A Comparative Study of Immune System Based Genetic Algorithms in Dynamic Environments

Shengxiang Yang
Department of Computer Science, University of Leicester
University Road, Leicester LE1 7RH, United Kingdom
s.yang@mcs.le.ac.uk

ABSTRACT

Diversity and memory are two major mechanisms used in biology to keep the adaptability of organisms in the ever-changing environment in nature. These mechanisms can be integrated into genetic algorithms to enhance their performance for problem optimization in dynamic environments. This paper investigates several GAs inspired by the ideas of biological immune system and transformation schemes for dynamic optimization problems. An aligned transformation operator is proposed and combined to the immune system based genetic algorithm to deal with dynamic environments. Using a series of systematically constructed dynamic test problems, experiments are carried out to compare several immune system based genetic algorithms, including the proposed one, and two standard genetic algorithms enhanced with memory and random immigrants respectively. The experimental results validate the efficiency of the proposed aligned transformation and corresponding immune system based genetic algorithm in dynamic environments.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

Immune system based genetic algorithms, transformation, memory, random immigrants, dynamic environments

1. INTRODUCTION

Optimization in dynamic environments has emerged into a rapidly growing subject of research for the community of genetic algorithms (GAs) due to the practical relevance [2]. In the real world, many problems are actually dynamic optimization problems (DOPs), where the optimization crite-ria, design variables, and environmental conditions may all change over time due to many factors, e.g., machine breakdown, stochastic orders from customers, and financial conditions. Traditional GAs, when applied for DOPs, face a big problem: the convergence problem. Though convergence at proper pace and precision toward the optimum is what we expect from GAs for stationary optimization problems, it becomes a significant limitation for traditional GAs in dynamic environments. This is because once converged, GAs are unable to response effectively to changes in the dynamic environment and track the moving optimum with time.

In order to address the convergence problem, i.e., to avoid all individuals in the population converging toward the optimum of a problem, researchers have developed quite a lot diversity approaches into GAs to enhance their performance, such as the random immigrants [6], memory-based immigrants [15], and hypermutation schemes [3, 9] (just to list a few). Besides diversity schemes, another major approach developed for GAs for DOPs is the incorporation of memory mechanisms, where useful information from the current environment is stored and reused when similar environment is observed later [1, 8, 13]. In fact, diversity and memory are two major mechanisms used in biology to adapt organisms in the ever-changing environment in nature. For example, in biology the immune system protects animal body against potentially harmful pathogens. The immune system can produce diverse antigen receptors that reside on the surface of immune cells to detect infinitely diverse pathogens and can retain a *memory* of the pathogens to speed up response when the pathogens invade the body again.

The mechanisms of the immune system have been applied into GAs to deal with dynamic environments. Relevant work can be found in [4, 5]. In [11], Simões and Costa proposed an immune system based genetic algorithm (ISGA) for dynamic environments. ISGA integrates the two main mechanisms of diversity and memory used in the natural immune system into GAs. In the ISGA, a biologically inspired genetic operator called *transformation* is used to create new individuals [10]. Transformation simulates the somatic hypermutation while an immune cell is cloned in the immune system.

In this paper, an *aligned transformation* operator is proposed for ISGAs to deal with dynamic environments. In the aligned transformation, each gene segment records a starting position when it is stored in the gene pool. And when transformation happens involving such a gene segment, it is transformed into an individual at the exact position indicated by its record. In Simões and Costa's ISGA, each individual in the memory is associated with the average fit-

ness of the population when it was stored. This value is used as the affinity measure to retrieve memory individuals to clone new individuals into the population when an environmental change is detected. In this paper, we utilize the re-evaluated fitness value of memory individuals as the measure to retrieve the best memory individual for cloning. Using the DOP generator proposed in [14, 16, 17], a series of DOPs are constructed as the test environments and experiments are carried out to compare the performance of several variants of ISGAs, including the one proposed in this paper, and two standard GAs enhanced with memory and random immigrants respectively. The experimental results validate the efficiency of the proposed transformation operator and corresponding ISGA in dynamic environments.

The rest of this paper is outlined as follows. The next section describes the fundamentals of the biological immune system and the ISGA proposed by Simões and Costa. Section 3 details the ISGA with proposed aligned transformation and other GAs investigated in this paper. Section 4 presents the experimental study and relevant analysis of comparing investigated ISGAs and two GA variants on a series of dynamic test environments. Section 5 concludes this paper with some discussions.

2. RELEVANT BACKGROUND

2.1 The Immune System

The immune system consists of cells, molecules, and organs that act together to defend the body against foreign invaders that may cause disease, such as bacteria, viruses, and fungi. The health of the body depends on the immune system's ability to recognize and eliminate foreign invaders called *antigens* [7]. The human body maintains a large number of immune cells. Some belong to the innate immune system, e.g., the macrophages that function by surrounding, ingesting, and destroying antigens. Those immune cells that belong to the acquired or adaptive immune system are called *lymphocytes*, which circulate through the body. There are two types of lymphocytes, namely *T-cells* and *B-cells*, which cooperate in the immune response with different roles. The T-cells can be separated into *helper T-cells*, which build up the immune response together with B-cells, and *killer T-cells*, which eliminate the antigens detected by the immune system. B-cells can produce and secret antibodies as a response to exogenous organisms.

When an antigen invades the body for the first time, a few B-cells can recognize the invader's peptides and will be activated to proliferate and produce a specific antibody, which can recognize and bind to the antigen. This process is called *primary response*, which is a slow process and is described as follows. When a B-cell is activated, it migrates to a lymph node, where it proliferates producing many short-lived clones through cell division. B-cell cloning is subject to a form of mutation termed *somatic hypermutation*. This increases the chance that the clones have different affinities from the parent to the antigen. The mutated B-cell clones will undergo a differentiation process. Those clones that have low affinity to the antigen will die while those with high affinity will survive and differentiate into *plasma* or *memory B-cells*. Plasma B-cells secrete antibodies that can bind to the antigen, destroying or neutralizing it.

Memory B-cells will retain in the circulation and constitute the "memory" of the immune system: if the same pathogens attack the body again in the future, the adapted memory B-cells can provide a *second response* that is much faster and more efficient than the primary response. In this sense, the immune system is said to learn to recognize specific kinds of antigens in the primary response and memorize the learned result via the memory B-cells.

The immune system can recognize a large number of pathogens. This is because it has a gene library that aggregates modular chunks of genes or gene segments. These gene segments can be recombined to build up diverse antibodies.

2.2 Simões and Costa's ISGA

The mechanisms of diversity and memory in the biological immune system can be integrated into GAs to enhance their performance in dynamic environments. In [11], Simões and Costa proposed an immune system based genetic algorithm for dynamic environments. The ideas of gene libraries, clonal selection with somatic hypermutation and memory B-cells were translated and integrated into the standard GA to deal with dynamic environments. The starting point is to view the environment as the antigen and the changes in the environment as the appearance of different antigens.

In their ISGA, there are two populations of individuals. The first one consists of plasma B-cells individuals that evolve as follows: the individuals with the best match to the optimum (antigen) are selected and cloned. During the cloning phase every individual has a chance of being modified by a mechanism similar to the somatic hypermutation of B-cells. A transformation genetic operator, which was biologically inspired and proposed by them in [10], is used here to create new individuals during the cloning process. The second population is a collection of memory B-cells individuals, which were the best ones at different moments in the past when they belong to the first population. Each individual of the second population has an attached value which corresponds to the average fitness of the first population in some particular situation, when that individual had good performance and was stored in the second population. This attached value will be used as the affinity measurement to match memory B-cells to a different environment (antigen).

The ISGA uses an environmental change detection mechanism by observing the degradation of population average fitness. When a change is detected, there may exist one individual in the second population that has proximity with the new environment. The proximity is measured by the average fitness of the first population of plasma B-cells and the value attached to the memory B-cells. The most suitable memory B-cells is then activated, cloned and reintroduced into the plasma B-cells population, replacing the worst ones.

The ISGA uses a set of gene libraries, each containing a set of gene segments of fixed size. The libraries are randomly initialized and then kept constant during the entire evolutionary process. They are used in the transformation process, as described in the next subsection.

2.3 The Transformation Operator

In nature, some bacteria can absorb small pieces of DNA or gene segments from the surrounding environment and re-integrate them into its own genetic material. These bacteria are called competent and the re-integration of gene segments may give them evolutionary advantages. In [10], Simões and Costa proposed a transformation operator to simulate the somatic hypermutation while an immune cell is cloned in

the immune system. The transformation operator was used as the main genetic operator in the GA instead of crossover and has shown to outperform traditional crossover operators for GAs for some stationary problems in [10]. The transformation process is described as follows.

The ISGA starts from an intital population of individuals (plasma B-cells) and an initial pool of gene libraries, both randomly created. Every generation individuals are selected to form the mating pool. Each individual in the mating pool will be transformed with a transformation probability p_t. If an individual is subject to transformation, it is transformed as follows. First, one gene segment is randomly selected from one random gene library. Then, a transformation locus is randomly chosen in the individual. Finally, the selected gene segment is incorporated in the genome of the individual, replacing the genes after the transformation locus.

3. GENETIC ALGORITHMS STUDIED

3.1 ISGA with Aligned Transformation

In Simões and Costa's transformation process, a selected gene segment is transferred into an individual at a random locus. This transformation has an effect of implanting a leg to replace an arm of a body. This leg-to-arm transformation may be beneficial due to the diversity introduced. A legged arm may be what is exactly required by the environment. In order to test whether an arm-to-arm transformation is better than the leg-to-arm transformation, this paper proposes an *aligned transformation* operator for ISGAs to deal with dynamic environments. The corresponding ISGA is called *ISGA with aligned transformation*, denoted *ISGAa* in this paper. The pseudo-code of ISGAa as well as other ISGAs studied in this paper is shown in Figure 1[1].

In ISGAa, the gene pool consists of a set of N gene segments, each with a fixed length L. Each gene segment consists of a trinary record $< random, \ locus, \ string >$. The first element *random* represents whether the gene segment is a random one or not, which will be further explained below. The second element *locus* records the starting locus when the gene segment is stored in the gene pool. The third elements *string* represent the real bitstring of the gene segment respectively. The N gene segments are grouped into two groups: one random group that consists of $p_r * N$ random gene segments and one non-random group that includes $p_n * N$ gene segments that are retrieved from the population.

The gene pool is randomly initialized and is updated every generation as follows. All the gene segments in the random group are replaced by new random ones while a part of gene segments in the non-random group, say $p_u * N$ ($p_u \le p_n$) gene segments, are updated according to the current population. The updating is implemented as follows. First, a gene segment is randomly selected from the non-random group to be updated. Then, an individual from the population is selected using a standard tournament selection with a tournament size of 2. Finally, the selected gene segment is replaced with a random segment of length L from the selected individual and the starting locus is recorded as the new *locus* of the gene segment.

Each time an individual, be it a plasma B-cell or memory

<hr>

[1]In Simões and Costa's original ISGA [10], mutation is not used. In this paper, mutation is switched on since it proves beneficial according to our preliminary experiments.

```
t := 0
initialize population P(0) randomly
initialize gene pool G(0) and memory M(0) randomly
evaluate population P(0) and memory M(0)
repeat
   P'(t) := selectForReproduction(P(t))
   clone(P'(t), G(t), p_t)   // p_t is the transformation prob.
   mutate(P'(t), p_m)        // p_m is the mutation prob.
   evaluate the interim population P'(t)
   evaluate memory M(t)

   if environmental change detected  then
      retrieve the best memory point B_M(t)
      clone r_c * n new individuals from B_M(t)
      replace the worst individuals in P'(t) by the clones

   if time to update memory  then
      update memory M(t)

   if not a constant gene pool  then
      update gene pool G(t)    // for ISGAd and ISGAa
   P(t + 1) := P'(t)
until terminated = true        // e.g., t > t_max
```

Figure 1: Pseudo-code of investigated ISGAs.

B-cell, is cloned, it undergoes transformation with a probability p_t as below. First, a gene segment is randomly selected from the gene pool. Then, we check whether it is from the random group or non-random group. If the gene segment is from the non-random group, it is transformed into the individual under consideration from the exact locus indicated by the variable *locus* in its record, i.e., aligned transformation. Otherwise, for a random group gene segment, the starting locus is depressed in the transformation process and it is transformed just as in Simões and Costa's transformation process. That is, the gene segment will be transformed into the recipient individual from a random locus.

ISGAa (and other GAs with memory studied in this paper) uses a memory of size $m = 0.1 * n$ where n is the size of the total population, including the plasma B-cells and memory B-cells. The memory is randomly initialized and evaluated every generation. The environment is detected as changed if at least one individual in the memory has been detected having changed its fitness. If an environment change is detected, the best memory point is retrieved and cloned to create $r_c * n$ (r_c is called the *clone ratio*) individuals to replace the worst ones in the population. The memory is updated in a stochastic time pattern: suppose that a memory updating happens at generation t, then the next memory updating time is $t_M = t + rand(5, 10)$. When the memory is due to update, if any of the randomly initialized points still exists in the memory, the best individual of the population will replace one of them randomly; otherwise, it will replace the closest memory point measured by Hamming distance if it is better than the closest memory point.

3.2 Other GAs Investigated

In addition to ISGAa, two other ISGA variants, denoted *ISGAc* and *ISGAd* respectively, are also investigated in this paper. ISGAc is similar to Simões and Costa's original ISGA: the gene pool is randomly initialized and then kept

constant during the run of ISGAc, and the average population fitness is attached with memory B-cells and used as affinity measure to retrieve memory point for cloning. One different thing is that ISGAc uses the classical mutation while Simões and Costa's ISGA does not.

ISGAd is similar to ISGAa but Simões and Costa's transformation is used instead of the proposed aligned transformation in the cloning process. The memory updating and retrieving mechanisms in ISGAd are the same as in ISGAa. As in ISGAa, the gene pool in ISGAd is *dynamic*, i.e., updated every generation. However, the difference lies in that in ISGAd all gene segments in the non-random group are updated according to the current population.

In order to verify the effect of using memory for ISGAs, we also study a variant of Simões and Costa's Transformation Based GA [12], denoted $TBGA^2$. TBGA is similar to ISGAd except that no memory is used.

Two standard GAs, the memory-enhanced GA (denoted *MEGA*) [1] and random immigrants GA (denoted *RIGA*) [6], are also investigated as peer GAs in this paper. For MEGA, the memory updating scheme is the same as in ISGAa. The memory is also evaluated every generation. When a change is detected, the memory is merged with the current population and the best individuals are selected as the new population while the memory remains unchanged. In RIGA, every generation $r_i * n$ random individuals are generated and replace the worst members in the current population, where r_i is the immigrants ratio.

4. EXPERIMENTAL STUDY

4.1 Dynamic Test Environments

The DOP generator proposed in [14, 16] can construct *random* dynamic environments from any binary-encoded stationary function $f(\vec{x})$ ($\vec{x} \in \{0,1\}^l$) by a bitwise exclusive-or (XOR) operator. We suppose the environment changes every τ generations. For each environment k, an XOR mask $\vec{M}(k)$ is incrementally generated as follows:

$$\vec{M}(k) = \vec{M}(k-1) \oplus \vec{T}(k), \qquad (1)$$

where "$\oplus$" is the XOR operator and $\vec{T}(k)$ is an intermediate binary template randomly created with $\rho \times l$ ones for environment k. For the first environment $k = 1$, $\vec{M}(1) = \vec{0}$.

An individual at generation t is evaluated as follows:

$$f(\vec{x}, t) = f(\vec{x} \oplus \vec{M}(k)), \qquad (2)$$

where $k = \lceil t/\tau \rceil$ is the environment index at time t. With this XOR generator, parameters τ and ρ control the speed and severity of environmental changes respectively. Smaller τ means faster changes while bigger ρ means severer changes.

The above generator has been extended to construct *cyclic* dynamic environments [17] as follows. First, $2K$ XOR masks $\vec{M}(0), \cdots, \vec{M}(2K-1)$ are constructed as the *base states* in the search space randomly. Then, the environment can change among these base states cyclically. Suppose the environment changes every τ generations, the individuals at generation t is evaluated as follows:

$$f(\vec{x}, t) = f(\vec{x} \oplus \vec{M}(I_t)) = f(\vec{x} \oplus \vec{M}(k\%(2K))), \quad (3)$$

^{2}In Simões and Costa's TBGA, how the gene pool is updated according to the current population was not stated. In this study, tournament selection of size 2 is used to select individuals from the population to update the gene pool.
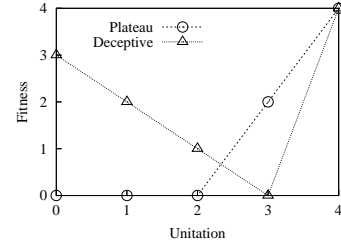

Figure 2: Building block of the stationary functions.

where $k = \lfloor t/\tau \rfloor$ is the index of current environment and $I_t = k\%(2K)$ is the index of the base state the environment is in at generation t, calculated by a $2K$ modulo division.

The $2K$ XOR masks can be generated as follows. First, we construct K binary templates $\vec{T}(0), \cdots, \vec{T}(K-1)$ that form a partition of the search space with each containing randomly but exclusively selected $\rho \times l = l/K$ bits of ones. Let $\vec{M}(0) = \vec{0}$ denote the initial state, the other XOR masks are generated iteratively as:

$$\vec{M}(i+1) = \vec{M}(i) \oplus \vec{T}(i\%K), i = 0, \cdots, 2K-1 \qquad (4)$$

The templates $\vec{T}(0), \cdots, \vec{T}(K-1)$ are first used to create K masks till $\vec{M}(K) = \vec{1}$ and then orderly reused to construct another K XOR masks till $\vec{M}(2K) = \vec{M}(0) = \vec{0}$. The Hamming distance between two neighbour XOR masks is the same and equals $\rho \times l$. Here, $\rho \in [1/l, 1.0]$ is the distance factor, determining the number of base states.

Two 100-bit binary functions, called *plateau* and *deceptive*, are selected as base stationary functions to construct dynamic test environments for this study. Both functions consist of 25 contiguous 4-bit building blocks (BBs) and have an optimum fitness of 100. Each BB is defined based on its unition function, i.e., the number of ones inside the BB, as shown in Figure 2. The BB for *plateau* contributes 4 (or 2) to the total fitness if its unition is 4 (or 3), otherwise, it contributes 0. The BB for *deceptive* is fully deceptive.

Two kinds of dynamic environments, random and cyclic, are constructed from each of the two base functions using the DOP generator mentioned above. For each DOP, the environment is periodically changed every τ generations during the run of a GA. The parameters τ is set to 10, 25 and 50 and ρ is set to 0.1, 0.2, 0.5, and 1.0 respectively.

Totally, a series of 24 DOPs, three τ's times four ρ's under two kinds of dynamic environments, are constructed from each stationary function.

4.2 Experimental Design

For all GAs, parameters are set as follows: generational, uniform crossover with the crossover probability $p_c = 0.6$, bit flip mutation with probability $p_m = 0.01$, fitness proportional selection, and elitism of size 1. The population size n, including the memory if memory is used, is set to 100 and the memory size is $m = 0.1 * n = 10$ if used. For all ISGAs and TBGA, the transformation probability p_t is set to 0.7 and the gene pool has $N = 200$ gene segments of fixed length $L = 5$. For ISGAd, ISGAa and TBGA, 60 gene segments form the random group while other gene segments belong to the non-random group, i.e., $p_r = 0.3$ and $p_n = 0.7$. For ISGAa, ISGAd and TBGA, the number of gene segments in the non-random group that are updated every generation by
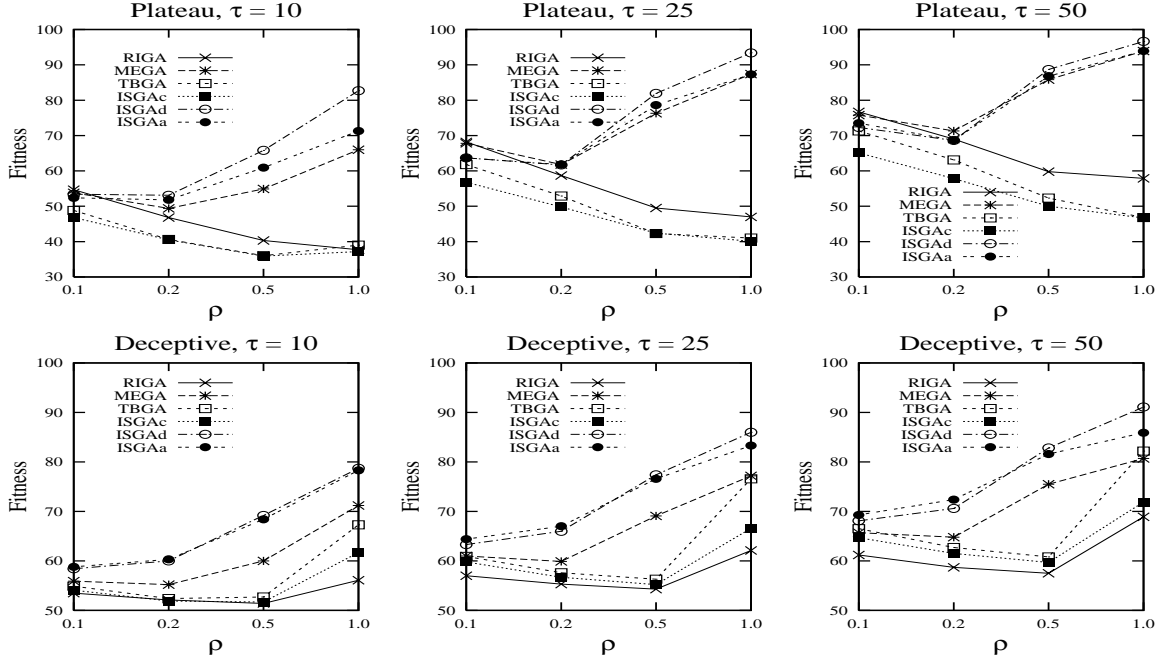
Figure 3: Experimental results of GAs on cyclic DOPs.

the current population are set to 40, 140, and 140 respectively, i.e., $p_u = 0.2$, 0.7, 0.7 respectively. For all ISGAs, the clone ratio r_c is set to 0.2 in order to compare with RIGA where the immigrants ratio r_i is also set to 0.2.

For each experiment of a GA on a DOP, 50 independent runs were executed with the same set of random seeds. For each run, 50 environmental changes were allowed and the best-of-generation fitness was recorded every generation. The overall performance of a GA on a DOP is defined as:

$$\overline{F}_{BOG} = \frac{1}{G} \sum_{i=1}^{G} \left(\frac{1}{N} \sum_{j=1}^{N} F_{BOG_{ij}} \right), \qquad (5)$$

where $G = 50 * \tau$ is the total generations for a run, $N = 50$ is the total runs, and $F_{BOG_{ij}}$ is the best-of-generation fitness of generation i of run j. The off-line performance $\overline{F}_{BOG}$ is the best-of-generation fitness averaged over 50 runs and then averaged over the data gathering period.

4.3 Experimental Results and Analysis

The experimental results of comparing GAs on the constructed cyclic and random DOPs are plotted in Figure 3 and 4 respectively. The statistical results of comparing GAs by one-tailed t-test with 98 degrees of freedom at a 0.05 level of significance are given in Table 1. In Table 1, the t-test result as to $Alg.\ 1 - Alg.\ 2$ is shown as "+", "−", "++", or "−−" when $Alg.\ 1$ is better than, worse than, significantly better than, or significantly worse than $Alg.\ 2$ respectively.

In order to better understand the performance of investigated GAs in dynamic environments, the dynamic behaviour of GAs with respect to best-of-generation fitness against generations on the DOPs with $\tau = 25$ and $\rho = 0.1$ and $\rho = 1.0$ is plotted in Figure 5. In Figure 5, the first 20 environmental changes (i.e., the first 500 generations) are shown and the data were averaged over 50 runs. From the tables and figures several results can be observed.

First, both ISGAa and ISGAd significantly outperform TBGA on almost all DOPs, the t-test results not shown. This result validates the benefit of combing memory scheme into ISGAs. The advantage is much more obvious in cyclic environments, see Figure 3.

Second, both ISGAa and ISGAd significantly outperform ISGAc on almost all DOPs, see the t-test results regarding $ISGAa - ISGAc$ and $ISGAd - ISGAc$ in Table 1. This result happens due to two factors: the affinity measure to retrieve the memory for clones when change occurs and the updating of gene pool. The affinity measurement by the average population fitness used in ISGAc can not guarantee to select good memory B-cell to create clones into the population. The effect can be observed from the dynamic behaviour of ISGAs on DOPs with $\rho = 1.0$ in Figure 5(e) and 5(f), where each time when change occurs, for ISGAa and ISGAd (and MEGA), the retrieved best memory point can quickly draw their performance back to a high level while for ISGAc the performance drops significantly because the affinity measurement fails to dig out the required memory B-cell. As to the gene pool contents factor, it can be observed from the comparison of ISGAc and TBGA. ISGAc is beaten by TBGA on most DOPs, see the t-test results in Table 1. This shows the advantage of dynamic gene pool over constant gene pool.

Third, both ISGAa and ISGAd outperform MEGA on most cyclic DOPs and on most random deceptive problems, see the t-test results regarding $ISGAa - MEGA$ in Table 1. This happens because the memory scheme in ISGAa and ISGAd is stronger than that in MEGA. In MEGA, when change occurs the memory is simply merged with the population. Many memory points in fact may be quite widely distributed in the new landscape and have low fitness. On the contrast, in ISGAa and ISGAd the best re-evaluated memory solution is retrieved to clone new individuals. Hence,
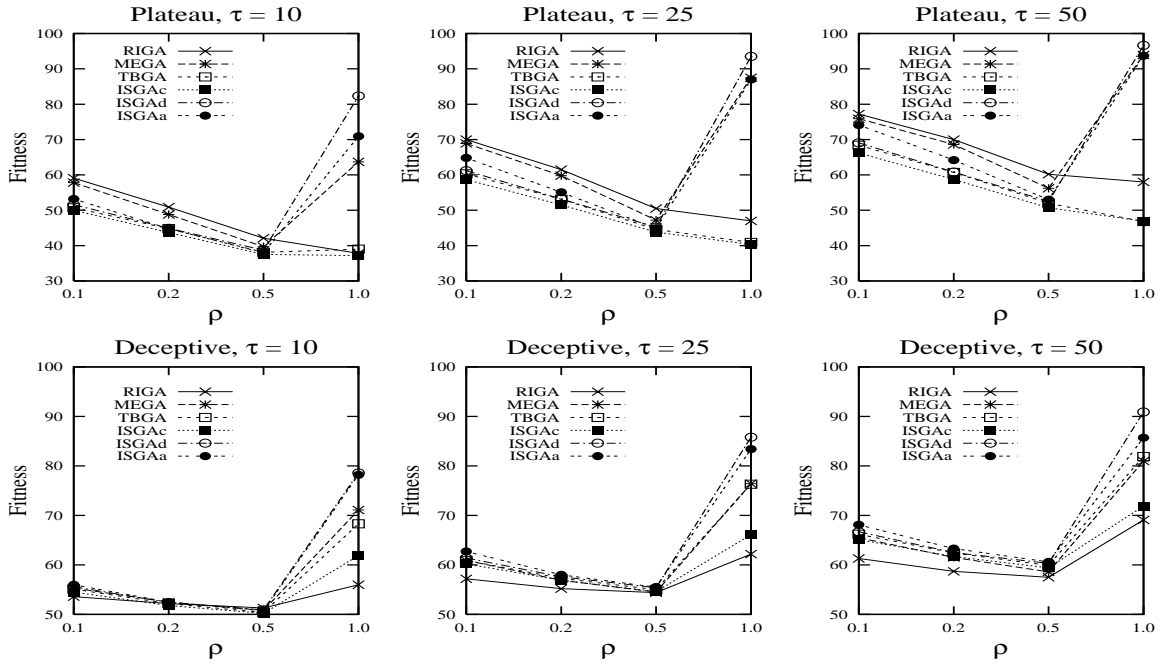
Figure 4: Experimental results of GAs on random DOPs.

Table 1: The t-test results of comparing GAs on cyclic DOPs.

Environment	Cyclic								Random							
t-test Result	*Plateau*				*Deceptive*				*Plateau*				*Deceptive*			
$\tau = 10,\ \rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
$ISGAa - RIGA$	−−	++	++	++	++	++	++	++	−−	−−	−−	++	++	++	−−	++
$ISGAa - MEGA$	−−	++	++	++	++	++	++	++	−−	−−	−−	++	++	+	−−	++
$TBGA - ISGAc$	++	+	+	++	++	++	++	++	++	++	++	++	++	++	++	++
$ISGAa - ISGAc$	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++
$ISGAd - ISGAc$	++	++	++	++	++	++	++	++	++	+	++	++	++	++	++	++
$ISGAa - ISGAd$	−−	−−	−−	−−	++	+	−−	−	++	−	−−	−−	++	+	−−	−
$\tau = 25,\ \rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
$ISGAa - RIGA$	−−	++	++	++	++	++	++	++	−−	−−	−−	++	++	++	++	++
$ISGAa - MEGA$	−−	+	++	−	++	++	++	++	−−	−−	−−	−	++	++	++	++
$TBGA - ISGAc$	++	++	−	++	++	++	++	++	++	++	++	++	++	++	++	++
$ISGAa - ISGAc$	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++
$ISGAd - ISGAc$	++	++	++	++	++	++	++	++	++	+	++	++	++	++	++	++
$ISGAa - ISGAd$	+	+	−−	−−	++	++	−−	−−	++	++	−−	−−	++	++	+	−−
$\tau = 50,\ \rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
$ISGAa - RIGA$	−−	−−	++	++	++	++	++	++	−−	−−	−−	++	++	++	++	++
$ISGAa - MEGA$	−−	−−	++	−	++	++	++	++	−−	−−	−−	−	++	++	++	++
$TBGA - ISGAc$	++	++	++	+	++	++	++	++	++	++	++	−	++	++	++	++
$ISGAa - ISGAc$	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++
$ISGAd - ISGAc$	++	++	++	++	++	++	++	++	++	+	++	++	++	+	++	++
$ISGAa - ISGAd$	++	+	−−	−−	++	++	−−	−−	++	++	++	−−	++	++	++	−−

the clones are more related to the new environment than the memory points in MEGA are. However, on random plateau problems the wide distribution of the memory points may give MEGA more adaptability to the new environment. But, on random deceptive DOPs, the potential diversity of the memory points in MEGA seems harmful due to the property of the deceptive function, which makes MEGA lose to ISGAa and ISGAd. This result can be further observed in Figure 5: on deceptive problems, ISGAa and ISGAd can climb to a higher fitness level that MEGA can do during each dynamic environment.

Fourth, both ISGAa and ISGAd outperform RIGA on most cyclic DOPs and random deceptive DOPs but are beaten by RIGA on random plateau DOPs, see the t-test results regarding $ISGAa - RIGA$ in Table 1. The result regarding cyclic DOPs is easy to understand. For cyclic DOPs the memory scheme usually works better than random immigrants scheme, see Figure 5(e) and 5(f). On random plateau DOPs, the diversity introduced by random immigrants is beneficial which gives RIGA an advantage over ISGAa and ISGAd, see Figure 5(c).

Finally, comparing ISGAa and ISGAd, it can be seen that neither of them is a clear winner over the other on all tested DOPs, see the t-test results regarding $ISGAa - ISGAd$. It is interesting to see that ISGAa dominates on DOPs with smaller ρ, e.g., 0.1 or 0.2, while ISGAd dominates on DOPs with bigger ρ. This can be further observed by comparing their dynamic behaviour in Figure 5(a) and 5(b) over 5(e)
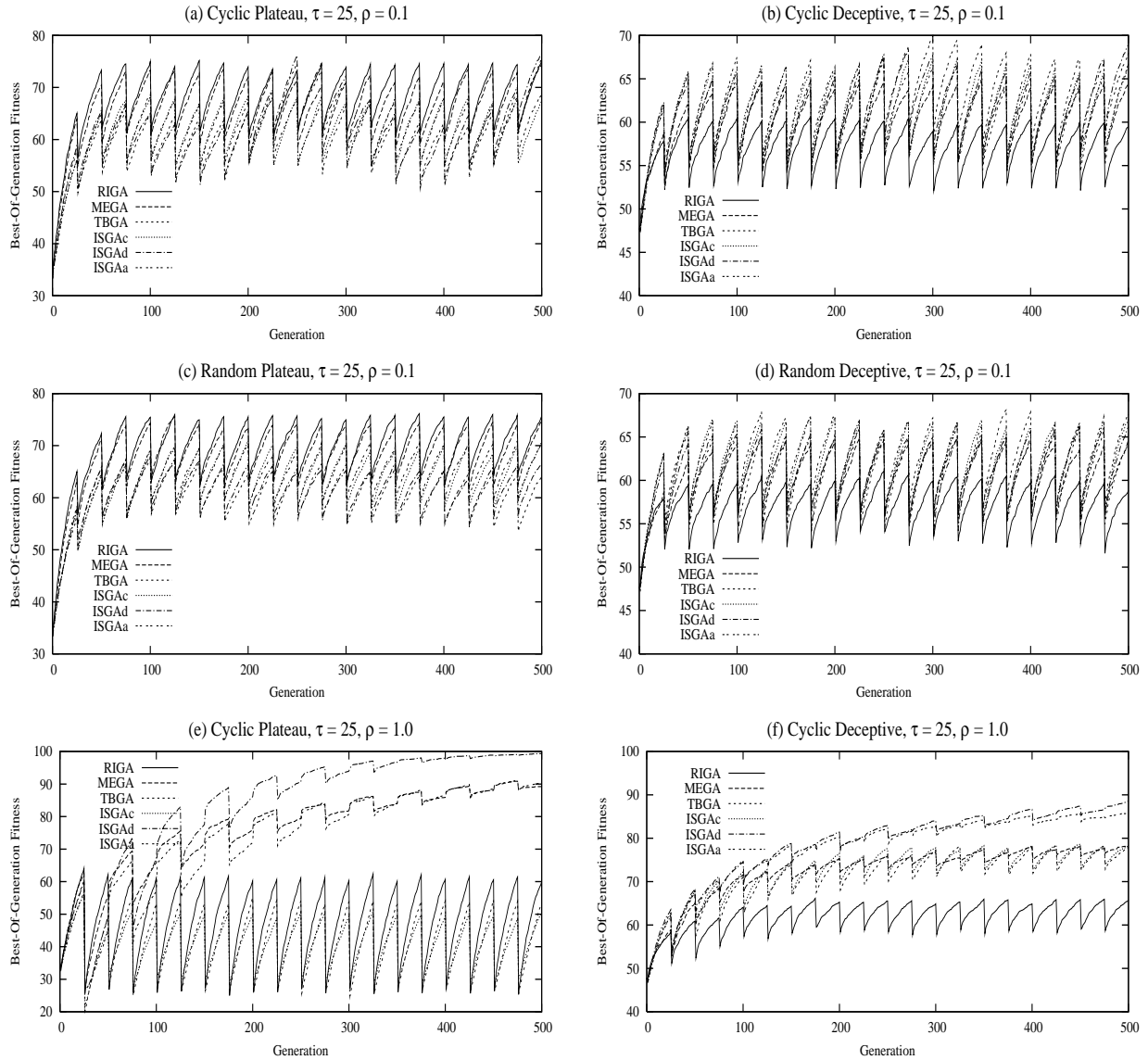
Figure 5: **Dynamic behaviour of GAs on DOPs with** $\tau = 25$ **and** $\rho = 0.1$ **and** 1.0.

and 5(f). This happens because significant environmental changes require a higher population diversity level to follow the moving optimum, which gives the leg-to-arm transformation an advantage. However, in slightly changing environments, when a change occurs, a stronger arm other than a leg is required to replace an old weakened arm. This leads to the better performance of ISGAa than ISGAd.

4.4 Sensitivity Analysis of the Transformation Probability on ISGAa

In order to study the effect of the transformation Probability p_t on the performance of ISGAa for DOPs, we further carried out experiments on ISGAa with different settings, 0.3, 0.5, and 0.9, for p_t on the dynamic test problems. The experimental design is the same as previous one for ISGAa except p_t. The experimental results are plotted in Figure 6 where the data were obtained and averaged over 50 runs and the ISGAa with p_t is marked as p_t-ISGAa accordingly.

From Figure 6, it can be seen that p_t does affect ISGAa's performance on some DOPs with small values of ρ and it seems ISGAa's performance rises with the value of p_t. For example, on cyclic plateau problem with $\tau = 25$ and $\rho = 0.1$, when p_t rises from 0.3 to 0.5, the performance of ISGAa improves from $\overline{F}_{BOG}(0.5\text{-ISGAa}) = 63.2$ to 64.1. When p_t is raised to 0.7, ISGAa's performance is further improved to 64.8 and when $p_t = 0.9$ the performance improves to 65.1. From Figure 6, it can also be seen that p_t has more sensitive effect on ISGAa on *plateau* DOPs than on *deceptive* DOPs.

5. CONCLUSIONS

The mechanisms of immune system and transformation of genetic materials improve organisms' adaptability in natural dynamic environments and can be borrowed into GAs to enhance their performance in dynamic environments. This paper proposes an aligned transformation operator and combines it to the immune system based genetic algorithm to
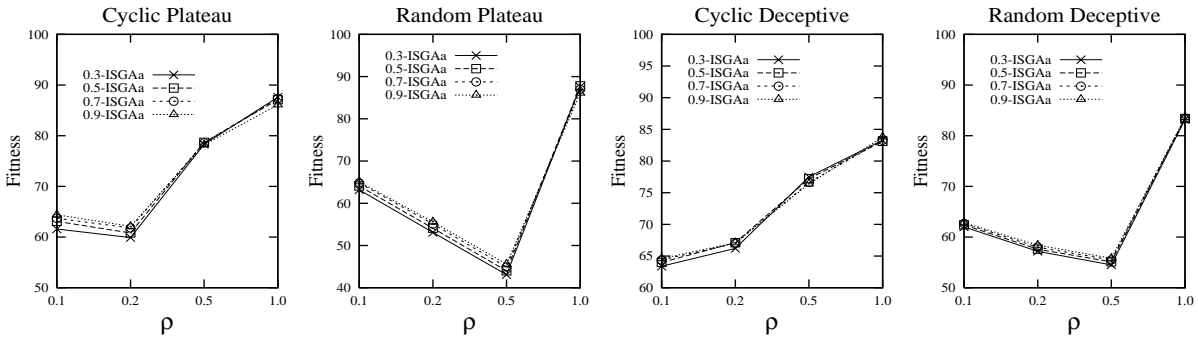
Figure 6: Experimental results of ISGAa with different transformation rates on DOPs with $\tau = 25$.

deal with dynamic environments. Based on a series of systematically constructed dynamic environments, experiments were carried out to compare several variants of immune system based genetic algorithms and traditional memory and random immigrants enhanced GAs. The experimental results validate the efficiency of the proposed transformation operator and corresponding immune system based GA in dynamic environments. From the experimental results and relevant analysis, several conclusions can be drawn on the dynamic test environments.

First, using gene pool that is dynamically updated according to the current population is better than a constant gene pool for ISGAs in dynamic environments. Second, the proposed aligned transformation works well for ISGAs, especially when the environmental changes are not very severe. When the environment involves severe changes it is better to use ISGA with Simões and Costa's transformation operator. Third, the parameter sensitivity analysis shows that the transformation probability p_t does affect the performance of ISGAa on some DOPs and setting p_t in the range of $[0.7, 0.9]$ seems a good choice for ISGAa. Finally, random immigrants improve GA's performance on random non-deceptive DOPs while work poorly on deceptive DOPs and on cyclic DOPs.

In this paper, some preliminary experiments were carried out, comparing several ISGAs for DOPs. Further comparing the investigated ISGAs with other approaches for GAs in dynamic environments is now under investigation. It is also an interesting future work to develop other advanced transformation schemes for ISGAs for DOPs. We believe that developing adaptive transformation schemes that respect the changing environment will further improve the performance of GAs in dynamic environments.

6. REFERENCES

[1] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. *Proc. of the 1999 Congr. on Evol. Comput.*, vol. 3, pp. 1875-1882, 1999.

[2] J. Branke. *Evolutionary Optimization in Dynamic Environments.* Kluwer Academic Publishers, 2002.

[3] H. G. Cobb and J. J. Grefenstette. Genetic algorithms for tracking changing environments. *Proc. of the 5th Int. Conf. on Genetic Algorithms*, pp. 523–530, 1993.

[4] A. Gaspar and P. Collard. From GAs to artificial immune systems: Improving adaptation in time dependent optimization. *Proc. of the 1999 Congr. on Evol. Comput.*, vol. 3, pp. 1859–1866, 1999.

[5] A. Gaspar and P. Collard. Two models of immunization for time dependent optimization. *Proc. of the 2000 IEEE Int. Conf. on SMC*, 2000.

[6] J. J. Grefenstette. Genetic algorithms for changing environments. *Proc. of the 2nd Int. Conf. on Parallel Problem Solving from Nature*, pp. 137-144, 1992.

[7] D. L. Hartl and E. W. Jones. *Genetics: Principles and Analysis.* Jones and Bartllet Publishers, Inc., 1998.

[8] N. Mori, H. Kita and Y. Nishikawa. Adaptation to changing environments by means of the memory based thermodynamical genetic algorithm. *Proc. of the 7th Int. Conf. on Genetic Algorithms*, pp. 299-306, 1997.

[9] R. W. Morrison and K. A. De Jong. Triggered hypermutation revisited. *Proc. of the 2000 Congress on Evol. Comput.*, pp. 1025–1032, 2000.

[10] A. Simões and E. Costa. On biologically inspired genetic operators: Using transformation in the standard genetic algorithm *Proc. of the 2001 Genetic and Evol. Comput. Conf.*, pp. 584-591, 2001.

[11] A. Simões and E. Costa. An immune system-based genetic algorithm to deal with dynamic environments: diversity and memory. *Proc. of the 6th Int. Conf. on Neural Networks and Genetic Algs.*, pp. 168-174, 2003.

[12] A. Simões and E. Costa. Improving the genetic algorithm's performance when using transformation. *Proc. of the 6th Int. Conf. on Neural Networks and Genetic Algs.*, pp. 175-181, 2003.

[13] K. Trojanowski and Z. Michalewicz. Searching for optima in non-stationary environments. *Proc. of the 1999 Congress on Evol. Comput.*, pp. 1843-1850, 1999.

[14] S. Yang. Non-stationary problem optimization using the primal-dual genetic algorithm. *Proc. of the 2003 Congress on Evolutionary Computation*, vol. 3, pp. 2246-2253, 2003.

[15] S. Yang. Memory-based immigrants for genetic algorithms in dynamic environments. *Proc. of the 2005 Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1115-1122, 2005.

[16] S. Yang and X. Yao. Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft Computing*, vol. 9, no. 11, pp. 815-834, 2005.

[17] S. Yang and X. Yao. Population-based incremental learning with associative memory for dynamic environments. Submitted to *IEEE Trans. on Evolutionary Computation*, 2005.