On Interactive Evolutionary Algorithms and Stochastic Mealy Automata

Günter Rudolph

Informatik Centrum Dortmund, Joseph–von–Fraunhofer–Str. 20, 44227 Dortmund

Abstract. Interactive evolutionary algorithms (IEAs) are special cases of interactive optimization methods. Potential applications range from multicriteria optimization to the support of rapid prototyping in the field of design. In order to provide a theoretical framework to analyze such evolutionary methods, the IEAs are formalized as stochastic Mealy automata. The potential impacts of such a formalization are discussed.

1 Introduction

The idea of interactive optimization in the field of multicriteria optimization can be traced back at least to the early 1970s [1] and it was developed constantly ever since. The type of interaction between user and optimization algorithm may be very different and its actual realization touches many questions ranging from computer graphics [2] to psychological matters [3].

The typical situation leading to multicriteria optimization problems arises when the goals of the decision maker are conflicting. The plain approach to aggregate a vector–valued objective function $F : X^n \to \mathbb{R}^m$ $(m > 1)$ into a scalar–valued objective function $f : X^n \to \mathbb{R}$ via $f(x) = w' F(x)$ with some weight vector $w \in \mathbb{R}^m$ introduces a not negligible degree of uncertainty for the decision maker whether the chosen weights do reflect the importance of each original goal appropriately. In fact, as soon as a specific weight vector has been chosen, the original decision space is considerably and prematurely cut down *before* enough information could be gathered that might justify such a reduction. As a consequence, a huge number of potential good decisions are precluded *a priori*.

Interactive optimization may offer a remedy: The weights and other parameters are not fixed in advance and the decision maker interacts with the optimization algorithm in order to guide the optimization process in such a way as to keep the 'man–in–the–loop', using the decision maker to make judgments regarding the weights, certain complex constraints and tradeoffs. Evolutionary algorithms (EAs) enriched with visualizing and interactive components can be designed and used for this purpose [4].

Another field of application of interactive evolutionary algorithms (IEAs) is opened when the objectives are not explicitly identifiable and/or not quantifiable. For example, this situation arises when the evaluation of an admissible solution is made on basis of human taste [5]. It is clear that the presence of a graphical interface is inevitably here. This applies as well in situations in which formalized strategies with additional human judgments build a hybrid and synergistic optimization method. Applications of this type are of course not restricted to evolutionary methods [6, 7].

Further situations that make IEAs applicable are imaginable. The general theme that will be explored here, however, is less related to questions regarding additional fields of application — rather, the central questions will be of theoretical nature: Can IEAs be modeled in a probabilistic framework? If so, which theoretical properties could be investigated? And even more fundamental: Is there any utility of such models at all?

Note that EAs without interactions can be modeled and analyzed in a Markov chain framework [8, 9, 10, 11, 12]. Typical theoretical investigations are devoted to the finite time and the limit behavior of these evolutionary methods. The utility of this theory is obvious.

The situation changes in case of IEAs. As will be demonstrated in section 2, the Markov chain framework is too restricted to model IEAs. Thus, a more general theoretical framework is required to represent the interactions between machine and environment appropriately. Such a framework is offered by a specific class of stochastic automata whose theory is well–developed in several directions. But the question of whether or not IEAs can benefit from the existing theory will deserve careful scrutiny and it is addressed in section 3. Finally, some conclusions are drawn in section 4.

2 Stochastic Automata and Interactive Evolution

Stochastic automata were independently defined by several authors in the early 1960s. Since then the theory developed steadily in many directions. Here, only the very basic terminology and theory will be necessary [13, 14].

The tuple $< S, X, Y, P\{\, s', y \,|\, s, x \} >$ represents a *stochastic automaton* where S denotes the set of *states*, X the set of *input symbols*, Y the set of *output symbols* and $P : S \times X \times S \times Y \rightarrow [\,0, 1]$ the *transition probabilities* that the stochastic automaton transitions from state $s \in S$ to state $s' \in S$ and outputs symbol $y \in Y$, provided that $x \in X$ was the input symbol. To keep the presentation simple it will be assumed that the sets S, X and Y are finite. Then it is clear that

$$\sum_{(s',y) \in S \times Y} P\{\, s', y \,|\, s, x \} = 1 \text{ for all } (s, x) \in S \times X$$

and that the conditional probabilities $P\{\, s', y \,|\, s, x \}$ can be gathered in a collection of *stochastic matrices* $A(y \,|\, x)$ with $a_{ss'}(y \,|\, x) = P\{\, s', y \,|\, s, x \}$. Some special cases of stochastic automata are presented below:

1. Markov chains:
   A stochastic automaton $< S, \emptyset, \emptyset, P\{\, s' \,|\, s \} >$ with empty input and output sets is a *Markov chain.*
2. Stochastic Mealy automaton:
   A stochastic automaton whose transition probabilities for a consecutive state

s' and a output symbol y are mutually independent for each input symbol x and each state s is a *stochastic Mealy automaton*. Thus, the relation $P\{\, s', y \,|\, s, x \} = P\{\, s' \,|\, s, x\} \cdot P\{\, y \,|\, s, x\}$ with

$$P\{\, s' \,|\, s, x\} = \sum_{y \in Y} P\{\, s', y \,|\, s, x\} \quad \text{and} \quad P\{\, y \,|\, s, x\} = \sum_{s' \in S} P\{\, s', y \,|\, s, x\}$$

is valid for all $(s', y, s, x) \in S \times Y \times S \times X$.

3. Stochastic automaton with deterministic output:
   A stochastic automaton with $P\{\, y \,|\, s, x\} \in \{0, 1\}$ is a *stochastic automaton with deterministic output*. Thus, there exists a (deterministic) function $g : S \times X \to Y$ with

$$P\{\, y \,|\, s, x\} = \begin{cases} 1 & , \text{if } y = g(s,x) \\ 0 & , \text{otherwise} \end{cases} .$$

   Evidently, every stochastic automaton with deterministic output is also a stochastic Mealy automaton.

Next, an abstract model of interactive evolutionary algorithms will be given in terms of the stochastic automata framework. It is obvious that the interactions between evolutionary algorithm and user can be represented by the input and output symbols. Since the input and output sets are empty for Markov chains it becomes clear that Markov chains cannot be used to model IEAs.

2.1 The abstract model

Evidently, it is sufficient to identify the state space, the input set, the output set and the transition matrices of an interactive evolutionary algorithm to specify an abstract stochastic automata model.

- The state space S:
  Suppose there is a population $(i_1, \ldots, i_N)$ of N individuals $i_n \in \mathcal{S}$ where $\mathcal{S}$ is some finite space. Then the set $\mathcal{S}^N$ of all possible populations is of cardinality $|\mathcal{S}|^N < \infty$. Therefore, the state space of the stochastic automaton is just $S = \mathcal{S}^N$ and individual i_n may be referred to via $s(n) = i_n$ for $n = 1, \ldots, N$.
- The output set Y:
  At some generation $t \geq 0$ the current population $s \in S$ is presented to the user in some manner. For example, each individual might be visualized by means of a graphical user interface. In any case the output is a deterministic function of the current state and may be omitted in the model. Thus, $Y = \emptyset$ and the transition matrix will not be parametrized by output symbols.
- The input set X:
  After the current population has been presented to the user he selects some individuals that will serve as parents for the next generation. Since the number of all possible selections is finite, any of these actions can be symbolized by an element of X. For example, the user might select as follows: Choose

individual i_1 three times, individual i_2 twice, individual i_3 not at all, individual i_4 six times, and so forth until N selections have been made. Thus, there are $N\,(2\,N-1)!/(N!)^2$ different selection operations.

– The transition matrices $A(x)$:
  Let s be the current population. For every selection operation $x \in X$ of the user the parent population s'' is a deterministic function of the current population s and input x. Therefore, the entries of the matrices $U(x)$ describing the state transitions caused by the user are of the type $u_{ss''}(x) \in \{0,1\}$. Next, the parent population s'' might be modified by crossover and mutation resulting in a population s'. These probabilistic operations can be modeled by a transition matrix M as known from Markov models of EAs. Consequently, the transition matrices $A(x)$ of the stochastic automaton are determined by $A(x) = U(x) \cdot M$.

Summing up: Interactive evolutionary algorithms can be described as stochastic automata of the type IEA $= < \mathcal{S}^N, X, \emptyset, U(x) \cdot M >$. Since the output set Y is empty IEAs are stochastic Mealy automata.

The probabilistic behavior of the IEA for a given input sequence can be calculated as follows: Let $W = X \times X \times X \times \ldots$ be the set of words over alphabet X. The length of $w \in W$, denoted as $|w|$, is the number of symbols from X in w. The length of the *empty word* ϵ is $|\epsilon| = 0$. Let $A(\epsilon) = I$ be the unit matrix and $vw \in W$ be the concatenation of the words $v \in W$ and $w \in W$. Then the transition probabilities under input vw can be calculated via $A(vw) = A(v) \cdot A(w)$. This equation may be seen as an equivalent to the Chapman–Kolmogorov equation of discrete Markov chains. Similarly, the probability distribution of the state of the automata is given by $p(xvw) = p(x) \cdot A(v) \cdot A(w)$ where $p(\epsilon)$ is the initial distribution.

2.2 An explicit model

Suppose that the space of individuals is $\mathcal{S} = \mathbb{B}^\ell$ with $\mathbb{B} = \{\mathtt{0}, \mathtt{1}\}$ and that the individuals are modified by parametrized uniform crossover and mutation. The input of the crossover operator is a pair $(i,j) \in \mathbb{B}^\ell \times \mathbb{B}^\ell$ of individuals while the output is an preliminary individual $h' \in \mathbb{B}^\ell$ which is composed from the input pair by choosing each entry h'_k with probability $\chi \in (0,1)$ from individual i and with probability $1-\chi$ from individual j. Then individual h' is mutated to offspring h by inverting each entry independently with probability $\mu \in (0,1)$.

The first step to determine the transition matrix M is the derivation of the probability to generate the zero string from parents (i,j) by crossover and mutation. To this end note that crossover as well as mutation operate on each vector entry independently. Therefore, the probabilities to obtain an entry $\mathtt{0}$ at position k from (i_k, j_k) are given by

$$
\begin{aligned}
P\{\, h_k = \mathtt{0} \,|\, (i_k, j_k) = (\mathtt{0},\mathtt{0}) \} &= 1-\mu\,, \\
P\{\, h_k = \mathtt{0} \,|\, (i_k, j_k) = (\mathtt{0},\mathtt{1}) \} &= \chi\,(1-\mu) + (1-\chi)\,\mu\,, \\
P\{\, h_k = \mathtt{0} \,|\, (i_k, j_k) = (\mathtt{1},\mathtt{0}) \} &= \chi\,\mu + (1-\chi)\,(1-\mu) \text{ and}
\end{aligned}
$$

$$P\{\, h_k = \mathtt{0} \,|\, (i_k, j_k) = (\mathtt{1}, \mathtt{1}) \} = \mu$$

for each $k = 1, \ldots, \ell$. Let $\oplus$ be the *exclusive–or* and $\otimes$ be the *logical and–operation* on bit strings while $|i|$ denotes the number of $\mathtt{1}$s in some bit string $i \in \mathbb{B}^\ell$. Then

$$\ell_{00} = \sum_{k=1}^{\ell} \bar{i}_k \, \bar{j}_k = |\bar{i} \otimes \bar{j}|,$$
$$\ell_{01} = \sum_{k=1}^{\ell} \bar{i}_k \, j_k = |\bar{i} \otimes j|,$$
$$\ell_{10} = \sum_{k=1}^{\ell} i_k \, \bar{j}_k = |i \otimes \bar{j}| \quad \text{and}$$
$$\ell_{11} = \sum_{k=1}^{\ell} i_k \, j_k = |i \otimes j|$$

denote the frequency of the positions (i_k, j_k) at which the parents have the values $(\mathtt{0}, \mathtt{0})$, $(\mathtt{0}, \mathtt{1})$, $(\mathtt{1}, \mathtt{0})$ and $(\mathtt{1}, \mathtt{1})$, respectively. The probability $m_{ij}(0)$ to generate the zero string from the input pair (i, j) is thusly

$$m_{ij}(0) = (1-\mu)^{\ell_{00}} \, [\chi\,(1-\mu) + (1-\chi)\,\mu]^{\ell_{01}} \, [\chi\,\mu + (1-\chi)\,(1-\mu)]^{\ell_{10}} \, \mu^{\ell_{11}}$$

which reduces to

$$m_{ij}(0) = 2^{-|i \oplus j|} \, \mu^{|i \otimes j|} \, (1-\mu)^{\ell - |i \otimes j| - |i \oplus j|}$$

for standard uniform crossover with $\chi = 1/2$. The probability to generate an arbitrary offspring $h \in \mathbb{B}^\ell$ from the pair of parents (i, j) can be obtained by using the relation [15, p. 33]

$$m_{ij}(h) = m_{i \oplus h, j \oplus h}(0).$$

Suppose that the pair (i, j) is drawn at random from population s by choosing two indices from $\{1, \ldots, N\}$ independently with uniform probability. Let

$$r_i(s) = \frac{1}{N} \sum_{n=1}^{N} 1_{\{i\}}(s(n))$$

denote the relative frequency of individual or pattern $i \in \mathbb{B}^\ell$ in population or state $s = (s(1), s(2), \ldots, s(N)) \in (\mathbb{B}^\ell)^N$. Then the probability that population s is modified by crossover and mutation to population s' is given by

$$m_{ss'} = \prod_{n=1}^{N} \sum_{i \in \mathbb{B}^\ell} \sum_{j \in \mathbb{B}^\ell} r_i(s) \, r_j(s) \, m_{ij}(s'(n)) \,.$$

It remains to model the interaction with the user. Let the input symbol be of the type $x = (x_1, x_2, \ldots, x_N)$ where x_n denotes the number of times the user has selected individual $s(n)$ from population s. It is clear that the sum over the x_n must be N. Then the entries of the matrices $U(x)$ are as follows:

$$u_{ss'}(x) = 1 \quad \text{if} \quad s' = \underbrace{s(1) \cdots s(1)}_{x_1 \text{ times}} \underbrace{s(2) \cdots s(2)}_{x_2 \text{ times}} \cdots\cdots \underbrace{s(N) \cdots s(N)}_{x_N \text{ times}}$$

and zero otherwise. For example, with $\mathcal{S} = \mathbb{B}$ and a population size $N = 2$ the input set is $X = \{(2,0),(1,1),(0,2)\}$ leading to the three matrices

$U(2,0)$	00	01	10	11
00	1	0	0	0
01	1	0	0	0
10	0	0	0	1
11	0	0	0	1

$U(1,1)$	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

$U(0,2)$	00	01	10	11
00	0	0	0	1
01	0	0	0	1
10	1	0	0	0
11	1	0	0	0

where the input symbol is written in the upper left corner. When continuing this example with $\chi = 1/2$ matrix M becomes

M	00	01	10	11
00	$(1-\mu)^2$	$\mu(1-\mu)$	$\mu(1-\mu)$	μ^2
01	$1/4$	$1/4$	$1/4$	$1/4$
10	$1/4$	$1/4$	$1/4$	$1/4$
11	μ^2	$\mu(1-\mu)$	$\mu(1-\mu)$	$(1-\mu)^2$

yielding the transition matrices

$A(2,0)$	00	01	10	11
00	$(1-\mu)^2$	$\mu(1-\mu)$	$\mu(1-\mu)$	μ^2
01	$(1-\mu)^2$	$\mu(1-\mu)$	$\mu(1-\mu)$	μ^2
10	μ^2	$\mu(1-\mu)$	$\mu(1-\mu)$	$(1-\mu)^2$
11	μ^2	$\mu(1-\mu)$	$\mu(1-\mu)$	$(1-\mu)^2$

$A(0,2)$	00	01	10	11
00	μ^2	$\mu(1-\mu)$	$\mu(1-\mu)$	$(1-\mu)^2$
01	μ^2	$\mu(1-\mu)$	$\mu(1-\mu)$	$(1-\mu)^2$
10	$(1-\mu)^2$	$\mu(1-\mu)$	$\mu(1-\mu)$	μ^2
11	$(1-\mu)^2$	$\mu(1-\mu)$	$\mu(1-\mu)$	μ^2

and $A(1,1) = U(1,1) \cdot M = M$. Evidently, the interactive evolutionary algorithm of this example is completely formalized by the description of the stochastic automaton given above.

3 What might the formalization be good for?

Stochastic automata are a generalization of deterministic automata. Therefore it is not surprising that many theoretical questions regarding deterministic automata were also treated in the theory of stochastic automata. Typical topics are the equivalence of automata, the minimization of states, decompositions of automata and stochastic languages. Moreover, stochastic automata may be seen as a generalization of Markov chains. This leads to questions regarding the limit behavior of stochastic automata.

In the remainder two parts of stochastic automata theory associated with the above mentioned questions will be investigated in order to rate their applicability and usefulness with regard to interactive evolutionary algorithms. It should be kept in mind that this list is not complete and that some suggestions are of speculative nature yet.

3.1 Decompositions of stochastic automata

There exist techniques to decompose a stochastic automaton into a sequential combination of automata [13, 14]. Owing to this theory every IEA can be decomposed into a *controlled random source* and a deterministic automaton. In particular, a controlled random source is a single state stochastic automaton $< s, X, Y, P\{\, y \,|\, x \} >$ that returns an output symbol $y \in Y$ provided that $x \in X$ was fed in. It may be interpreted as follows: A realization of a random variable ξ and the input symbol $x \in X$ is passed to function R that returns $y = R(x, \xi)$. This symbol y is the input symbol of the deterministic automaton that calculates the new state.

At a first glance this result seems to be remarkable, but a closer look reveals that IEAs can be implemented in this manner easily. Consider the explicit model given in the previous section: In each generation two binomial random variables are drawn for mutation and additional two for crossover. Thus, there are 16 different potential realizations. Since the user may choose among three different selection operations, the size of the support of random variable ξ need not be larger than $3 \times 16 = 48$. These 48 potential realizations of ξ are the input to the deterministic automaton. It is clear that the new state/population can be calculated deterministically now. But when using the decomposition method described in [14, pp. 30–35] and choosing $(\chi, \mu) = (1/2, 1/8)$, it turns out that the size of the support of ξ need not exceed 11. Consequently, this IEA can be equivalently realized with less randomness in the operations than it was presented originally and one may speculate that this theory may give information about the extent of potential de–randomizations of the strategies.

3.2 Convergence of stochastic automata

Similar to the Markov theory of EAs one might inquire for the limit behavior of IEAs represented by stochastic Mealy automata. It is clear that the limit behavior must depend on the input sequences — besides the structure of the transition matrices. For example, let $f : S \to \mathbb{R}$ be some function with $\max\{|f(s)| \,:\, s \in S\} < \infty$ and let $(Z_t : t \in \mathbb{N})$ be the random sequence of states attained by the stochastic automaton. Then there exist conditions [16] that ensure convergence in expectation, i.e.,

$$E[\, f(Z_t) \,|\, Z_1 = s, w_1\, w_2\, \cdots\, w_t\,] = \sum_{s' \in S} f(s')\, a_{ss'}(w_1\, w_2\, \cdots\, w_t) \;\rightarrow\; E[\, f(Z_\infty)\,]$$

as $t \to \infty$ for a sequence of input words $(w_t : t \in \mathbb{N})$ and where $A(w) = (a_{ss'}(w))$. But it is likely that convergence issues will not play a major role with regard to IEAs since it is reasonable to assume that the user's interventions in the behavior of the optimization algorithm are intended to increase the flexibility of the entire man–machine system and that these interventions are not predictable with regard to the current state or time step. If they were predictable then the inputs $x \in X$ would be a function of the state/time and the stochastic automaton would reduce to an ordinary Markov chain.

4 Conclusions

It was shown that interactive evolutionary algorithms can be modeled as stochastic Mealy automata without output. At the current state of investigation, however, it is inconceivable yet to which extent a potential theory of IEAs can profit from stochastic automata theory, although some clues were given. Apparently, this work does not exhibit immediate utility with regard to practical applications — but this ought not to be surprising for the first steps towards a theory of IEAs.

Acknowledgment

Financial support by project EVOLALG, grant 01 IB 403 A, from the German Federal Ministry of Education, Science, Research and Technology (BMBF) is gratefully acknowledged.

References

1. A. Geoffrion, J. Dyer, and A. Feinberg. An interactive approach for multi–criterion optimization, with an application to the operation of an academic department. *Management Science*, 19:357–368, 1972.
2. P. Korhonen. Solving discrete multiple criteria problems by using visual interaction. In G. Fandel, M. Grauer, A. Kurzhanski, and A.P. Wierzbicki, editors, *Large–Scale Modelling and Interactive Decision Analysis*, pages 176–185. Springer, Berlin, 1986.
3. A. Lewandowski. User–machine interface and intelligent decision support. In G. Fandel, M. Grauer, A. Kurzhanski, and A.P. Wierzbicki, editors, *Large–Scale Modelling and Interactive Decision Analysis*, pages 161–175. Springer, Berlin, 1986.
4. C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann, San Mateo (CA), 1993.

5. J. Graf. Interactive evolutionary algorithms in design. In D.W. Pearson, N.C. Steele, and R.F. Albrecht, editors, *Proceedings of the 2nd International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 227–230. Springer, Vienna, 1995.
6. H. Adeli and K.V. Balasubramanyam. A synergetic man–machine approach to shape optimization of structures. *Computers and Structures*, 30(3):553–561, 1988.
7. W.B. Powell and Y. Sheffi. Design and implemetation of an interactive optimization system for network design in the motor carrier industry. *Operations Research*, 37(1):12–29, 1989.
8. A.E. Nix and M.D. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1992.
9. T.E. Davis and J. Principe. A markov chain framework for the simple genetic algorithm. *Evolutionary Computation*, 1(3):269–288, 1993.
10. D.B. Fogel. Asymptotic convergence properties of genetic algorithms and evolutionary programming: Analysis and experiments. *Cybernetics and Systems*, 25(3):389–407, 1994.
11. G. Rudolph. Convergence properties of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101, 1994.
12. J. Suzuki. A markov chain analysis on simple genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(4):655–659, 1995.
13. A. Paz. *Introduction to Probabilistic Automata*. Academic Press, New York and London, 1971.
14. R.G. Bukharaev. *Theorie der stochastischen Automaten*. Teubner, Stuttgart, 1995.
15. M.D. Vose and G.E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5(1):31–44, 1991.
16. E.-E. Doberkat. Convergence theorems for stochastic automata and learning systems. *Mathematical Systems Theory*, 12:347–359, 1979.

This article was processed using the LaTeX macro package with LLNCS style