On the Evolution of Evolutionary Computation

Hans–Paul Schwefel
University of Dortmund
D–44221 Dortmund, Germany

Keywords: *history of EC, differences and similarities between Genetic Algorithms, Evolutionary Programming, and Evolution Strategies, outlook and suggestions for further work in the field*

Abstract — This paper does not present any new developments in the field of Evolutionary Computation – neither concerning theory nor applications. It tries, however, to give an overview of motivations in the field and to conclude from the differences between the three main species of algorithms that there are a lot of open questions how to model evolution for the sake of increasing the algorithms' usefulness, which even now is undoubted despite of their simplicity.

I. Introduction or: Is it allowed to imitate life ?

It is a matter of fact that some living beings sometimes happen to imitate other living beings. In the famous case of mimicry e. g. butterflies, which are tasteful for predating birds, have been observed to resemble forms or patterns of other, less tasteful, even inedible, butterflies. They do so, as we say, in order to get some protection from their enemies. But they don't do it consciously on the level of the individuals involved. It is our simplified interpretation on the level of populations or species over several generations of what is really happening as an evolutionary process. Imitators, which resemble the model more closely, simply live longer and have more descendants, which become parents of the next generation. Thus their genetic information spreads within the pool of living genes, as long as environmental conditions remain the same.

There are a lot of examples for human trials to imitate forms, structures, or even processes from various domains of natural prototypes. One example shall be looked at here more closely, i. e. the way we have learned to fly – artificially, by means of a machine. The first men, who were reported to have tried it, Deadalus and Icarus, took birds as blueprints, and failed. Even Lilienthal during this century made the same mistake, when he designed the wings of his flying machine too closely according to the profiles of small birds' wings. He paid that error with his life. Nevertheless, we survivors have learned a lot from such explorations in the field. Nowadays we understand the physics, especially aerodynamics, and need no longer rely upon trial-and-error in the large. We can deduce from our knowledge that an optimal profile of a large wing must be thicker, and are able to build quite reliable aircrafts without much testing of the prototypes. I wonder, however, whether anybody would have hit upon the idea to try flying if birds would not exist. Even very recent improvements like winglets at the end of airplanes' wings and shark–like riblets on the surface of their bodies do not stem from deductive use of hydrodynamic theory, but from inspirations (at least) gained from observing natural life.

Whether it is allowed to imitate life should not be questionable, therefore. The purposes pursued, of course, must be checked thoroughly against ethical principles. And it always remains an open question, how closely one should imitate a living model. The answer generally will depend on the effectivity and efficiency of the result. Sometimes one may even win better insight into the mimicked system as a byproduct from using nature's patents.

II. Evolutionary Computation — an outline

A more conscious look for natural metaphores to solve (mainly, but not only) technical problems has been taken since the onset of such inter-, multi- or transdisciplinary endeavours like cybernetics and bionics in the 1940ies and 1960ies, respectively. I do not want to go into a detailed history of those areas here, but closely related to the former have been some preliminary ideas to create intelligent automata/computers by looking for natural processes with seemingly adaptive or even creative capacities (see e. g. [1]).

The (human?) brain, the immune system of living beings, even simple living beings as a whole, and the process of organic evolution, which has led to such products worthy of imitation, currently are of broad interest as models for creating such 'intelligent' machines, like neural nets, cellular automata, or animats, or at least software for existing computers. New computing architectures ask and allow for such new kinds of programs even more.

This should be allowed, as we have argued before, even if it were for the sake of adding to our so far not yet excellent knowledge about the mimicked phenomena only. But why do we do it, how, and for which purposes? These are questions, which are worthwhile to be treated.

A. Why?

My speculative answer to why we imitate life on computers stems from an observed mismatch between the range of traditional crisp computing methods and the tasks we want to tackle today. That does not mean to abandon traditional methods, but merely to add new ones to our toolboxes for solving problems. In order to become more concrete, I shall address one type of problem, which is not the only but a dominant one in case of Evolutionary Computation [EC], i. e. optimization.

Nobody should make use of EC in cases where the good old methods like Linear and Dynamic Programming, quasi–Newton, or other theoretically well underpinned methods work. None of the Evolutionary Algorithms (EAs) would do the job better nor even as good as those. EC should be taken into consideration if and only if classical methods for the problem at hand don't exist, are not applicable, or obviously fail. Even then, at least two alternative approaches should be discussed: either total enumeration or other brute force methods, when the necessary computing power is available, and, last not least, the development of a specific method, which makes full use of the knowledge about the problem's structure, like in expert systems. EAs are weak methods, which should be handled as a last resort. Their success and attractivity has two main roots: One is the vastness of the field of applications after subtracting those areas covered by strong or specific methods, the other is the fact that EAs inspire many researchers as well as users to invent their own instantiations or even completely new versions. It is true, indeed, that so far no EA has reached a state of maturity. In comparison to organic evolution all computer analogues are still quite primitive. Nevertheless they can be and have been applied to many hard problems. Recently Alander [2] has collected nearly 1600 references on EC.

B. How?

Trials to imitate the function of a (human) brain have been embraced much easier than trials to imitate organic evolution. Human hubris has made it possible to believe in an intelligent late product, i. e. mankind, from a stupid process, i. e. organic evolution. Even today, many people think that evolution is a prodigal process and they would model it similar to Ashby's homeostat [1], i. e. as a pure random or Monte–Carlo (MC) method. They do not realize the fact that life exists on earth since 10^{17} seconds only and that this would by far not have been sufficient to solve the combinatorial task of putting together the DNA of the simplest yeast or bacterium. This argument was used sometimes to deny the fact of evolution. If a model's result is in discrepancy with the real world, one must look for a better model, however. Parallelity is just one trick but not the only one, and limited as well by only 10^{80} quarks available in the universe.

More serious are objections to comparing evolution with an optimization procedure. Let's take aside those pessimists which believe that the world turns to the worse and worse – cultures believing that the golden age lay in their past always have vanished. But even from an optimistic point of view that evolution gradually, punctualistically, or at least in the long run creates new individuals, species, taxa, or ecosystems, and prefers the survival of the better ones, the term 'better' is ambiguous and questionable.

If we optimize a device or system we usually do have a prefixed evaluation criterion with which we can decide upon improvements and deteriorations, and in most cases there is just one globally optimal solution – a distinct point in the space of the decision variables. The situation becomes more difficult, if noise or deterministic chaos and decision uncertainty comes into play, if the optimum changes its location over time due to uncontrollable environmental influences, and especially if there are several non-cooperative criteria. In the latter case the sought for optimum is no longer represented by a single point but by a subspace of efficient or Pareto-optimal solutions.

All these and even more complications are given in the real world where evolution works. Even the numbers of variables, objectives, and constraints are variable over time, and due to backcouplings between the current positions of all living individuals to the response surfaces given by the selection criteria, the search for improvements has to be performed on some kind of huge trampoline with many, partly invisible people and a long memory of their actions. If we look onto our long-term search for technical, organizational, and other progress, the situation is the same! That is one more reason for trying to understand how nature has overcome these problems. A good Evolutionary Algorithm should be able to do its work even under such turbulent conditions. A video film will demonstrate it. Instead of 'optimization' I prefer the term 'meliorization', therefore, but then I do see many similarities between organic evolution and meliorization procedures which we could, may be should, make use of.

C. What for?

The real world is nonlinear and dynamic, and thus full of phenomena known as deterministic chaos, (structural) instabilities, and fractal geometries. Ever more realistic computer simulation models, which no longer ignore such phenomena, must lead to complicated input-output relations. If we try to find out optimal parameters within such models minimizing or maximizing some quality measure, given as response from running the model, traditional strong methods for doing so will have their problems. They were conceived for linear, continuous and at least (twice) differentiable worlds. Evolutionary Algorithms are a means to overcome these limitations, not by guaranteeing the exact solution with a few (numerical) experiments, but by finding a good approximation in less than exponentially increasing time, when the number of decision variables goes up.

One should not forget, on the other hand, that EAs can do such work (and did so in their past) not only in a computer modelled world, but in the real world itself, as well. Experimental optimization still is necessary in some domains, where our knowledge is too poor for formalizing all relationships appropriately, or where exact calculations are too costly.

III. Evolutionary Computation — a brief history

Nobody really can say who devised the first EA. Was it Fraser [3], Friedberg [4], Anderson [5], Bremermann [6], or somebody else (for some more forerunners see e. g. Schwefel [7])? It is for sure, however, that today there are three different schools whose roots have been developed independently from each other. Only recently they have found together and agreed upon the common denominators EC and EAs for:

- Genetic Algorithms (GAs), first introduced by [8], and first used for optimization tasks by [9];
- Evolutionary Programming (EP), first roots of which were laid by [10], and designed by [11] in its currently practiced form;
- Evolution Strategies (ESs), first introduced by [12] and further developed by [13].

Genetic Programming (GP) and other special subbranches of the GA philosophy like Classifier Systems (CSs) are out of the scope of this short article.

Two facts are astonishing: The foundations of all three kinds of EAs have been laid in the 60ies, and up until now they have maintained their considerable differences in modelling evolution. Since the different versions of EAs will be handled in more detail within other contributions to this conference, I shall try to give an overview only with respect to similarities and differences in general. A more detailed analysis has been given in [14]. Some of the differences result from the situation under which the algorithms were devised. Others lie in the representation form of the individuals. More striking, however, are the differences with respect to the reproduction cycle and the variation operators. The latter comprise mutation and recombination, the relative importance of both being controversially discussed as well. In the following only canonical versions of all EAs will be considered. The vast variety of special incarnations must be ignored here for the purposes of clarity and pinpointing main issues.

A. Original purposes

The first GA was devised by its originator as an adaptation process, only later on special GAs were used as optimization tools for computer models.

EP, it has been called so later on, stems from the idea of building a finite state machine to solve prediction tasks. Nowadays it is used for numerical optimization mainly.

ESs were devised as experimental optimum-seeking procedures in the beginning. Only later on they found their way to computers.

B. Representation of individuals

GAs started and still mainly operate with binary strings for representing individuals (their genotype). If the objective function is not a pseudo-Boolean one, each string has to be decoded into a set of appropriate decision variables (the phenotype), before the fitness of the individuals can be evaluated.

ESs started with integer variables as an experimental optimum-seeking method, but turned to real variables when used on computers. The individuals are not only represented by the set of decision or object variables, but additionally by a set of strategic parameters controlling the variation process, i.e. variances and covariances. This latter set has been called ‘internal model’ of the environmental laws by the author and is learned on-line during the search for optima.

EP in its current form relies upon real variables, both for the object variables and the strategy parameters, which are adapted according to exogenous rules.

The main iteration loop of all EAs is the same, may it start with the selection/reproduction or the recombination/mutation steps.

C. Mutation

In the bit-string world of GAs, mutations are purely random bit inversions, occurring with low frequencies generally. Some experts deny their necessity.

ESs and EP use Gaussian noise with zero mean to perturb all object variables, ESs additionally logarithmic normal distributions for the standard deviations of the mutation step sizes (for their scaling) and normal distributions for changing the covariances (which may lead to correlated mutations).

D. Recombination

It's very interesting to observe that GAs emphasize the role of recombination, e. g. in the form of two- or multi-point crossover, whereas EP rejects this form of variation as useless or sometimes even harmful. An explanation can be found if one looks into probability distributions for changes of the object variables on the level of phenotypes, i. e. after decoding the bit strings. Then one realizes that crossover in GAs may lead to recombinants which lie outside of the hypercube spanned by their parental positions. Thus recombination in GAs contains an element, which in ESs and EP is achieved by mutations only.

ESs rely on both mutation and recombination. Discrete recombination, recommended for object variables, is similar to uniform crossover if the crossover points lie on the boundaries of the partial bit strings, which encode the different phenotypic object variables. Intermediate recombination, recommended for strategy parameters, helps to avoid over-adaptation, but may lead to a loss of diversity of 'internal models' of the individuals and must be counterbalanced by mutation.

E. Selection

The most striking differences exist between GAs and EP on the one hand and ESs on the other hand with respect to the selection procedures. But it's not merely the scheme of assessing the individuals for their environmental compatibility, which plays a role here. Two other processes are intermingled and mostly modelled as one complex: the generation transition and the mating behaviour.

If one excludes elitist variants, where good parental positions cannot get lost (that is good for proving global convergence [15]), all three classes of canonical EAs give their individuals a life span of one generation, or in other words, they let their individuals have children only once at the same time. Only ESs operate with a surplus of descendants, the (μ, λ) version with $\lambda > \mu$ children from μ parents. This helps in handling inequality constraints, the violation of which leads to lethal descendants. Canonical forms of GAs and EP let their parents have just one descendant on average. This is true for GAs with crossover as well, since only one of the two recombinants is used later on, generally at random, i. e. without comparing fitnesses.

With proportional selection as well as most other forms like (linear) ranking, all individuals produced during generation g within GAs and EP have a chance to have children themselves in the next generation $g + 1$. ESs, however, discard the $\lambda - \mu$ worst descendants, what can be achieved without really sorting them. The remaining μ individuals become parents of the next generation and do have equal chances to mate and have children. GAs and EP allocate mating as well as reproduction probabilities to their individuals according to the relative fitness (e. g. objective function) values or the relative position in the ranking process.

IV. Evolutionary Algorithms — open questions and future work

From the diversity of the three approaches mentioned above one should not conclude that just one of them finally will become the one and only survivor – may it be on the basis of some selection criterion like degree of current popularity or even quality with respect to some more or less preliminary test series. Recombination and mutation of ideas is the best we could achieve, gaining some perhaps completely new and better approaches. There is no 'true' model of reality, nor of organic evolution. Still all three species of EAs are rather primitive and hopefully prone to improvement.

As mentioned in the introduction, the hope was formulated that from modelling evolutionary processes one should get some feedback from the results to a better understanding of organic evolution itself. One must be very careful, of course, in comparing EA behaviour with the real world, especially with human affairs. Nevertheless, it is attractive.

I want to mention just two similarities, which I found between ES behaviour and reality. There is a famous theoretical result of Rechenberg, that the optimal standard deviation of mutations corresponds to a success probability of about one fifth. The industries, which are operating in the fields of fashion products and music records, are truly groping in the dark and cannot rely upon robust prediction methods. From newspapers I recently learned that about 80% of the disks produced are not sold, but come back to be recycled, and only 20% of the new dresses presented during fashion parades are really ordered, while the rest never finds a customer. Isn't that astonishing?

The second experience was that during the self-adaptation of the strategy parameters within ESs, a population can achieve nearly maximal convergence rates as if it knew the optimal parameters a priori. But if one looks, whether at least the best individuals have adapted their internal models consistently to their 'real world', this is not the case. The near optimal behaviour of the whole stems from the diversity of non-optimal internal models of the individuals [16]! Isn't that even more astonishing and reason for hopes that collective intelligence is possible without perfect individual knowledge?

Nevertheless, EAs are still extremely primitive models of organic evolution. Even single-cell organisms are more sophisticated then EA individuals. They do have a complicated epigenetic apparatus, which transforms the genetic information into amino acids, proteins and enzymes, and finally to their phenotype. ESs and EP model the variation process in a more descriptive way, i. e. by omitting the lower level mechanisms. GAs on the other hand use a more explicative model, but do not vary the transmission rules to the phenotype. Organic evolution nowadays operates at least with a fixed, only near optimal, genetic code, the first transmission step from triplets of four nucleotides to twenty amino acids. But this will have changed itself in early steps of evolution! We should not compare EAs for new application tasks with a steady-state near status of organic evolution. In many cases we begin from scratch. The fact that today mutation rates are very low in most cases observed, may be due to having reached (local) optima or stationary states with respect to current environmental conditions only. In earlier stages it might have been much higher. From theory we learn that optimum mutation rates are inversely proportional to the number of decision variables involved and proportional to the distance from the optimum. Every good EA should include a variable variation scheme, the (internal) parameters of which have to adapted on-line – not by rerunning again and again several evolutions sequentially. At least that is not nature's way.

There are many open questions with respect to mutation and recombination operators. Even the latter could be parameterized and the parameters become part of the individuals' genome, for example. This is a vastly open field.

Looking onto the selection/reproduction operators in EAs, it is just the other way round: GAs and EP model it in a descriptive or more macroscopic way, whereas ESs model it more explicatively. I don't know why, but I observe that many EA researchers are not willing to experiment more rigorously with the form of the mating/reproduction probability over fitness curve. For ESs it is constant for the μ fittest individuals and zero for the others, for GAs and EP it is monotonically decreasing from higher to lower fitness values. Both forms are somehow artificial. From empirical observations one knows that sometimes the strongest males in a population do have the smallest number of descendants. They use most of their energy to attack their weaker colleagues. In other cases one observed that most of the descendants (75%) came from rather few (25%) parents, and the latter were those with medium properties. Perhaps the form of the selection/reproduction profile should be parameterized as well (may be on the level of competing sub-populations). The outcome would be a variable selection pressure at least, and this is very important for the balance between exploration and exploitation, or in other words between effectivity and efficiency of all EAs. There remains much to be done.

Multicellular organisms with somatic mutations during cell-divisions, species with two sexes, which are genetically different and represent different skills, diploid or even polyploid instead of only haploid individuals, and many other seemingly important features have been taken into account within EAs rather sparsely or not at all until now. There should be much more research into those directions, e. g. in order to solve multicriteria optimization, self-adaptation of internal parameters in general, and optimum control problems as well.

Much more could be done also into the direction of parallel EAs. Especially the selection process must become asynchronous in order to make better use of MIMD computer architectures. Here the balance must be found between communication and processing load on all nodes. Migratory as well as diffusion models and may be even predator-prey mechanisms should be investigated in more detail. May be that one day someone will hit upon the idea to fit the architecture of a parallel computer to the rules of a multi-purpose flexible Evolutionary Algorithm. Then the age of Evolutionary Computation would have reached its climax.

References

[1] Ross W. Ashby. *Design for a brain*. Wiley, New York, 2nd edition, 1960.

[2] Jarmo T. Alander, editor. *Proc. 1st Finnish workshop on genetic algorithms and their applications, Helsinki, November 4-5, 1992*. Bibliograhy, pages 203–281, Helsinki University of Technology.

[3] A.S. Fraser. Simulation of genetic systems by automatic digital computers. *Australian J. of Biol. Science*, 10:484–499, 1957.

[4] R.M. Friedberg. A learning machine: part I. *IBM Journal*, pages 2–13, January 1958.

[5] R.L. Anderson. Recent advances in finding best operating conditions. *J. Amer. Stat. Assoc.*, 48:789–798, 1953.

[6] H.J. Bremermann. Optimization through evolution and recombination. In M.C. Yovits, G.T. Jacobi, and D.G. Goldstein, editors. *Self–organizing systems*, Spartan, Washington D.C., 1962, pages 93–106.

[7] Hans-Paul Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, 1981. (2nd edition in preparation).

[8] John H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.

[9] Kenneth De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. Thesis, University of Michigan, 1975.

[10] Lawrence J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial intelligence through simulated evolution*. Wiley, New York, 1966.

[11] David B. Fogel. *Evolving artificial intelligence*. Ph.D. Thesis, University of California, San Diego, 1992.

[12] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann–Holzboog, Stuttgart, 1973.

[13] Hans-Paul Schwefel. *Evolutionsstrategie und numerische Optimierung*. Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering, 1975.

[14] Th. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[15] Günter Rudolph. Parallel simulated annealing and its relation to evolutionary algorithms. In I. Maros, editor, *Symposium on Applied Mathematical Programming and Modeling, APMOD 93*, volume of extended abstracts, pages 508–515, Budapest, January 6–8, 1993. Hungarian Academy of Sciences.

[16] Hans-Paul Schwefel. Collective phenomena in evolutionary systems. In P. Checkland and I. Kiss, editors, *Problems of Constancy and Change — the Complementarity of Systems Approaches to Complexity, Papers presented at the 31st Annual Meeting of the Int'l Soc. for General System Research*, volume 2, pages 1025–1033, Budapest, June 1–5, 1987. Int'l Soc. for General System Research.