

## Introduction

The idea of this Dagstuhl-Seminar was born during the Third International Conference on Parallel Problem Solving from Nature (PPSN III) at Jerusalem in 1994. Among those methods to solve problems by means of computers using natural metaphors, Evolutionary Algorithms (EAs) are dominant, nowadays. They play an important role in two new fields of computer science called Computational Intelligence (CI) and Artificial Life (AL), which emerged aside of the now classical realm of Artificial Intelligence (AI). Whereas AI uses symbolic knowledge processing techniques, CI and AL emphasize the use of subsymbolic computing, sometimes also called Soft Computing (SOCO).

The terms Evolutionary Algorithms (EAs) and Evolutionary Computing (EC) were coined at the beginning of the 1990ies, i.e., during the first PPSN (Parallel Problem Solving from Nature) conference at Dortmund and the fourth International Conference on Genetic Algorithms (ICGA) at San Diego, in order to have a common denominator for at least three different, independently of each other and at different places but nearly at the same time, the 1960ies, created methods to solve problems, mostly optimization problems, by means of mimicking mechanisms gleaned from organic evolution:

- Evolutionary Programming (EP), first introduced by Lawrence Fogel at San Diego and recently revived by David Fogel;
- Evolution Strategies (ESs), developed by Ingo Rechenberg and Hans-Paul Schwefel at Berlin;
- Genetic Algorithms (GAs), proposed by John Holland at Ann Arbor and first used as an optimization algorithm by Kenneth De Jong.

It took nearly a human generation until these methods became accepted and used for a variety of tasks for which classical methods are insufficient and exact (enumerative) methods too time-consuming. So far, applications of EAs have been more convincing than their theory, which is still weak, though under progress.

There were about 40 presentations which were subsumed under the following topics: theory of EAs, self adaptation, parallelity, application (industry, economics, and others), computational intelligence, genetic programming, and natural and artificial evolution. The participants got deeper insights into methods, mechanisms, foundations, and application areas, but there remained a lot of new questions which we could not answer in this week. Evolutionary Algorithms are forming still a young and rich field of scientific

research and application oriented experiments with surprises and disadvantages, where foundations, useful methods, codings, fitness and their landscapes, and experiences by applications and experiments must and will be developed intensively in the next decade.

These considerations led to the idea of emphasizing EA applications during this seminar. Nevertheless, we started with a survey of the state of art concerning theoretical results about the convergence and convergence rates as far as available for EP, ESs, and GAs.

It was a pity that those invited to come here from the U.S.A. finally could not attend due to the budget struggles over there. However, some leading scientists from other European countries came here and provided their recent results and experiences. Though the German scene was not completely present, as well, many groups from academia as well as industry sent delegates and thus contributed to the exchange of knowledge and ideas as well as many lively discussions about open questions.

The ambients of this unique place, i.e. Schloß Dagstuhl, was optimal, indeed, to take a step forward into the exciting field of Evolutionary Algorithms and — may be even more important — towards closer personal connections between those interested in it.

# Program

## Monday morning

Chair: H.-P. Schwefel

Welcome *H.-P. Schwefel*  
Artificial and Natural Evolution - Some Remarks *J. Hopf*

## Monday afternoon “Theory”

Chair: W. Banzhaf

Theory of EA: State of the Art *G. Rudolph*  
Basic Principles for a Unified EA-Theory *H.-G. Beyer*

Chair: G. Rudolph

Evolution von Netztopologien mit PBIL *M. Höhfeld*  
Critical Remarks on Performance Issues of Genetic Algorithms *R. Salomon*

## Tuesday morning (part 1) “Self adaptation”

Chair: T. Bäck

Breeding Good ES - First Results *F. Kursawe*  
One Result from the Analysis of  $(1, \lambda) - \sigma$  -Self-Adaption *H.-G. Beyer*  
The Role of Sex in EAs *A.E. Eiben*

## Tuesday morning (part 2) “Parallelism”

Chair: A. Zell

The EVA Project: Parallel GAs and Parallel ESs on SIMD and  
MIMD Parallel Computers - A Comparison *A. Zell*  
Parallel Models In Evolutionary Computing *M. Tomassini*

## Tuesday afternoon “Applications I”

Chair: J. Kok

Experiences with Applications of GAs to NP-Hard Problems *T. Bäck*  
Evolutionary Planning for the Exchange of Elements in  
Nuclear Power Plants *J. Heistermann*  
Scheduling of Power Plant Units with Coupled Heat &  
Power Production *M. Höhfeld*  
Evolving and Optimizing Braitenberg Controllers for  
Autonomous Agents *R. Salomon*  
Application of EAs to the Free Flight Concept for Aircraft *I. Gerdes*

Evolutionary Air Traffic Flow Management in the Free Routing Concept	<i>J. Kok</i>
Demo: Free Flight Tool	<i>I. Gerdes</i>
Video: Prediction Of Pilot Bid Behaviour; Free Routing Tool	<i>J. Kok</i>

**Wednesday morning** “Applications II”

Chair: A. E. Eiben

An Overview of EAs in Management Applications	<i>V. Nissen</i>
Evolving Trading Models with Parallel EAs	<i>M. Tomassini</i>
Comparing EC, NN, Logistic Regression and CHAID on a Direct Marketing Problem	<i>A. E. Eiben</i>
Data Mining with EC	<i>J. Kok</i>
An ES-Approach for Facility Layout with Unequal Areas and Undetermined Locations for Single- and Multi-Floor Problems	<i>V. Nissen</i>
Optimizing Photomask-Layout for Grey-Tone Lithography	<i>J. Hopf</i>
System-Level Synthesis using EAs	<i>T. Blickle</i>
Applications of EAs at Daimler Benz	<i>U. Kreßel</i>

**Thursday morning** “Computational Intelligence”

Chair: M. Höhfeld

An Evolution Model for Integration Problems and its Application to Lighting Simulation	<i>B. Lange</i>
Fuzzy Controllers & EA	<i>F. Klawonn</i>
Evolutionary Inverse Problem Solving: Identification of the Isotherm Function by Recurrent Neural Networks	<i>M. Schoenauer</i>
EAs and Neural Networks	<i>M. Mandischer</i>
On a GA for the Selection of Optimally Generalizing Neural Network Topologies	<i>S. Rudolph</i>

**Thursday afternoon** “Genetic Programming (GP)”

Chair: J. Heistermann

GP Controlling - a Miniature Robot	<i>W. Banzhaf</i>
Programmic Compression	<i>P. Nordin</i>
Finding a Formula using GP	<i>J. Heistermann</i>
Evolutionary Inverse Problem Solving: GP in Structural Mechanics	<i>M. Schoenauer</i>

**Friday morning** “Natural and Artificial Evolution”

Chair: H.-P. Schwefel

Two Remarks on Theory	<i>V. Claus</i>
Evolutionary Algorithms and Artificial Life	<i>W. Banzhaf</i>
Missing Features in Current EAs	<i>H.-P. Schwefel</i>
General Discussion	
Manoeuvre Critique and	
Farewell	

# Artificial and Natural Evolution - Some Remarks

Jörn Hopf, Max-Planck-Institute for Computer Science, Germany

Both, natural and artificial evolution try to find optimum solutions in our dynamic and nonlinear world. Evolution ensures the possibility to leave a current local optimum to find a global optimum solution. Natural evolution handles its task quite successfully, so why don't we imitate nature?

Different strategies of EAs, different concepts concerning genetic operators, recombination and selection have been developed. But all of them don't bear close comparison with its natural counterparts. Even simplified concepts like typographical genetic with its main dogma of information processing have not been taken into consideration so far.

Evolutionary biologists use self-replicating RNA-species for their experiments. For the interpretation of a spectrum of mutants, biologists consider them quasi-species. A small part of the quasi-species contains the master sequence, surrounded by many mutants which are represented only a few times. In this case selection prevents the degeneration of the master species. Does - as a consequence - "Selection of the Fittest" have to give way to "Selection of the Master Sequence"?

Simulating evolutionary models on computers we use generated random numbers to implement selection and recombination. Complexity theory helps us to define non-random numbers, it doesn't tell us that a sequence is random. Thus, random numbers have to satisfy several tests. The more tests we apply, our sequence behave accurately due to these tests and the generated numbers become predictable. For real coincidence this prediction is not possible.

In addition, more general questions are to be answered: Is natural evolution capable of finding global optima at all and how many of them that could have been found were missed already? Are there optimum solutions which are doomed to failure because there is no possibility to encode them on the nucleotide strings?

The more research is done in this new field, the more questions arise. With our work, we just make the first steps imitating natural evolution.

# Theory of Evolutionary Algorithms — State of the Art

Günter Rudolph, Informatik Centrum Dortmund, Germany

The theoretical properties of evolutionary algorithms are presented in terms of Markov and Martingale theory. It is shown that the conditions for stochastic convergence to the global optimum are satisfied by almost all evolutionary algorithms. The actual expected time to reach the minimum can be determined only for selected problem classes: Linear, unimodal and submodular 0/1-problems; and strongly convex problems with real variables. A negative result concerns submodular functions: Although the minimization of submodular functions is in P, there exist problem instances for which an evolutionary algorithm requires at least exponentially many function evaluations.

Finally, some recent criticisms about the soundness and utility of Schema and Breeding theory applied to evolutionary algorithms are discussed.

## Basic Principles for a Unified EA-Theory

Hans-Georg Beyer, Universität Dortmund, Informatik LS11, Germany

An alternative explanation for the working of Evolutionary Algorithms (EAs) is presented, focusing on the collective phenomena taking place in populations performing recombination. The new approach is mainly based upon three ‘basic principles’: the *evolutionary progress principle (EPP)*, the *genetic repair (GR)* hypothesis, and the *mutation induced species by recombination (MISR)* principle. GR and MISR are especially intended to give insight into the mechanism of recombination.

Within the GA-community it seems to be common believe that the building block hypothesis (BBH) accounts for the exceptional role of recombination. It will be shown that a *totally* different explanation for the benefit of recombination can be possible - the GR-hypothesis.

The new principles are derived from the investigation of different variants of the Evolution Strategy (ES). There are certain evidences that the principles found are very general, i. e., they should hold for all kinds of (optimizing)

Evolutionary Algorithms (EAs) like Genetic Algorithms (GAs) and Evolutionary Programming (EP). Furthermore, they seem to be in agreement (in the sense of an analogy) with recent theories on the benefits of sex in biology.

## **Evolving Neural Network Topologies with Population–Based Incremental Learning**

**Elvis Galić and Markus Höhfeld, Department of Theoretical Physics, Würzburg, Germany and Corporate Research Siemens AG, München, Germany**

Based on Population–Based Incremental Learning (PBIL) we present a new approach for the evolution of neural network architectures and their corresponding weights. The main idea is to use a probability vector rather than bit strings to represent a population of networks in each generation. We show that crucial issues of neural network training can effectively be integrated into the PBIL framework. First, the Quasi Newton method for local weight optimization is integrated and the moving average update rule of the PBIL is extended to continuous parameters in order to transmit the best network to the next generation. Second, and more important, we incorporate cross–validation to focus the evolution towards networks with optimal generalization performance. A comparison with standard pruning and stopped–training algorithms shows that our approach effectively finds small networks with increased generalization ability.

## **Critical Remarks on Performance Issues of Genetic Algorithms**

**Ralf Salomon, University of Zurich, Department of Computer Science, Switzerland**

In recent years, genetic algorithms (GAs) have become more and more robust and easy to use. Current knowledge and many successful experiments suggest that the application of GAs is not limited to easy-to-optimize unimodal functions that contain only one optimum. Several results and GA theory



give the impression that GAs easily escape from millions of local optima and reliably converge to a single global optimum. However, recent results raise serious doubts that this explanation is correct. The performance of a prototypical GA when applied to several of the widely-used test functions is presented. It is shown that a rotation of the coordinate system leads to a catastrophic performance loss; the GA under investigation simply fails. This performance loss raises several new, yet unanswered questions. Future work should find a detailed clarification of the underlying reasons for this kind of drastic performance loss. As long as these problems have not been solved, the application of GAs will be limited to the optimization of easy-to-optimize (decomposable) functions.

## Breeding ES – First Results

**Frank Kursawe, University of Dortmund, Dept. of Computer Science, Systems Analysis Group, Germany**

When observing the convergence speed of Evolution Strategies on very easy test functions one can observe all sorts of behaviour ranging from very fast convergence even to divergence in the case of the sphere model. This behaviour mainly depends on the type of recombination used. Furthermore, the parameters controlling the adaptation speed of the stepsizes ( $\tau_0, \tau_i$ ) influence the velocity and robustness of convergence. Depending on the problem the topology of the  $\tau_0 - \tau_i$ -‘mountain’ changes. And even for one problem this topology may change with increasing problem size.

But using a meta–approach tuning 8 parameters of a multi–membered ES one can breed good strategies for a given problem. Using this approach the meta–ES comes up with surprising results, e.g. when looking for the global optimum in a multimodal environment one should use a very weak selection and a setting of  $\tau_0$  and  $\tau_i$  with  $\tau_0 > \tau_i$ . For other functions the meta–ES leads to strategies with a convergence speed being 2 to 5 times faster than a standard ES on the same problem.

# One Result from the Analysis of the $(1, \lambda)$ - $\sigma$ -Self-Adaptation

Hans-Georg Beyer, Universität Dortmund, Informatik LS11, Germany

The power of an Evolution Strategy (ES) is mainly based upon its ability to perform a ‘second order’ evolutionary process. This process adapts internal strategy parameters, especially the mutation strength, in such a way that the whole algorithm exhibits near optimal performance. That is, an ES *drives itself* into its optimal working regime. One way of obtaining nearly optimal ES-performance is the *self-adaptation* (SA) technique developed by SCHWEFEL in the late sixties.

Starting from an introduction into the basic concept and the algorithm of self-adaptation-ESs, a framework for the analysis of the  $\sigma$ SA is developed which has two stages: a microscopic level concerning the description of the stochastic changes from one generation to the next, and a macroscopic level describing the evolutionary dynamics of the  $\sigma$ SA over the time (generations). The behavior of the  $\sigma$ SA depends on the  $\sigma$ -learning rules used and the *learning parameters* to be chosen appropriately.

This talk concentrates on one result obtained from the analysis of the sphere model concerning the different learning rules proposed. At the microscopic level (i. e., for the driving forces) it can be shown that for small learning parameters it does not make any difference which learning rule is actually used. This gives rise to the astonishing conclusion that, e. g., Schwefel’s ‘Log-Normal’-rule and Rechenberg’s ‘Two-Point’-rule as well as Fogel’s meta-EP-rule should exhibit the same evolutionary dynamics. It is an open question whether this theoretical prediction does hold for non-spherical success domains too.

## The Role of Sex in EAs

A.E. Eiben, Leiden University, The Netherlands

Every Evolutionary Algorithm applies reproduction operators to generate new candidate solutions. These operators are either asexual, i.e. unary, or sexual, traditionally binary. The viability of binary reproduction operators, and thereby the usefulness of sex in EAs, is seriously being questioned

lately. For instance, Evolutionary Programming rejects sexual reproduction and claims that sex (binary operators) is inferior to no sex (unary mutation only). In the meanwhile, recent research showed that GAs with n-ary (orgy) operators consistently perform better for higher n's than for n=2, suggesting that more sex is better. This lecture gives an overview of n-ary reproduction operators and discusses recent research on 'orgies'. In particular, the performance of uniform scanning crossover and diagonal crossover is shown on 3 types of problems, numerical optimization, NK-landscapes and a constraint satisfaction problem (graph coloring). The results demonstrate the advantages of operators with high arity.

## **The EvA Project: Parallel GAs and Parallel ESs on SIMD and MIMD Parallel Computers - A Comparison**

**Andreas Zell, Univ. Tübingen, WSI, Germany**

We give an overview of the EvA (Evolution Algorithms) project, a project to compare parallel implementations of Genetic Algorithms (GAs) and Evolutions Strategies (ESs) on massively parallel SIMD computers and on large MIMD computers. EvA currently consists of the following parts: VEGA: distributed GAs on the MIMD computer Intel Paragon, MPGA: massively parallel GAs on the SIMD computer MasPar MP-1, VEES: distributed ESs on the Intel Paragon, MPES: massively parallel ESs on the MasPar MP-1. All modules are callable from a common X11/Motif based user interface and use standardized I/O formats.

We describe the different methods to map a population of GA individuals to processors and the means taken to limit or regularize communication. We report performance figures of the different implementations on standard benchmark problems and compare them with other parallel and sequential GA implementations.

Our results indicate that all four massively parallel and distributed GAs and ESs have very good scaling behaviour and are significantly faster than sequential GAs. The large population sizes possible on the parallel computers are especially useful for large multi-modal optimization problems. The parallel GAs have a higher convergence rate, due to the prevention of premature convergence in multiple subpopulations or due to the spatial separation of individuals. The parallel versions also frequently find better optima than the

sequential algorithms. While the SIMD implementations have a slight performance advantage for large population sizes, the distributed MIMD algorithms have the benefit that one can obtain a sequential GA or ES implementation as a by-product. Also processor memory size is not as critical as in massively parallel SIMD implementations.

In summary, the results show that the SIMD and MIMD implementations of GAs and of ESs have complementary strengths and weaknesses. All systems described are among the fastest general purpose implementations of evolution algorithms for large population sizes currently available.

## Parallel Models in Evolutionary Computing

**Marco Tomassini, Swiss Federal Institute of Technology, Switzerland.**

Evolutionary algorithms (EAs) lend themselves quite naturally to parallel implementations. There are two main reasons for parallelizing an evolutionary algorithm: the first is to achieve time savings by distributing the computational effort and the second is to benefit from a parallel setting from the algorithmic point of view. Indeed, with the exception of the trivial parallelization of the fitness calculation, all the known parallel algorithm models are in fact novel evolutionary algorithms, whose properties are not very well known. However, from an experimental point of view, the advantage of using parallel EAs has been sufficiently demonstrated. There is a large number of possible parallel topologies and parameters that can be chosen. A first classification distinguishes between coarse-grained, subpopulation based EAs (also called *island* models) and fine-grained, massively parallel or *cellular*.

The *island* model features geographically separated subpopulations of relatively large size. Subpopulations may exchange information from time to time by allowing some individuals to migrate from one subpopulation to another according to various patterns. The main reason for this approach is to periodically reinject diversity into otherwise converging subpopulations. When the migration takes place between nearest neighbor subpopulations the model is called *stepping stone*. Within each subpopulation a standard sequential evolutionary algorithm is executed between migration phases.

In the *cellular* model individuals are placed on a large one or two-dimensional toroidal grid, one individual per grid location. Fitness evaluation is done simultaneously for all individuals and selection, reproduction and mating take place locally within a small neighborhood. In time, semi-isolated niches

of genetically homogeneous individuals emerge across the grid as a result of slow individual diffusion. This phenomenon is called *isolation by distance* and is due to the fact that the probability of interaction of two individuals is a fast-decaying function of their distance. The model can be made more dynamical by adding provisions for longer range individual movement through random walks. Hybrid models are also possible; for example, one might consider an island model in which each island is structured as a grid of individuals interacting locally.

From the point of view of time, current parallel EAs are synchronous or quasi-synchronous, with algorithmic phases taking place at the same time everywhere in the system. This does not need to be so. Asynchronous EAs are possible and even more natural. They should be particularly advantageous in the case of genetic programming where individuals might be of widely different complexity. A kind of parallel steady-state algorithm would offer advantages since new individuals can enter the population at arbitrary times, thus eliminating the need for load-balancing the system.

## Experiences with Applications of Genetic Algorithms to NP-Hard Problems

**Thomas Bäck, Informatik Centrum Dortmund, Centrum für angewandte Systemanalyse (CASA), Germany**

Using a binary representation  $\vec{x} = \{0, 1\}^\ell$  of search points, a mutation operator that inverts bits with a certain mutation probability  $p_m = 1/\ell$ , a two-point crossover operator and proportional selection, *canonical genetic algorithms* are applied to NP-hard optimization problems such as subset sum [5], 0/1 multiple knapsack [6], set covering [2], minimum tardy task [5], minimum vertex cover [4], maximum independent set [1], and maximum cut [5]. Techniques such as the problem representation by binary strings and the development of appropriate penalty terms to handle feasibility constraints are illustrated by these applications.

A comparison of the results achieved by the canonical genetic algorithm and results obtained by greedy heuristics clarifies that, at the cost of much larger running times of the algorithm, a canonical genetic algorithm often yields better solutions than a greedy heuristic.

An extension of the canonical genetic algorithm is presented, which incorporates a deterministic, time-decreasing mutation rate schedule of the form

$p(t) = (2 + \frac{\ell-2}{T-1} \cdot t)^{-1}$  (where  $T$  denotes the predefined maximum number of generations) as well as a *self-adaptation* mechanism of individual mutation rates according to the logistic transformation  $p' = (1 + \frac{1-p}{p} \cdot \exp(-\gamma \cdot N(0,1)))^{-1}$  for the mutation of mutation rates [3]. Comparing these two mechanisms to the mutation rate  $1/\ell$ , the deterministic time-decreasing schedule clearly performs best. Self-adaptation also outperforms the setting  $p_m = 1/\ell$ , but it does not find solutions of similar quality as the time-decreasing schedule does. Preliminary investigations indicate that this is caused by the adaptation speed (controlled by the learning rate  $\gamma$ ), which needs to be tuned towards slower self-adaptation to achieve better results with this mechanism.

## References

- [1] Th. Bäck and S. Khuri. An evolutionary heuristic for the maximum independent set problem. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 531–535. IEEE Press, 1994.
- [2] Th. Bäck, M. Schütz, and S. Khuri. A comparative study of a penalty function, a repair heuristic, and stochastic operators with the set-covering problem. In E. Ronald, editor, *Evolution Artificielle*, 1996. (In Print).
- [3] Th. Bäck and Martin Schütz. Intelligent mutation rate control in canonical genetic algorithms. In Z. Ras, editor, *Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems*, Lecture Notes in Computer Science. Springer, Berlin, 1996.
- [4] S. Khuri and Th. Bäck. An evolutionary heuristic for the minimum vertex cover problem. In J. Kunze and H. Stoyan, editors, *KI-94 Workshops (Extended Abstracts)*, pages 83–84. Gesellschaft für Informatik e. V., Bonn, 1994.
- [5] S. Khuri, Th. Bäck, and J. Heitkötter. An evolutionary approach to combinatorial optimization problems. In D. Cizmar, editor, *Proceedings of the 22nd Annual ACM Computer Science Conference*, pages 66–73. ACM Press, New York, 1994.
- [6] S. Khuri, Th. Bäck, and J. Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In E. Deaton, D. Oppenheim, J. Urban, and H. Berghel, editors, *Proceedings of the 1994 ACM Symposium on Applied Computing*, pages 188–193. ACM Press, New York, 1994.

# **An Application of Genetic Algorithms and Neural Networks to Scheduling Power Generating Systems**

**Claus Hillermeier and Joachim Keppler (Speaker: Markus Höhfeld), Siemens AG, Corporate Research (ZFE), 81730 München, Germany and Siemens AG, Power Generation (KWU), 91050 Erlangen, Germany**

This paper presents an effective strategy to schedule fossil-fuel power plant systems with so-called power-heat coupling. Due to the simultaneous production of electricity, heat and steam such systems reach a higher efficiency than pure electric power plants. The goal is to minimize the total costs for the production of the predicted load demand of the next day. We employ a genetic algorithm to determine the unit commitment, while the economically optimal load distribution among the operating units is performed by a conventional constraint optimization method. Our approach is based on exact thermodynamic simulations of the unit efficiency, taking into account the current plant state and environmental conditions. In order to make this high modeling precision available within short computation times we employ neural networks for the storage and interpolation of simulation data.

# **Evolving and Optimizing Braitenberg Controllers for Autonomous Agents**

**Ralf Salomon, University of Zurich, Department of Computer Science, Switzerland**

The application of the evolution strategy to the evolution and optimization of Braitenberg vehicles is presented. Braitenberg vehicles are a special class of autonomous agents. Autonomous agents are embodied systems that behave in the real world without any human control. One major goal of research on autonomous agents is to study intelligence as the result of a system environment interaction, rather than understanding intelligence on a computational level. Braitenberg vehicles are controlled by a number of parameters, which are mostly determined by hand in a trial and error process. It is shown that a simple evolution strategy evolves Braitenberg vehicles very efficiently. Other

research has used genetic algorithms for very similar tasks. A comparison of both approaches shows that the evolution strategy approach is much faster; the evolution strategy approach takes about one and a half hours, whereas the genetic algorithm approach takes more than two and a half days. Since autonomous agents are very important in the field of new AI, the results presented in this paper suggest that this research field should spend more attention to evolution strategies.

## **Application of Evolutionary Algorithms to the Free Flight Concept for Aircraft**

**Ingrid Gerdes, German Aerospace Research Establishment, Institute for Flight Guidance, Traffic Systems Analysis, Braunschweig, Germany**

Because of the increasing air traffic over the last decades new concepts for the utilization of the airspace were developed in Europe and in the USA. One of these is the Free Flight Concept. In contrary to the presently applied strategy where predefined flight routes with empty airspace between them are used for all aircraft within the new concept the pilots will be able to select there own routes, flight levels and speeds.

This will not only lead to a better usage of the airspace and a self-optimization by the pilots but also to the problem of avoiding conflicts between aircraft. In case of not using predefined routes conflicts between aircraft can occur everywhere in the airspace instead only at the crossing points of predefined routes. Therefore, a possibility must be found to create conflict-free routes in an easy way, e.g. with the application of evolutionary algorithms.

For the implementation of such a tool the airspace was overlaid by a grid with numbered crossing points and routes were defined as sequences of those points as way points between starting and destination point. Additional performance information about the aircraft type is taken into account. All aircraft are assumed to try to use the direct link between starting and destination point as route. In case of conflicts between those routes the information about the later aircraft is handed over to an evolutionary algorithm where 60 routes with up to ten way points for the observed aircraft are created. Then they are evaluated in dependence of the number of conflicts with already moving aircraft, the length of the routes and the ability of the aircraft type to perform these routes. After this the selection process is carried out and



the crossover and mutation operators are applied to the new generation of routes.

Several types of these evolutionary algorithms with different evaluation functions were tested and the results presented. The algorithms with higher penalty factors for conflicts were not able to create as good routes as the algorithm with the lowest factor. This shows the importance of not removing routes with conflicts from the population because they are often very close to the optimum.

As mentioned before there are routes with different numbers of relevant nodes (way points) in the population. Therefore, it is important in which way the number of relevant points is mutated. Tests have shown that increasing the number by one for a small number of routes and inserting the new point at a randomly selected position leads to the best results.

At last the combination of hillclimbing with the evolutionary algorithm was tested for two different main cases: Before and after the EA with different numbers of runs for both algorithms. The results were not as good as expected. They were neither able to create short routes with the necessary high number of nodes at the beginning nor adjust the routes created by the EA closer to the optimal route.

## **Evolutionary Air Traffic Flow Management in the Free Routing Concept**

**Joost N. Kok, Leiden University, The Netherlands**

We present an evolutionary tool to solve free-route Air Traffic Flow Management problems within a three-dimensional air space. The tool solves free-route planning problems involving hundreds of aircraft in three dimensional sectors of large size (a few thousand kilometers). We observe that the importance of the recombination operator increases as we scale to larger problem instances. The evolutionary algorithm is based on a variant of the elitist recombination algorithm. We give a theoretical analysis of the problem, and present the results of experiments.

## **An ES-based Approach to Facility Layout**

**Christoph Gorien, Volker Nissen, Universität Göttingen, Abt. Wirtschaftsinformatik I, Germany**

We present an evolutionary approach to solving facility layout problems with unequal areas and undetermined locations for single- and multi-floor problems. The method is based on a modification of Evolution Strategy (ES) and employs the concept of a space-filling curve. We compare our approach to previous work based on Simulated Annealing and Genetic Algorithms. The results confirm the potential of ES in this domain. An extension of this work to account for other practical constraints is planned.

## **Evolving Trading Models with Parallel EAs**

**Marco Tomassini, Swiss Federal Institute of Technology, Switzerland.**

Artificial evolutionary processes, such as genetic algorithms, are based on reproduction, recombination and selection of the fittest members in an evolving population of candidate solutions. Koza extended this genetic model of learning into the space of programs and thus introduced the concept of genetic programming. Each solution, in the search space, to the problem to solve is represented by a genetic program (GP), traditionally using the Lisp syntax. Genetic programming is now widely recognized as an effective search paradigm in artificial intelligence, databases, classification, robotics and many other areas .

We present a parallel genetic programming system (PGPS) which speeds up the searching process. In complex applications, like financial market forecasting, the evaluation of each GP is very lengthy while the time spent in the selection and reproduction phases is practically negligible compared with the population evaluation time. Parallelizing the evaluation phase on a parallel coarse-grain machine such as the IBM SP-2, can be done obviously. After the reproduction phase, each GP is simply sent to a processing node for evaluation, independently of operations in other processing nodes. However, the run time GP complexity with programs of widely differing sizes makes the design of a parallel algorithm and its implementation to obtain large speedups a nontrivial task. This irregularity causes some processing nodes to be idle while others are active. This run-time load imbalance lead us to design a

simple greedy task scheduling algorithm. For problems of large enough size, the parallelization of genetic programming on distributed memory machines can deliver a nearly linear speedup. The code was written in C++, using PVM3 message passing library and can be run on other parallel machines like the Cray T3D

A trading model is a system of rules catching the movement of the market and providing explicit trading recommendations for financial assets. Trading rules act on and logically combine so-called technical indicators, which are complicated functions (sophisticated moving averages) of the price time history. To simplify the learning process, we have used pre-optimized technical indicators that were obtained with niching genetic algorithms in a previous phase of our work. Trading models are represented as decision trees, each of which is a member of the GP population. The function set used in the construction of the trading strategies is composed of logical operators and the terminal set is composed of the pre-optimized indicators and the values of the trading recommendation (essentially sell, buy or do nothing). The results on the financial problem of inducing trading models for predicting the currency exchange market are encouraging. Currently, the best automatically evolved trading models have a performance on market data comparable to that of the hand-crafted models of the same complexity class. We hope that in the future we will be able to derive even more sophisticated models by GP induction.

## **Comparing EC, NN, Logistic Regression and CHAID on a Direct Marketing Problem**

**A.E. Eiben, Leiden University, The Netherlands**

EC entered the application area of classification problems by genetic programming. EC became thereby a competitor of NNs, logistic regression, or decision trees techniques. In marketing there are numerous classification, clustering and modelling problems that can be solved by either of the above techniques. In this talk we report on an application oriented research comparing these four techniques on a direct mailing selection problem at a multinational hardware and software company. We compare different features, such as accuracy of the prediction, interpretability of the predictor models, time needed to perform the data analysis and the required level of expertise to apply each technique. The comparison shows no clear winner; GP is slightly better in

terms of accuracy, but it costs more time to apply and at the moment there are very few off-the-shelf packages available on the market.

Note: this research has been conducted by the Marketing Intelligence and Technology Foundation coordinating joint research by the University of Amsterdam, Free University Amsterdam and the Leiden University.

## **Data Mining with EC**

**Joost N. Kok, Leiden University, The Netherlands**

More and more data about consumer-behaviour is becoming available through the use of database systems and electronic-banking systems. We describe an evolution-based technique to extract some useful information from a database holding data about customers. In this approach we try to learn more about the needs and wishes that lead consumers towards specific behaviour, so that better service can be provided in the future. We formulate a formal version of this problem, describe the evolutionary algorithm used, and show results on artificial and real data.

## **An Overview of EA in Management Applications**

**Volker Nissen, Universität Göttingen, Abt. Wirtschaftsinformatik I, Germany**

An overview of the state of art in management applications of Evolutionary Algorithms (EA) is presented. Interest in EA has risen considerably since 1990, but no true breakthrough on the level of practical applications could be observed so far. Genetic Algorithms are dominant with a focus on combinatorial problems. Apparently, there are substantially more prototypes than running systems. Results w.r.t. the competitiveness of EA are mixed. This, and the large number of design options of EA confuse potential users. There is a need for powerful and user-friendly tools as well as a demonstration center for EA-applications. Also, a solid basis for meaningful comparisons with other heuristics is lacking. In general, a stronger collaboration between science and practice would be highly desirable to advance the level of practical

applications. Of particular interest are hybrid systems such as combinations and integrations of EA with neural nets or fuzzy systems.

## Optimizing Photo Mask Layout for Grey-tone Lithography

Jörn Hopf, Max-Planck-Institute for Computer Science, Germany

Silicon micro machining uses equipment and processes of IC-technology to produce highly miniaturized three-dimensional micro structures like sensors, actuators, etc. The process of layer patterning usually is performed by photo lithography followed by dry etching. The photo lithographic step requires masks which control the grey-tone level during the exposure process. Hole arrangement with its restrictions caused by mechanical limits during the production process and diffraction during the exposure process could be improved by giving up the fixed raster conventional optimization processes need. This work presents the application of an Evolutionary Algorithm optimizing a "free" hole arrangement. First results show that the average deviation for even difficult structures like a Fresnel-lens can be kept below the wavelength of light. Future improvements will concern the number of actually four lithographic steps.

## System-Level Synthesis Using Evolutionary Algorithms

Tobias Blickle, Jürgen Teich and Lothar Thiele, Institute TIK, ETH Zurich, Switzerland

In this paper, we consider *system-level synthesis* as the problem of optimally mapping an algorithm-level specification onto a heterogeneous hardware/software architecture. This problem requires (1) the *selection of the architecture* (allocation) including general purpose and dedicated processors, ASICs, buses and memories, (2) the *mapping* of the algorithm onto the selected architecture in space (binding) and time (scheduling) and (3) the design space exploration with the goal to find a set of implementations that

satisfy a number of constraints on cost and performance. Existing methodologies often consider a fixed architecture, perform the binding only, do not reflect the tight interdependency between binding and scheduling, require long run-times preventing design space exploration or yield only one implementation with optimal cost. Here, a model is introduced that handles all mentioned requirements and allows the task of system-synthesis to be specified as an optimization problem.

This optimization problem is then solved by means of an Evolutionary Algorithm. A coding of the individuals and a fitness function are given and shown to be accurate on two case studies. The first study considers the H261 codec for image compression as an example for system-level synthesis. The second study is taken from the domain of high-level synthesis.

This paper will be available as TIK-Report No. 16 from

<http://www.tik.ee.ethz/Publications/TIK-Reports/TIK-Reports.html>

## Applications of EAs at Daimler–Benz

U. Kreßel, Daimler–Benz AG, Research Center Ulm, Germany

An overview of research and application activities at Daimler–Benz is presented, which is sorted according to the three application fields: pattern recognition, automation and control, and operation research. Also a few remarks are given, how a new technology such as evolutionary algorithms is introduced in a large concern like Daimler–Benz (e.g. BMBF project: HYBRID, since 1994; innovation project: GenOPro, since 1996).

In the second part of the talk it is shown by the example *‘recovering = writing traces in off-line handwriting recognition’* (work down by S. Jäger), how a specific application problem was tackled = with different evolutionary and genetic approaches=20 (firstly, local rules based on classifier systems and machine learning; secondly, global optimization by representing the task as a = travelling–salesman–problem; and finally, replacing the recombination = operator by a deterministic segmentation rule) and successfully solved.

# An Evolution Model for Integration Problems and its Application to Lighting Simulation

Brigitta Lange, Fraunhofer Institute for Computer Graphics, Darmstadt, Germany

A general evolution model for integration problems is introduced, and applied to solve a multidimensional integral equation modelling radiant light transfer. In lighting simulation the amount of light reflected from a point in a particular direction is determined by integrating the irradiance over all directions of the hemisphere. Unfortunately the irradiance distribution is a unknown, multimodal, and discontinuous function that has to be evaluated by a set of discrete samples, i.e. by applying computational expensive ray-tracing techniques. Hence the integral is approximated by a sum of weighted function values. The goal is to find the best approximation with the fewest number of function evaluations, but this can only be achieved by an adaptive sampling strategy. Since evolution has to be considered as an efficient adaptation process the optimization of numerical integration by means of evolution seems to be promising.

The concept is to maximize the confidence in the estimate by an evolutionary stratification of the integration domain producing successively better approximations. In contrast to classical evolutionary algorithms here every individual is only a part of the solution and represents a cell (weighted sample). Thus the whole population forms the estimated integral. Initially the integration domain is subdivided by few samples into cells. The fitness of each cell is given by an associated confidence value. Each cell has a chance of being selected for reproduction by a probability proportional to its confidence. The mutation operator models population growth by cell-splitting, i.e. subdivision of cells through insertion of new samples and the process terminates if all individuals are of equal confidence.

It is shown that this evolution process has the potential to achieve good approximations to the solution of the integral equation. In contrast to stochastic integration radiance information gained during sampling is exploited efficiently. Thus the simulation process becomes self-organizing and is not limited to any a priori assumptions on irradiance distribution which allows the system to adapt optimally to a particular lighting situation and results in a better convergence with a significant reduction in sample set size. In the presence of multimodal integrands it even may outperform numerical techniques since it has an innovation component. Thus the result does not depend on the initial spacing of knots and the algorithm has a chance to escape from local optima.

# Fuzzy Controllers and Evolutionary Algorithms

**Frank Klawonn, Department of Computer Science, Technical University of Braunschweig, Braunschweig, Germany**

The possibilities for applications of evolutionary algorithms to tuning and optimizing fuzzy controllers, or even to generate fuzzy controllers automatically, are investigated. There are various ad-hoc approaches to using evolutionary algorithms for the design of fuzzy controllers, which already indicated good results. However, there is a need for systematic techniques that take the properties of fuzzy controllers and evolutionary algorithms into account in order to obtain fast convergence and to be able to tackle more complex control problems. Therefore, special initialization heuristics, problem specific crossover and mutation operators and a changing fitness function, which in the beginning allows even very rough control strategies, are designed.

## Evolutionary Inverse Problem Solving

**Marc Schoenauer, CMAP – CNRS URA 756, Ecole Polytechnique, Palaiseau, FRANCE**

Numerical simulations of physical processes very often involve some internal state law to compute the behavior of the physical system under experimental conditions. When new systems are set up, these internal laws have to be identified before any future reliable simulation can take place. Expertise and slight modifications of existing laws is often the actual way of deriving accurate models for internal laws of new systems.

Experimental results regarding the behavior of some new system under known experimental condition offer another possibility for the identification of the internal law of that system. Whereas standard (e.g. gradient-based) technique should be used whenever possible, the nature of these inverse problem often leads to ill-posed mathematical problems, resulting in divergent or unstable numerical methods. This is where Evolutionary Computation can bring real breakthroughs: the comparison of the results of a direct simulation involving a potential model for the target internal law and the actual experimental results on the physical system give a quality of the model at hand, and allows to derive a fitness function for an evolutionary algorithm. The critical



issue is the choice of the search space, i.e. the choice of a representation for the function to identify.

Parametric models (e.g. polynomials of degree less than  $d$ ) are either limited in approximation capabilities, or lead to a very large number of parameters which makes the identification task very difficult. On the other hand, non-parametric models need to specify both the architecture and the parameters of the model, requiring specific evolution operators. It is however, the most promising approach when no a priori information on the analytic formulation of the target law is available.

Two examples illustrate this approach:

### **Part I: Identification of the isotherm function by recurrent neural networks**

Chromatography is a separation process of a chemical mixture based on the different affinities of the components of the mixture for a porous medium. The velocity of one component into the chromatographic column therefore depends on the concentration of all other components. This dependency is given by what the chemical engineers call the *isotherm function*. The mathematical model of the chromatographic process, obtained through the equation of mass conservation, is a nonlinear hyperbolic system of partial differential equations for which standard robust and accurate numerical schemes exist: this allows the straightforward application of the general principles stated in the introduction.

The chosen model for the isotherm function is recurrent neural networks. The evolutionary algorithm is based on Angeline's GNARL algorithm, where both the topology and the weights of the network are evolved through specific mutations. Original strategy for the application of the different mutation have been defined.

The first results (for one-component "mixtures") highlighted the need to take into account more than one set of experiments during the identification process for the sake of generalization. For instance, very good accuracy was reached using 9 experiments, the experimental conditions being equally distributed in the experimental domain. However, this approach does not scale up, and coevolution of isotherm function and experimental conditions will be investigated to tackle this problem.

### **Part II: Genetic Programming in Structural Mechanics**

The macroscopic mechanical behavior of a material can be predicted numerically (e.g. by solving a system of partial differential equations by Finite Element method) using a model for its microscopic properties, called the *constitutive law* of the material. The accuracy of these simulations, of utter importance in engineering design, depends on the adequacy of the constitutive law. When a new material is designed (e.g. composite, laminated

materials), the constitutive law is generally extrapolated from that of neighbor materials, whose parameters have to be modified through trial and errors by experts.

Hence, the identification of the constitutive law, from experimental results of the material actual behavior in given situations (forces, pressure, ...) is a good candidate for evolutionary identification as defined in the introduction. We restrict to the mechanical context of hyperelastic materials, where the constitutive law is given as a real-valued function of three variables, the strain energy function of the material.

However, the numerical simulation of the mechanical behavior by Finite Element methods requires the computation of values of the first- and second-order derivatives of the strain energy function, which makes Genetic Programming a perfect model – the derivation of a GP tree is straightforward. Nevertheless, the first results proved to be a complete failure, as random GP trees are not valid strain energy function. We needed to incorporate mechanical knowledge in the identification process, ending up in identifying Ogden materials, for which the strain energy function has a restricted form. Moreover, a careful strategy in the choice of the experimentations used for the identification had to be defined: increase the intensity of the loadings along the evolution proved necessary. This finally lead to the first successes in the identification of Mooney-Rivlin materials. The GP approach was validated against multi-start hill-climbing, and gave better results up to one order of magnitude in the error on the experiments.

## EAs and Neural Networks

**Martin Mandischer, University of Dortmund, Department of Computer Science XI, Germany**

We present an evolutionary approach for the design of feed-forward and recurrent neural networks. Therefore we developed a Genetic Algorithm with a non-binary and dynamic data-structure encoding. The modified GA can be used for the construction of networks for tasks from various domains like classification, control, function approximation and time-series prediction. The performance of the evolved networks is measured on standard benchmark data as well as on our own applications. Results are compared with other evolutionary created networks and the constructive Cascade-Correlation algorithm. In most cases the evolved networks showed better (and more stable) learning and approximation capabilities than standard networks. Their

performance on approximation was not as good as the performance of the Cascade-Correlation algorithm but yields always the best generalization results.

## **On a GA for the Selection of Optimally Generalizing Neural Network Topologies**

**Stephan Rudolph, Institute of Statics and Dynamics of Aerospace Structures, University of Stuttgart, Germany**

A genetic algorithm is used for the topology design of feed-forward neural networks. The generated neural network topology populations are then trained to approximate non-linear relationships of multiple variables. A specifically designed fitness function using the epistemological principle of dimensional homogeneity is used for the evaluation of the individual neural network generations and for the selection of the best generalizing neural network topologies. A general theory for the topology design and the explanation of the correct generalization capability of non-linear feed-forward neural networks is developed, mathematically proved and explained using simulation results.

**Reference:** Proceedings of the 2nd International Conference on Adaptive Computing in Engineering Design and Control'96, edited by I.C.Parmee, University of Plymouth, March 26th-28th, Plymouth, United Kingdom, 1996, 79-86.

## **Genetic Programming controlling a miniature robot**

**Wolfgang Banzhaf, University of Dortmund, Germany**

A very general form of representing and specifying an autonomous agent's behavior is by using a computer language. The task of planning feasible actions could then simply be reduced to an instance of "automatic programming". In this talk I report on our recent work in this area. We have evaluated the use of an evolutionary technique for automatic programming called Genetic Programming (GP) to directly control a miniature robot. Whereas simulations

have been done elsewhere, this is the very first attempt to control a physical robot with a GP-based learning method in real-time. Two schemes are presented. The objective of the GP-system in our first approach is to evolve real-time obstacle avoiding behavior from sensorial data. This technique allows training of stimulus-response tasks. It has, however, the drawback of the learning time being limited by the response dynamics of the environment. To overcome this problems we have devised a second method, learning from past experiences which are stored in memory. The new system results in a strong acceleration of learning. As compared to the previous system, the emergence of the obstacle avoiding behavior is speeded up by a factor of 40, enabling learning of this task in 1.5 minutes. This learning time is several orders of magnitudes faster than comparable experiments with other control architectures. We discuss the advantages and disadvantages of the new method and introduce the range of behaviours that have been examined so far.

Work in collaboration with Peter Nordin, Markus Olmer

## Programmatic Compression

**Peter Nordin and Wolfgang Banzhaf, Fachbereich Informatik, Universität Dortmund, Germany**

Programmatic compression is a very general form of compression. The basic idea behind this technique is that any system, which evolves programs or algorithms for generating data, can be viewed as a data compression system. The data that should be compressed are presented to the Genetic Programming system as fitness cases for symbolic regression. After choosing a function set that facilitates an accurate reproduction of the uncompressed data, the system then tries to evolve an individual program that, to a certain degree of precision, outputs the uncompressed data. If the evolved program solution can be expressed by fewer bits than the target data, then we have achieved a compression. In principle, this technique could address any compression problem with an appropriate choice of the function set and the method has a potential for both lossy and lossless compression. We have applied a Compiling Genetic Programming System (CGPS) to the programmatic compression problem. A CGPS is up to 2000 times faster than a LISP Genetic Programming system which enables the application of programmatic compression to real-world problems.

## Two remarks about Theory

Volker Claus, University of Stuttgart, Germany

First remark: When you teach topics of evolutionary algorithms, the convergence theorems are usually proved in an analogous way. We present a theorem for a large class of heuristics which are based on variation and selection and which uses random numbers. This theorem is due to M. Kohler (1995) and extends the convergence of genetic algorithms of Eiben and al. (PPSN I). We are using this theorem in our lectures for proving the convergence of genetic algorithms, evolutionary strategies, and simulated annealing. The theorem covers even the case that an optimal value will be forgotten during the iterations.

Second remark: If you want to test the quality of any heuristic, you may take one of the usual functions (see de Jong), or there are tables of datas for the TSP or other problems. We propose to use the so called C-function if you have to deal with permutations. This function sums up all the gradients of the graph of a permutation, i.e.,  $C(g)$  is the sum of the absolute values of  $(g(j)-g(i))/(j-i)$  for a permutation  $g$  of order  $n$  and all different  $j$  and  $i$  (ranging from 1 to  $n$ ). Though there are good bounds for the maximum value  $\text{Max}C(n)$  (maximum for all permutations of order  $n$ ), the exact value of  $\text{Max}C(n)$  is unknown for  $n \geq 40$ , and the C-function has many local optima with respect to the environment defined by the LIN-2-operator. Therefore, the calculation of  $\text{Max}C(n)$  may serve as a good testfunction.

## Genetic Programming controlling a miniature robot

Wolfgang Banzhaf, University of Dortmund, Germany

A very general form of representing and specifying an autonomous agent's behavior is by using a computer language. The task of planning feasible actions could then simply be reduced to an instance of "automatic programming". In this talk I report on our recent work in this area. We have evaluated the use of an evolutionary technique for automatic programming called Genetic Programming (GP) to directly control a miniature robot. Whereas simulations have been done elsewhere, this is the very first attempt to control a physical robot with a GP-based learning method in real-time. Two schemes are

presented. The objective of the GP-system in our first approach is to evolve real-time obstacle avoiding behavior from sensorial data. This technique allows training of stimulus-response tasks. It has, however, the drawback of the learning time being limited by the response dynamics of the environment. To overcome this problems we have devised a second method, learning from past experiences which are stored in memory. The new system results in a strong acceleration of learning. As compared to the previous system, the emergence of the obstacle avoiding behavior is speeded up by a factor of 40, enabling learning of this task in 1.5 minutes. This learning time is several orders of magnitudes faster than comparable experiments with other control architectures. We discuss the advantages and disadvantages of the new method and introduce the range of behaviours that have been examined so far.

Work in collaboration with Peter Nordin, Markus Olmer

## **Missing Features in Current Evolutionary Algorithms**

**Hans-Paul Schwefel, University of Dortmund, Germany**

While current Evolutionary Algorithms (EAs) mostly handle static single-criteria optimization problems and want to achieve ultimate solution accuracy just once in a run, organic evolution happens continuously in a dynamic environment (influenced itself recursively by all actors of the game), is always governed by multiple criteria (e.g. predators with different skills), and maintains evolvability. Meliorization thus may be a more appropriate term than optimization to describe the algorithmic model. These and other observations like multi-cellularity, individual ontogeny, and the formation of genetically controlled genotype-phenotype mappings as well as individual brains and cooperative phenomena between several individuals and species might lead to scepticism whether EAs really mimic natural evolution processes.

These observations may, however, as well give wings to the development of better and more generally applicable EAs that can handle broader classes of optimization problems. Some examples are given with respect to dynamic optimization, multi-criteria decision making, endogenous self-adaptation,

and more.

Next, the question of appropriateness of the natural metaphor to look for specialized solution algorithms is handled. The answer is that there is no necessity to mimic nature. Besides of the NFL (no free lunch) theorem which states that any algorithm that gains some superiority in solving specific problems at the same time loses w.r.t. other problem classes, there will always exist optimal non-natural specific solution concepts for specific tasks. EAs may be good for an ever diminishing rest (still huge in the next decades and centuries) of problems, only.

However, models of natural evolutionary processes and observations about their computational effectiveness and efficiency under artificial evaluation criteria might help in better understanding a variety of ecological as well as economic phenomena and thus be helpful for other scientific disciplines than merely numerical optimization. This is the ultimate reason why they will retain their attractiveness.