Annales Mathematicae et Informaticae
44 (2015) pp. 177–185
http://ami.ektf.hu

Optimization of coefficients of lists of polynomials by evolutionary algorithms

J. Rafael Sendra[a], Stephan M. Winkler[b]

[a]University of Alcalá, Department of Physic and Mathematics
Rafael.Sendra@uah.es

[b]Upper Austria University of Applied Sciences,
Heuristic and Evolutionary Algorithms Laboratory
Stephan.Winkler@fh-hagenberg.at

Submitted July 30, 2014 — Accepted December 12, 2014

Abstract

We here discuss the optimization of coefficients of lists of polynomials using evolutionary computation. The given polynomials have 5 variables, namely t, a_1, a_2, a_3, a_4, and integer coefficients. The goal is to find integer values α_i, with $i \in \{1, 2, 3, 4\}$, substituting a_i such that, after crossing out the gcd (greatest common divisor) of all coefficients of the polynomials, the resulting integers are minimized in absolute value. Evolution strategies, a special class of heuristic, evolutionary algorithms, are here used for solving this problem. In this paper we describe this approach in detail and analyze test results achieved for two benchmark problem instances; we also show a visual analysis of the fitness landscapes of these problem instances.

Keywords: Optimization of parametrizations, symbolic computation, evolutionary computation, evolution strategies.

MSC: 65K10, 68T05, 68W30

1. Problem statement

In this section, trying to avoid as much as possible mathematical technicalities, we describe the problem, and we explain its interest in the field of mathematics.

The problem statement. We are given a list with infinitely many (at least 3) non-constant polynomials ($L = [p_1; p_2; \dots; p_n]$). These polynomials have 5 variables, namely t, a_1, a_2, a_3, a_4, and integer coefficients. The problem consists in finding integer values α_1, α_2, α_3, and α_4 for a_1, a_2, a_3, and a_4 such that:

1. $\alpha_1\alpha_4 - \alpha_2\alpha_3 \neq 0$
2. We substitute $a_1 = \alpha_1$, $a_2 = \alpha_2$, $a_3 = \alpha_3$, $a_4 = \alpha_4$, in L. This yields $L'(\alpha)$, a list of polynomials with one variable, namely t, and integer coefficients. We cross out the greatest common divisor (gcd) of all non-zero coefficients of the polynomials in $L'(\alpha)$ to get a new list $L''(\alpha)$.

The goal is to find that substitution $a_i = \alpha_i$, $i \in \{1, 2, 3, 4\}$, so that the maximum of the absolute values of all the coefficients of all polynomials in $L''(\alpha)$ is minimum.

An illustrating example. We consider the list with three polynomials $L = [p_1; p_2; p_3]$ where

$$\begin{aligned}
p_1(t) = {} & 13923t^2{a_1}^2 + 5474t^2a_1a_3 - 1904t^2{a_3}^2 + 27846ta_1a_2 + 5474ta_1a_4 \\
& + 5474ta_2a_3 - 3808ta_3a_4 + 13923{a_2}^2 + 5474a_2a_4 - 1904{a_4}^2 \\
p_2(t) = {} & 7564t^2{a_1}^2 - 10298t^2a_1a_3 - 990t^2{a_3}^2 + 15128ta_1a_2 - 10298ta_1a_4 \\
& - 10298ta_2a_3 - 1980ta_3a_4 + 7564{a_2}^2 - 10298a_2a_4 - 990{a_4}^2 \\
p_3(t) = {} & 15845t^2{a_1}^2 - 106t^2a_1a_3 + 2146t^2{a_3}^2 + 31690ta_1a_2 - 106ta_1a_4 \\
& - 106ta_2a_3 + 4292ta_3a_4 + 15845{a_2}^2 - 106a_2a_4 + 2146{a_4}^2
\end{aligned}$$

If we substitute $a_1 = 1, a_2 = 1, a_3 = 1, a_4 = 1$ we get

$L'(\alpha) = [17493t^2 + 34986t + 17493; -3724t^2 - 7448t - 3724; 17885t^2 + 35770t + 17885]$

Since $\gcd(17493, 34986, 17493, -3724, -7448, -3724, 17885, 35770, 17885) = 49$, one gets $L''(\alpha) = 1/49L'(\alpha)$, that is

$$L''(\alpha) = [357t^2 + 714t + 357; -76t^2 - 152t - 76; 365t^2 + 730t + 365]$$

and the maximum, in absolute value, is 730. However, if we take $a_1 = 45, a_2 = 11, a_3 = 31, a_4 = -122$ the new list is

$$L'(\alpha) = [34000561t^2 - 34000561; 68001122t; 34000561t^2 + 34000561].$$

The corresponding gcd is now 34000561. Therefore

$$L''(\alpha) = \frac{1}{34000561} L'(\alpha) = [t^2 - 1; 2t; t^2 + 1]$$

whose maximum, in absolute value, is 2.

The mathematical origin of the problem. This optimization question, we are dealing with, comes from a central problem in the field of the symbolic computation of algebraic curves (see [6] for further details), appears in many computational aspects of the practical applications of curves and is, to our knowledge, not solved. Let us first motivate the problem: In many practical applications, such as in computed aided geometric design, in physics, etc., one deals with parametric representations of a curve. For instance, if we have to compute a line integral along an arc of the curve of equation $y^3 = x^2$, we might use the parametric representation $x = t^3, y = t^2$ of the curve. In general a rational parametrization of a curve, say for simplicity planar, is a nonconstant pair

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

where p_1, p_2, q are polynomials in the variable t. The difficulty here is the following: If we replace t by a polynomial or by a rational function, then we get another parametrization of the same object; for instance, in the example above, $(1000t^3, 100t^2)$ and $((t^2+1)^3, (t^2+1)^2)$ are also parametrizations of $y^3 = x^2$. Thus, we have infinitely many possibilities, but some parametrizations are more complicated and increase the computational time when using them. The question is how to choose the simplest parametrization. Achieving an optimal degree in the polynomials is solved by means of symbolic deterministic algorithms (see [6]). However, the question of determining a parametrization with the smallest (in absolute value) integer coefficients is open. Here, in this paper, we show how to approach the problem by means of evolutionary algorithms.

In order to translate the original parametrization problem into the the problem stated above, we use Lüroth's theorem that establishes how all parametrizations, with optimal degree, are related. More precisely, if $\mathcal{P}(t) = (\frac{P_1(t)}{Q(t)}, \frac{P_2(t)}{Q(t)})$ is a parametrization with optimal degree and integer coefficients, then all the other parametrizations with optimal degree and integer coefficients are of the form

$$\mathcal{P}\left(\frac{a_1t+a_2}{a_3t+a_4}\right) = \left(\frac{P_1\left(\frac{a_1t+a_2}{a_3t+a_4}\right)}{Q\left(\frac{a_1t+a_2}{a_3t+a_4}\right)}, \frac{P_2\left(\frac{a_1t+a_2}{a_3t+a_4}\right)}{Q\left(\frac{a_1t+a_2}{a_3t+a_4}\right)}\right),$$

where a_1, a_2, a_3, a_4 are integers such that $a_1a_4 - a_2a_3 \neq 0$. Simplifying this expression, we get the three polynomials in the variables t, a_1, a_2, a_3, a_4.

Revisiting the illustrating problem. We are given the parametrization

$$\mathcal{P}(t) = \left(\frac{13923\,t^2 + 5474\,t - 1904}{15845\,t^2 - 106\,t + 2146}, \frac{7564\,t^2 - 10298\,t - 990}{15845\,t^2 - 106\,t + 2146}\right)$$

Performing the formal substitution $t = \frac{a_1t+a_2}{a_3t+a_4}$, simplifying expressions and collecting numerators and denominators in a list we get the list $L = [p_1; p_2; p_3]$ of the three polynomials shown above. Now, after taking $a_1 = 45, a_2 = 11, a_3 = 31, a_4 = -122$, we get the parametrization

$$\mathcal{P}\left(\frac{45t+11}{31t-122}\right) = \left(\frac{t^2-1}{t^2+1}, \frac{2t}{t^2+1}\right).$$

The parametrization in this example corresponds to the unit circle $x^2 + y^2 = 1$.

2. Parameter optimization by evolutionary algorithms

Evolution strategies (ES; [4], [5]), beside genetic algorithms (GA; [2], [1]) the second major representative of evolutionary computation, are here used for optimizing $\alpha_1, \ldots, \alpha_4$. ES are population based, i.e., each optimization process works with a population of potential solution candidates that are initially created randomly and then iteratively optimized. In each generation, new solution candidates are generated by randomly selecting parent individuals and forming new individuals applying mutation and (optionally) crossover operators; λ children are produced by μ parent individuals.

By offspring selection, the best children are chosen and become the parents of the next generation. Typically, parent selection in ES is performed randomly with no regard to fitness; survival in ESs simply saves the μ best individuals, which is only based on the relative ordering of their fitness values. Basically, there are two selection strategies for ESs:

- The (μ, λ)-strategy ("comma selection"): μ parents produce λ children; the best μ children are selected and form the next generation's parents.

- The $(\mu+\lambda)$-strategy ("plus selection"): μ parents produce λ offspring; parents and children form a pool of potential new parents, and the best μ individuals are selected from this pool to become the next generation's parents.

Thus, the main driving forces of optimization in ESs are offspring selection and mutation. For the optimization of vectors of real values, mutation is usually implemented as additive Gaussian perturbation with zero mean or multiplicative Gaussian perturbation with mean 1.0. Mutation strength control [4] is based on the quotient of the number of the successful mutants (i.e., those that are better than their parents): If this quotient is greater than 1/5, then the mutation variance is to be increased; if the quotient is less than 1/5, the mutation variance should be reduced.

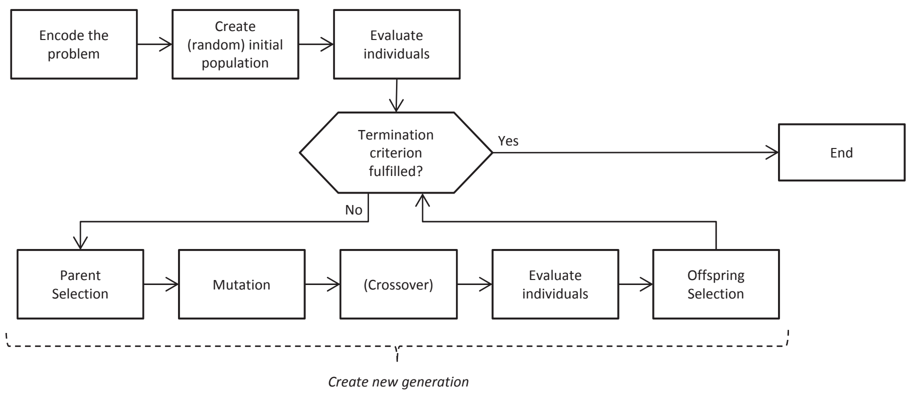

Figure 1: The main workflow of an evolution strategy

3. Test series

3.1. Problem instances

We have used the following two test instances:

- The example (*Ex1*) introduced in Section 1
- The second example (*Ex2*) is defined as $L2 = [p1; p2; p3]$ with

$$\begin{aligned}
p_1 &= 1685t^2{a_1}^2 + 2252t^2a_1a_3 + 769t^2{a_3}^2 + 3370ta_1a_2 + 2252ta_1a_4\\
&\quad + 2252ta_2a_3 + 1538ta_3a_4 + 1685{a_2}^2 + 2252a_2a_4 + 769{a_4}^2\\
p_2 &= -627t^2a1^2 - 1148t^2a_1a_3 - 481t^2{a_3}^2 - 1254ta_1a_2 - 1148ta_1a_4\\
&\quad - 1148ta_2a_3 - 962ta_3a_4 - 627{a_2}^2 - 1148a_2a_4 - 481{a_4}^2\\
p_3 &= 2467t^2{a_1}^2 + 3235t^2a_1a_3 + 1069t^2{a_3}^2 + 4934ta_1a_2 + 3235ta_1a_4\\
&\quad + 3235ta_2a_3 + 2138ta_3a_4 + 2467{a_2}^2 + 3235a_2a_4 + 1069{a_4}^2
\end{aligned}$$

For this example, taking $\alpha_1 = -25, \alpha_2 = 12, \alpha_3 = 34, \alpha_4 = -23$ (note that $\alpha_1\alpha_4 - \alpha_2\alpha_3 = 167 \neq 0$) we get

$$L' = [27889t^2 + 27889; 27889t^2 - 27889; 27889t^2 + 27889t + 27889].$$

The gcd is 27899 and $L'' = [t^2 + 1; t^2 - 1; t^2 + t + 1]$ and the maximum of the absolute values is 1, which is clearly optimal.

3.2. Algorithm configurations

The following 10 algorithm variants have been used for solving the problems defined in the previous section:

	Population size (μ)	Number of children (λ)	Selection mechanism
Settings 1	100	1,000	comma
Settings 2	100	10,000	comma
Settings 3	100	1,000	plus
Settings 4	100	10,000	plus
Settings 5	1,000	10,000	comma
Settings 6	1,000	100,000	comma
Settings 7	1,000	10,000	plus
Settings 8	1,000	100,000	plus
Settings 9	10,000	100,000	comma
Settings 10	10,000	100,000	plus

Table 1: Algorithm parameter settings used for solving the here discussed coefficients optimization problem.

The range of values for initial solution candidates was set to ± 200. For creating offspring we have used multiplicative mutation: The average value of the multiplication factors μ was set to 1.0, the standard deviation σ was initially set to 1.0 and according to the 1/5 success rule updated after each generation (with multiplicative factor / divisor 0.9).

3.3. Results

We have executed ES test series using all parameter configurations defined previously; each algorithm configuration was executed 5 times independently, and for guaranteeing a fair comparison of results the maximum number of evaluations used as termination criterion was set to 1,000,000. Thus, the number of generations executed was not equal for all test configurations.

The results achieved in these test series are summarized in Table 2.

We see that the success rate for small populations is very low, when using bigger populations (with size 1,000 or 10,000) the results are significantly better; when using populations of size 1,000, then significantly better results are achieved using higher selection pressure, i.e. selecting the 1,000 best out of 100,000 offspring each generation.

Problem *Ex1* seems to be harder for the algorithm than *Ex2*. For *Ex1* the algorithm was able to find the optimal solution at least once using settings 8 and 10; for *Ex2* the algorithm was successful in finding the optimum in 4 or 5 out of 5 runs using the settings 6, 8, 9, and 10.

	Problem instance *Ex1*	Problem instance *Ex2*
Settings 1	4807.4	483.6
Settings 2	1270.6	402.2
Settings 3	2136.6	671.0
Settings 4	438.2	3.8
Settings 5	854.4	110.6
Settings 6	160.0	1.0
Settings 7	120.4	21.2
Settings 8	58.2	1.2
Settings 9	230.8	1.4
Settings 10	35.4	1.6

Table 2: Test results. For each algorithm configuration we give the average result qualities achieved for problem instances *Ex1* and *Ex2*.

Fitness Landscape analysis [3] methods can be used for estimating an optimization problem's hardness. As we see in Figures 2 and 3, the fitness landscape of the here used problem instances *Ex1* and *Ex2* are very rugged, which makes it very hard for optimization algorithms to find optimal solutions.

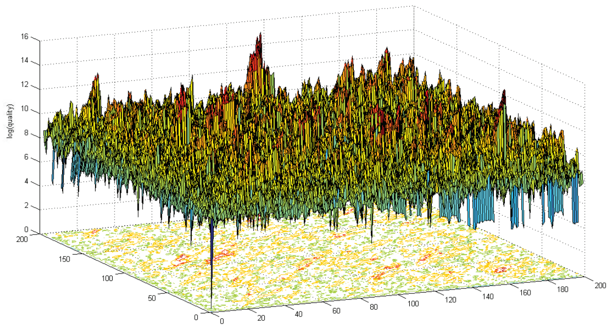

Figure 2: Fitness landscape analysis for example *Ex1*. We have created 40,000 solution candidates for *Ex1* that are arranged on the x-y-plane; the optimal solution discussed in Section 1 ($a_1 = 45, a_2 = 11, a_3 = 31, a_4 = -122$) is positioned at (1,1), and at all other cells are assigned solution candidates that are produced by mutating one of their neighbors (using $\sigma = 1.0$). On the z-axis we draw the fitness of the so created solution candidates for *Ex1*. We see high fluctuations of the fitness values which indicates that fitness values of neighboring solutions vary significantly.

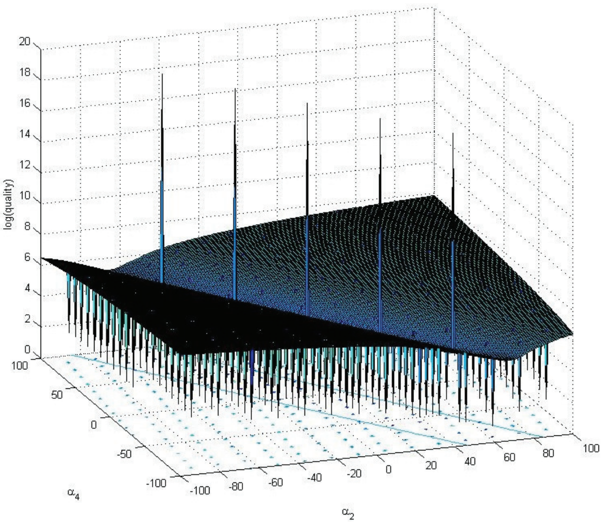

Figure 3: Fitness landscape analysis for example *Ex2*. All possible solution candidates for the here used problem with α_1 and α_3 set optimally ($\alpha_1 = -25$, $\alpha_3 = -34$) are created, their fitness is drawn on the z-axis. We see that even when setting two of four parameters optimally, the resulting fitness landscape is very rugged.

4. Conclusion, outlook

Future work will concentrate on the improvement of mutation and selection operators for this problem class in order to solve problem instances involving significantly bigger coefficients. Additionally, we are working on strategies to decrease the search space. We are also working on the integration of the here discussed class of problems in HeuristicLab [7], a framework for heuristic and evolutionary algorithms that is developed by members of the Heuristic and Evolutionary Algorithms Laboratory (HEAL).

Acknowledgements. The authors thank Franz Winkler at the Research Institute for Symbolic Computation, Johannes Kepler University Linz, for his advice. R. Sendra is partially supported by the Spanish Ministerio de Economía y Competitividad under the project MTM2011-25816-C02-01 and is a member of the Research Group ASYNACS (Ref. CCEE2011/R34). The authors also thanks members of the Heuristic and Evolutionary Algorithms Laboratory as well as of the Bioinformatics Research Group, University of Applied Sciences Upper Austria, for their comments.

References

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

[2] J. H. Holland. *Adaption in Natural and Artifical Systems.* University of Michigan Press, 1975.

[3] E. Pitzer. *Applied Fitness Landscape Analysis.* PhD thesis, Institute for Formal Models and Verification, Johannes Kepler University Linz, 2013.

[4] I. Rechenberg. *Evolutionsstrategie.* Friedrich Frommann Verlag, 1973.

[5] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhäuser Verlag, Basel, Switzerland, 1994.

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

[7] S. Wagner, G. Kronberger, A. Beham, M. Kommenda, A. Scheibenpflug, E. Pitzer, S. Vonolfen, M. Kofler, S. Winkler, V. Dorfer, and M. Affenzeller. *Advanced Methods and Applications in Computational Intelligence*, volume 6 of *Topics in Intelligent Engineering and Informatics*, chapter Architecture and Design of the HeuristicLab Optimization Environment, pages 197–261. Springer, 2014.