A Systematic Study of Genetic Algorithms with Genotype Editing

Chien-Feng Huang and Luis M. Rocha

Modeling, Algorithms, and Informatics Group (CCS-3),
Computer and Computational Sciences,
Los Alamos National Laboratory, MS B256,
Los Alamos, NM 87545, USA
{cfhuang, rocha}@lanl.gov

Abstract. This paper continues our systematic study of an RNA-editing computational model of Genetic Algorithms (GA). This model is constructed based on several genetic editing characteristics that are gleaned from the RNA editing system as observed in several organisms. We have expanded the traditional Genetic Algorithm with artificial editing mechanisms as proposed in [11] and [12]. The incorporation of editing mechanisms, which stochastically alter the information encoded in the genotype, provides a means for artificial agents with genetic descriptions to gain greater phenotypic plasticity, which may be environmentally regulated. The systematic study of this artificial genotype editing model has shed some light into the evolutionary implications of RNA editing and how to select proper genotype editors to design more robust GAs. Our results also show promising applications to complex real-world problems. We expect that the framework here developed will both facilitate determining the evolutionary role of RNA editing in biology, and advance the current state of research in Evolutionary Computation.

1 Introduction

Evidence for the important role of non-protein coding RNA (ncRNA) in complex organisms (higher eukaryotes) has accumulated in recent years. "ncRNA dominates the genomic output of the higher organisms and has been shown to control chromosome architecture, mRNA turnover and the developmental timing of protein expression, and may also regulate transcription and alternative splicing." ([9], p 930).

RNA Editing ([2]; [1]), a process of post-transcriptional alteration of genetic information, can be performed by ncRNA structures (though it can also be performed by proteins). The term initially referred to the insertion or deletion of particular bases (e.g. uridine), or some sort of base conversion. Basically, RNA Editing instantiates a non-inheritable stochastic alteration of genes, which is typically developmentally and/or environmentally regulated to produce appropriate phenotypical responses to different stages of development or states of the environment.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 1233–1245, 2004.

The most famous RNA editing system is that of the African Trypanosomes [2]. Its genetic material was found to possess strange sequence features such as genes without translational initiation and termination codons, frame shifted genes, etc. Furthermore, observation of mRNA's showed that many of them were significantly different from the genetic material from which they had been transcribed. These facts suggested that mRNA's were edited post-transcriptionally. It was later recognized that this editing was performed by guide RNA's (gRNA's) coded mostly by what was previously thought of as non-functional genetic material [13]. In this particular genetic system, gRNA's operate by inserting, and sometimes deleting, uridines. To appreciate the effect of this edition let us consider Fig. 1. The first example (p. 14 in [2]) shows a massive uridine insertion (lowercase u's); the amino acid sequence that would be obtained prior to any edition is shown on top of the base sequence, and the amino acid sequence obtained after edition is shown in the gray box. The second example shows how, potentially, the insertion of a single uridine can change dramatically the amino acid sequence obtained; in this case, a termination codon is introduced. It is important to retain that a mRNA molecule can be more or less edited according to the concentrations of the editing operators it encounters. Thus, several different proteins coded by the same gene may coexist in an organism or even a cell, if all (or some) of the mRNA's obtained from the same gene, but edited differently, are meaningful to the translation mechanism.

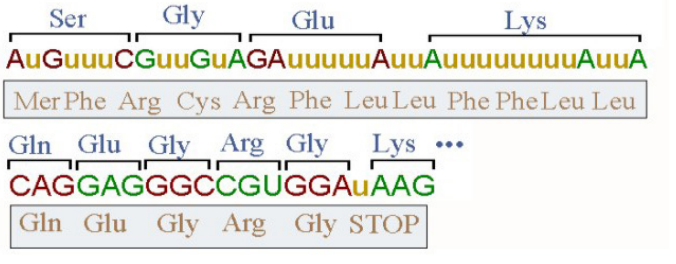

Fig. 1. U-insertion in Trypanosomes' RNA

The role of RNA editing in the development of more complex organisms has also been shown to be important. Lomeli et al. [8] discovered that the extent of RNA editing affecting a type of receptor channels responsible for the mediation of excitatory postsynaptic currents in the central nervous system, increases in rat brain development. As a consequence, the kinetic aspects of these channels differ according to the time of their creation in the brain's developmental process. Another example is that the development of rats without a gene (ADAR1) known to be involved in RNA editing, terminates midterm [14]. This showed that RNA Editing is more prevalent and important than previously thought. More recently, Hoopengardner et al. [5] found that RNA editing plays a central role in nervous system function. Indeed, many edited sites recode conserved and

functionally important amino acids, some of which may play a role in nervous system disorders such as epilepsy and Parkinson Disease.

Although RNA editing seems to play an essential role in the development of some genetic systems and more and more editing mechanisms have been identified, not much has been advanced to understand the potential evolutionary advantages, if any, that RNA editing processes may have provided. To acquire insights for answering this question, we started the systematic study of a Genetic Algorithm with Edition (GAE) initially proposed by Rocha [11], [12]. Specifically, we reported in [7] some preliminary results on how Genotype Editing may provide evolutionary advantages. Here, we continue this study by presenting results based on simulations with much larger numbers of runs with randomized parameters, yielding a more statistically significant treatment of the conclusions reached in [7] from individual examples of genotype editing. Our goal is to gain a deeper understanding of the nature of RNA editing and exploit its insights to improve evolutionary computation tools and their applications to complex problems. Before delving fully into this paper, the next section summarizes our prior work in Genetic Algorithms with Genotype Edition in [7].

2 Prior Work on Genetic Algorithms with Edition

In science and technology Genetic Algorithms (GA) [4] have been used as computational models of natural evolutionary systems and as adaptive algorithms for solving optimization problems. Table 1 depicts the process of a simple genetic algorithm.

Table 1. Mechanism of a simple GA.

1. Randomly generate an initial population of l n-bit chromosomes.
2. Evaluate each individual's fitness.
3. Repeat until l offspring have been created.
a. select a pair of parents for mating;
b. apply crossover operator;
c. apply mutation operator.
4. Replace the current population with the new population.
5. Go to Step 2 until terminating condition.

GAs operate on an evolving population of artificial organisms, or agents. Each agent is comprised of a genotype and a phenotype. Evolution occurs by iterated stochastic variation of genotypes, and selection of the best phenotypes in an environment according to a fitness function. In machine learning, the phenotype is a candidate solution to some optimization problem, while the genotype is an encoding, or description, of that solution by means of a domain independent representation, namely, binary symbol strings (or chromosomes). In traditional GAs, this code between genotype and phenotype is a direct and unique mapping.

In biological genetic systems, however, there exists a multitude of processes, taking place between the transcription of genes and their expression, responsible for the establishment of a one-to-many relation between genotype and phenotype. For instance, it was shown that RNA editing has the power to dramatically alter gene expression [10] (p. 78): "cells with different mixes of (editing mechanisms) may edit a transcript from the same gene differently, thereby making different proteins from the same opened gene."

In a genetic system with RNA editing, in other words, 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 [11], [12] proposed an expanded framework of GA with a process of stochastic edition of the genetic descriptions (chromosomes) of agents, prior to being translated into solutions. The editing process is implemented by a set of editors with different editing functions, such as insertion or deletion of symbols in the original chromosomes. Before these descriptions can be translated into the space of solutions, they must "pass" through successive layers of editors, present in different concentrations. In each generation, each chromosome has a certain probability (given by the concentrations) of encountering an editor in its layer. If an editor matches some subsequence of the chromosome when they encounter each other, the editor's function is applied and the chromosome is altered. The implementation of a GA with Edition (GAE) is described in the following:

The GAE model consists of a family of r m-bit strings, denoted as $(E_1, E_2, \ldots,$
$E_r)$, that is used as the set of editors for the chromosomes of the agents in a GA population. The length of the editor strings is assumed much smaller than that of the chromosomes: $m << n$, usually an order of magnitude. An editor E_j is said to match a substring, of size m, of a chromosome, 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 chromosomes: when the editor matches a portion of a chromosome, a number of bits are inserted into or deleted from the chromosome.

For instance, if the editing function of editor E_j is to add one randomly generated 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, this editor will instead delete the allele at s_{k+m+1} 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 concentrations 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. With these settings, the algorithm for the GA with genotype editing is essentially the same as the regular GA, except that step 2 in Table 1 is now redefined as:

"For each individual in the GA population, apply each editor E_j with probability v_j (i.e., concentration). If E_j matches the individual's chromosome S, then edit S with the editing function associated with E_j and evaluate the resulting individual's fitness."

It is important to notice that the "post-transcriptional" edition of genotypes is not a process akin to mutation, because editions are not inheritable. Just like in biological systems, it is the unedited genotype that is reproduced. One can also note that Genotype Editing is not a process akin to the Baldwin effect as studied by, e.g., Hinton and Nowlan [3]. The phenotypes of our agents with genotype edition, do not change (or learn) ontogenetically. In Hinton and Nowlan's experiments, the environment is defined by a very difficult ("needle in a haystack") fitness function, which can be made more amenable to evolutionary search by endowing the phenotypes to "learn" ontogenetically. Eventually, they observed, this learning allows genetic variation to discover, and genetically encode fit individuals. In contrast, genotype edition does not grant agents more "ontogenetic learning time", it simply changes inherited genetic information ontogenetically but the phenotype, once produced, is fixed. Also, as we show below, it is advantageous in environments very amenable to evolution, such as Royal Road functions (the opposite of "needle in a haystack") [7].

In [7], based on specific examples, we have demonstrated how the editing mechanism can improve the GA's search performance by suppressing the effects of hitchhiking. We have also showed that editing frequency plays a critical role in the evolutionary advantage provided by the editors – only a moderate degree of editing processes would facilitate organisms' exploration of the search space. Therefore, one needs to choose proper editor parameters to avoid over or under-editions in order to develop more robust GAs. In this paper, we conduct a larger statistical exploration using numerous sets of families of editors to elaborate on the conditions where genotype editing truly enhances the GA's search power.

3 Effects of Genotype Editing

How rapid is evolutionary change, and what determines the rates, patterns, and causes of change, or lack thereof? Answers to these questions can tell us much about the evolutionary process. The study of evolutionary rate in the context of GA usually involves defining a performance measure that captures the idea of rate of improvement, so that its change over time can be monitored for investigation. In many practical problems, a traditional performance metric is the "best-so-far" curve that plots the fitness of the best individual that has been seen thus far by generation n. As a step towards a deeper understanding of how Genotype Editing works, we employ a testbed, the small Royal Road S_1, which is a miniature of the class of the "Royal Road" functions [7].

Table 2 illustrates the schematic of the small Royal Road function S_1. This function involves a set of schemata $S = \{s_1, \ldots, s_8\}$ and the fitness of a bit string (chromosome) x is defined as $F(x) = \sum_{s \in S} c_s \sigma_s(x)$, where each c_s is a value assigned to the schema s as defined in the table; $\sigma_s(x)$ is defined as 1 if

Table 2. Small royal road function S_1

s_1 = 11111**********; $c_1 = 10$
s_2 = *****11111**********; $c_2 = 10$
s_3 = **********11111**********; $c_3 = 10$
s_4 = **********11111**********; $c_4 = 10$
s_5 = **********11111**********; $c_5 = 10$
s_6 = **********11111**********; $c_6 = 10$
s_7 = **********11111*****; $c_7 = 10$
s_8 = **********11111; $c_8 = 10$

x is an instance of s and 0 otherwise. In this function, the fitness of the global optimum string (40 1's) is $10 \times 8 = 80$.

There are several factors that play a role in the GAE's search power – e.g., *size of the family of editors*, *editor length*, *editor concentration* and *editor function* [7]. Our aim here is to investigate these four parameters. Since a multitude of parameter combinations are possible, we conduct numerous GAE runs and focus on a single parameter while other parameters are randomly generated in the beginning of each GAE run and then fixed until the end of that run. The results are then averaged over the number of the GAE runs so that we may zero in on the effects of that parameter.

3.1 Effects of Size of the Family of Editors

To study the effect of the size of editor family parameter r, two sets of values, $r \in$ 1,2,3,4,5 and $r \in$ {6,7,8,9,10}, are tested. The GAE was run 100 times for each set and in each GAE run the value of r is randomly chosen from the respective sets.[1] Figure 2.a and 2.b display the results on averaged best-so-far performance and averaged editing frequency (the total number of times all editors edited chromosomes in a generation) over 100 runs, respectively.[2] One can see that the GAEs with less editors (i.e., 1 to 5) clearly outperform the GAEs with more editors (i.e., 6 to 10).[3] The results also show that the editing frequency for the GAEs with less editors is substantially smaller than that of the GAEs with more

[1] The settings of other editor parameters are: each editor is a randomized bit-string of a randomly chosen number of bits from {1,10}; the editor concentration is randomly generated from [0,1]; and the editor function inserts or deletes a randomly chosen number of bits from {1,10}, as well. For the GA part, throughout this section, we use a population of 40 chromosomes, a binary tournament selection, and crossover and mutation rates of 0.7 and 0.005, respectively.

[2] The value of the averaged best-so-far performance is calculated by averaging the best-so-fars obtained at each generation for all 100 runs; and so is the averaged editing frequency, where the vertical bars overlaying the metric curves represent the 95-percent confidence intervals. This applies to all the results presented in this paper.

[3] We do not contrast the performance of traditional GAs with that of the GAE here, since the purpose in this section is to study the effects of the editor parameters per se. Please see [7] for specific examples of the GAE outperforming GA, as well as guidelines for choosing proper editors so that the GAEs can outperform the GAs.

editors. These results are intuitive, since more editors naturally tend to incur more editing processes.

To further elucidate the effects of this parameter, Figure 2.c displays the results of editing frequency in individual runs for r: 2, 5 and 10 editors. The corresponding maximal fitness reached by the GAE with 2, 5, and 10 editors is 70, 80 and 50, respectively (the detailed results are not displayed here due to the limit of the paper length). One can notice that in the run of the GAE with 10 editors, where the maximal fitness attained is far from the optimum, the editing frequency does not significantly drop to zero near the end of the experiments. It appears that the GAE's population continues utilizing the editors to explore the search space. This is the reason why the corresponding population diversity displayed in Figure 2.d is far from zero in the case of the GAE with 10 editors.[4] For the GAE with 2 editors, the best-so-far fitness located is close to the optimum – the results in Figure 2.c and 2.d show that the degree of editing is then reduced and the population is not as diverse as that of the GAE with 10 editors. All this indicates that the system settles into a dynamic equilibrium in which the exploratory power of the editing process is balanced by the exploitative pressure of selection.

In the case of the GAE with 5 editors, whose best-so-far fitness reaches the optimum, the striking difference is that the corresponding editing frequency declines dramatically as the GAE's population evolves, and tends to drop to zero at the end of the experiments. This shows that the editing process ultimately comes to an end and the population diversity is lost (as shown in Figure 2.d). Based on the effects of editor length and concentration, in the next two subsections we will present more results to support this observation.

3.2 Effects of Editor Length

To test the effect of the editor length parameter m, two sets of values, $m \in \{1,2\}$ and $m \in \{3,4,5\}$, are investigated. The GAE was run 100 times for each set and in each GAE run the value of m is randomly chosen from respective sets.[5] Figure 3.a illustrates the results for these two sets of GAEs, in which the GAEs with longer editor length (3 to 5 bits) outperform the other. This is

[4] To measure diversity at the i^{th} locus of a GA string, a simple bitwise diversity metric is defined as [7]: $D_i = 1 - 2|0.5 - p_i|$, where p_i is the proportion of 1s at locus i in the current generation. Averaging the bitwise diversity metric over all loci offers a combined allelic diversity measure for the population: $D = \frac{\sum_{i=1}^{l} D_i}{l}$. D has a value of 1 when the proportion of 1s at each locus is 0.5 and 0 when all of the loci are fixed to either 0 or 1. Effectively it measures how close the allele frequency is to a random population (1 being closest).

[5] The size of the editor family in this subsection is randomly chosen from {1,10}. The other two parameters (concentration and function) of an editor are generated by the same way as in the preceding subsection.

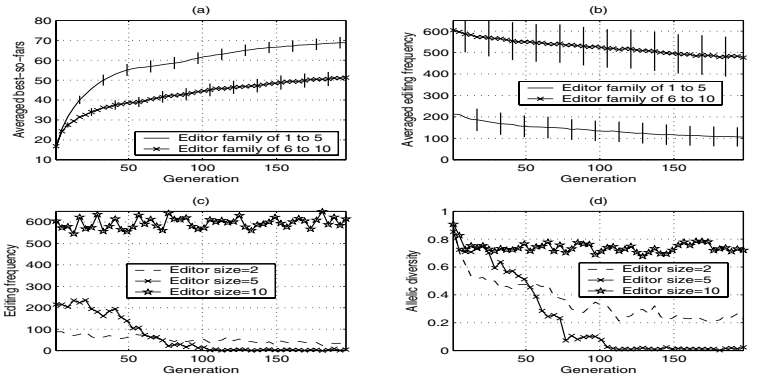

Fig. 2. Effects of size of the family of editors

also an intuitive result, since when the length of editors is too short, numerous matchings occur and the GAEs' population undergos too many editing processes. This typically results in serious disruptive effect on fit individuals.

In other words, the performance discrepancy of the GAEs with different editor length again depends on editing frequency. The empirical results for editing frequency shown in Figure 3.b confirm our assertion. The editing frequency for the GAEs with 1 to 2-bit editors is much higher than that of the GAEs with 3 to 5-bit editors. Therefore, beneficial genotype editing requires moderate editing frequency.

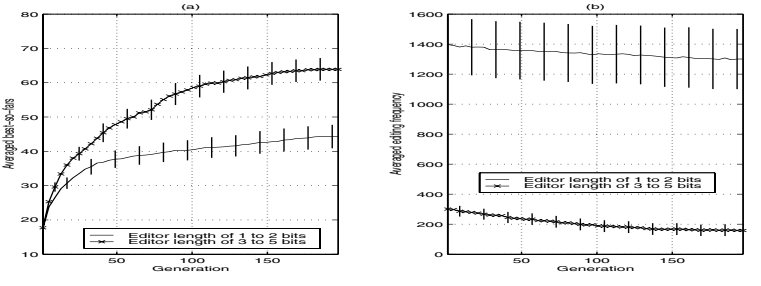

Fig. 3. Effects of editor length

3.3 Effects of Editor Concentration

To test the effect of the editor concentration parameter v_j, we again ran the GAE 100 times for two different sets of values of this parameter for each editor

E_j: $v_j \in$ [0,0.5] and $v_j \in$ in [0.5,1].[6] Thus, the probability that chromosomes encounter editors in the second set of GAEs is higher than in the first set of GAEs. Figure 4.a and 4.b display the results. Since the probability of the chromosomes meeting with editors is higher in the second set of GAEs, the population naturally undergos more editions than in the first set of GAEs.

These results again indicate that the performance difference lies in the number of the performed editions. When the GAE's population is considerably edited by the editors, too much exploration of the search space generates deleterious effects on performance advancement. Appropriate editor concentration is thus essential for the GAE, since beneficial genotype edition requires a moderate quantity of editions.

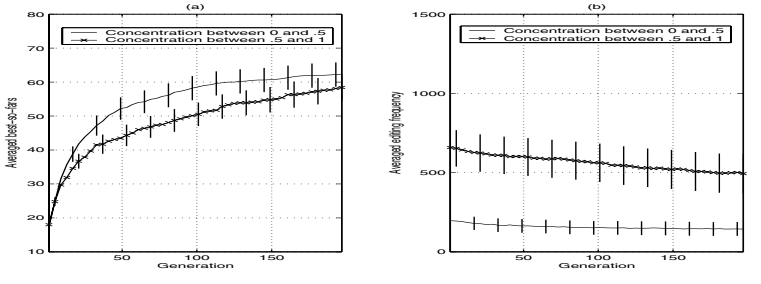

Fig. 4. Effects of editor concentration

3.4 Effects of Editor Function

To test the effect of the editor functions F_j, we again ran the GAE 100 times for two different sets of functions for each editor E_j. The scope of possible functions is open-ended, but here we contrast moderate edition with massive edition. The first set of functions F_j insert or delete a randomly chosen small number of bits in {1,2,3}. The second set of functions F_j insert or delete a randomly chosen larger number of bits in {10,11,12,13,14,15}.[7] Figure 5.a and 5.b display the corresponding results. Since the gene deletion or insertion frequency in chromosomes is now much higher in the second set of GAEs, the population naturally undergos more disruptive processes than in the first set of GAEs.

These results demonstrate that the performance difference lies in the degree of gene deletion (or insertion) in chromosomes. Appropriate editor function is thus also very important for the GAE to gain substantial search progress.

[6] The size of the editor family in this subsection is randomly chosen from {1,10}. The other two parameters (length and function) of an editor are generated by the same way as in Section 3.1.

[7] The size of the editor family in this subsection is randomly chosen from {1,5}; an editor's concentration is randomly generated from [0,1] and its length is a randomly chosen number of bits from {1,10}.

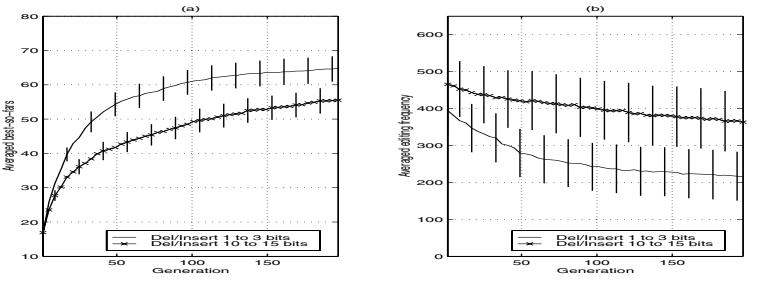

Fig. 5. Effects of editor function

4 Design of Robust GA

The study of Genotype Editing has provided us with insights into how to choose editor parameters for developing more robust GAs. Basically, in order to facilitate the GAE's search process, the guidelines are: the size of the editor family, the length and concentration of the editors need to be moderate so as to avoid over or under-editing processes; the editor function should not lead to massive deletions (or insertions).

In this section we apply these rules to select proper genotype editors for the design of more robust GAEs, and test them on a multimodal, non-building-block-based test function – the modified Schaffer's function F_7 [6]:

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

where $-1 \leq x_i \leq 1$ for $1 \leq i \leq 2$. A sketch of this function is displayed in Figure 6. To attain the global optimum at the center of the search space, the population has to cross over many deep wells and high barriers. Since there are many local optima in the search space, a traditional GA's population can easily converge on any of them. The multimodality of this testbed is hence expected to present substantial difficulty to the GA's search.

Each of the two variables is encoded by 30 bits, and thus each individual is a binary string of length 60. We use a GA of population size 50, a binary tournament selection, and crossover and mutation rates of 0.7 and 0.005, respectively. We contrast the traditional GA with a GAE with the same parameters, but with genotype edition performed by a family of five editors as shown in Table 3. The experiments are conducted for 100 runs, each run with 200 generations.

Figure 7.a displays the averaged best-so-far performance, where one can see that the search performance of the GAE is better than that of the traditional GA. We also record the value of best-so-far attained at the end of each run and generate histograms as illustrated in Figure 7.b (for the GA) and 7.c (for the GAE). The results show that the GAE tends to locate more best-so-fars that are close to the optimum. One can also notice that there are several runs in which the traditional GA does not even locate best-so-fars of more than 2.3, meaning

Table 3. Parameters of the five editors

	editor 1	editor 2	editor 3	editor 4	editor 5
length	5	4	5	3	6
alleles	{0,0,1,1,0}	{1,0,0,1}	{0,1,1,0,1}	{0,1,1}	{1,1,1,1,0,0}
concentration	0.1410	0.7936	0.2524	0.5885	0.0871
function	delete 2 bits	delete 1 bit	add 3 bits	add 2 bits	add 5 bits

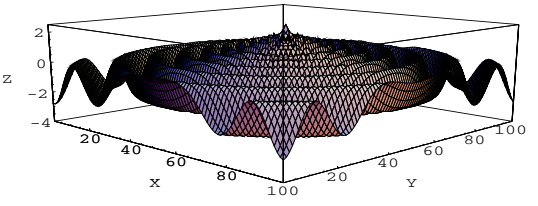

Fig. 6. Modified Schaffer function F_7

that the population in these runs prematurely converge on these "lower" local optima.

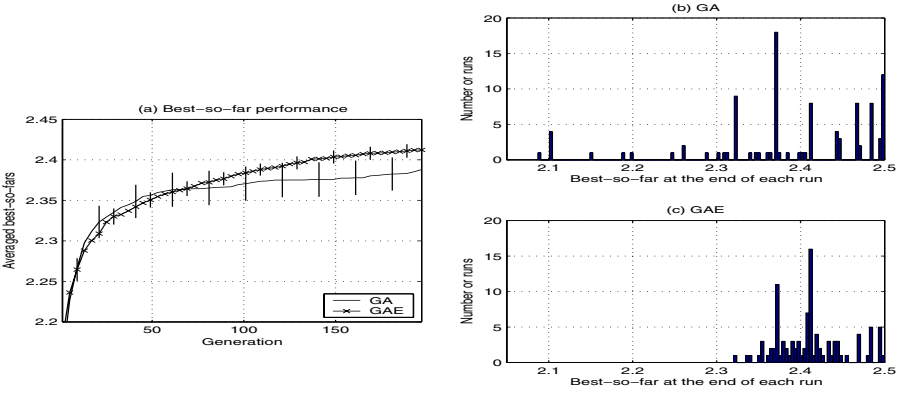

Fig. 7. Best-so-far performance and distribution on the modified Schaffer function F_7

5 Conclusion and Future Work

We have continued our systematic investigation of Genotype Editing in GA and tested several evolutionary scenarios. The results obtained have provided the following insights:

Editing frequency plays a critical role in the evolutionary advantage provided by the editors – only a moderate degree of editing processes can facilitate

organisms' exploration of the search space. Our results also indicate that editing frequency declines dramatically as the population diversity is lost, indicating that the editing process ultimately comes to an end. If the editing frequency does not substantially decrease, the system settles into a dynamic equilibrium where the exploratory power of the editing process is balanced by the exploitative pressure of selection.

We have also learned some rules for setting up editors' parameters to develop robust GAs. The results obtained show promising applications to practical problems. Indeed, Genotype Editing demonstrates the capability of substantially improving the GA's search power.

In this paper we have thus far discussed GAs with edition solely with constant parameters, such as fixed concentrations, of editors and a stable environment defined by a fixed fitness function. That is, the edition parameters are fixed at the start of a given run. They do not change or adapt in the evolutionary process. Our preliminary tests (not discussed here), however, also show that constant concentrations of editors may not grant the system any evolutionary advantage when the environment changes. In order to simulate a genetic system in which the linking of editors' concentrations with environmental states may be advantageous in time-varying environments, Rocha [11], [12] proposed a new type of GA known as Contextual Genetic Algorithms (CGA). In this class of algorithms, the concentrations of editors change with the states of the environment, thus introducing a control mechanism leading to phenotypic plasticity and greater evolvability. We are currently working on this model and, together with the insights acquired previously, in future work we aim at (1) conducting more biologically realistic experiments which may lead us towards a better understanding of the advantages of RNA editing in nature, and (2) developing novel evolutionary computation tools for dealing with complex, dynamic real-world problems.

References

1. Bass, B.L. (Ed.) (2001). RNA Editing. Frontiers in Molecular Biology Series. Oxford University Press.
2. Benne, R. (Ed.) (1993). RNA Editing: The Alteration of Protein Coding Sequences of RNA. Ellis Horwood.
3. Hinton, G. E. and Nowlan, S. J. (1987). "How learning can guide evolution." Complex Systems. Vol. 1, pp. 495-502.
4. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.
5. Hoopengardner, B., Bhalla, T., Staber, C., and Reenan, R. (2003). "Nervous System Targets of RNA Editing Identified by Comparative Genomics." Science 301: 832-836.
6. 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.
7. Huang, C-F. and Rocha, L. M. (2003). "Exploration of RNA Editing and Design of Robust Genetic Algorithms." Proceedings of the 2003 IEEE Congress on Evolutionary Computation, Canberra, Australia, December 2003. R. Sarker et al (Eds). IEEE Press, pp. 2799-2806.

8. Lomeli, H. et al. (1994). "Control of Kinetic Properties of AMPA Receptor Channels by RNA Editing." Science, 266: 1709-1713.
9. Mattick, J. S. (2003). "Challenging the Dogma: the Hidden Layer of Non-protein-coding RNAs in Complex Organisms." BioEssays. 25: 930-939.
and Scott, H. S. (1997). "Localization of a Novel Human RNA-editing Deaminase (hRED2 or ADARB2) to Chromosome 10p15." Human Genetics, 100: 398-400.
10. Pollack, R. (1994). Signs of Life: The Language and Meanings of DNA. Houghton Mifflin.
11. Rocha, Luis M. (1995). "Contextual Genetic Algorithms: Evolving Developmental Rules." Advances in Artificial Life. Moran, J., Moreno, A., Merelo, J. J., and Chacon, P. (Eds.). Springer Verlag, pp. 368-382.
12. Rocha, Luis M. (1997). Evidence Sets and Contextual Genetic Algorithms: Exploring Uncertainty, Context and Embodiment in Cognitive and biological Systems. PhD. Dissertation. State University of New York at Binghamton. Science.
"RNA Editing." Annual Review of Neuroscience, 19: 27-52.
13. 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.
14. 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.