Rocha, L.M., A. Maguitman, C. Huang, J. Kaur, and S. Narayanan. [2006]."An Evolutionary Model of Genotype Editing". In: *Artificial Life 10: Tenth International Conference on the Simulation and Synthesis of Living Systems*, L.M. Rocha, M. Bedau, D. Floreano, R. Goldstone, A. Vespignani, and L. Yaeger (Eds.). MIT Press, pp. 105-111.

# An Evolutionary Model of Genotype Editing

Luis M. Rocha[1], Ana Maguitman[1], Chien-feng Huang[2], Jasleen Kaur[1], and Sheetal Narayanan[1]
[1]Indiana University, Bloomington, IN 47406
[2]Los Alamos National Laboratory, Los Alamos, NM 87545
rocha@indiana.edu

## Abstract

Evolutionary algorithms rarely deal with ontogenetic, non-inherited alteration of genetic information because they are based on a simple, direct genotype-phenotype distinction. In contrast, in Nature several processes have been discovered which alter genetic information encoded in DNA before it is translated into amino-acid chains. Ontogenetically altered genetic information is not inherited but extensively used in regulation and development of phenotypes. An example of post-transcriptional alteration of gene-encoding sequences is the process of RNA Editing. Here we introduce a novel Agent-based model of genotype editing and a computational study of its evolutionary performance in static and dynamic environments. This model builds on our previous Genetic Algorithm with Edition, but presents a fundamentally novel architecture in which coding and non-coding genetic components are allowed to co-evolve. Our goal is twofold: (1) to study the role of RNA Editing regulation in the evolutionary process, and (2) to investigate the conditions under which genotype edition improves the performance of evolutionary algorithms. We show that edition allows evolving agents to perform better in several classes of fitness functions, both in static and dynamic environments.

## Introduction

Although RNA editing (Bass, 2001) seems to play an essential role in regulation and development of biological organisms, not much has been advanced to understand the potential evolutionary advantages, if any, that RNA editing processes may have provided. Here we continue our study of the evolutionary advantages of genotype editing by introducing an *Agent-Based Model of Genotype Editing* (ABMGE), which marks a substantial departure from our previous Genetic Algorithm with Edition (GAE) (Huang and Rocha, 2003; Rocha and Huang, 2004; Huang and Rocha, 2004). While the GAE allowed a population of genotypes to evolve under fixed editing constraints, the more realistic ABMGE here presented allows editor parameters to co-evolve with genotypes in a population of agents defined by a genome with both coding and non-coding genetic material.

In previous work we had already shown that the GAE, with appropriate editor parameters, can outperform the traditional Genetic Algorithm (GA) (Holland, 1975) in static (Huang and Rocha, 2003; Huang and Rocha, 2004) and dynamic environments (Rocha and Huang, 2004). But for the GAE to outperform the GA, we needed to define good editor parameters by hand. Thus, while we showed that genotype edition can *in principle* be beneficial in an evolutionary process, we did not show how such a process could on its own discover the benefits of genotype edition. With the ABMGE presented in this paper, we show that beneficial editor parameters can be co-evolved to provide evolutionary advantages in both static and dynamic environments — without much user intervention other than the usual variation probabilities. In (Huang and Rocha, 2005) we presented a preliminary study of the ability of the ABMGE to track changing extrema in dynamic environments using a simple mutation on editor parameters. Here we explore a new form of editor variation similar to genetic crossover, as well as more extensive results on longer runs for static and dynamic environments. The overall goal of our research is to gain a deeper understanding of the evolutionary nature of genotype editing as well as to exploit its insights to improve evolutionary algorithms and their applications to complex problems.

## RNA Editing

RNA Editing (Bass, 2001) is a process of post-transcriptional alteration of genetic information. Perhaps the most famous form of RNA editing operates by deletion and/or insertion of bases in the messenger RNA (mRNA) molecules of organisms such as African Trypanosomes (Stuart, 1993). In this case, insertion/deletion editing is performed by small guide RNA's (gRNA's) encoded mostly by what was previously thought of as non-functional or non-coding genetic material (Sturn and Simpson, 1990). [1] Guide RNA's (reviewed in (Simpson, 1999) ) are usually small sequences (compared to pre-edited mRNA's) that are complementary to the region around the site to be edited. gRNA molecules base-pair with mRNA regions to be edited and then insert, and sometimes delete, uridines into the mRNA (for examples see (Bass, 2001)). This edition alters the

[1]By non-coding genetic material we mean DNA not used to encode proteins (no open reading frame) or known RNA products.

aminoacid chain encoded in the edited mRNA, sometimes extensively.

Another common type of RNA editing is edition via base substitution. This type of RNA editing exists in mitochondrial and chloropast RNA of many higher plants (with mostly C-to-U substitution) and in the genomes of higher eukaryotes such as mammals (with A-to-I substitution) (for reviews see (Blanc and Davidson, 2003) and (Maas et al., 2003)). This form of RNA editing is known to be important in the development of more complex organisms. For instance, the development of rats without a gene (ADAR1) known to be involved in RNA editing, terminates midterm (Wang et al., 2000). RNA editing processes have also been identified in human brains (Mittaz et al., 1997). More recently, (Hoopengardner et al., 2003) found that RNA editing plays a central role in nervous system function and may play a role in nervous system disorders such as epilepsy and Parkinson Disease. Furey et al (Furey et al., 2004) have also found statistical evidence that a significant amount of nucleotide discrepancies in the human genome are due to RNA Editing (both A to I/G substitution and the lesser known T to C substitution).

The importance of RNA Editing is thus unquestionable, since it has the power to dramatically alter and regulate gene expression. We have presented more thorough introductions of RNA editing in previously published Artificial Life models of genotype edition (Rocha, 1995; Huang and Rocha, 2003; Rocha and Huang, 2004), therefore here we only highlight some of its most salient features which are important for the model we propose:

- A mRNA molecule may be more or less edited according to the concentrations of the editing operators it encounters. Thus, several different proteins encoded by the same gene may coexist in an organism or even a cell, if all (or some) of the mRNAs obtained from the same gene, but edited differently, are meaningful to the translation mechanism. This way, genotype editing can be successfully used for gene expression regulation from external or developmental cues (Rocha, 1995; Mattick, 2003).

- Genotype edition is not equivalent to mutation. In all cells, prokaryotic and eukaryotic, RNA is derived from DNA. Mutations in RNA can appear if the DNA polymerase makes mistakes during the DNA replication or if the RNA polymerase makes mistakes during the RNA. However, only mutations that occur during DNA replication can become permanent and inheritable. If mistakes occur during transcription, they get incorporated into that single transcript but not into other ones. Likewise, edited mRNA transcripts are not allowed inheritable variation. **What is inheritable, and subjected to variation, is the original non-edited gene**(Rocha, 1995; Rocha and Huang, 2004).

## Modeling Genotype Editing

Genetic Algorithms (GA) (Holland, 1975) are based on an idealized model of Natural Selection. GA operate on an evolving population of artificial organisms, or agents. Each agent is comprised of a genotype (encoding a solution to some problem, typically in binary symbol strings) and a phenotype (the solution itself). Evolution occurs by iterated stochastic variation of genotypes (or chromosomes), and selection of the best phenotypes in an environment according to how well the respective solution solves a problem (or fitness function). While idealized, GA, capture the process of adaptation of agents under genetic variation and phenotypic selection.

In traditional GA, this code between genotype and phenotype is a direct and unique mapping. In biological genetic systems, however, before a gene is translated into the space of proteins it may be altered through interactions with other types of molecules, namely RNA editors such as gRNA's. Based upon this analogy, Rocha (Rocha, 1995) proposed the expansion of the traditional GA with a process of stochastic edition of genotypes, prior to them being translated into solutions. In this model, genotype edition is performed by a small set of smaller genetic strings, the *editors*, which stochastically base-pair with genotypes. Each editor is associated with a different editing function, such as insertion, deletion or substitution of symbols into the original genotypes. In each generation, before translation into the space of solutions, each genotype has a certain probability (defined by the editor's "concentration") of encountering an editor. When there is an encounter, if the editor matches some subsequence of the genotype, the editor's function is applied and the genotype is altered.

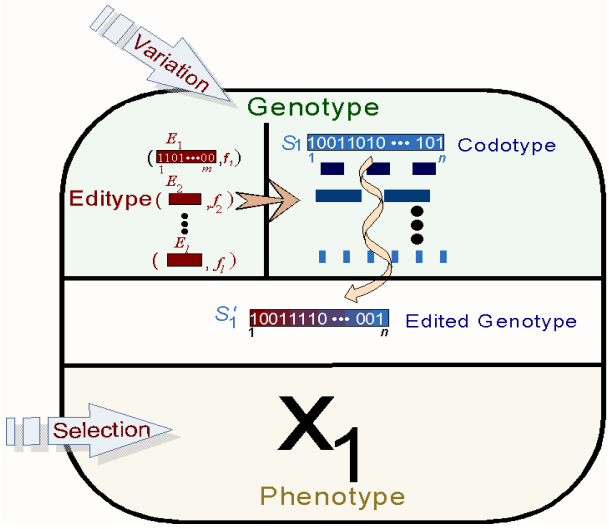


Figure 1: Individual Agent in the ABMGE.

In our previous implementations of this artificial genotype editing system, the set of editors did not evolve and

Table 1: The ABMGE Algorithm

1. Randomly generate an initial agent population, each agent consisting of a codotype (a $n$-bit string) and an editype.
2. Edit each agent's codotype using the agent's editype and evaluate each agent's fitness.
3. Repeat until $l$ offspring have been created.
   a. select a pair of parents for mating;
   b. apply codotype variation operators (mutation and crossover);
   c. apply editype variation operators (editor mutation and crossover).
4. Replace the current population with the new one.
5. Go to Step 2 until terminating condition.

was the same for every agent in the population (Huang and Rocha, 2003; Rocha and Huang, 2004; Huang and Rocha, 2004). Our novel Agent-based Model of Genotype Editing (ABMGE), that we present here, associates a different set of editors to each agent, and furthermore allows them to co-evolve with the genotypes of their respective agents. In this new model, the agents in the population are defined by an artificial genome that contains both *coding* and *non-coding* components. The coding component encodes solutions to a particular fitness function or environment, while the non-coding component defines a set of editors which act on the coding component. Let us refer to the coding portion of the artificial genome as the *codome*, and to the non-coding portion as the *editome*. In each generation, the coding component of an agent's genotype, the *codotype* may be stochastically edited by the agent's non-coding editors, the *editype*, and produce a solution/phenotype different from what is encoded. Figure 1 depicts an agent in the ABMGE; table 1 depicts its algorithm.

Notice that our agents possess functionally and operationally distinct codomes and editomes — for instance with separate variation operations (more details below). It would probably be more biologically realistic to extract both functions from a common artificial genome (e.g. as defined by (Reil, 1999)). However, here we want precisely to explore the influence of an editome in the evolutionary process, therefore we functionally separate it from the codome to better test its relative importance.

## The Editype

The editome of our agents consists of a family of $r$ editors each defined by a $m$-bit string: $(E_1, E_2, \ldots, E_r)$. The length of the editor strings is defined much smaller than that of the codotype strings: $m << n$, usually an order of magnitude. An editor $E_j$ is said to match a substring, of size $m$, of a codotype string, $S$, at position $k$ if $e_i = s_{k+i}, i = 1, 2, \ldots, m$, $1 \leq k \leq n-m$, where $e_i$ and $s_i$ denote the $i^{th}$ bit value of $E_j$ and $S$, respectively. For each editor, $E_j$, there exists an associated editing function, $F_j$, that specifies how a particular editor edits the codotypes it matches. For instance, when the editor matches a portion of a codotype string, a number of bits may be inserted into or deleted from it.

If the editing function of editor $E_j$ is to add one specific allele at $s_{k+m+1}$ when $E_j$ matches $S$ at position $k$, then all alleles of $S$ from position $k+m+1$ to $n-1$ are shifted one position to the right (the allele at position $n$ is removed). Analogously, if the editing function of editor $E_j$ is to delete an allele, an allele at $s_{k+m+1}$ is deleted when $E_j$ matches $S$ at position $k$. All the alleles after position $k+m+1$ are shifted in the inverse direction (one randomly generated allele is assigned at position $n$). Finally, let the concentration of the editor family be defined by $(v_1, v_2, \ldots, v_r)$; i.e., the concentration of editor $E_j$ is denoted as $v_j$. Then the probability that $S$ encounters $E_j$ is given by $v_j$. In the remaining of this article, when we refer to an editor we mean a tuple $(E_j, F_j, v_j)$.

It is important to note that **the "post-transcriptional" edition of codotypes is not akin to mutation, because editions are not inheritable**. Just like in biological organisms, in our model, it is the unedited genotype (codotype plus editype) that is reproduced, while agent fitness is calculated using the phenotype produced from the edited codotype. Therefore, the unedited and edited codotypes can be viewed as mimicking coding (functional) DNA and edited mRNA's in biological organisms, respectively, even though this model does not include a true DNA/RNA distinction. Furthermore, just like a mRNA molecule may be edited in different degrees according to the concentrations of editing operators it encounters, in our model the same codotype may be edited differently because editor concentration is a stochastic parameter that specifies the probability of a given editor encountering the codotype before translation.

In Rocha's formulation (Rocha, 1995), any bit-string editor function is possible, including substitution. Here we use only insertion and deletion functions. However, given that in evolutionary algorithms there is no equivalent of the transcription of RNA from DNA, what we are modeling is a **generic process of non-inheritable alteration of an agent's genetic information** (codotype) via edition, before it is translated into a solution (phenotype) — not a specific type of RNA Editing.

## Editype Variation

The variation operations of codotypes (mutation and crossover) operate just like in a regular GA. Therefore, here we reserve detailed explanations only for the variation operations of the editype. When two parents are selected for reproduction in our algorithm (step 3 in table 1), in addition to variation of codotypes as it is commonly done in GA, the editype is also subjected to variation. In the current model,

we only apply variation to the set of editor strings $E_j$, while the associated functions $F_j$ and concentrations $v_j$ remain unchanged; we will consider variation on these parameters in future work.

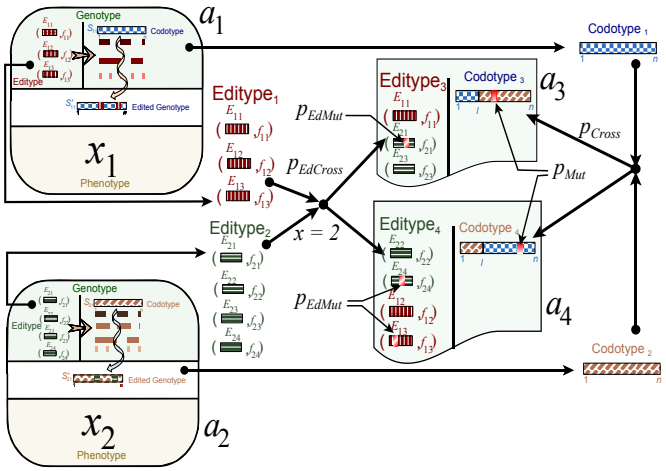


Figure 2: Agent variation in the ABMGE for two selected parents with $r_1 = 3$, $r_2 = 4$, and $x = 2$.

We implemented *editype mutation* as occurring simply on editor bit-strings in the same way they occur in codotype bit-strings: a bit-mutation probability, $P_{EdMut}$, of each individual bit of editor strings being flipped, which is independent from bit-mutation on codotype strings with probability $P_{Mut}$. We have introduced the ABMGE with editype mutation in (Huang and Rocha, 2005) where we presented some preliminary results. Here we introduce an additional editype variation operator: *editype crossover*,which implements an exchange of editors between a pair of parent agents. We start with two parent agents $a_1$ and $a_2$, with $r_1$ and $r_2$ editors in their editypes, respectively. From this pair of agents, two offspring agents, $a_3$ and $a_4$, are produced whose editypes also contain $r_1$ and $r_2$ editors, respectively. However, $x$ editors, chosen randomly from the editype of each agent, are swapped between the parent agents to produce the offspring, where $x$ is a random integer (sampled from a uniform distribution) from the interval $[1, MIN(r_1, r_2)]$. Editype crossover occurs with a probability $P_{EdCross}$, which is independent from the typical codotype crossover of GA which occurs with a probability $P_{Cross}$. Therefore, when two parents are selected in step 3 of table 1, we may have that (1) no crossover of any kind occurs, in which case the parents are reproduced as they are; (2) codotype crossover occurs with no editype crossover; (3) editype crossover occurs with no codotype crossover; and (4) both types of crossover occur. In future work we will explore making both types of crossover dependent. See figure 2 for a depiction of agent variation mechanisms in the ABMGE.

Table 2: Small royal road function $L_1$

$t_1$ = 11111**********; $c_1$ = 10
$t_2$ = *****11111**********; $c_2$ = 10
$t_3$ = **********11111**********; $c_3$ = 10
$t_4$ = **********11111**********; $c_4$ = 10
$t_5$ = **********11111**********; $c_5$ = 10
$t_6$ = **********11111**********; $c_6$ = 10
$t_7$ = **********11111*****; $c_7$ = 10
$t_8$ = **********11111; $c_8$ = 10

## Experiments with the ABMGE

### Static Environments

The performance of evolutionary algorithms is typically evaluated by monitoring improvement of the solutions discovered by the population of evolving agents in each generation. In many practical problems, a traditional performance measure is the "best-so-far" curve that plots the fitness of the best individual that has been seen thus far by generation $n$. We tested the behavior of the ABMGE with various fitness functions which we have previously used to test the simpler GAE (Huang and Rocha, 2003; Rocha and Huang, 2004; Huang and Rocha, 2004); for lack of space, here we describe the ABMGE results on only two of these functions.

The first of these functions is a miniature of the class of the "Royal Road" functions (Forrest and Mitchell 1993): the *small Royal Road* $SRR_1$, as depicted in Table 2. This function is defined by a set of schemata $T = \{t_1, \ldots, t_8\}$. The fitness of a bit string (codotype) $S$ is defined as $F(S) = \sum_{t \in T} c_t \sigma_t(S)$, where each $c_t$ is the value assigned to schema $t$ as defined in Table 2; $\sigma_t(S) = 1$ if schema $t$ exists in $S$ and 0 otherwise. The optimum fitness for $SRR_1$ is ascribed to a single string with 40 1's, and its value is $10 \times 8 = 80$. This Royal Road function is selected as a testbed because it serves as an idealized fitness environment with a single optimum, particularly amenable to evolutionary search .

We contrasted the traditional GA with two versions of the ABMGE: with and without editype crossover. Our experiments with $SRR_1$ use binary tournament selection, a population of 40 agents over 200 generations for 50 runs. Codotype variation (in both the ABMGE and the simple GA) is implemented with one-point crossover and mutation rates of $P_{Cross} = 0.7$ and $P_{Mut} = 0.005$, respectively. For the editome parameters (of the ABMGE), we employ the guidelines discovered in (Huang and Rocha, 2004) to randomly generate editor families. The size of the family of the editors, $r$, for each agent is randomly sampled from $\{1, \cdots, 5\}$; the editors are generated as randomized bit-strings $S_j$ of size $m$ also randomly sampled from $\{2, \cdots, 4\}$; the editor concentration is randomly generated from [0,1]; and the editor function inserts or deletes a number of bits randomly chosen from $\{1,3\}$ (which are fixed once the editor is generated).

We conducted our experiments on $SRR_1$ for various values of editype variation: $P_{EdMut} \in \{0.01, 0.05\}$ and $P_{EdCross} \in \{0.3, 0.5, 0.7, 0.9\}$. Figure 3 depicts the results for $P_{EdMut} =$

0.05 and $P_{EdCross} = 0.5$. One can see that the averaged best-so-far reached by both versions of the ABMGE is much closer to 80 (the maximum) than what the traditional GA obtains at the end of the experiments, and significantly better for the ABMGE with both editype mutation and crossover. This was observed for all values of $P_{EdCross}$ tested.[2]

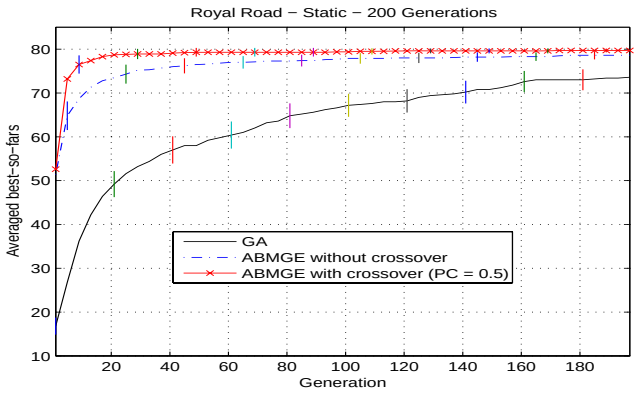


Figure 3: Performance of GA, ABMGE without crossover, and ABMGE with crossover ($P_{EdCross} = 0.5$) on $SRR_1$.

In contrast to the amenable fitness function $SRR_1$, we also tested a much more complicated testbed, the modified *Schaffer's function* $F_7$ (Huang, 2002) :

$$f(\bar{x}) = 2.5 - (x_1^2 + x_2^2)^{0.25}[sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1],$$

where $-1 \leq x_i \leq 1$ for $i = 1, 2$. A sketch of this function is displayed in Figure 4. To attain the global optimum (2.5) at the center of the search space, the population has to move across many deep wells and high barriers. Since there are many local optima in the search space, the population of an evolutionary algorithm can easily converge to any of them. The multimodality of the problem is hence expected to present substantial difficulty to evolutionary search.

Each of the two variables of $F_7$ is encoded by 50 bits, and thus each individual is a binary string of length 100. We contrasted the traditional GA with two versions of the ABMGE: with and without editype crossover. We used the same parameters as those used for $SRR_1$ except the agent population size is 100 agents, the size of the family of the editors, $r$, for each agent which was randomly sampled from $\{1, \cdots, 20\}$, and we computed statistics for 100 runs. Figure 5 depicts the results for $P_{EdMut} = 0.05$ and $P_{EdCross} = 0.5$. One can see that the averaged best-so-far reached by both versions of the ABMGE clearly outperform the traditional GA, and the ABMGE with both editype mutation and crossover also outperforms the ABMGE without editype crossover. This was observed for all values of $P_{EdCross}$ tested.

[2]The value of the averaged best-so-far performance metric is calculated by averaging the best-so-fars (the fitness of the best individual that has been seen thus far by generation $n$) obtained at each generation for all runs, where the vertical bars overlaying the metric curves represent the 95-percent confidence intervals. This applies to all the experimental results obtained in this paper.

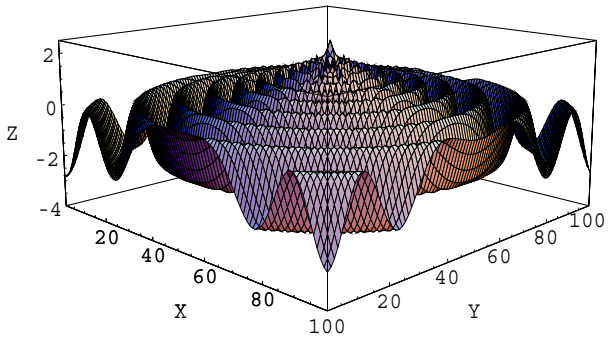


Figure 4: Modified Schaffer function $F_7$.

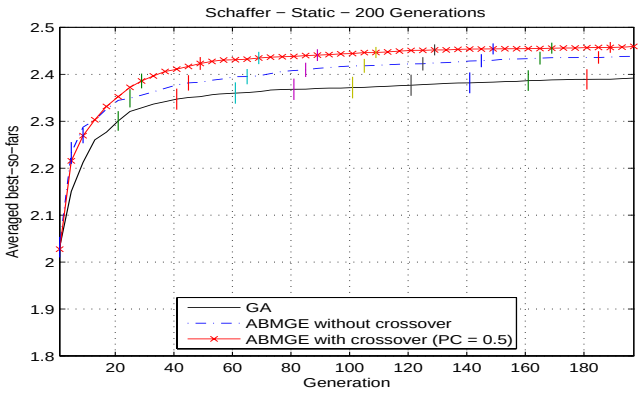


Figure 5: Performance of GA, ABMGE without crossover, and ABMGE with crossover ($P_{EdCross} = 0.5$) on $F_7$.

## Dynamic Environments

Evolutionary optimization in static environments, such as those of the previous subsection, involve the search of the extrema of functions. For dynamic environments, where the fitness function changes in time, the interest is not so much to locate the extrema but to follow it as closely as possible. This section compares the extrema-tracking performance of the traditional GA and the two versions ABMGE: with and without editype crossover. To perform this study we adapted the static fitness functions used above to a dynamic setting.[3]

Consider another Small Royal Road function, $SRR_0$, in which each schema is comprised of all 0's rather than 1's as $SRR_1$, but with all other parameters the same as $SRR_1$. Our first dynamic fitness function, the *oscillatory royal road* (*ORR*), oscillates between $SRR_1$ and $SRR_0$at every $p$ generations. Because $SRR_0$ and $SRR_1$ are maximally different, we

[3]Notice that the best-so-far measurement we used in stationary environments is problematic in dynamic environments since it has to be assured that the best solution found thus far is the best solution for the current environment. Therefore, the best-so-far solution we track is re-evaluated whenever a change in the environment occurs.

are able to study the effects of drastic environmental changes when we oscillate them. The parameters of the dynamic environment simulations using *ORR* are the same as those used for the static $SRR_1$. In addition to various editype mutation and crossover probabilities, we tested different oscillation periods: $p \in \{50, 100, 200, 250\}$. Figure 6 depicts the results for $P_{EdMut} = 0.05$ and $P_{EdCross} = 0.5$ and period $p = 100$ for 1000 generations.

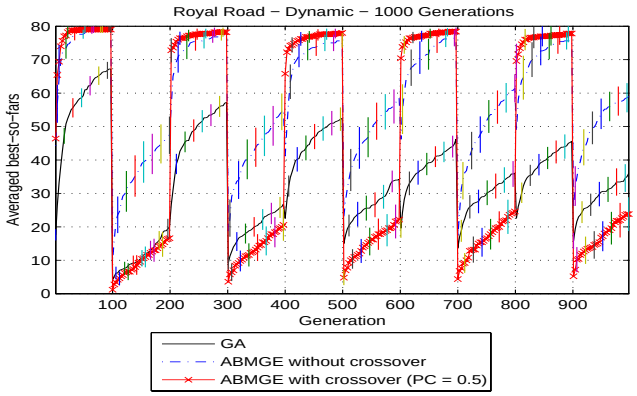


Figure 6: Performance of GA, ABMGE without crossover, and ABMGE with crossover ($P_{EdCross} = 0.5$) on *ORR*, $p = 100$.

It is clear that it is difficult for a population to re-adapt to an entirely new environment, as the best-so-far solution significantly declines when the environment changes. The traditional GA, as time progresses degrades its performance even when the first fitness environment ($SSR_1$) returns. Indeed, we ran the GA for 4000 generations for various oscillation periods (plot not shown for lack of space) and its performance on both environments eventually reaches the same level, slightly under 40 — which means the GA is converging to a population with a balanced combination of schemas of all 1's and schemas of all 0's.

As for the ABMGE, interestingly, the version with both editype crossover and mutation performs best on the first fitness environment ($SSR_1$), and every time it repeats, but it performs very poorly on the second environment — even worse than the GA as time progresses. In contrast, the ABMGE version with editype mutation alone, is almost as good as the other version on the first environment, but much better on the second environment where it progressively improves its performance — well beyond that of the regular GA. Editype crossover, as here implemented, does not allow agents to adapt to a drastic environmental change, though, unlike the GA, it is capable of recovering its performance once the environment changes back to the first state. Therefore, editype mutation alone seems to offer a much more flexible agent architecture in drastic environmental changes, far outperforming the traditional GA.

The *ORR* implements a drastically changing environment. In general, when we study dynamic environments, we consider less drastic changes. (Angeline, 1997) and (Bäck, 1998) reported a study of dynamic problems with three different modes of severity of changes. We use this idea to build a *dynamic version of the modified Schaffers function*, $DF_7$, controlled by a parameter $s$ to specify the severity of fitness change:

$$f(\overline{X}) = 2.5 - (X_1^2 + X_2^2)^{0.25}[sin^2(50(X_1^2 + X_2^2)^{0.1}) + 1],$$

where $X_i = x_i + \delta(t)$, $-1 \leq x_i \leq 1$ for $i = 1, 2$. Due to lack of space, here we only depict the results for severity $s$ of 0.1 (though we have also computed values of 0.5 and 1):

$$\begin{aligned} \delta(0) &= 0, \\ \delta(t) &= \delta(t-1) + s. \end{aligned} \quad (1)$$

Note that $t$ is used as an index for the environmental state, whenever the environment changes (e.g., every $p = 100$ generations), $t$ is increased by 1. The parameters of the dynamic environment simulations using $DF_7$ are the same as those used for the static $F_7$. We tried various editype mutation and crossover probabilities, for fitness function change periods: $p = 100$. Figure 7 depicts the results for $P_{EdMut} = 0.05$ and $P_{EdCross} = 0.5$ and period $p = 100$ for 400 generations.

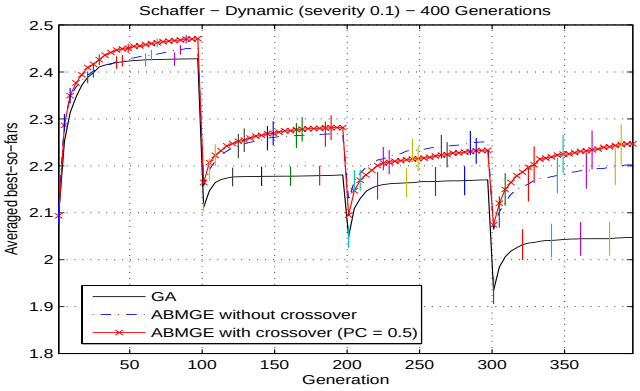


Figure 7: Performance of GA, ABMGE without crossover, and ABMGE with crossover ($P_{EdCross} = 0.5$) on $DF_7$.

Both versions of the ABMGE significantly outperform the GA in this dynamic environment. However, given the confidence intervals, there is not a significative difference between both versions of the ABMGE. Therefore we cannot establish if editype crossover is beneficial as an editype variation mechanism, or if mutation alone is preferable. We are currently running longer simulations of $DF_7$, also with different fitness update periods. We expect to have those very soon after submission to help us issue a more definitive conclusion on this matter.

## Discussion

We introduced an evolutionary model of genotype editing based on agents endowed with coding and non-coding portions of their artificial genome: a codome and an editome.

The non-coding editome is used to alter encoded solutions ontogenetically, without inheritance. We furthermore tested different forms of variation on the editome, independently from the codome, on several static and dynamic environments, and showed that it always significantly outperforms the traditional GA which does not include an editome.

Thus, we conclude that the co-evolving genotype editing mechanism offers a significant evolutionary advantage. This advantage is particularly interesting in dynamic, complex environments as agents become better equipped to deal with changing environments. Indeed, they both react quicker to the change and produce fitter agents. While our highly idealized models do not capture the reality of Biology, they do imply that the process of RNA editing in nature is, likewise, advantageous in evolution. Indeed, our results emphasize the importance of genetic regulation by non-coding genetic components. Additionally, the performance of our algorithms positions them as useful and important novel methods for evolutionary computation and machine learning. We have thus advanced the understanding of the evolutionary role of RNA Editing, and we have developed a new biologically-inspired algorithm with powerful search capabilities — meeting the two goals we set up for this project.

There are many other aspects of the ABMGE that one can study to further enhance the power of this agent-based evolutionary algorithm. Naturally, we intend to apply them to additional problems, but we also intend to study other forms of editor variation, such as allowing the variation of other editor parameters such as editor function, concentration, etc. There is also much to investigate regarding dynamic environments. Not only do we expect to study other dynamic landscapes, but we need to investigate more deeply the influence of the changing period on performance.

## Acknowledgements

The Indiana University's supercomputing facilities used in our analysis are funded in part by NSF under Grant No. 0116050 and Grant CDA-9601632. We are grateful to IU's Research and Technical Services for technical support.

## References

Angeline, P. J. (1997). Tracking extrema in dynamic environments. In *6th Int. Conf. on Evolutionary Programming*, p. 335–345.

Bass, B. (2001). *RNA Editing. Frontiers in Molecular Biology Series*. Oxford University Press.

Bäck, T. (1998). On the behavior evolutionary algorithms in dynamic environments. In *IEEE Int. Conf. on Evolutionary Computation*, p. 446–451.

Blanc, V. and Davidson, N. O. (2003). C-to-U RNA editing: Mechanisms leading to genetic diversity. *Journal of Biological Chemistry*, 278(3):1395–1398.

Forrest, S. and Mitchell, M. (1993). Relative Building Block Fitness and The Building Block Hypothesis. *Foundations of Genetic Algorithms* , 2: pp. 109-126.

Furey, T. S. et al. (2004). Analysis of human mRNAs with the reference genome sequence reveals potential errors, polymorphisms, and RNA editing. *Genome Ressearch*, 14(10B):2034–40.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Hoopengardner, B., Bhalla, T., Staber, C., and Reenan, R. (2003). Nervous system targets of RNA editing identified by comparative genomics. *Science*, 301(5497):832–836.

Huang, C.-F. (2002). *A Study of Mate Selection in Genetic Algorithms*. Doctoral dissertation. Ann Arbor, MI: University of Michigan, Electrical Engineering and Computer Science.

Huang, C.-F. and Rocha, L. M. (2003). Exploration of RNA editing and design of robust genetic algorithms. In *IEEE Congress on Evolutionary Computation*, p. 2799–2806. IEEE Presss.

Huang, C.-F. and Rocha, L. M. (2004). A systematic study of genetic algorithms with genotype editing. In *GECCO2004: Genetic and Evolutionary Computation Conference*, v. 1, p. 1233–1245.

Huang, C.-F. and Rocha, L. M. (2005). Tracking extrema in dynamic environments using a coevolutionary agent-based model of genotype edition. In *GECCO '05: Genetic and evolutionary computation conference*, pages 545–552, New York, NY, USA. ACM Press.

Maas, S., Rich, A., and Nishikura, K. (2003). A-to-I RNA editing: Recent news and residual mysteries. *Journal of Biological Chemistry*, 278(3):1391–1394.

Mattick, J. S. (2003). Challenging the dogma: the hidden layer of non-protein-coding RNAs in complex organisms. *BioEssays*, 25:930–939.

Mittaz, L., Antonarakis, S. E., Higuichi, M., and Scott, H. S. (1997). Localization of a novel human RNA-editing deaminase (hred2 or adarb2) to chromosome 10p15. *Human Genetics*, 100:398–400.

Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In *ECAL '99*, p. 457–466, London, UK. Springer-Verlag.

Rocha, L. M. (1995). Contextual genetic algorithms: Evolving developmental rules. *ECAL 1995: Advances in Artificial Life*, LNCS 929, p. 368–382.

Rocha, L. M. and Huang, C.-F. (2004). The role of RNA editing in dynamic environments. In *ALIFE9*, p. 489–494. MIT Press.

Simpson, L. (1999). RNA editing-An evolutionary perspective. In Atkins, J. and Gesteland, R., editors, *The RNA World*, p. 585–608. Cold Spring Harbor.

Stuart, K. (1993). RNA editing in mitochondria of african trypanosomes. In Benne, R., editor, *RNA Editing : The Alteration of Protein Coding Sequences of RNA*, p. 26–52. Ellis Horwood Publishers.

Sturn, N. R. and Simpson, L. (1990). Kinetoplast DNA minicircles encode guide RNA's for editing of cytochrome oxidase subunit Iii mRNA. *Cell*, 61:879–884.

Wang, Q., Khillan, J., Gadue, P., and Nishikura, K. (2000). Requirement of the RNA editing deaminase adar1 gene for embryonic erythropoiesis. *Science*, 290(5497):1765–1768.