[99, 26.457]	[99, 62.104]	[99, 18.814]
$\mathbf{P}_{10}$	[94, 37.581]	[94, 37.987]	[94, 81.277]	[94, 25.085]
$\mathbf{P}_{11}$	$>$ **900**	[95, 59.265]	[95, 130.542]	[95, 24.039]
$\mathbf{P}_{12}$	[81, 18.471]	[81, 25.272]	[81, 52.791]	[81, 19.968]
$\mathbf{P}_{13}$	[98, 17.409]	[98, 17.472]	[98, 39.343]	[98, 10.468]
$\mathbf{P}_{14}$	$>$ **900**	[92, 113.085]	[92, 107.937]	$>$ **900**
$\mathbf{P}_{15}$	[96, 9.859]	$>$ **900**	[96, 31.637]	[96, 9.391]
$\mathbf{P}_{16}$	$>$ **900**	[89, 22.027]	$>$ **900**	[89, 21.544]
$\mathbf{P}_{17}$	[97, 7.987]	[97, 13.463]	$>$ **900**	[97, 3.776]
$\mathbf{P}_{18}$	[87, 8.565]	[87, 19.516]	[87, 45.022]	[87, 9.422]
$\mathbf{P}_{19}$	[98, 81.215]	$>$ **900**	[98, 116.159]	[98, 28.923]
$\mathbf{P}_{20}$	[97, 21.154]	$>$ **900**	[97, 80.138]	[97, 19.126]

Table 7: Results for Test 4.2 achieved with $\mu = 10$, $\lambda = 50$; $T = true, F = false$.

m	$\mathrm{H}(\mathbf{P}_m)$	$[\mathrm{H}(\mathbf{Q}_m^j)]_{j=1,\dots,5}$	$\mathrm{Impr}_{10}(\mathbf{P}_m, \mathbf{Q}_m^j)[\mu \pm \sigma]$
6	15150416256 (1.5×10^{10})	$[1734, 6, 24, 24, 642]$	4.683 ± 3.202
7	5406945213124 (5.4×10^{12})	$[45, 880, 626, 752, 45]$	4.2 ± 1.187
8	2527530405855232 (2.5×10^{15})	$[2560, 85, 8, 8, 8]$	11 ± 5.657
9	1498601124347956605 (1.5×10^{18})	$[3186, 639, 9, 802, 9]$	10.083 ± 7.256
10	1098765432100000000000 (1.1×10^{21})	$[308, 10, 308, 10, 10]$	8.533 ± 2.008

Table 8: Results of Test 4.3 in Subsection 4.3.

m	$\mathrm{H}(\mathbf{P}_m)$	$[\mathrm{H}(\mathbf{Q}_m^j)]_{j=1,\dots,5}$	$\mathrm{Impr}_{10}(\mathbf{P}_m, \mathbf{Q}_m^j)[\mu]$
6	417753473088 (4.2×10^{11})	$[6, 6, 6, 6, 6]$	11
7	3967210831773824 (4.0×10^{15})	$[7, 7, 7, 7, 7]$	15
8	148080050112590643456 (1.5×10^{20})	$[8003, 8003, 8003, 8003, 8003]$	4.25
9	217985083567931284882723 84 (2.2×10^{25})	$[9, 9, 9, 9, 9]$	25
10	126876571423134838624248460 74880 (1.3×10^{31})	$[10, 10, 10, 10, 10]$	15

Table 9: Results of Test 4.4 in Subsection 4.3.

m	$\mathrm{H}(\mathbf{P}_m)$	$[\mathrm{H}(\mathbf{Q}_m^j)]_{j=1,\dots,5}$	$\mathrm{Impr}_{10}(\mathbf{P}_m, \mathbf{Q}_m^j)[\mu \pm \sigma]$
6	23611832414348226068480 (2.4×10^{22})	$[6, 4374, 1216, 6, 64]$	12.8 ± 8.72
7	12676506002282294014967032 05376 (1.3×10^{30})	$[7, 7, 7, 7, 7]$	30
8	17014118346046923173168730 37158841057280 (1.7×10^{39})	$[8, 8, 8, 8, 8]$	39
9	29230032746618058364073696 65432566039311 8650859520 (2.9×10^{49})	$[9, 9, 9, 9, 9]$	49
10	80346902212949513777098104 61705813012611 014968913964176506880 (8.0×10^{60})	$[10, 10, 10, 10, 10]$	29.5

Table 10: Results of Test 4.5 in Subsection 4.3.

Average runtime , $\#(\Omega_e)$						
Test		m=6	m=7	m=8	m=9	m=10
4.3	time	46.5	51.16	51.16	73.12	65.57
	$\#(\Omega_e)$	*9250*	*1320*	*16320*	*20880*	*25400*
4.4	time	28.03	28.85	37.8	91.24	160.96
	$\#(\Omega_e)$	*13680*	*5960*	*4720*	*18000*	*29480*
4.5	time	36.92	63.53	61.67	68.57	79.21
	$\#(\Omega_e)$	*1000*	*14120*	*13720*	*16280*	*18120*

Table 11: Average time of the computations and cardinality of the used seed sets in Tests 4.3, 4.4, and 4.5.