

Tracking Extrema in Dynamic Environments using a Coevolutionary Agent-based Model of Genotype Edition

Chien-Feng Huang
Center for Nonlinear Studies,
Los Alamos National Laboratory, MS B258, Los
Alamos, NM 87544
cfhuang@lanl.gov

Luis M. Rocha
School of Informatics and Cognitive Science
Program,
Indiana University, 1900 East Tenth Street,
Bloomington, IN 47406
rocha@indiana.edu

ABSTRACT

Typical applications of evolutionary optimization in static environments involve the approximation of the extrema of functions. For dynamic environments, the interest is not to locate the extrema but to follow it as closely as possible. This paper compares the extrema-tracking performance of a traditional Genetic Algorithm and a coevolutionary agent-based model of Genotype Editing (ABMGE). This model is constructed using several genetic editing characteristics that are gleaned from the RNA editing system as observed in several organisms. The incorporation of editing mechanisms provides a means for artificial agents with genetic descriptions to gain greater phenotypic plasticity. By allowing the family of editors and the genotypes of agents to co-evolve using the re-generation of editors as a control switch for environmental changes, the artificial agents in ABMGE can discover proper editors to facilitate the tracking of the extrema in dynamic environments. We will show that this agent-based model, together with a coevolutionary mechanism, is more adaptive and robust than the GA. We expect the framework proposed in this paper to advance the current state of research of Evolutionary Computation in dynamic environments.

Categories and Subject Descriptors: I.2.11 Distributed Artificial Intelligence: multiagent systems; I.6.5 Model Development: modeling methodologies; I.6.8 Types of Simulation: distributed.

General Terms: Algorithms, experimentation, performance.

Keywords: RNA Editing, genotype editing, agent-based model, coevolution, editing frequency, dynamic environments.

1. INTRODUCTION

Coevolutionary algorithms, methods by which multiple populations of agents (or species) are adapting to each other, have been studied in the field of Evolutionary Computation

([5]; [12]). They are popular augmentations of traditional evolutionary algorithms. In nature, coevolution is the process of reciprocal genetic change in one species in response to another. The reciprocal change observed in coevolution can be considered either as a competitive arms race, or cooperative methodology where separate population evolve components of the solution [13].

In essence, the coevolution of multiple populations of agents collectively acquire the information for the location of the extrema of static functions. When the function being optimized is dynamic, the goal is not to acquire the extrema but to track their progression through the space as closely as possible.

This paper compares the extrema-tracking performance of a traditional Genetic Algorithm and a coevolutionary agent-based model of Genotype Editing (ABMGE). We have previously shown that the incorporation of genotype editing mechanisms provides a means for artificial agents with genetic descriptions to gain greater phenotypic plasticity ([14]; [9]; [10]; [16]). Here we show that, by allowing the family of editors and the genotypes of agents to co-evolve using the re-setting of editors as a control switch for environmental changes, the ABMGE is successful in tracking extrema when facing dynamic environments. We demonstrate that this agent-based model, together with a coevolutionary mechanism, is more adaptive and robust than the GA.

The next section discusses the biological background of RNA editing that lays out the foundation for our model. We then review a model of Genotype Edition using Genetic Algorithms and how it is extended to build an agent-based model of Genotype Editing. This is followed by several non-trivial dynamic testbeds and the corresponding experimental results.

2. RNA EDITING

RNA Editing ([4]; [2]), a process of post-transcriptional alteration of genetic information, can be performed by non-coding RNA (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 at different stages of development or to states of the environment.

Copyright 2005 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the U.S. Government. As such, the Government retains a non-exclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

Table 1: Mechanism of a simple GA.

1. Randomly generate an initial population of l agents, each being an n -bit genotype (chromosome).
2. Evaluate each agent’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.

through interactions with other types of molecules, namely RNA editors such as gRNA’s. Based upon this analogy, Rocha ([14]; [15]) proposed an expanded framework of GA with a process of stochastic edition of the genotypes (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, deletion or substitution 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, \dots, 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: $m \ll n$, 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 $e_i = s_{k+i}, i = 1, 2, \dots, m, 1 \leq k \leq n-m$, 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. For instance, when the editor matches a portion of a chromosome, a number of bits are inserted into or deleted from the chromosome. In Rocha’s formulation ([14]; [15]), any string function is possible, including substitution. Here we use only insertion and deletion functions.

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, \dots, 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 . Figure 2 depicts the whole process of this model. With these settings, the algorithm for the GA with Genotype Edition is essentially the same as the regular GA, except that step 2 in Table 1 is now redefined as:

“For each chromosome 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.”

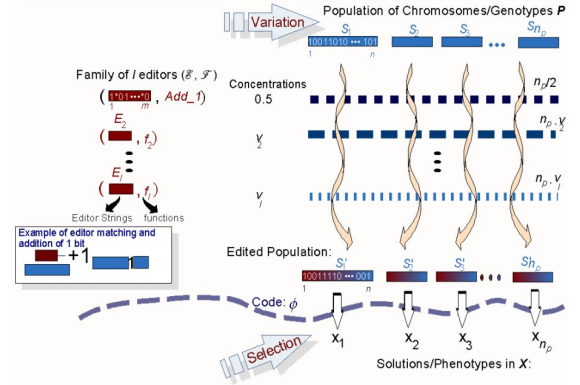


Figure 2: Schematic of GAE

It is important to note 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. As a result, what is being reproduced in our model is the unedited genotypes, but each individual’s fitness is calculated using the phenotype produced from its edited genotype. Therefore, the unedited and edited genotypes can be viewed as mimicking functional DNA and edited coding mRNA in biological organisms, respectively, though our model does not include a true DNA/RNA distinction.

One can also notice that this Genotype Editing model is not a process akin to the Baldwin effect as studied by, e.g., Hinton and Nowlan [6]. 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.

It is also important to retain that just like an mRNA molecule may be edited in different degrees according to the concentrations of editing operators it encounters, in the GAE the same chromosome may be edited differently because the editor concentration is a stochastic parameter that specifies the probability of a given editor encountering a chromosome. Thus, if the same genotype is repeated in the population, it may actually produce different solutions (or phenotypes). This is similar to what happens with RNA editing in biological organisms where, at the same time, several different proteins coded by the same gene may coexist.

Table 2: Mechanism of ABMGE.

1. Randomly generate an initial agent population, each agent consisting of a n -bit chromosome and a family of editors.
2. Edit each agent’s chromosome using the agent’s editor family and evaluate each agent’s fitness.
3. Repeat until l offspring have been created.
 - a. select a pair of parents for mating;
 - b. apply genotype crossover operator;
 - c. apply genotype mutation operator.
4. Replace the current population with the new population.
5. Go to Step 2 until terminating condition.

3.2 A Coevolutionary Agent-based Model of Genotype Edition

In this paper, we discuss an extended version of the GAE using a more realistic co-evolutionary agent model as proposed in [16]. Whereas the GAE model defines a single family of editors for the entire population, the agent-based model we introduce here allows for heterogeneous agents, each with a distinct editor family. Therefore, instead of every chromosome encountering the same editors with the same probability, in the agent-based model of Genotype Editing (ABMGE) the genotype of each agent is edited by its own editor family. Figure 3 depicts an agent in the ABMGE model.

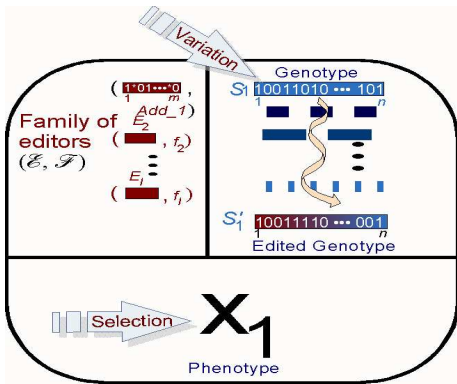


Figure 3: Schematic of an agent in ABMGE

Table 2 shows the algorithm for the ABMGE. In this model, the sequences of editors in the editor family for each agent, once generated, are fixed. Variation operations are applied to the agent’s genotypes (step 3.b and 3.c).

One way to highlight the difference between the two models is to notice that in the GAE model, there are essentially two separate populations: an evolving population of genotype strings and a fixed, small, editor population (family). Moreover, in the simple GAE the entire population of agents faces exactly the same “post-transcriptional” editors. In contrast, in the ABMGE model, the evolving agents face heterogeneous post-transcriptional editors. As fit agents are selected for reproduction, their editor families propagate to the next generation.

In static and dynamic environments the results in [9], [10] and [16] have demonstrated how genotype editing with

Table 3: Small royal road function L_1

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

an homogenous editor family, as implemented in the GAE model, can improve the standard GA search performance. It was shown that editing frequency (the total number of times all editors edited chromosomes in a generation) plays a critical role in the evolutionary advantage provided by the editors: only a moderate degree of editing processes facilitates the exploration of the search space. Those results also indicate that, as the population of individuals converges to a single phenotype (or a few phenotypes), editing frequency may dramatically decrease so that the editing process ultimately comes to an end.

In this paper, we will further show that a significant drop of the editing frequency may also occur when the editors do not play a beneficial role in the agents’ adaption to the changing environments, indicating that the editors being evolved with the population are not able to offer adaptive advantage when the population continues to face considerable changes in environments. As a result, by allowing the family of editors and the genotypes of agents to co-evolve using the re-setting of editors as a control switch for environmental changes, the ABMGE is successful in tracking extrema in dynamic environments. We demonstrate that this agent-based model, together with this coevolutionary mechanism, is more adaptive and robust than the GA.

4. EMPIRICAL RESULTS

4.1 Oscillatory Royal Road Functions

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 embodies the idea of rate of improvement, so that its change over time can be monitored for investigation. 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 .¹ As a step towards the understanding of how the coevolutionary ABMGE works, an oscillatory Royal Road function is studied, which is an oscillatory version of the simple “Royal Road” L_1 function used in [9].

Table 3 illustrates the schematic of the small Royal Road function L_1 . This function involves a set of schemata $S = \{s_1, \dots, s_8\}$ and the fitness of a genotype bit string (chromosome) x is defined as $F(x) = \sum_{s \in S} c_s \sigma_s(x)$, where each

¹The best-so-far measurement commonly 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 will be re-evaluated whenever a change in the environment occurs.

c_s is a value assigned to the schema s as defined in the table; $\sigma_s(x)$ is defined as 1 if x is an instance of s and 0 otherwise. In this function, the fitness of the global optimum string (40 1's) is $10 \times 8 = 80$. The oscillatory royal road function consists of two fitness landscapes, L_0 and L_1 , in which each schema is comprised of all 0's in L_0 and the other parameters remain the same as used in L_1 . These two landscapes are maximally different in the configurations of their fitness landscapes. By oscillating these two landscapes, we are able to study the effects of drastic environmental changes.

We conduct 100 runs for the GA and the ABMGE. The GA experiments conducted in this section are based on a binary tournament selection, one-point crossover and mutation rates of 0.7 and 0.005, respectively; population size is 100 for each of the GA and ABMGE run. For the ABMGE, an editor family is randomly generated for each agent.² Table 4 shows the parameters of an example editor family with two editors: length, alleles, concentration and editing function. For example, editor 1 is a bit-string of length 5 (01011); its concentration, or the probability that the agent's genotype string will encounter this editor is 0.9073; its editing function is to delete 1 bit, meaning that this editor deletes one allele of the genotype string at the position following the genotype's substring that matches the editor's string. The ultimate goal of the ABMGE is to facilitate the discovery of beneficial family of editors for each agent.

Table 4: Parameters for an example editor family.

	editor 1	editor 2
length	5	3
alleles	{0,1,0,1,1}	{0,1,0}
concentration	0.9073	0.2691
function	delete 1 bit	add {1,1,0}

Consider now a dynamic environment which oscillates periodically between the landscapes L_1 and L_0 . This oscillation models an environment with recurring dramatic changes in conditions. Figure 4 displays the results of the GA and ABMGE (without re-setting the families of editors) using the oscillation of landscapes L_1 and L_0 , at every 250 generations. As can be seen, the population of the ABMGE can better track the extrema in the first 500 generations (the first two oscillatory periods), but fails to outperform the GA after that.³ Further inspection of the corresponding editing frequency in Figure 5 indeed shows a significant drop of the editing frequency, thereby indicating that the editors do not play a beneficial role in the agents' adaption to the changing environments after the first few oscillations of the fitness function. This shows that the editors being evolved with the population are not able to offer adaptive advan-

²The size of the editor family ranges from $\{1,5\}$. Each editor is a randomized bit-string of a randomly chosen number of bits from $\{2,5\}$. The editor concentration is randomly generated from $[0,1]$; and the editor function inserts or deletes a randomly chosen number of bits from $\{1,3\}$. (See [10] for the guidelines concerning choosing these parameters of the editors.)

³The value of the averaged best-so-far performance metric is calculated by averaging the best-so-fars obtained at each generation for all 100 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.

tage when continuing to face several rounds of the drastic environmental changes.

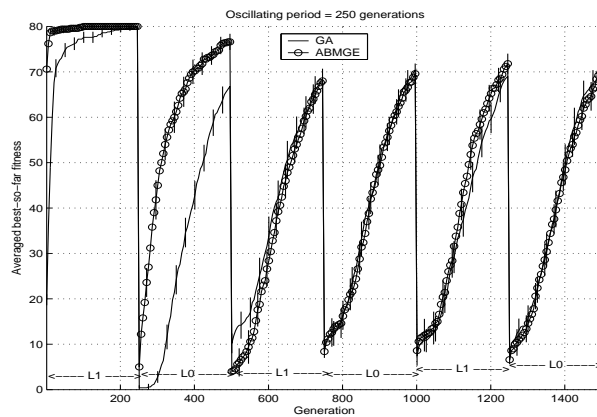


Figure 4: Averaged best-so-far performance on the oscillating Royal Road function of the GA and the ABMGE.

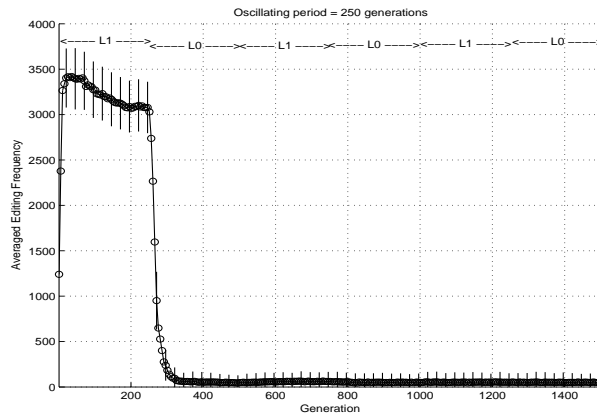


Figure 5: Editing frequency of the ABMGE on the oscillatory Royal Road function.

One can therefore consider to allow the family of editors and the genotypes of agents to co-evolve using the regeneration of editors as a control switch, so that the artificial agents in ABMGE can discover proper editors to facilitate the tracking of the extrema. That is, when the population detects a significant decline of the editing frequency, new families of editors are randomly re-generated, with the goal of discovering beneficial editors.

Figure 6 shows the best-so-far performance when the editing frequency is being reset whenever its values drop below 50 (this number is arbitrarily chosen for illustration, based on the heuristic of it being 10 times of the maximum number of editors in an agent, which is 5.) The results displayed in Figure 6 shows significant improvement for the ABMGE in following the extrema.

4.2 Dynamic Schaffer Function F_7

The preceding subsection discusses our study of the ABMGE in a drastic changing environment. The degree of the changes in an environment has been studied in EC research

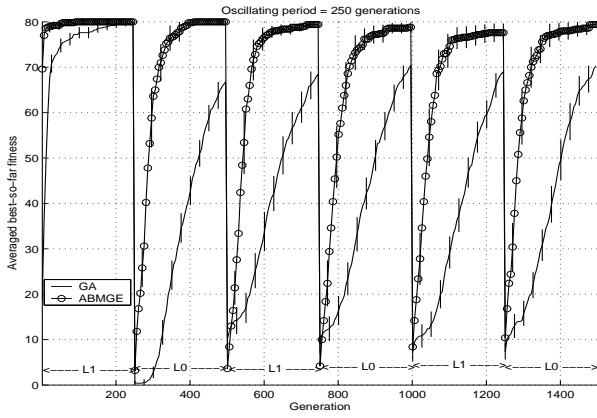


Figure 6: Best-so-far performance on the oscillatory Royal Road function of the GA and the coevolutionary ABMGE.

– how strongly the system changing is and how severe this change is going to constrain the search of an algorithm? Angeline [1], Bäck [3] reported a study of dynamic problems with three different modes of severity of changes: linear, circular, and random. In this subsection, with s representing a parameter to set the severity, one illustration can be described by:

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

where $X_i = x_i + \delta(t)$, $-1 \leq x_i \leq 1$ for $i = 1, 2$.

This testbed is a dynamic version of the modified Schaffer's function F_7 studied in [8]. A sketch of the original static function is displayed in Figure 7. X and Y-axis represent the index of the sample points in parameters x_1 and x_2 that are used to compute $f(\bar{x})$, which is then represented on Z-axis.

The example uses linear dynamics with severity s of 0.1:

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

Note that T is used as index for the environmental state; whenever the environment changes (e.g., every 50 generations in this subsection), T is increased by 1. This equation will be used for the dynamic test function studied in the next subsection, as well.

In this subsection, the same GA parameters as in the preceding subsection are used. For the ABMGE, the settings of the editor parameters are: the size of the family of the editors for each agent ranges from {1, 10} and other editor parameters remain the same as those employed in the preceding subsection. (We use a larger size of family of editors here since our results show that as the complexity of the problem increases, a larger size of editor family is required to offer adaptive advantage to the system.) The editing frequency is required to be reset whenever its values drop below 100 (this number is chosen based on the heuristic of it being 10 times of the maximum number of editors in an agent, which is 10.) Figure 8 displays the averaged best-so-far performance for this dynamic problem. These results are encouraging since the coevolutionary ABMGE consistently outperforms the traditional GA in tracking the extrema.

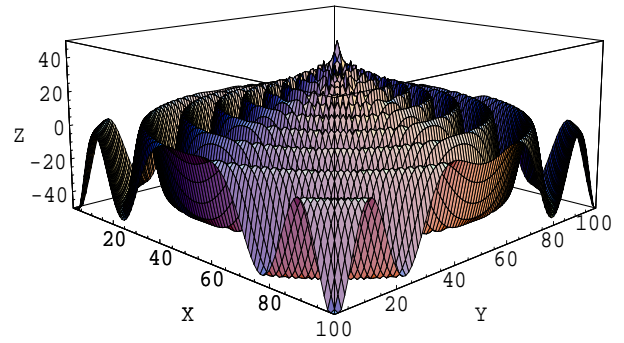


Figure 7: A sketch of the Schaffer Function F_7 .

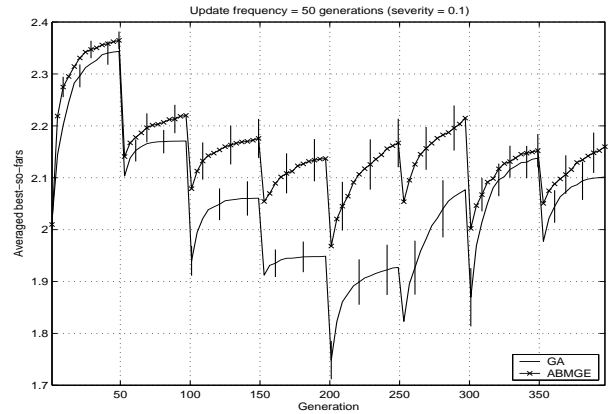


Figure 8: Best-so-far performance of the GA and the coevolutionary ABMGE on the linear dynamic Schaffer function (severity = .1).

4.3 Dynamic Optimal Control Testbed

Optimal Control problems often arise in many different fields of engineering and sciences. This class of problems has been well studied from both theoretical and computational perspectives. The models used to describe optimal control problems almost always involve more or less nonlinearity in nature. This often results in the existence of multiple local optima in the area of interest.

In this subsection we employ an artificial optimal control problem designed in [8]. The constraints of the artificial optimal control problem are:

$$\begin{aligned} \frac{d^2 z(t)}{dt^2} + \sin(z(t)) \frac{dz(t)}{dt} + \sin(t) \cos(z(t)) z(t)^3 \\ = \sin(t) u_1^2 + \cos(t) u_2^2 + \sin(t) u_1 u_2, \\ z(t_0) = 2, \dot{z}(t_0) = 2, t \in [0, 1], \end{aligned}$$

where $U_i = u_i + \delta(T)$, $-1 \leq u_i \leq 1$ for $i = 1, 2$.

Two examples are studied in this subsection that use linear dynamics with severity s of 0.1 and 1 according to Equation (1), respectively.

The goal is to maximize $z(t_f)^2$ by searching for two control variables, u_1 and u_2 ($-5 \leq u_1, u_2 \leq 5$). A sketch of this dynamic function for $\delta = 0, 3, 7$ are illustrated on Figure 9, 10 and 11, respectively. X and Y axes represent the index of sample points in parameters u_1 and u_2 that are used to

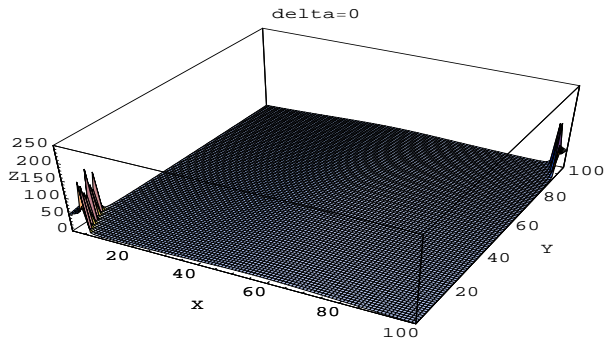


Figure 9: Schematic of the linear dynamic optimal control testbed ($\delta = 0$).

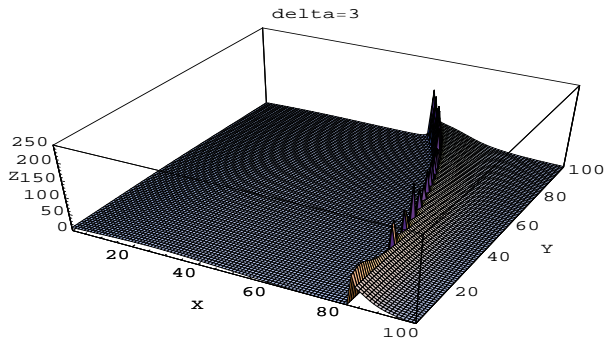


Figure 10: Schematic of the linear dynamic optimal control testbed ($\delta = 3$).

compute $z(t_f)^2$, which is then plotted on Z-axis.

In this subsection, each of the two variables is encoded by 50 bits, and thus each individual is a binary string of length 100. We again contrast the traditional GA with a coevolutionary ABMGE with the same parameters as those used for the Schaffer dynamic function.

Figure 12 and 13 display the averaged best-so-far performance for severity = .1 and 1, respectively, which shows that the editors provide adaptive advantage for the ABMGE in tracking the extrema of the search space. Along with the results obtained for the previous testbeds, we have been able to show that the coevolutionary ABMGE can be more

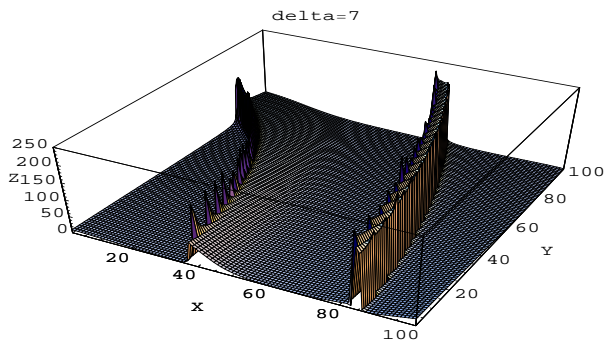


Figure 11: Schematic of the linear dynamic optimal control testbed ($\delta = 7$).

adaptive than the traditional GA when facing dynamic environments.

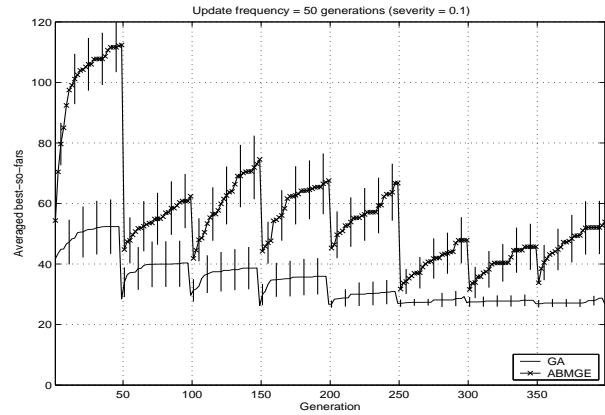


Figure 12: Best-so-far performance of the GA and the coevolutionary ABMGE on the linear dynamic optimal control testbed (severity = .1).

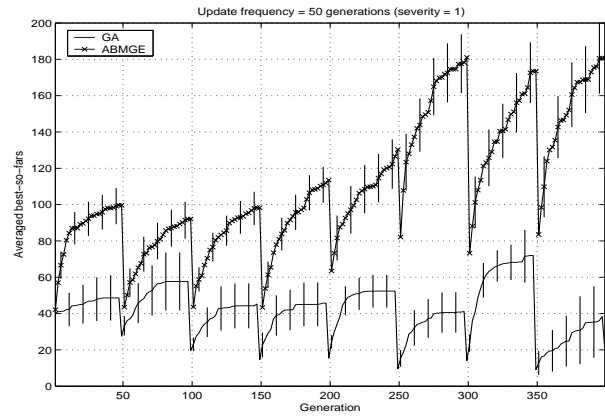


Figure 13: Best-so-far performance of the GA and the coevolutionary ABMGE on the linear dynamic optimal control testbed (severity = 1).

5. CONCLUSION AND FUTURE WORK

This paper presents a comparative study of traditional GA and a coevolutionary agent-based model of Genotype Editing for dynamic environment. This genotype edition model is constructed using several genetic editing characteristics that are gleaned from the RNA editing system as observed in several organisms. The incorporation of editing mechanisms provides a means for artificial agents with genetic descriptions to gain greater phenotypic plasticity. By using the re-generation of editors (based on levels of editing frequency) as a control switch for environmental changes, the artificial agents with heterogeneous genotype edition in ABMGE can discover proper editors to facilitate the tracking of the extrema. When facing several non-trivial dynamic problems, the results show that this agent-based model, along with a coevolutionary mechanism, is more adaptive than the traditional GA.

There are other ways to introduce coevolution in the AB-MGE systems. In our future work, we will study the effects of using other editor parameters as a coevolutionary mechanism. For example, mutation on editors, change of editors' function, or crossing-over between editors can cause the family of editors to coevolve with the genotype population. We expect that the framework proposed here can advance the current state of research of Evolutionary Computation in dynamic environments.

6. REFERENCES

- [1] P. J. Angeline. Tracking extrema in dynamic environments. In *Proceedings of the Sixth International Conference on Evolutionary Programming*, pages 335–345, 1997.
- [2] B. Bass. *RNA Editing. Frontiers in Molecular Biology Series*. Oxford University Press, 2001.
- [3] T. Bäck. On the behavior evolutionary algorithms in dynamic environments. In *IEEE International Conference on Evolutionary Computation*, pages 446–451, 1998.
- [4] R. Benne. *RNA Editing: The Alteration of Protein Coding Sequences of RNA*. Ellis Horwood, 1993.
- [5] S. Ficici and J. B. Pollack. A game-theoretic memory mechanism for ecevolution. In *Proc. of 2003 Genetic and Evolutionary Computation Conference*, pages 286–297, 2003.
- [6] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987.
- [7] B. Hoopengardner, T. Bhalla, C. Staber, and R. Reenan. Nervous system targets of rna editing identified by comparative genomics. *Science*, 301(5497):832–836, 2003.
- [8] C.-F. Huang. *A Study of Mate Selection in Genetic Algorithms*. Doctoral dissertation. Ann Arbor, MI: University of Michigan, Electrical Engineering and Computer Science., 2002.
- [9] C.-F. Huang and L. M. Rocha. Exploration of rna editing and design of robust genetic algorithms. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, pages 2799–2806. IEEE Press, 2003.
- [10] C.-F. Huang and L. M. Rocha. A systematic study of genetic algorithms with genotype editing. In *Proc. of 2004 Genetic and Evolutionary Computation Conference*, volume 1, pages 1233–1245, 2004.
- [11] H. Lomeli et al. Control of kinetic properties of ampa receptor channels by rna editing. *Science*, 266:1709–1713, 1994.
- [12] L. Panait, R. P. Wiegand, and S. Luke. A sensitivity analysis of a cooperative coevolutionary algorithm biased for optimization. In *Proc. of 2004 Genetic and Evolutionary Computation Conference*, volume 1, pages 573–584, 2004.
- [13] M. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation Journal* 8(1):1–29, 2000.
- [14] L. M. Rocha. Contextual genetic algorithms: Evolving developmental rules. *Advances in Artificial Life*, pages 368–382, 1995.
- [15] L. M. Rocha. *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., 1997.
- [16] L. M. Rocha and C.-F. Huang. The role of rna editing in dynamic environments. In *The Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, MIT press, pages 489–494, 2004.
- [17] N. R. Sturn and L. Simpson. Kinetoplast dna minicircles encode guide rna's for editing of cytochrome oxidase subunit iii mrna. *Cell*, 61:879–884, 1990.
- [18] Q. Wang, J. Khillan, P. Gadue, and K. Nishikura. Requirement of the rna editing deaminase adar1 gene for embryonic erythropoiesis. *Science*, 290(5497):1765–1768, 2000.