On Interactive Evolution Strategies

Thomas Bäck[1,2], Ron Breukelaer[1] and Michael Emmerich[1]

[1] Natural Computing Group, University of Leiden, Niels Bohrweg 1
2333-CA Leiden, The Netherlands
{baeck, rbreukel, emmerich}@liacs.nl
http://www.liacs.nl/ emmerich
[2] NuTech Solutions GmbH, Martin Schmeisser Weg 15,
44227 Dortmund, Germany

Abstract. In this paper discuss evolution strategies in the context of interactive optimization. Different ways of interaction will be compared and classified. A focus will be on the suitability of the approach in cases, where the selection of individuals is done by a human expert based on some subjective criteria. First of all, this paper will study the behavior of the step-size adaptation mechanism, which might be seen as the most distinguishing feature of evolution strategies as compared to other evolutionary algorithms. Moreover, we compare the convergence dynamics of different approaches, and discuss typical patterns of user interactions as observed in empirical studies.

The discussion of empirical results will be based on a survey conducted via the world wide web. Color (pattern) redesign problems were chosen as test case. The simplicity of the chosen problems allowed us to let a large number of people participate in our study. The amount of data collected made it possible to add statistical support to our hypothesis about the performance and behavior of different interactive evolution strategies, and last but not least helped us to figure out high-performing instantiations of the approach.

1 Introduction

The research field of human-algorithm interaction (HAI) puts forward the involvement of human beings in the algorithmic solution processes. In contrast to human computer interaction the focus of this technology is on computational processes that are assisted by users. In contrast to interactive software like text processing systems or drawing software, the main structure of the solution process for the higher level task is still governed the algorithm, The user has to do assist the algorithm at some stages, that call for decisions based on subjective preferences, or that require the insights of experts in a problem, the formalization of which is often very difficult.

On a very global level we propose to distinguish between *reactive* or *proactive* interaction, i.e. user feedback requested by the algorithm, or optional interventions by the users into an autonomously running algorithm. An example for reactive feedback could be the request of an optimization algorithm on the subjective evaluation of solutions by means of the user. Contrarily, an example for proactive feedback would be given, if the user halts an optimization algorithm that is in a phase of stagnation, changes some parameters, and lets it continue with the changed settings. A boundary case for proactive feedback, would be, if the user simply decides to finish an algorithm and pushes some 'stop' button that terminates it.

Among the few algorithmic classes that already integrates the user in the computational process, interactive evolutionary algorithms are one of the most well known. Applications range from arts [1] and music [13], to industrial engineering applications [14, 9], mixture optimization [12], and prototyping in product design [6]. An excellent overview on applications of IEA was given by Banzhaf [6] and more recently by Takagi et al. [19].

In this paper we will mainly focus on the discussion of interactive variants of evolution strategies (ES) [18, 16]. ES are instantiations of evolutionary algorithms that are mainly used for the purpose of parameter optimization. In particular they feature self-adaptive parameters of the stochastic distribution used in the mutation. This allows to minimize the effort of the user when working with these evolutionary algorithms, as for many other EA the choice of the adequate parameters can cause a significant problem for the unexperienced user. Moreover, the self-adaptation makes it possible to automatically scale the behavior of the variation operator between a more exploratory coarse sampling or a finer sampling, which is needed to achieve a high approximation accuracy to the optimum in the end.

Evolution strategies have been already successfully applied for interactive optimization in parametric design. In particular the pioneering work of Herdy [11, 12] in this field should be mentioned here, who applied interactive variants of the evolution strategy to various problems ranging from the design of color mixtures to the search for coffee mixtures that meet a desired taste.

However, we belief that there are still many open questions with regard to interactive evolution strategies. For example the step-size adaptation deserves further attention, and the typical behavior of the user. Moreover, it is an interesting question how a theory of interactive evolution strategies might look like. In this paper, we intend to provide contributions to these questions. In particular we discuss new methods of how to conduct research in interactive evolutions strategies, analyze the user behavior in the selection process and study the feasibility of the self-adaptive step size adaptation within this context.

We will base our discussion on a representative problem for interactive evolution. The problem we have chosen the *re-design of RGB colors* by means of subjective evolution. The problem can be easily increased by using color patterns instead of single patterns. Moreover, it can be easily explained to people participating in experimental studies, and thus can be readily used for collecting statistical data in math experiments. In the studies presented in this paper, a survey conducted via the world wide web served us to gain a larger amount of data, which allowed us to better support the hypothesis about the algorithms' behavior and performance.

As it can be concluded from this introduction, the contribution of this paper not only to be seen in the presentation of new empirical results, but also to a fair extend on the discussion of a general research methodology in the field of interactive evolutionary algorithms.

The structure of our paper is as follows: After a short introduction to evolution strategies (Section 2) we will present a survey on interactive evolution strategies (Section 3). We continue with a discussion of self-adaptive features 4 in evolution strategies and discuss their role in interactive evolution strategies. Finally, in section 5, we report on first statistical studies of self-adaptive IES on color (pattern) redesign problems. The paper concludes (Section 6) with a summary of first results and an outline of some open questions for future research.

2 Evolution Strategies

The main loop of the evolution strategy is displayed in algorithm 1. Firstly, the algorithm initializes a population (multi-set) P_0 of μ individuals (objective function vectors + mutation parameters) randomly (e.g. uniformly distributed within the parameter space). P_0 forms a starting populations, and within the subsequent *generational loop* a series of new populations $(P_t)_{i=1,2,\ldots}$ is generated by means of a stochastic process:

Algorithm 1 (μ, κ, λ)-Evolution Strategy

$t \leftarrow 0$
$P_0 = \mathbf{init}_\mu()$
while terminate$() = false$ **do**
  $Q_t \leftarrow \text{generate}_{\mu\rightarrow\lambda}(P_t)$ /* Generate λ new variations based on P_t */
  $P_{t+1} \leftarrow \text{select}^{\kappa}_{\lambda\rightarrow\mu}(P_t \cup Q_t)$ /* Select μ 'best' individuals */
  $t \leftarrow t + 1$
end while
return P_t

First λ random variations of individuals in P_t are generated by means of a variation operator, the details of which will be described later. The new variants form the population Q_t, the so-called offspring population. Then, among all individuals in P_t and Q_t the best μ individuals are selected by means of a selection criterion. Note, that in case of $\kappa < \infty$ only those individuals are taken into account that have been generated in an iteration t_0 with $t_0 > t - \kappa$. The selection process is usually governed by an objective function $f : \mathbb{I} \rightarrow \mathbb{R}$, i.e. the μ best solutions with regard to this function are selected. However, it is not always necessary to have an objective function, and it suffices to establish a ranking on the merged population or just to have some criterion that can extract the μ best solutions from all other solutions.

Note, that the stochastic process defined by the series $(P_1)_{i=1,2,\ldots}$ has the Markov property, if we consider the selection criterion to be fixed. This means that the stochastic distribution of P_{t+1} is determined by the population P_t.

The variation-selection process is meant to drive the populations into regions of better solutions as t increases. However, there is no criterion that can be used to determine whether the best region has been found (except in cases with a pre-defined goal or bound on the objective space). Hence the process is usually terminated in case of stagnation or if the user decides to stop it, because of his/her time constraints.

An operator that deserves some further attention in the evolution strategy is the variation operator that is used to generate offsprings. Let us first repeat that individuals within the evolution strategy consist of a vector of decision variables $\mathbf{x} = (x_1, \ldots, x_{n_x}) \in \mathbb{R}$ and a vector of parameters of the mutation distributions (often referred to as step-size vector) $\mathbf{s} = (s_1, \ldots, s_{n_s}) \in \mathbb{R}^+$.

A variation of this vector is generated via a mutation of the step-size vector and the subsequent mutation of the vector of decision variables using the new mutation parameters. As an example let us discuss the so called 3-point mutation operator, that works with a single step-size, i.e. $n_s = 1$ is described in algorithm 2.

Algorithm 2 Generate λ offspring via 3-point mutation

$Q = \emptyset$
for $i \in 1 \ldots \lambda$ **do**
  choose $(\mathbf{x}, \mathbf{s})$ randomly out of P_t
  $u \leftarrow \mathrm{uniform}(0,1)$ // uniformly distributed random number between 0 and 1
  $s_1' \leftarrow \begin{cases} s_1\alpha & \text{if } u < \frac{1}{3} \\ s_1/\alpha & \text{if } u > \frac{2}{3} \\ s_1 \text{ otherwise} & \end{cases}$
  for $j \in \{1, \ldots, n_x\}$ **do**
    $x_j' = x_j + s_1 \mathrm{normal}(0,1)$
    /* normal(0,1) generates standard normal distributed random number */
  end for
end for
$Q = Q \cup \{(\mathbf{x}', \mathbf{s}')\}$

In order to generate a new individual, first the step-size of the given individual is multiplied by a constant factor, the value of which can be 1, α or $1/\alpha$ depending on a random number. Then this step-size of the new individual is used to obtain the decision variables of the new individual. These are obtained by adding an offset to the corresponding value of the original individual. The value of this offset is determined by a random number that is gaussian distributed with mean value 0 and standard deviation s_1. The idea behind this mutation operator is that decision variable vectors that are generated with a favorable step-size are more likely to be part of the next generation, and thus also the information about the step size that was used to generate them is transferred to that generation.

The process of mutative step-size adaptation was investigated in detail by Beyer et al. [8]. Due to his findings, simple adaptation rules like the 2-point or 3-point mutatation for the step sizes serve well, if the population is small and only a few iterations of the algorithm can be afforded. For a higher number of iterations, say $t_{max} >> 100$, more sophisticated adaptation mechanisms should be considered for the parameters of the mutation. State of the art techniques include the individual step size adaptation by Schwefel [18] and the covariance matrix adaptation (CMA) by Hansen and Ostermeier [10]. Note, that in order to allow for a mutative step-size adaptation, a surplus of offspring individuals needs to be generated in each generation. The recommended ratio of $\mu/\lambda \simeq 1/7$ leads to a good average performance of evolution strategy in many cases [18].

3 Interactive Evolution Strategies

There are many possibilities to integrate user interaction in the evolution strategy. In general, we can distinguish between reactive and proactive feedback. Reactive feedback is feedback requested by the algorithm, e.g.

- the user might be asked for evaluation (grading) of offspring individuals
- the user is asked for selecting individuals
- the user is asked for generating variants

In contrast to this, proactive feedback denotes an optional intervention by the user, e.g.:

- he/she might change the step-size parameter actively, after watching the search process stagnate)
- he/she might insert a new individual into the population or actively change the parameters of an individuals (manual mutation)

In this paper we are more interesting in strategies with reactive feedback and the only proactive feedback will be given, when the user decides to stop the search process.

Probably the most simple form of reactive feedback that might be given by the user is to simple select the best individual(s) out of a population, as suggested by Herdy [11]. A more complicated scheme of subjective evaluation would be a grading procedure, where the user has to provide a grade to each individual. Note, that the information the user has to provide in each iteration, consists of λ numbers in case of grading, and μ numbers, namely the number of the best solution, in case of selecting the best variant. As $\lambda \gg \mu$, the latter seems to be a favorable choice.

A selection procedure following this strategy is described by the simple algorithm:

- The algorithm presents the λ new solutions from Q_t and the μ solutions from P_t that have not exceeded a maximal age of κ generations to the user

– The user decides which one of them are the best μ solutions and these form the new population

A theoretical analysis of such kind of processes involves different kind of complications. First of all we need to find an adequate model. A model that is frequently used for the analysis of evolution strategies is that of a Markov chain. A markov chain can be viewed as an autonomous stochastic automaton $(S, Pr : S \times S \to [0,1])$, where S denotes a state space, and Pr denotes a function that assigns a probability to each state transition, such that $\forall s \in S : \sum_{s' \in S} Pr(s, s') = 1$. By setting $S = \mathbb{I}^{\mu}$ evolutionary algorithms on a finite search space can be modelled as Markov chains. This allows to obtain results about the limit behavior and average behavior on some test problems.

It was suggested by Rudolph [17], to extend this model to a stochastic mealy automaton with deterministic output, in order to model interactive evolution strategies. Such an automaton would be denoted with $(S, X, Pr : S \times S \times X \to [0,1])$, where X denotes a set of input symbols. Now, the probability function $Pr(S, X, Pr : S \times S)$ assigns a probability value to each state, input pair $(s, x) \in S, X$. Accordingly, the function $Pr : S \times X \to S$ must obey $\sum_{(s,x) \in S \times X} Pr(s, x, s') = 1$.

An interesting observation is now, that given a stream of inputs, the behavior of this strategy breaks down to a markov chain, where by the input stream becomes part of the deterministic formulation of Pr. This, for example, provides us with an means to analyze the best case behavior of a strategy for some target function f, by minimizing the mean convergence time t_{conv} over all possible user inputs:

$$E(t^*_{conv}(f)) = \min_{\mathbf{w} \in X^*} E(t_{conv}(\mathbf{w}, f)) \tag{1}$$

This convergence time of an 'ideal user' could be compared to the real behavior of a user in order to assess the performance of the user interaction and judge if this is the weak point of the algorithm. If so, further measures to support the user interaction might be considered, like user modelling, a better presentation of the variants or even the request of further information from the user.

However, even for quite simple variant of interactive evolution strategies, the computation of an ideal user behavior might be a challenging task. From a more abstract point of view, the input of the user in an interactive evolution strategy with subjective selection is a set of disjoint indices $\{i_1, \ldots, i_\mu\} \subset \{1, \ldots, \mu + \lambda\}$. Accordingly, he/she has $\binom{\lambda+\mu}{\mu}$ possibilities of choice, the value of which reduces to λ in case of a 'comma' strategy with $\mu = 1$. Hence, in case of t time steps there are already λ^t possible input streams that need to be considered. Hence, without any simplifying assumptions it will the determination of the ideal user behavior be intractable, even in cases where $E(t_{conv}(\mathbf{w}, f))$ can be obtained, the computation of which might in itself also cause severe computational effort.

In summary, it seems that only in a few, very simple cases it will be possible to get meaningful results from a convergence theory of interactive evolution strategies and empirical results will likely play an important part in the dynamic convergence theory of these algorithms, even if we assume the 'ideal user'.

Another theoretical question would be to analyze, if the user can, in principle obtain any given starting point obtain any other state with a finite probability and a finite number of inputs, independent of the settings of the strategy parameters. Assuming an 'ideal user', this would be a necessary and sufficient condition for global convergence. In case this condition can be simply proven for strategies that use a mutation distribution of infinite support. In that case, the probability of obtaining the optimum within μ optimal solutions in one step is always finite and the ideal user will of course detect and select the optimal solution. However, such kind of considerations tell us nothing about the practically important convergence time. However, they can well serve, to discard certain variants of interactive evolution strategies, e.g. those where the step-size might converge to a value of zero.

4 Self-adaptation and Interaction

One of the main research questions addressed in this paper is, whether self-adaptive mechanisms of the ES can be utilized also for the interactive ES. With regards to this, there are some important differences between the standard ES and the interactive ES.

First of all, for the standard ES in continuous spaces, the precision of an optimum approximation can, in principle, get arbitrarily close. In applications of the interactive evolution strategy, the subjective nature of the objective function usually forbids an arbitrary close approximation of some solution. The reason for this is that in many cases the user will not be able to measure arbitrarily small differences in quality. For example, when comparing two colors, a human user will perceive two colors as equal if their distance is below a *just noticeable difference.* The concept of JNDs is quite frequently discussed in the field of psycho-physics, a subbranch of cognitive psychology [2]. It is notable, that the JND depends on the intensity and complexity of the stimulus presented to the user. Moreover, it has been found that the lower the difference between two stimuli and the more complex the stimuli are, the more time it takes for the user to decide upon the similarity of two patterns. We will come back to this results, when we discuss the empirical results of our experiments.

Another difference between the standard ES and the ES with subjective selection criterion is that the user's attention level will decrease after a while. For a theory of attention we refer to the work of [15]. Hence, the number of experiments is usually very limited and very fast step-size adaptation mechanisms have to be found, and only a few parameters of the mutation distribution can be adapted.

Moreover, the amount of interaction should be minimized, e.g. by choosing a simple selection scheme. This might prevent the use of step-size adaptation strategies that demand for numerical values of the fitness function value. A performance measure would be based on the number of selections made by the user, rather than on the number of function evaluations, and even the time spend on the selection need to be considered.

5 A color redesign test-case

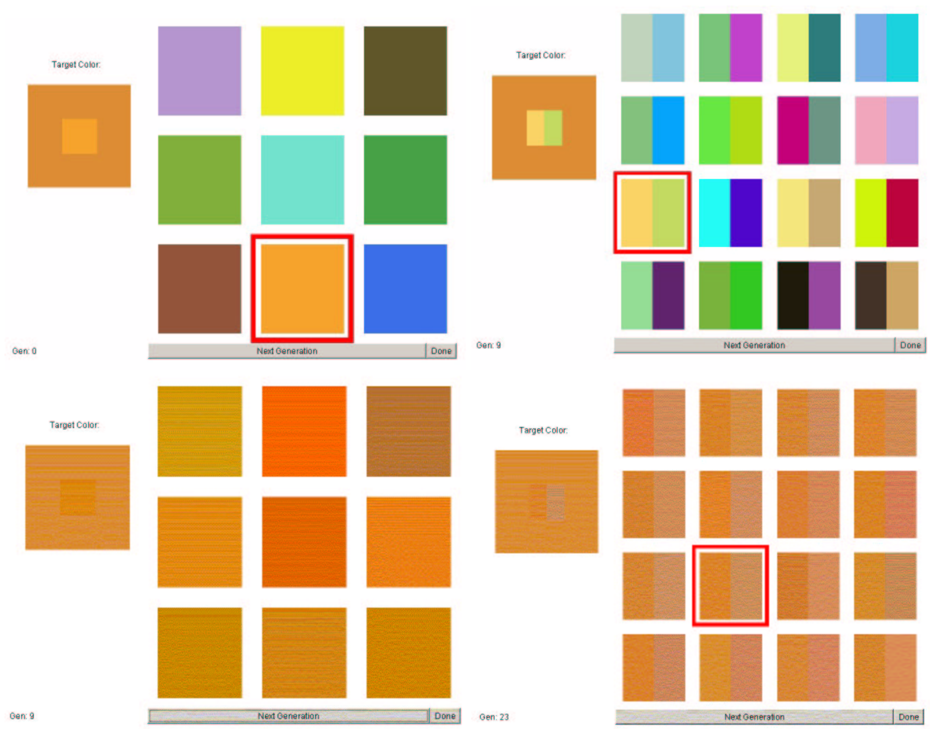

Fig. 1. Subjective selection dialogue with user: The upper figures show the initial color patterns (single color (l) and two-color test case (r)) and the lower figures show color patterns at a later stage of the evolution. The bigger box on the left hand side displays the respective target color, and in its center a small box is placed displaying the selected color. Once the user presses the NEXT bottom a selection gets confirmed and a new population will be generated and displayed. If the user is finally satisfied with the result he/she presses the Done button, in order to stop the process.

First results we obtained from our internet application are displayed in figures 3 to 3. Next, we will discuss these results one by one. Figure 3 shows the average function value for the different strategies. [....]

It was also studied if the step-size adaptation can achieve a refinement of the step-size as the optimization approaches the target color. The results of this study are plotted in 4. As expected the step size reduces in the course of the evolution, indicating that the step-size adaptation mechanism can work sufficiently fast to support the optimization process

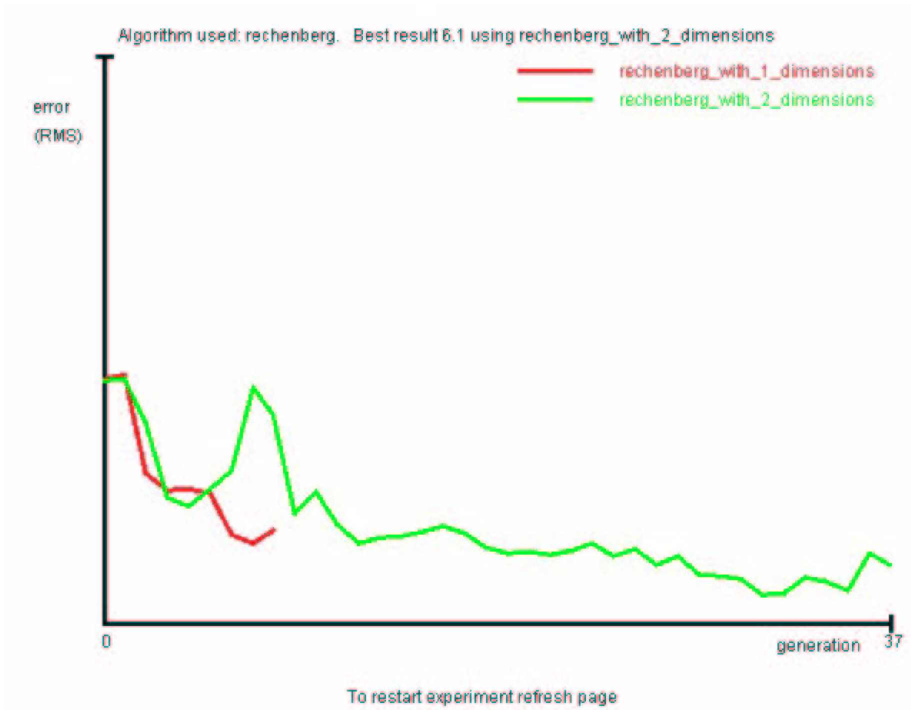

Fig. 2. After completing the experiment the tester got presented some results, showing how the distance to the target color (in RGB) changed over time for the two test-cases. An example is displayed in the figure above.

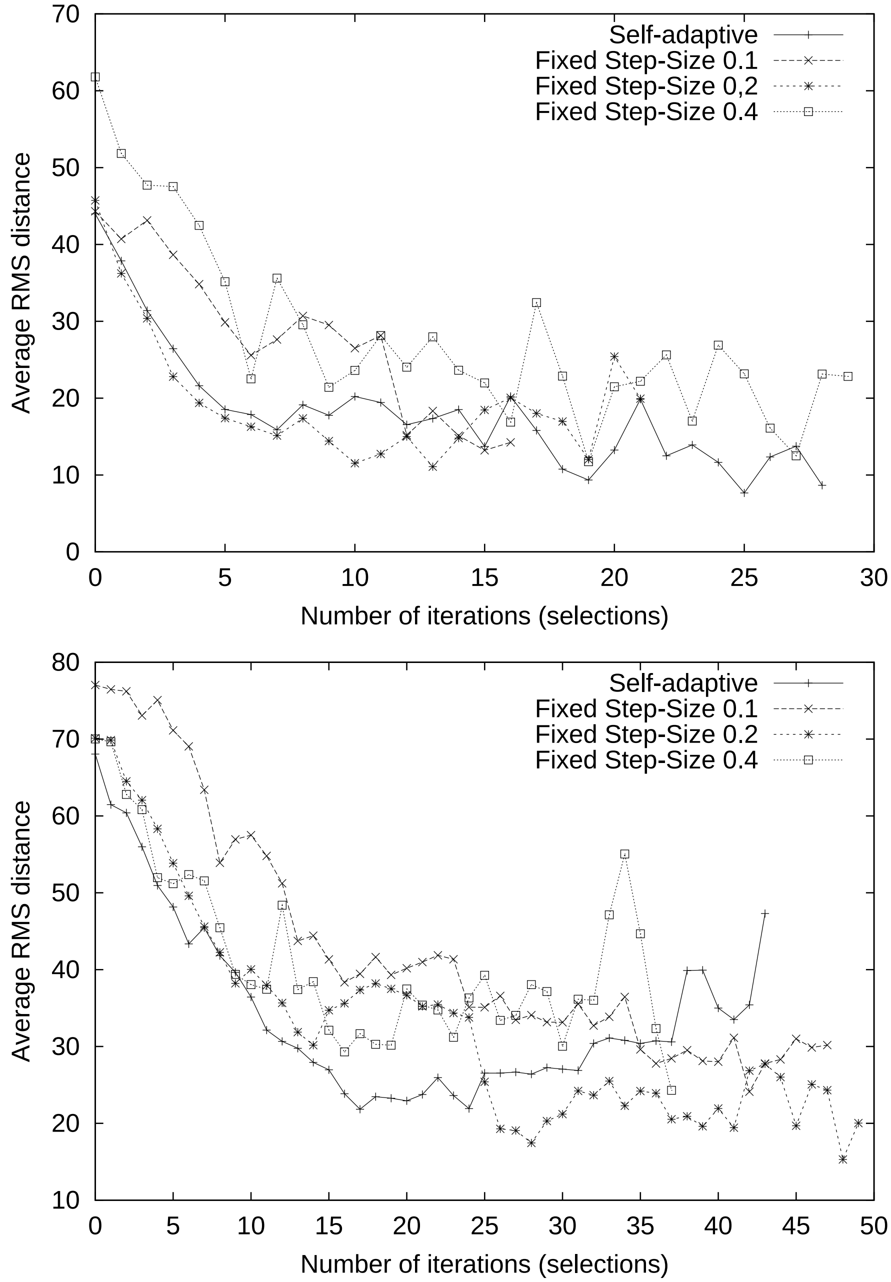

Fig. 3. The convergence behavior of different ES obtained in the online experiment. The upper figure displays results for a single RGB color, and the lower figure results for two different RGB colors. The number of iterations depends on the number of iterations the at least two of the users spend until terminating the experiment.

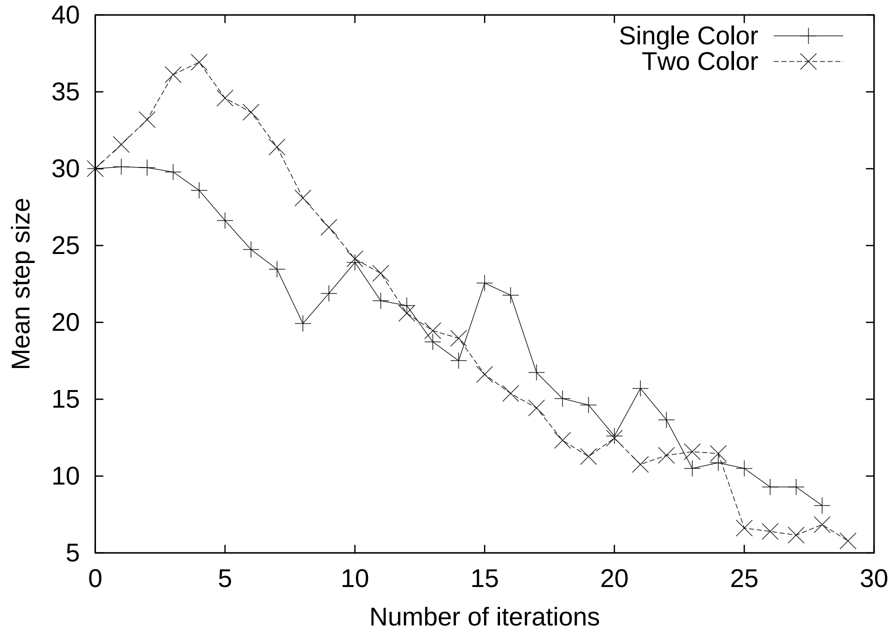

Fig. 4. Step-size adaptation in interactive evolution strategy for single color and two-color example.

6 Conclusion

References

1. P.J. Angeline (1996): Evolving fractal movies", 1st annual conference on generic programming (Stanford, CA, USA), pp. 503-511
2. Anderson J.R. (2004): Cognitive Psychology and its implications, Worth Publishers, UK
3. Arnold, D.V. , Beyer, H.-G. (2003). On the Benefits of Populations for Noisy Optimization. Evolutionary Computation 11(2):111-128, 2003.
4. Bäck, T. (1998a). An overview of parameter control methods for self-adaptation in evolutionary algorithms. Fundamenta Informaticae, 35:51-66, 1998
5. Bäck, T. (1996). Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.
6. Banzhaf, W.: Interactive evolution. In: Handbook of Evolutionary Computation (T. Bäck,, D. Fogel, and Z. Michalewicz, ch C2.10, pp. 1-5, Oxford University Press, 1997
7. Dix, A., Finlay, J., Abouwd, G.D., Beale, R. (2003). Human Computer Interaction (3rd ed.). Pearson Education, 2003.
8. Beyer, H.-G. (2001). The Theory of Evolution Strategies. Springer, Berlin, 2001
9. B. Filipic, D. Juricic (2003): An interactive genetic algorithm for controller parameter optimization, Intl. Conf. on Artificial Neural Nets and Genetic Algorithms, Innsbruck, Austria, pp 458–462

10. Hansen, N. and Ostermeier, A. (2001). Completley Derandomized Selfadaptation in Evolution Strategies, Evolutionary Computation, 9(2):159-195, 2001.
11. M. Herdy (1996): Evolution strategies with subjective selection, PPSN IV, Berlin, Germany, 1996
12. M. Herdy (1997) Evolutionary optimization based on subjective selection - evolving blends of coffee, In: 5th european congress on Intelligent Techniques and Soft Computing EUFIT'97, pp.640-644
13. Horowitz (1994) Generating rhythms with genetic algorithms, in Int. Computer Music Conference, (ICMC'94) (Aarhus, Denmark), pp. 142– 143, 1994
14. I.C. Parmee, C.R. Bonham: Cluster oriented genetic algorithms to support designer/evolutionary computation, Proc. of CEC'99, Washington D.C., USA, 546-55
15. Reason J. (1990): Human Error, Cambridge University Press, Cambridge UK, 1990
16. Rechenberg, I. (1994). Evolutionsstrategie '94. Frommann-holzboog, Stuttgart, 1994.
17. Rudolph, G.: On Interactive Evolutionary Algorithms and Stochastic Mealy Automata. PPSN 1996: 218-226.
18. Schwefel, H.-P. (1995), Evolution and Optimum Seeking, Wiley, NY
19. Takagi H.(2001) "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation", Proceedings of the IEEE, vol.89, no.9, pp.1275-1296, 2001.