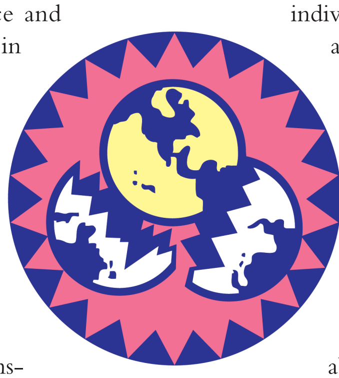
Thomas Bäck, M. Emmerich,
and O.M. Shir, *Leiden University, THE NETHERLANDS*

Evolutionary Algorithms for Real World Applications

A Personal Introduction

How can a set of three authors, two and three generations after evolutionary computation pioneers, seriously attempt to write an article in memoriam of one of these pioneers? That is the question to start with, and it is not an easy one. Fortunately, the first author had the pleasure to be in the middle of the evolutionary computation storm in the early '90s when it became very active and established itself as its own segment within computational intelligence and natural computing. Later in the mid '90s, M. Emmerich joined the field, and O. Shir is now a young Ph.D. student at our group. Moreover, the first two authors had the pleasure to work with one of the pioneers in evolutionary computation, namely Hans-Paul Schwefel, thus experiencing the spirit, creativity, motivation and passion of one of these innovators through an almost daily interaction over a number of years. This may qualify us to at least try writing such an article.



© IMAGE CLUB

To be very open, the evolutionary strategy researchers in Germany had not been aware of Larry Fogel's work until, through intensive literature research in the early '90s, we came across David Fogel's Ph.D. thesis [Fog92]. It was a huge surprise to see an algorithm outlined there—evolutionary programming for continuous search spaces—which was essentially working with continuous representations, a mutation operator, no recombination operator, and a stochastic variant of tournament selection. Even more interesting, however, was the fact that this version of evolutionary programming used mutations sampled from a normal distribution and incorporated the variance of the normal distribution into the representation of individuals to allow for self-adaptation of those parameters. Independent of evolution strategies, evolutionary programming researchers had obviously invented a very similar idea, namely evolving object variables (the solution vector itself) and strategy parameters (i.e., variances of normal distributions) simultaneously. In other words, this looked like a variant of the well known and well investigated evolution strategy when we discovered it in the early '90s. The pioneers of evolution strategies, I. Rechenberg and H.-P. Schwefel, had developed those algorithms also in the early '60s for tackling hard engineering optimization problems; and in the '70s, Hans-Paul had added the concept of self-adaptation to overcome the step size control problem [Schw77].

This was the first author's starting point in learning about evolutionary programming and its origins, and to discover, surprisingly, that evolutionary programming was clearly the third original branch of evolutionary computation, dating back to the early '60s like the work on genetic algorithms and evolution strategies. To the first author of this paper, this insight meant a dramatic change of his understanding of the field as a whole, with an idea that independently evolved at different geographic locations on earth, and turned into similar algorithms—evolutionary programming, genetic algorithms, and evolution strategies. And it also had a dramatic impact on the first author's work on his Ph.D. thesis, because a lot of additional material had to be reviewed, additional experiments made, and some of the existing material rewritten. In retrospect, the contemporary evolutionary programming work was the start of strong and fruitful interactions of the different groups in the field and the initial phase of a worldwide emerging trend to view the field as a whole rather than a group of independent, different algorithms. Finally, evolutionary computation was born. Now, still in the early '90s, the European "Parallel Problem Solving from Nature" Workshop explicitly opened the scope for all branches of natural computing approaches, in particular including genetic algorithms, evolution strategies, and evolutionary programming.

Digital Object Identifier 10.1109/MCI.2007.913378

1556-603X/08/$25.00©2008IEEE

It took a while for me, however, to meet Larry Fogel for the first time. It is always a different thing whether you read the articles by a researcher, or whether you meet that researcher personally. In this case, however, the personality of this man was even more remarkable than I had imagined. The presentation he gave at the meeting where I first saw him was the most entertaining, exciting, artistic and touching way to give a scientific talk I had ever experienced. Larry told a wonderful story about his research, combined it with artistic components such as singing (which he did very well), was very humorous, and had my full attention during the whole presentation. He was truly performing, rather than just presenting. Talking to him personally later on, for me as a young Ph.D. student, was just an amazing experience showing how creative and inspiring he was as he quickly related the evolution strategy concept to evolutionary programming and gave ideas about combining concepts and expanding them into new dimensions. Clearly, Larry had an inspiring, creative mind that was not only able to invent a new algorithm in the '60s, but also to apply those concepts very early on to industrial tasks. This other ability led to the foundation of Natural Selection, his company where evolutionary programming is always in the center of real world problem solving applications.

Unfortunately, as always with great, famous, busy people, I did not get enough time to spend with him. It would have been wonderful to learn more from an inventor like him, and I am truly happy that I got to know him and his family personally, and still have the pleasure to be able to interact with his sons David and Gary.

With this paper, we want to give an overview of some of the applications evolutionary computation is good for—in this case evolutionary strategies—which have been extended in our group to cover high dimensional continuous and mixed-integer optimization tasks. These algorithms have so much in common with evolutionary programming, such that we can nicely demonstrate through those applications how powerful evolutionary computation is today. In the following section, we give an overview of three different applications, namely, molecular alignment in quantum control, feature detector optimization, and chemical plant optimization.

Evolutionary Algorithms in Practice

In this section, we will report on three applications of evolutionary algorithms (EA) in real world parameter optimization. These applications show how the principles, such as robustness and representation independence, make it possible to extend the scope of optimization algorithms available so far and find answers to problems where classical solution methods fail.

Quantum Control: Molecular Alignment

As a first example of a challenging task where evolutionary algorithms are used as solution methods, we will introduce a specific application from the quantum control field, the so-called dynamic molecular alignment problem. This research is part of collaboration between LIACS and the XUV group of Marc Vrakking at the AMOLF-FOM institute in Amsterdam [SEB+07a, SEB+07b].

In quantum control, the goal is to control the motion or position of molecules in space. This is typically done by means of shaped laser pulses in the regime of femtosecond lasers. Though the pulse of the laser can be controlled and the resulting *yield* can be measured—ranging from energy excitation level to orientation or in our case, molecular alignment with respect to a certain axis—it is a very difficult task to find a waveform for the laser that results in a desired yield. This is a task where evolutionary algorithms can be very useful problem solvers. The problem of finding an optimal pulse shape can be recasted as a real-valued optimization problem. The shape of the waveform is governed by a number of control points, the coordinates of which are then the input variables of a continuous optimization problem solved by means of the EA. The high number of control points needed to control the shape of a waveform with the needed accuracy and the nonlinear underlying process (the computation of the alignment yield requires solving the Schrödinger's equation, which is expensive in computational terms) makes this problem a very difficult—high dimensional and nonlinear—optimization task [SEB+07].

In the joint research project, the group achieved groundbreaking new insights into the shape of lasers that lead to the desired alignment. Not only single optimized results have been found, but also a method was developed to obtain different promising results. Again, this method was inspired by nature. Looking at natural environments, it is quite obvious that solutions to one problem do not always look the same, and different animals and plants can coexist with each other. A fundamental principle found by biologists is that of niching, i.e., that coexisting animals and plants are assigned to different loci in an ecosystem. The difference can be spatial as well as in behavioral patterns. Abstracting from this principle of niching in nature, niching in evolutionary algorithms allows populations to coexist in the attractors of different local optimal (peaks) of the yield function—that in the case of two input variables

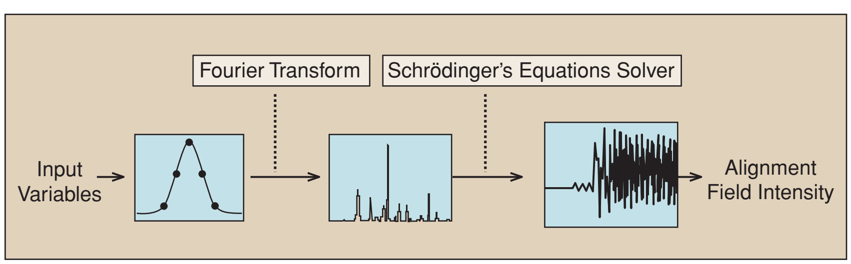


FIGURE 1 Schematic sketch of the computations in the molecular alignment problem.

can be intuitively visualized as a landscape—a so-called fitness landscape.

Using niching evolutionary algorithms [SEB07], it was possible to find conceptually different solutions to the same laser pulse shaping problem [SEB+07a]. The value of the gained insight is great for the physical point of view. In addition, the niching methods derived seem to be very beneficial for other problem domains, such as the optimal design of buildings. Future research has just started to explore these new possibilities.

Optimization of Feature Detectors in Medical Images

Heart diseases are among the most frequent causes of death in the western world, and they often strike their victims unforeseen and in early ages. The Leiden University Medical Center of the University of Leiden is one of the leading researchers in the field of using image analysis techniques for detecting heart diseases. One of the advanced methods used in this field is ultrasound scanning of coronary heart arteries. The result of such a scanning is a high resolution movie displaying the inner part of the heart artery. However, these movies appear to be very noisy to laymen and only medical experts are capable of determining calcifications and their precise thickness. This information is important for treating calcified blood vessels by means of a stent, which is used to eliminate the plaque and the size of which is to be determined precisely in order to avoid serious complications of the operations.

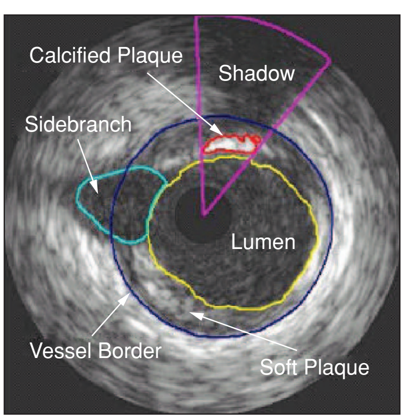


FIGURE 2 IVUS image of a heart artery with annotated regions.

Despite a small number of experts, treatment based on IVUS movies is done quite frequently in practice and technical support in the form of image analysis tools. The LKEB group of the LUMC, led by Dr. Hans Reiber, has developed image-analysis software that can detect structures with high accuracy. However, due to the high importance of quality in this sensitive domain, it is their desire to further improve their software, and in a collaboration with researchers from LIACS, it was found that evolutionary algorithms can help to achieve this goal [LEE+06].

The procedure of measuring the quality of the image analysis software is to compare classifications of image regions (e.g. lumen, calcified plaques) predicted with the software analysis software with annotations of medical experts—so called training data. The better the results of the expert comply with the results of the software, the better the software has been calibrated. It turned out that once the basic image processing system has been set up, the tuning of its many (up to 20) parameters of different types turns out to be a tedious and difficult task, if carried out by hand. Here evolutionary algorithms can help the program designer by automatically exploring the huge space of parameter combinations using all computers available. This can result in a large number of several thousand scheduled experiments, in each of which the quality of the software measured on the basis of the training data. In the end of the automatic optimization, it is almost certain that the parameter settings improve the manual settings, as has been shown in various studies carried out by the researchers. One could be satisfied with this improvement compared to the manual settings, but the hope for perfection led to improving evolutionary algorithms by incorporating specialized search operators for the different parameter types, which led to even better results than the standard methods. The idea is to use problem specific data-structures that are then modified by means of data-structure specific mutation operators, a principle that dates back to evolutionary programming, emphasizing the importance of mutation in evolutionary algorithms.

Future work concentrates on the integration of parameter dependencies, and it is hoped that by integrating knowledge on the structure of the image processing pipeline, not only will a further improvement be achieved, but also an increased efficiency of the parameter tuning process that allows the expert to experiment more easily with different structures of the processing pipeline and different sets of training data with more difficult to detect structures.

Chemical Engineering Plant Optimization

A large application domain of computer aided optimization is chemical engineering. It is thus no surprise that many leading researchers in classical optimization theory are and have been chemical engineers. In the field of chemical engineering, equation based modeling of processes has achieved a very high standard, and alongside this, nonlinear optimization methods were developed that provide solutions with a proof of optimality for optimization problems, such as energy minimization of systems or profit maximization. Due to the availability of well-elaborated classical optimization methods, it took a relatively long period of time for heuristic methods, such as genetic and evolutionary algorithms, to be accepted by the community. Today, it seems clear that the complexity of many relevant large-scale optimization problems forbid the precise solution, and evolutionary algorithms have been proven to find good approximations to rigorous lower bounds in such cases.

One example of an optimization task where evolutionary algorithms are used in combination with a rigorous solution method is the scheduling of chemical batch processes. This problem has been investigated by the process control group of the chemical engineering department of the University of Dortmund in collaboration with the LIACS [UET+07]. The problem is to find the optimal task for the production of polystyrene scheduled to different chemical reactors. This problem is a problem with uncertainty, as the demand of products by the marked cannot be foreseen. There are decisions (coded as variables) that must be dealt with here-and-now, and other decisions can be postponed to a later time when there is more information available. The expected profit for a here-and-now decision can be estimated by solving a mathematical program that assumes a best possible decision politic in the later decisions. Because the solution of this problem is already structurally complex and time demanding, evolutionary algorithms are used as the search method for optimal here-and-now decisions. The number of possibilities for setting the vector of here and now decisions is huge, and it exceeds the number of 10^{32}. A simple integer EA has been used to search this space and it found satisfactory solutions. However, the time needed was too long, as the software has to be used in an online scheduling system, and only about four hours of time are available to tune the decision variables.

A problem for the search was that many of the solutions evaluated during the search turned out to be infeasible, due to the many constraints on the integer parameter space. A careful analysis of the structure of constraints and encoding of the search space as a decision tree made it possible to reduce the number of alternative solutions significantly to a number of 10^{18}. Now, the size of the search space contains only the feasible solutions—and exactly these solutions. The price that was paid for this enormous reduction in search space size was the increased complexity of the search space representation. Again, the powerful concept of problem specific mutation operators, as introduced in evolutionary programming, was helpful to deal with this complexity. Following the guidelines of metric-based evolutionary programming, scalable mutation operators were designed leading to an engineered evolutionary algorithm that solved the problem in much less time and ended with significantly better approximations of the lower bound for the objective function value, that was computed by means of a mixed-integer programming method using relaxation of variables.

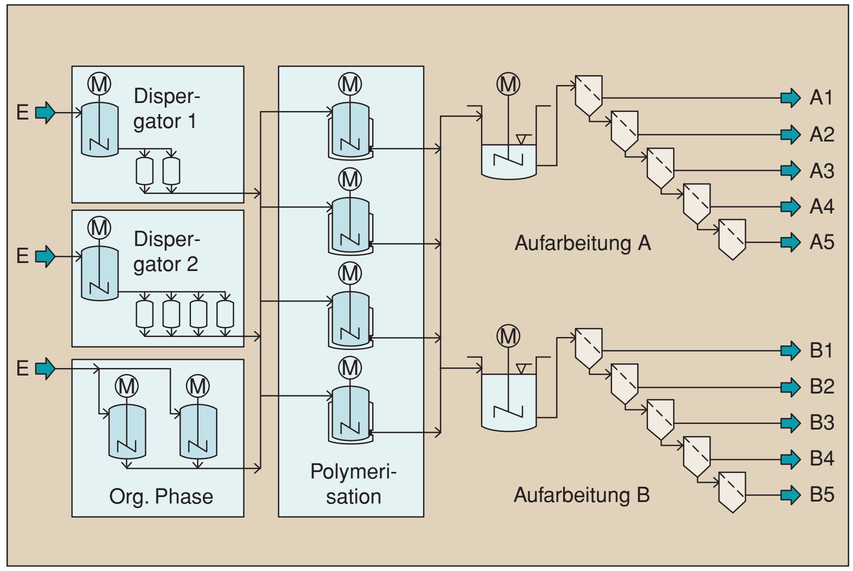


FIGURE 3 Chemical semi-continuous batch-process for the production of polystyrene. A1–A5 and B1–B5 represent products with different grain size. The number of active process units at a given time is limited.

Conclusions

Evolutionary algorithms are today a state-of-the-art methodology in solving hard optimization problems, and are regularly being used in industries such as automotive and aerospace. In fact, these algorithms have revolutionized the way hard problems are being solved today. The pioneers of evolutionary computation, including Larry Fogel as one the key founders of the field, have been laying the foundations with their creativity and visionary approach toward science. The few examples given in this paper are just giving an idea of what is possible with evolutionary computation today, and we are certainly going to see a lot more exciting work in the field in the future.

We will always remember Larry Fogel as a wonderful scientist and human character.

References

[1] D.B. Fogel, *Evolving Artificial Intelligence*. PhD Thesis, University of California, San Diego, 1992.

[2] H.-P. Schwefel, "Numerische optimierung von computer-modellen mittels der evolutionsstrategie," *Interdisciplinary Systems Research*, vol. 26, Birkhäuser, Basel, 1977.

[3] O.M. Shir, M. Emmerich, and T. Bäck, "Self-adaptive niching CMA-ES with mahalanobis metric," *Proceedings of IEEE-CEC 2007*, Singapore, IEEE Press (CDROM).

[4] O.M. Shir, M. Emmerich, T. Bäck, and M.J.J. Vrakking, "Conceptual Designs in Laser Pulse Shaping Obtained by Niching in Evolution Strategies," *EUROGEN 2007*, Finland (CD-ROM).

[5] O.M. Shir, M. Emmerich, T. Bäck and M.J.J. Vrakking, "The application of evolutionary multi-criteria optimization to dynamic molecular alignment," *Proceedings of IEEE-CEC 2007*, Singapore, IEEE Press (CD-ROM).

[6] R. Li, M.T.M. Emmerich, J. Eggermont, E.G.P. Bovenkamp, T. Bäck, J. Dijkstra, and J.H.C. Reiber, "Mixed-integer optimization of coronary vessel image analysis using evolution strategies," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2006*.

[7] M. Urselmann, M.T.M. Emmerich, J. Till, G. Sand, S. Engell, "Design of problem-specific EA/MIP hybrids: Two-stage stochastic integer programming applied to chemical batch scheduling," *Engineering Optimization*, vol. 39, no. 5, pp. 529–549, 2007.