

# A Hybrid Immigrants Scheme for Genetic Algorithms in Dynamic Environments

Shengxiang Yang<sup>1,\*</sup> Renato Tinós<sup>2</sup>

<sup>1</sup> Department of Computer Science, University of Leicester, University Road, Leicester LE1 7RH, United Kingdom

<sup>2</sup> Departamento de Física e Matemática, FFCLRP, Universidade de São Paulo (USP), Ribeirão Preto, 14040-901, Brazil

---

**Abstract:** Dynamic optimization problems are a kind of optimization problems that involve changes over time. They pose a serious challenge to traditional optimization methods as well as conventional genetic algorithms since the goal is no longer to search for the optimal solution(s) of a fixed problem but to track the moving optimum over time. Dynamic optimization problems have attracted a growing interest from the genetic algorithm community in recent years. Several approaches have been developed to enhance the performance of genetic algorithms in dynamic environments. One approach is to maintain the diversity of the population via random immigrants. This paper proposes a hybrid immigrants scheme that combines the concepts of elitism, dualism and random immigrants for genetic algorithms to address dynamic optimization problems. In this hybrid scheme, the best individual, i.e., the elite, from the previous generation and its dual individual are retrieved as the bases to create immigrants via traditional mutation scheme. These elitism-based and dualism-based immigrants together with some random immigrants are substituted into the current population, replacing the worst individuals in the population. These three kinds of immigrants aim to address environmental changes of slight, medium and significant degrees respectively and hence efficiently adapt genetic algorithms to dynamic environments that are subject to different severities of changes. Based on a series of systematically constructed dynamic test problems, experiments are carried out to investigate the performance of genetic algorithms with the hybrid immigrants scheme and traditional random immigrants scheme. Experimental results validate the efficiency of the proposed hybrid immigrants scheme for improving the performance of genetic algorithms in dynamic environments.

**Keywords:** Genetic algorithms, random immigrants, elitism-based immigrants, dualism, dynamic optimization problems.

---

## 1 Introduction

Genetic algorithms (GAs) are a class of stochastic meta-heuristic optimization methods that model the biological principles of Darwinian theory of evolution and Mendelian principles of inheritance [1, 2]. Due to their ease of use, GAs have been widely applied for solving many optimization problems with promising results. Most of the optimization problems studied so far by the GA community are stationary problems where it is assumed that the problems are given in advance and are not subject to changes over time. However, many real world problems are dynamic optimization problems (DOPs) where change may occur over time with respect to all aspects of the problem being solved. For example, the problem-specific fitness evaluation function and constraints, such as the design variables and environmental conditions, may change over time due to all kinds of reasons, e.g., machine breakdown, resource availability and economic factors.

Addressing DOPs has been a challenging task for the GA community since the early days. There were some preliminary works appeared in this area in the early days of GA research [3]. However, due to the difficulty of DOPs, only in recent years, with the growing interest of studying GAs for real world problems, there has been a growing interest in studying GAs for DOPs [4–7].

For stationary optimization problems, our goal is to develop GAs that can quickly and precisely locate the op-

tima of the fitness landscape. However, for DOPs quickly and precisely locating the optimum solution(s) of a snapshot optimization problem is no longer the unique goal. Instead, tracking the changing environment becomes a more important issue. This challenges traditional GAs due to the convergence problem because, once converged, GAs cannot adapt well to the changing environment.

The simplest approach to deal with DOPs is to re-start GAs whenever a change in the problem is detected. This re-optimization process, however, generally requires substantial computational time and effort. If the new problem is related to the previous one, which is usually true, knowledge obtained during the search for the solution for the old problem can be used to search for a new solution for the new problem in order to save processing time [8]. Over the years, several approaches have been developed into GAs to address DOPs, such as maintaining diversity via random immigrants [9, 10], increasing diversity after a change via hypermutation [11, 12], memory schemes [13–18] and multi-population and species schemes [19, 20].

Among the approaches devised for GAs for DOPs, the random immigrants scheme has proved beneficial for GAs for many DOPs. It works by maintaining the diversity of the population by replacing individuals from the population with randomly created individuals. Recently, a hybrid random immigrants and elitism approach, called *elitism-based immigrants*, has been proposed for GAs in dynamic environments with some promising results [21]. In the elitism-based immigrants scheme, the elite from the previous generation is retrieved as the base to create immigrants via mutation to replace the worst individuals in the current population.

---

Manuscript received March 5, 2007; revised date May 13, 2007

This work was supported by UK EPSRC (No. EP/E060722/01) and Brazil FAPESP (Proc. 04/04289-6).

\*Corresponding author. E-mail address: s.yang@mcs.le.ac.uk

This paper further investigates the performance of the elitism-based immigrants scheme for GAs in dynamic environments. Inspired by the dualism and complementary principle in nature, this paper also proposes a hybrid immigrants scheme, which combines the elitism-based immigrants scheme, the traditional random immigrants scheme, and dualism, for GAs to deal with DOPs. In this hybrid immigrants scheme, the best individual, i.e., the elite, from the previous generation and its dual individual are retrieved as the bases to create immigrants via the traditional mutation scheme. These elitism-based and dualism-based immigrants together with some random immigrants are substituted into the current population, replacing the worst individuals in the population. These three kinds of immigrants aim to address environmental changes of slight, medium and significant degrees respectively and hence efficiently adapt GAs to dynamic environments that may be subject to different severities of changes.

Based on the DOP generator proposed in [22, 23], a series of dynamic test problems are constructed from several stationary functions and experiments are carried out to compare the performance of several GA variants with different immigrants schemes. Based on the experimental results, we analyze the performance of GAs regarding the weakness and strength of different immigrants schemes for GAs in dynamic environments. The experiment results show that the proposed hybrid immigrants scheme efficiently improves the performance of GAs in dynamic environments.

The rest of this paper is organized as follows. Section 2 briefly reviews the traditional random immigrants scheme for GAs in dynamic environments. Section 3 presents the proposed hybrid immigrants scheme for GAs in dynamic environments. The dynamic test environments used for this study are described in section 4. The experimental results and relevant analysis are presented in section 5. Finally, section 6 concludes this paper with discussions on relevant future work.

## 2 The Random Immigrants Scheme

The standard GA, denoted *SGA* in this paper, maintains and evolves a population of candidate solutions through selection and variation. New populations are generated by first selecting relatively fitter individuals from the current population and then recombining them via crossover and mutation to create new off-spring. This process continues until some stop condition is met, e.g., the maximum allowable number of generations  $t_{max}$  is reached. The pseudo-code for SGA is shown as follows, where  $p_c$  and  $p_m$  are the probability of crossover and mutation respectively.

Pseudo-code for SGA and RIGA. Here, elitism of size one is used in both SGA and RIGA

```

begin
   $t := 0$  and initializePopulation( $P(0)$ ,  $n$ )
  evaluatePopulation( $P(0)$ )
  repeat
     $t := t + 1$ 
     $E(t-1) := \text{extractElite}(P(t-1))$ 
    replace the worst individual in  $P(t)$  by  $E(t-1)$ 
    if random immigrants used then // for RIGA
      generate a set  $S_{ri}(t)$  of  $n_i$  random immigrants

```

```

    evaluateImmigrants( $S_{ri}(t)$ )
    replace the worst individuals in  $P(t)$  with the
      generated immigrants in  $S_{ri}(t)$ 
  end if
   $P'(t) := \text{selectForReproduction}(P(t))$ 
  crossover( $P'(t)$ ,  $p_c$ )
  mutate( $P'(t)$ ,  $p_m$ )
  evaluatePopulation( $P'(t)$ )
   $P(t+1) := P'(t)$ 
until a stop condition is met // e.g.,  $t > t_{max}$ 
end

```

Usually, with the iteration of SGA, individuals in the population will eventually converge to optimum or near optimum solutions in stationary environments due to the selection pressure. For stationary optimization problems, the convergence of the population, at a proper pace instead of pre-mature, is usually beneficial for GAs to search for optimum or near optimum solutions. However, convergence becomes a big problem for GAs in dynamic environments. When a change occurs in the problem, the solution given by GAs before the change may no longer be effective and a new solution should be found. But, because GAs may have already converged when a change occurs, it is hard for them to search for a new solution in the new environment. Hence, dynamic environments require GAs to remain a certain level of population diversity to maintain their adaptability.

In order to avoid convergence, several approaches have been proposed in the literature over recent years. Typical examples are the random immigrants scheme [9], sharing or crowding mechanisms [24], thermodynamic GA [25, 26], and hypermutation [11]. The random immigrants approach is quite simple and natural. It was proposed by Grefenstette with the inspiration from the flux of immigrants that wander in and out of a population between two generations in nature [9]. It maintains the diversity level of the population through substituting a portion of individuals in the current population with randomly generated individuals, called *random immigrants*, every generation. As to which individuals in the population should be replaced, there are two strategies: replacing random individuals or replacing the worst ones [10]. In order to avoid that random immigrants disrupt the ongoing search progress too much, especially during the period when the environment does not change, the ratio of the number of random immigrants  $n_i$  to the population size  $n$  is usually set to a small value.

The pseudo-code of the GA with the traditional random immigrants scheme with replacement of the worst individuals, denoted *RIGA* in this paper, is also shown with the pseudo-code of SGA. RIGA differs from SGA only in that in each generation, a set of  $n_i$  worst individuals in the population are replaced by random immigrants.

## 3 The Hybrid Immigrants Scheme

As discussed above, the traditional random immigrants scheme works by replacing random individuals into the population. This may increase the diversity level of the population and hence benefit the performance of GAs in dynamic environments. However, in a slowly changing environment, the introduced random immigrants may divert the searching force of GAs during each environment before a change occurs and hence may degrade the performance. On

the other hand, if the environment only changes slightly in terms of the severity of changes, random immigrants may have no actual effect even when a change occurs because individuals in the previous environment may still be quite fitter than random immigrants in the new environment.

Based on the above consideration, an elitism-based immigrants scheme has been recently proposed for GAs to address DOPs. The pseudo-code for the GA with the elitism-based immigrants scheme, denoted *EIGA* in this paper, is shown as follows.

Pseudo-code for EIGA, ERIGA, and HIGA

```

begin
   $t := 0$  and initializePopulation( $P(0)$ ,  $n$ )
  evaluatePopulation( $P(0)$ )
  repeat
     $t := t + 1$ 
    // generate elitism-based immigrants
     $E(t-1) := \text{extractElite}(P(t-1))$ 
    generate a set  $S_{ei}(t)$  of  $n_{ei}(t)$  immigrants by
      mutating  $E(t-1)$  with  $p_m^i$ 
    evaluateImmigrants( $S_{ei}(t)$ )

    // generate random immigrants for ERIGA & HIGA
    if random immigrants used then
      generate a set  $S_{ri}(t)$  of  $n_{ri}(t)$  random immigrants
      evaluateImmigrants( $S_{ri}(t)$ )
    end if

    // generate dualism-based immigrants for HIGA
    if dualism based immigrants used then
      generate a set  $S_{di}(t)$  of  $n_{di}(t)$  immigrants by
        mutating the dual of  $E(t-1)$  with  $p_m^{di}$ 
      evaluateImmigrants( $S_{di}(t)$ )
    end if

    replace the worst individuals in  $P(t)$  with the
      generated immigrants in  $S_{ri}(t)$ ,  $S_{ei}(t)$  and  $S_{di}(t)$ 

     $P'(t) := \text{selectForReproduction}(P(t))$ 
    crossover( $P'(t)$ ,  $p_c$ )
    mutate( $P'(t)$ ,  $p_m$ )
    evaluatePopulation( $P'(t)$ )
     $P(t+1) := P'(t)$ 
  until a stop condition is met // e.g.,  $t > t_{max}$ 
end

```

Within EIGA, for each generation  $t$ , before the normal genetic operations (i.e., selection and recombination), the elite  $E(t-1)$  from previous generation is used as the base to create immigrants. From  $E(t-1)$ , a set  $S_{ei}$  of  $n_{ei}(t)$  individuals are iteratively generated by mutating  $E(t-1)$  bitwise with a probability  $p_m^{ei}$ . The generated individuals then act as immigrants and replace the worst individuals in the current population. It can be seen that the elitism-based immigrants scheme combines the idea of elitism with the traditional random immigrants scheme. It uses the elite from the previous population to guide the immigrants toward the current environment. The elitism-based immigrants scheme has shown some promising results in improving the performance of GAs in dynamic environments, especially when the environment involves slight degree of changes [21].

In order to address higher degree of changes, the elitism-based immigrants scheme can be combined with the traditional random immigrants scheme for GAs to deal with DOPs. The pseudo-code for the GA with elitism-based immigrants and random immigrants, denoted *ERIGA*, is also shown with the pseudo-code of EIGA. Within ERIGA, in

addition to the set of  $n_{ei}(t)$  immigrants created from the elite of the previous generation, a set  $S_{ri}(t)$  of  $n_{ri}(t)$  immigrants are also randomly generated at each generation  $t$ . These two sets of immigrants then replace the worst individuals in the current population. Within ERIGA, in order to give the immigrants scheme that performs better (in terms of whose best immigrant generated has a higher fitness) more chance to generate immigrants, the sizes of the two immigrants schemes are adaptively adjusted according to their relative performance within the range of  $[n_{min}, n_i - n_{min}]$ , where  $n_{min}$  is the minimum number of immigrants of one type per generation and  $n_i$  is the fixed total number of immigrants per generation, i.e.,  $n_i = n_{ei}(t) + n_{ri}(t)$ . If one immigrants scheme performs worse than the other, its immigrants size  $n_{xi}(t)$  (i.e.,  $n_{ei}(t)$  or  $n_{ri}(t)$ ) will be reduced by  $n_{xi}(t) - \max\{n_{min}, n_{xi}(t) - \alpha\}$ , where  $\alpha$  is a constant value, while the immigrants size of the winner is increased by a corresponding value. If the two immigrants schemes tie, there is no change to the immigrants sizes.

Dualism and complementary principle are widely existent mechanisms in nature. For example, in biology the DNA molecule consists of two complementary strands that are twisted together into a duplex chain. In order to address DOPs with significant or extremely high degree of environmental changes, a *dualism-based immigrants* scheme is also proposed in this paper for GAs with the inspiration from the dualism and complementary principle in nature. For the convenience of description, we first introduce the definition of dualism here. Given a binary-encoded individual  $\vec{x} = (x_1, \dots, x_l) \in I = \{0, 1\}^l$  of length  $l$ , its dual individual  $\vec{x}^d$  is defined as:

$$\vec{x}^d = \text{dual}(\vec{x}) = (x_1^d, \dots, x_l^d) \in I, \quad (1)$$

where  $x_i^d = 1 - x_i$  ( $i = 1, \dots, l$ ). That is, an individual's dual individual is the one that is symmetric to it with respect to the central point of the search space. With this definition, the dualism-based immigrants scheme can be described as follows. The dual individual of the elite  $E(t-1)$  from the previous generation is used as the base to generate a set  $S_{di}(t)$  of  $n_{di}(t)$  immigrants every generation via the traditional mutation scheme with a probability  $p_m^{di}$  to replace the worst individuals in the population.

The dualism-based immigrants scheme together with the elitism-based immigrants scheme and the traditional random immigrants scheme form the *hybrid immigrants* scheme and the corresponding GA is denoted *HIGA*. The pseudo-code of HIGA is also shown together with the pseudo-code of EIGA and ERIGA. As in ERIGA, the sizes of the three kinds of immigrants schemes are adaptively adjusted based on their relative performance. Each immigrants scheme generates immigrants within the range of  $[n_{min}, n_i - 2n_{min}]$ . For the worst two immigrants schemes, their immigrants size  $n_{xi}(t)$  (i.e.,  $n_{ei}(t)$ ,  $n_{ri}(t)$  or  $n_{di}(t)$ ) will be reduced by  $n_{xi}(t) - \max\{n_{min}, n_{xi}(t) - \alpha\}$  while the immigrants size of the winner immigrants scheme will be increased accordingly to make the total number of immigrants for the next generation fixed to  $n_i = n_{ei}(t) + n_{ri}(t) + n_{di}(t)$ . If the three immigrants schemes tie, there is no change to the immigrants sizes.

In summary, the key idea behind the hybrid immigrants scheme is to address dynamic environments with different

degrees of changes. The elitism-based immigrants aim to address slight changes or stationary environments in between changes. The random immigrants aim to deal with medium degree of changes while the dualism-based immigrants aim to handle significant changes. Together, these three kinds of immigrants are expected to improve the performance of GAs in different kinds of dynamic environments. This will be tested in the experimental study presented below.

## 4 Dynamic Test Environments

In order to study the performance of GAs in dynamic environments, researchers have developed a number of DOP generators to create dynamic test environments over the years, see [4, 8, 23, 27]. Generally speaking, DOPs are constructed via changing stationary base problem(s). Through properly tuning some parameters of the DOP generators, different dynamic environments can be constructed from the stationary base problem(s) regarding the characteristics of the environmental dynamics, such as the frequency (or speed) and severity of environmental changes.

In this paper, in order to compare different immigrants schemes for GAs in dynamic environments, a set of four well studied stationary problems is selected as the test suite, which is presented in section 4.1. A series of dynamic optimization problems are constructed from these stationary problems using the DOP generator proposed in [22, 23], which is briefly described in section 4.2.

### 4.1 Stationary Test Problems

#### 4.1.1 The OneMax Function

The OneMax function is a well-studied simple benchmark problem in the GA community. It aims to maximize the number of ones in a binary string. In this paper, we use a OneMax problem with 100-bit encoding, defined as follows:

$$\max f(\vec{x}) = \sum_{i=1}^{i=100} x_i, \quad (2)$$

where  $f(\vec{x})$  is the fitness of a bit string  $\vec{x} = (x_1, \dots, x_{100}) \in I = \{0, 1\}^{100}$ .

#### 4.1.2 The Royal Road Function

This function is similar to the royal road function introduced by Mitchell et al. [28]. It is defined on a 100-bit string that consists of 25 contiguous building blocks, each of which is 4-bit long and contributes  $c_i = 4$  ( $i = 1, \dots, 25$ ) to the total fitness only if all of the four bits are set to one. The fitness of a bit string  $\vec{x}$  is computed by summing the coefficients  $c_i$  corresponding to each given building block or schema  $s_i$ , of which  $\vec{x}$  is an instance (denoted by  $\vec{x} \in s_i$ ). That is, the royal road function is defined as follows:

$$\max f(\vec{x}) = \sum_{i=1}^{i=25} c_i \delta_i(\vec{x}), \quad (3)$$

where

$$\delta_i(\vec{x}) = \begin{cases} 1, & \text{if } \vec{x} \in s_i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

#### 4.1.3 The Deceptive Function

Deceptive functions are devised as difficult test functions for GAs [29]. They are a family of functions where there exist low-order building blocks that do not combine to form higher-order building blocks: instead they form building blocks resulting in a solution, called *deceptive attractor* [30], which is sub-optimal itself or near a sub-optimal solution. It is even claimed that the only challenging problems for GAs are problems that involve some degree of deception. A 4-bit fully deceptive problem can be defined based on the unitation function as follows:

$$f(\vec{x}) = \begin{cases} 4, & \text{if } u(\vec{x}) = 4 \\ 3 - u(\vec{x}), & \text{otherwise} \end{cases} \quad (5)$$

where  $u(\vec{x})$  is the unitation function, which returns the number of ones in the string  $\vec{x}$ .

In this study, we construct a deceptive function that consists of 25 copies of the above 4-bit fully deceptive function (order-4 subproblem). The fitness of a bit string is the sum of contributions from all the subproblems. The maximum fitness is 100 for the deceptive function as well as for the OneMax and Royal Road functions studied in this paper.

#### 4.1.4 The 0-1 Knapsack Problem

The 0-1 knapsack problem, a well known NP-complete combinatorial optimization problem, aims to select items from a set of items with varying weights and profits to fill in a knapsack without exceeding its limited weight capacity in order to yield the maximal summed profit. Given a set of  $m$  items and a knapsack, the 0-1 knapsack problem can be described as follows:

$$\max p(\vec{x}) = \sum_{i=1}^{i=m} p_i x_i \quad (6)$$

subject to the weight constraint

$$\sum_{i=1}^{i=m} w_i x_i \leq C, \quad (7)$$

where  $\vec{x} = (x_1, \dots, x_m)$ ,  $x_i = \{0, 1\}$ ,  $w_i$  and  $p_i$  are the weight and profit of item  $i$  respectively, and  $C$  is the knapsack's capacity. If  $x_i = 1$ , the  $i$ th item is selected for the knapsack.

In this paper, a 0-1 knapsack problem with 100 items is constructed as follows:

$$w_i = \text{uniformly distributed random integer in } [1, 30] \quad (8)$$

$$p_i = \text{uniformly distributed random integer in } [1, 30] \quad (9)$$

$$C = 0.5 \sum_{i=1}^{i=100} w_i \quad (10)$$

Given a solution  $\vec{x}$ , its fitness  $f(\vec{x})$  is evaluated as follows. If the sum of the weights of the selected items is within the capacity of the knapsack, the sum of the profits of the selected items is used as the fitness. If a solution overfills the knapsack, its fitness is set to the difference between the total weight of all items and the weight of the selected items, multiplied by a small factor  $10^{-5}$  to make it in-competitive

with those solutions that do not overfill the knapsack. Together, the fitness of a solution  $\vec{x}$  is defined as:

$$f(\vec{x}) = \begin{cases} \sum_{i=1}^{i=100} p_i x_i, & \text{if } \sum_{i=1}^{i=100} w_i x_i \leq C \\ 10^{-5} (\sum_{i=1}^{i=100} w_i - \sum_{i=1}^{i=100} w_i x_i), & \text{otherwise} \end{cases} \quad (11)$$

## 4.2 Constructing Dynamic Test Problems

The DOP generator proposed in [22, 23] can construct dynamic environments from any binary-encoded stationary function  $f(\vec{x})$  ( $\vec{x} \in \{0, 1\}^l$ ) by a bitwise exclusive-or (XOR) operator. The environment is changed every  $\tau$  generations. For each environmental period  $k$ , an XOR mask  $\vec{M}(k)$  is incrementally generated as follows:

$$\vec{M}(k) = \vec{M}(k-1) \oplus \vec{T}(k), \quad (12)$$

where “ $\oplus$ ” is the XOR operator (i.e.,  $1 \oplus 1 = 0$ ,  $1 \oplus 0 = 1$ ,  $0 \oplus 0 = 0$ ) and  $\vec{T}(k)$  is an intermediate binary template randomly created with  $\rho \times l$  ones for the  $k$ -th environmental period. For the first period  $k = 1$ ,  $\vec{M}(1) = \vec{0}$ . Then, the population at generation  $t$  is evaluated as below:

$$f(\vec{x}, t) = f(\vec{x} \oplus \vec{M}(k)), \quad (13)$$

where  $k = \lceil t/\tau \rceil$  is the environmental index.

With this generator, the parameter  $\tau$  controls the speed of changes while  $\rho \in (0.0, 1.0)$  controls the severity of changes. A bigger  $\rho$  means severer changes while a smaller  $\tau$  means faster changes. Using this DOP generator, we can study the performance of GAs on the dynamic version of many benchmark problems well studied in the GA community. For example, we can construct DOPs from the four stationary problems described in section 4.1 as dynamic test environments for the experimental study in this paper. The dynamic test environments are constructed as follows.

The fitness landscape of each stationary problem is periodically changed every  $\tau$  generations during the run of algorithms. In order to study each algorithm’s capability of adapting to dynamic environment under different degree of convergence or searching stage, the environmental change speed parameter  $\tau$  is set to 10, 50 and 100 respectively. The parameter  $\tau$  is set to these values because on the stationary problems all algorithms are sort of consistently on different search stages at generations of these values. For example, on the stationary problems almost all algorithms are at a quite early searching stage at generation 10, at a medium searching stage at generation 50, and at a late or converged stage at generation 100. In order to test the effect of the degree of environmental changes on the performance of algorithms, the value of  $\rho$  is set to 0.05, 0.2, 0.5, 0.6, 0.95, and 1.0 respectively for each run of an algorithm on a problem. These values represent different severities of environmental changes, from very light shifting ( $\rho = 0.05$ ) to medium variation ( $\rho = 0.2, 0.5, 0.6$ ) to significant change ( $\rho = 0.95, 1.0$ ).

Totally, we systematically construct a series of 18 DOPs, 3 values of  $\tau$  combined with 6 values of  $\rho$ , from each stationary test problem.

## 5 Experimental Study

### 5.1 Experimental Design

In the experiments, five GAs were investigated on the above constructed DOPs. They are SGA, RIGA, EIGA, ERIGA, and HIGA. All GAs are set with typical generators and parameters as follows: generational, 2-point crossover with  $p_c = 0.7$ , flip mutation with  $p_m = 0.01$ , and tournament selection with tournament size 2 and elitism of size 1. In order to have fair comparisons among GAs, the population size  $n$  and the number of immigrants are set such that each GA has 130 fitness evaluations per generation as follows:  $n = 130$  for SGA and  $n = 100$  for other GAs and the total number of immigrants  $n_i = 30$  for RIGA, EIGA, ERIGA and HIGA. For EIGA,  $n_{ei}(t) = n_i = 30$ . For ERIGA,  $n_{ei}$  and  $n_{ri}$  are initialized to 15 respectively and then adjusted adaptively later on in the range of  $[n_{min}, n_i - n_{min}] = [4, 26]$ . Similarly, for HIGA,  $n_{ei}$ ,  $n_{ri}$ , and  $n_{di}$  are initialized to 10 respectively and then adjusted adaptively later on in the range of  $[n_{min}, n_i - 2n_{min}] = [4, 22]$ . For ERIGA and HIGA,  $\alpha$  is set to 2. For EIGA, ERIGA, and HIGA,  $p_m^{ei}$  and/or  $p_m^{di}$  are set to 0.01.

For each GA on a DOP, 30 independent runs were executed with the same set of random seeds. For each run of a GA on a DOP, 50 environmental changes were allowed and the best-of-generation fitness was recorded every generation. The offline performance of a GA on a DOP is defined as the best-of-generation fitness averaged over 30 runs and over the data gathering period, as formulated below:

$$\bar{F}_{BOG} = \frac{1}{G} \sum_{i=1}^G \left( \frac{1}{N} \sum_{j=1}^N F_{BOG_{ij}} \right), \quad (14)$$

where  $G = 50\tau$  is the total number of generations for a run,  $N = 30$  is the total number of runs, and  $F_{BOG_{ij}}$  is the best-of-generation fitness of generation  $i$  of run  $j$ .

### 5.2 Basic Experimental Results

The experimental results of investigated GAs on the DOPs are plotted in Fig. 1. The statistical results of comparing GAs by one-tailed  $t$ -test with 58 degrees of freedom at a 0.05 level of significance are given in Table 1. In Table 1, the  $t$ -test result regarding algorithm 1 – algorithm 2 is shown as “s+”, “s−”, “+”, “−”, or “=” when algorithm 1 is significantly better than, significantly worse than, insignificantly better than, insignificantly worse than, or statistically equivalent to algorithm 2 respectively. The dynamic performance of GAs for the first 10 environments with respect to the best-of-generation fitness against generation on the DOPs with  $\tau = 50$  and  $\rho = 0.05, 0.5$ , and  $0.95$  is plotted in Figs. 2, 3, and 4 respectively, where the data were averaged over 30 runs. From the figures and Table 1, several results can be observed and are analyzed as follows.

First, RIGA significantly outperforms SGA on many dynamic test problems, see the  $t$ -test results regarding RIGA - SGA in Table 1. This result validates the benefit of introducing random immigrants into GAs for DOPs. However, RIGA is beaten by SGA on most DOPs with  $\rho = 0.05$  and  $0.2$ . This confirms our prediction made in section 3: when the environment changes slightly, random immigrants may not be beneficial. It happens because under slightly changing environments, random immigrants may divert the major

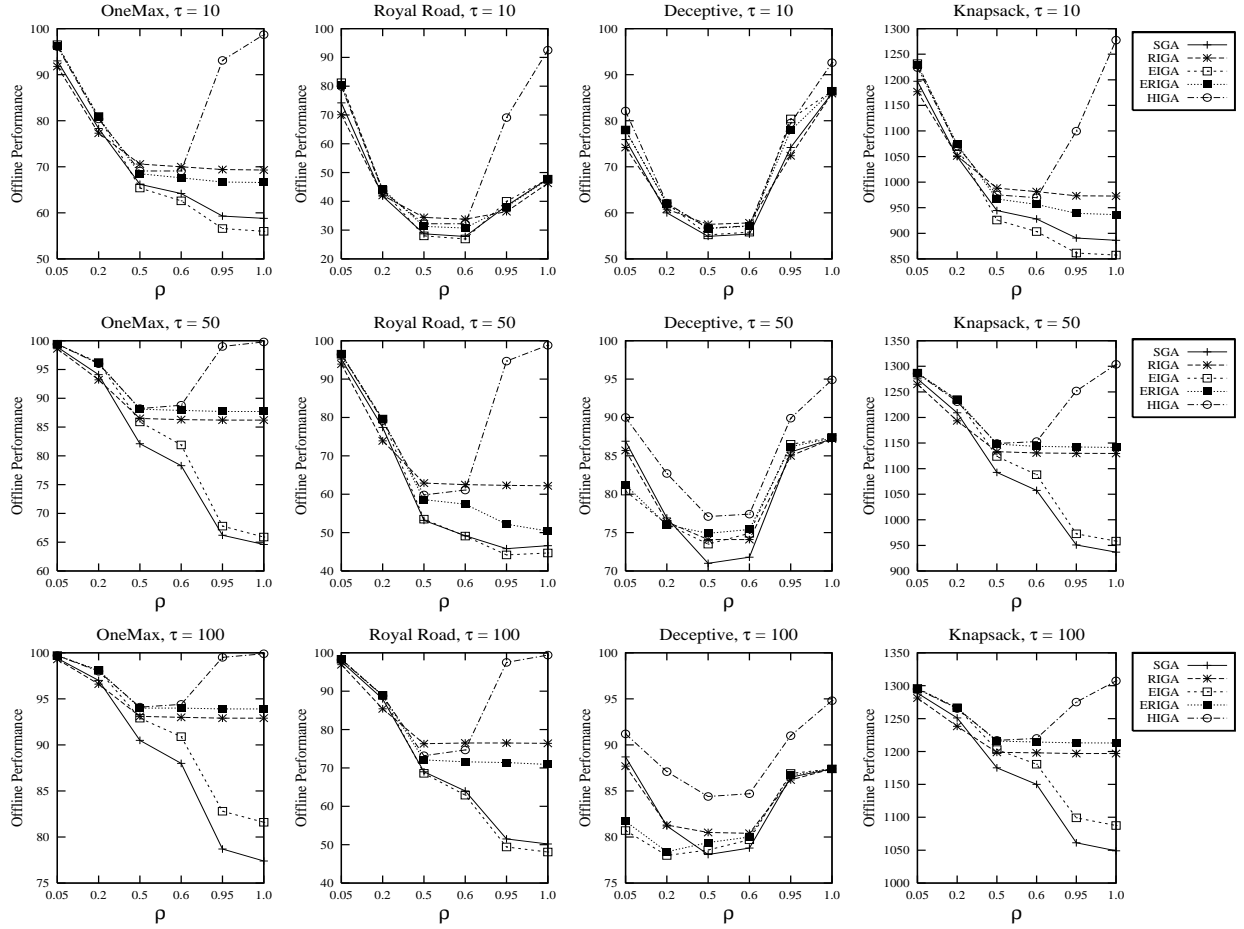


Fig. 1 Experimental results of GAs on dynamic test problems

searching force of the GA. This result can be further observed from the dynamic performance of SGA and RIGA. When  $\rho = 0.05$  RIGA maintains a lower fitness level than SGA (see Fig. 2) while when  $\rho = 0.5$  and  $0.95$ , RIGA maintains a higher fitness level (see Figs. 3 and 4).

Second, EIGA outperforms SGA and RIGA on most DOPs with  $\rho = 0.05$  and  $0.2$ , see the  $t$ -test results regarding  $EIGA - SGA$  and  $EIGA - RIGA$  in Table 1. This result confirms our expectation of the elitism-based immigrants for GAs in dynamic environments. When the environment changes slightly, it would be better to introduce immigrants guided toward the environment via the elite. However, when the environment changes significantly, e.g.,  $\rho = 0.95$  and  $1.0$ , EIGA is beaten by SGA and RIGA on some DOPs. This is because whenever the environment changes significantly, the elite from the previous generation may become significantly unfit in the new environment and hence will guide the immigrants to unfit area. This can be observed from the sharp drop of the dynamic performance of EIGA on dynamic *OneMax* and *Knapsack* problems with  $\rho = 0.95$  when a change occurs in Fig. 4.

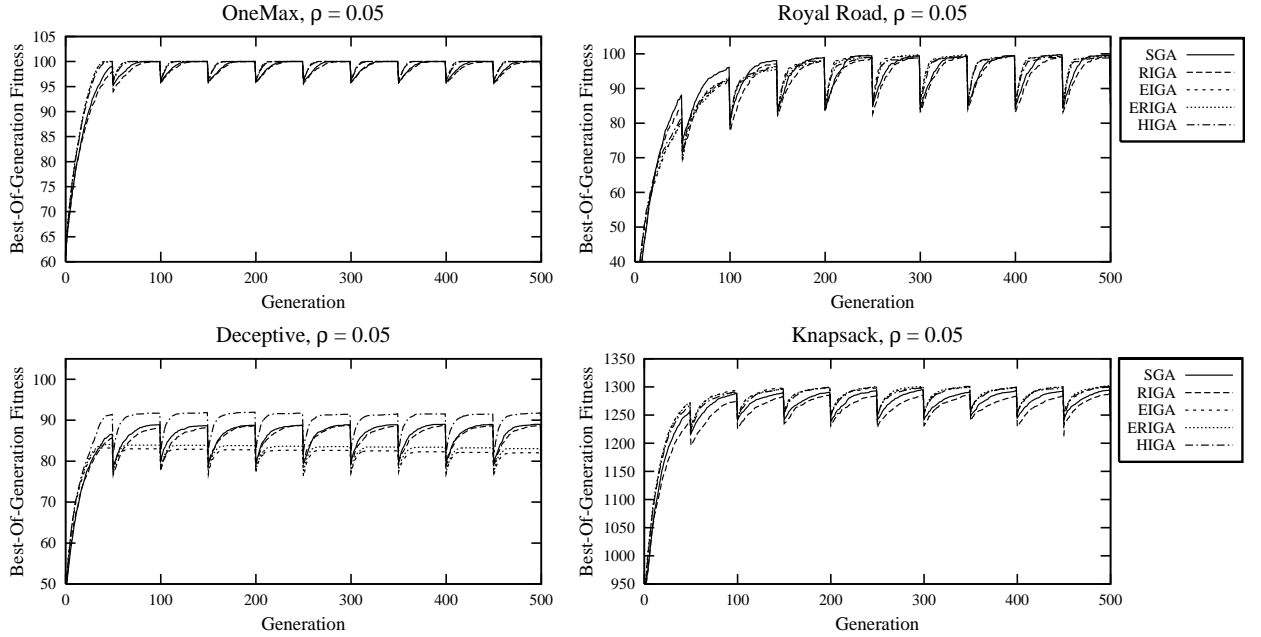
Third, regarding the effect of combining the elitism-based immigrants and random immigrants schemes to GAs, it can be seen that ERIGA outperforms SGA and RIGA on many

DOPs, see the  $t$ -test results regarding  $ERIGA - RIGA$  in Table 1. When comparing the performance of ERIGA over EIGA, it can be seen that ERIGA beats EIGA on DOPs with  $\rho$  set to bigger values  $0.6$ ,  $0.95$  and  $1.0$  while is beaten by EIGA on DOPs with  $\rho = 0.05$  and  $0.2$ . The random immigrants scheme improves the performance of ERIGA over EIGA in significantly changing environments at the price of degrading the performance in slightly changing environments. This result can be more clearly observed from the dynamic performance of ERIGA and EIGA in Fig. 6. The random immigrants added in ERIGA prevent the performance of ERIGA from a sharp drop when the environment significantly changes with  $\rho = 0.95$ .

Fourth, now consider the effect of the hybrid immigrants scheme for GAs for DOPs. It can be seen that HIGA now outperforms SGA and RIGA on most DOPs, see the  $t$ -test results regarding  $HIGA - RIGA$  in Table 1. When comparing the performance of HIGA over ERIGA and EIGA, it can be seen that HIGA beats ERIGA and EIGA on almost all DOPs with  $\rho$  set to  $0.5$ ,  $0.6$ ,  $0.95$  and  $1.0$  while is beaten on dynamic *OneMax*, *Royal Road*, and *Knapsack* problems with  $\rho = 0.05$  and  $0.2$ . The dualism-based immigrants scheme further improves the performance of HIGA over ERIGA and EIGA in significantly changing environ-

Table 1 The  $t$ -test results of comparing GAs on random DOPs.

$t$ -test Result	<i>OneMax</i>						<i>Royal Road</i>						<i>Deceptive</i>						<i>Knapsack</i>					
$\tau = 10, \rho \Rightarrow$	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0
<i>RIGA</i> – <i>SGA</i>	s–	s–	s+	s+	s+	s+	s–	+	s+	s+	s–	s–	s–	s+	s+	s+	s–	–	s–	+	s+	s+	s+	s+
<i>EIGA</i> – <i>SGA</i>	s+	s+	s–	s–	s–	s–	s+	s+	s–	s–	s+	–	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s–	s–
<i>EIGA</i> – <i>RIGA</i>	s+	s+	s–	s–	s–	s–	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s–	s–
<i>ERIGA</i> – <i>RIGA</i>	s+	s+	s–	s–	s–	s–	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s–	s–
<i>ERIGA</i> – <i>EIGA</i>	s–	s–	s+	s+	s+	s+	–	+	s+	s+	s–	–	–	+	s+	s+	s–	+	s–	s+	s+	s+	s+	s+
<i>HIGA</i> – <i>RIGA</i>	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+
<i>HIGA</i> – <i>EIGA</i>	s–	s–	s+	s+	s+	s+	s–	–	s+	s+	s+	s+	s+	+	s+	s+	s–	s+	s–	+	s+	s+	s+	s+
<i>HIGA</i> – <i>ERIGA</i>	s–	s–	s+	s+	s+	s+	–	s–	s+	s+	s+	s+	s+	+	+	+	+	s+	s+	s–	–	s+	s+	s+
$\tau = 50, \rho \Rightarrow$	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0
<i>RIGA</i> – <i>SGA</i>	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s–	–	s–	s–	s+	s+	s+	s+
<i>EIGA</i> – <i>SGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s+	–	s–	s–	s–	s–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>EIGA</i> – <i>RIGA</i>	s+	s+	s–	s–	s–	s–	s+	s+	s–	s–	s–	s–	s–	s–	s–	s–	s+	s+	s+	s+	s–	s–	s–	s–
<i>ERIGA</i> – <i>RIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s–	s–	s–	s–	s–	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>ERIGA</i> – <i>EIGA</i>	s–	s–	s+	s+	s+	s+	–	–	s+	s+	s+	s+	+	–	s+	s+	s–	–	–	–	s+	s+	s+	s+
<i>HIGA</i> – <i>RIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>HIGA</i> – <i>EIGA</i>	s–	s–	s+	s+	s+	s+	–	s–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+
<i>HIGA</i> – <i>ERIGA</i>	s–	s–	+	s+	s+	s+	–	–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	+	s+	s+	s+
$\tau = 100, \rho \Rightarrow$	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0	0.05	0.2	0.5	0.6	0.95	1.0
<i>RIGA</i> – <i>SGA</i>	s–	s–	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+	s–	+	s+	s+	s–	s+	s–	s–	s+	s+	s+	s+
<i>EIGA</i> – <i>SGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s–	s–	s–	s–	s–	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>EIGA</i> – <i>RIGA</i>	s+	s+	s–	s–	s–	s–	s+	s+	s–	s–	s–	s–	s–	s–	s–	s–	s–	s+	s+	s+	s–	s–	s–	s–
<i>ERIGA</i> – <i>RIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s–	s–	s–	s–	s–	s–	s–	s+	s+	s+	s+	s+	s+	s+
<i>ERIGA</i> – <i>EIGA</i>	s–	s–	s+	s+	s+	s+	+	=	s+	s+	s+	s+	s+	s+	s+	s+	s–	–	s–	–	s+	s+	s+	s+
<i>HIGA</i> – <i>RIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>HIGA</i> – <i>EIGA</i>	s–	s–	s+	s+	s+	s+	–	–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+
<i>HIGA</i> – <i>ERIGA</i>	s–	s–	+	s+	s+	s+	–	–	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s–	s–	s+	s+	s+	s+

Fig. 2 Dynamic performance of GAs on DOPs with  $\tau = 50$  and  $\rho = 0.05$  for the first 10 environments

ments. This result can be more clearly observed from the dynamic performance of HIGA on DOPs with  $\rho = 0.95$  in Fig. 4, where the dualism-based immigrants added in HIGA prevent its performance from a sharp drop when the environment changes.

Finally, in order to understand the effect of investigated immigrants schemes on the population diversity, we recorded the diversity of the population every generation for each run of a GA on a DOP. The mean population diversity of a GA on a DOP at generation  $t$  over 30 runs is

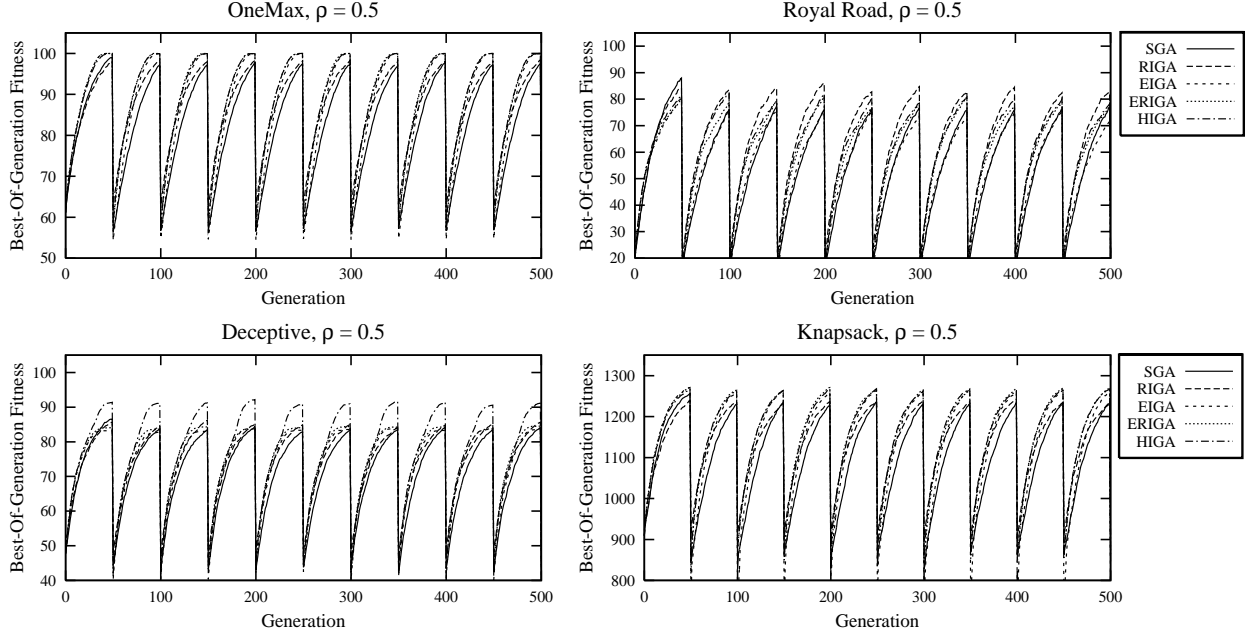


Fig. 3 Dynamic performance of GAs on DOPs with  $\tau = 50$  and  $\rho = 0.5$  for the first 10 environments

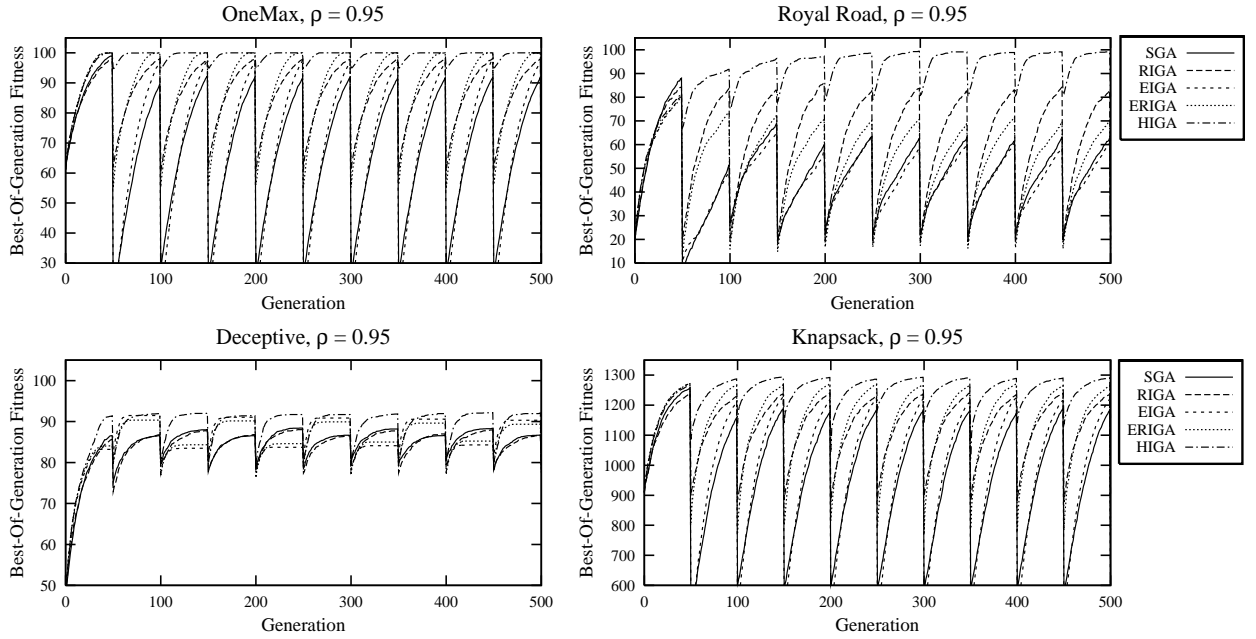


Fig. 4 Dynamic performance of GAs on DOPs with  $\tau = 50$  and  $\rho = 0.95$  for the first 10 environments

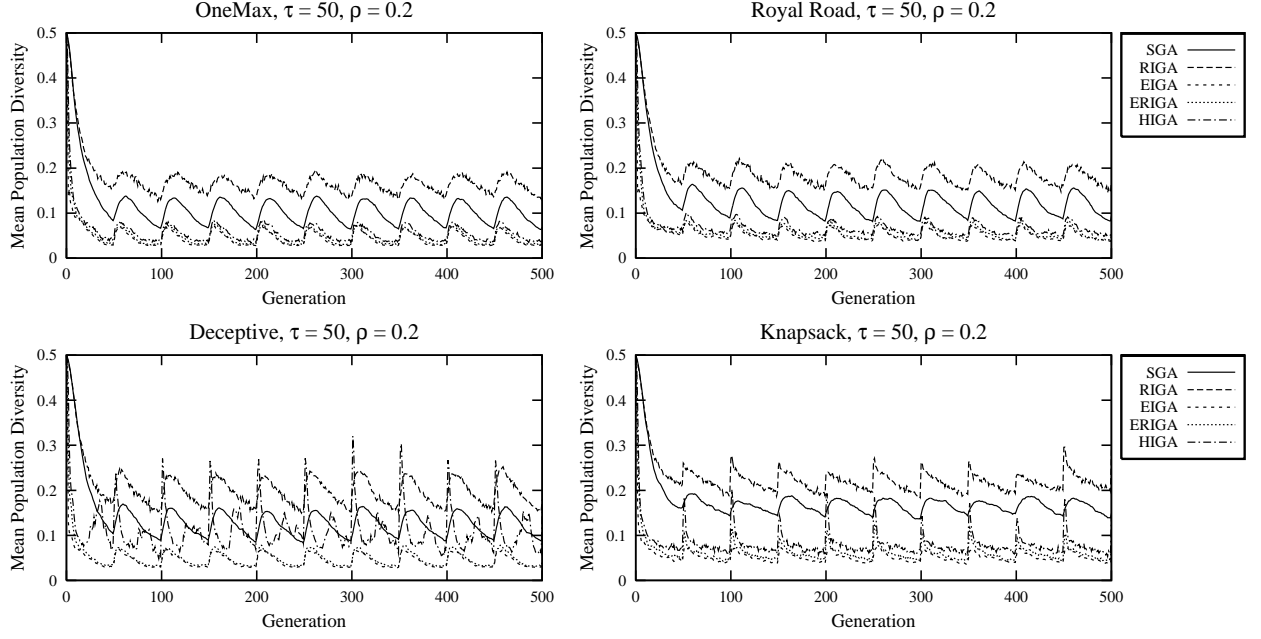
calculated according to the following formula:

$$\overline{Div}(t) = \frac{1}{30} \sum_{k=1}^{30} \left( \frac{1}{ln(n-1)} \sum_{i=1}^n \sum_{j \neq i}^n HD_{ij}(k, t) \right), \quad (15)$$

where  $l = 100$  is the encoding length and  $HD_{ij}(k, t)$  is the Hamming distance between the  $i$ -th and  $j$ -th individuals at generation  $t$  of the  $k$ -th run. The diversity dynamics over

generation for GAs on DOPs with  $\tau = 50$  and  $\rho = 0.2$  is shown in Fig. 5. From Fig. 5, it can be seen that RIGA does maintain the highest diversity level in the population while EIGA maintains the lowest diversity level. This interesting result shows that approaches that aim at maintaining a high diversity level in the population, though usually useful, do not naturally achieve better performance than other approaches for GAs in dynamic environments.



Fig. 5 Diversity dynamics of GAs on DOPs with  $\tau = 50$  and  $\rho = 0.2$  for the first 10 environmentsTable 2 The experimental results of GAs on DOPs with  $\rho$  randomly set for each change.

Function	<i>OneMax</i>			<i>Royal Road</i>			<i>Deceptive</i>			<i>Knapsack</i>		
$\rho = rand(0.0, 1.0), \tau \Rightarrow$	10	50	100	10	50	100	10	50	100	10	50	100
<i>SGA</i>	65.9	81.9	90.1	32.0	56.3	70.1	59.5	76.9	81.8	943.9	1093.3	1174.9
<i>RIGA</i>	73.0	89.1	94.4	37.7	67.8	80.1	61.1	77.6	82.3	1008.0	1158.4	1215.2
<i>EIGA</i>	65.3	84.7	92.3	32.1	55.7	68.8	61.3	79.4	82.1	929.9	1120.0	1200.2
<i>ERIGA</i>	71.8	91.2	95.5	34.9	64.6	78.2	62.0	79.7	82.3	993.0	1181.1	1235.2
<i>HIGA</i>	76.1	94.0	96.8	40.5	73.8	84.5	62.4	82.4	87.0	1027.3	1205.6	1248.6
<i>t</i> -test Result												
$\rho = rand(0.0, 1.0), \tau \Rightarrow$	10	50	100	10	50	100	10	50	100	10	50	100
<i>RIGA</i> – <i>SGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>EIGA</i> – <i>SGA</i>	s–	s+	s+	+	–	s–	s+	s+	+	s–	s+	s+
<i>EIGA</i> – <i>RIGA</i>	s–	s–	s–	s–	s–	s–	+	s+	s–	s–	s–	s–
<i>ERIGA</i> – <i>RIGA</i>	s–	s+	s+	s–	s–	s–	s+	s+	–	s–	s+	s+
<i>ERIGA</i> – <i>EIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	+	s+	s+	s+
<i>HIGA</i> – <i>SGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>HIGA</i> – <i>RIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>HIGA</i> – <i>EIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+
<i>HIGA</i> – <i>ERIGA</i>	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+	s+

### 5.3 Experimental Results under Dynamic Environments with Random Severities of Changes

The above basic experimental results show that the three kinds of immigrants schemes, i.e., random immigrants, elitism-based immigrants, and dualism-based immigrants, have different but complementary effects regarding the severity of environmental changes on improving the performance of GAs in dynamic environments. In the real world problems, it would be expected that the environment may be subject to different degrees of changes over time. In order to study the performance of GAs in dynamic environments with random degrees of changes, experiments are further carried out on the DOPs where the value of  $\rho$  is

randomly generated with a uniform distribution in  $[0.0, 1.0]$  (i.e.,  $\rho = rand(0.0, 1.0)$ ) for each environmental change. Here, the experimental settings, including generators and parameters for GAs and the performance measure, are the same as those for the basic experiments.

The experimental results of GAs on the DOPs are presented in Table 2. The statistical results of comparing GAs by one-tailed *t*-test with 58 degrees of freedom at a 0.05 level of significance are also given in Table 2. The dynamic performance of a typical run of GAs on the dynamic knapsack problem with  $\tau = 50$  and  $\rho = rand(0.0, 1.0)$  for the first 20 environments is plotted in Fig. 6(a). The value of  $\rho$  for each environmental change and the number of immigrants of the three kinds of immigrants schemes within HIGA over generation (i.e.,  $n_{ri}(t)$ ,  $n_{ei}(t)$ , and  $n_{di}(t)$ ) for this typical run

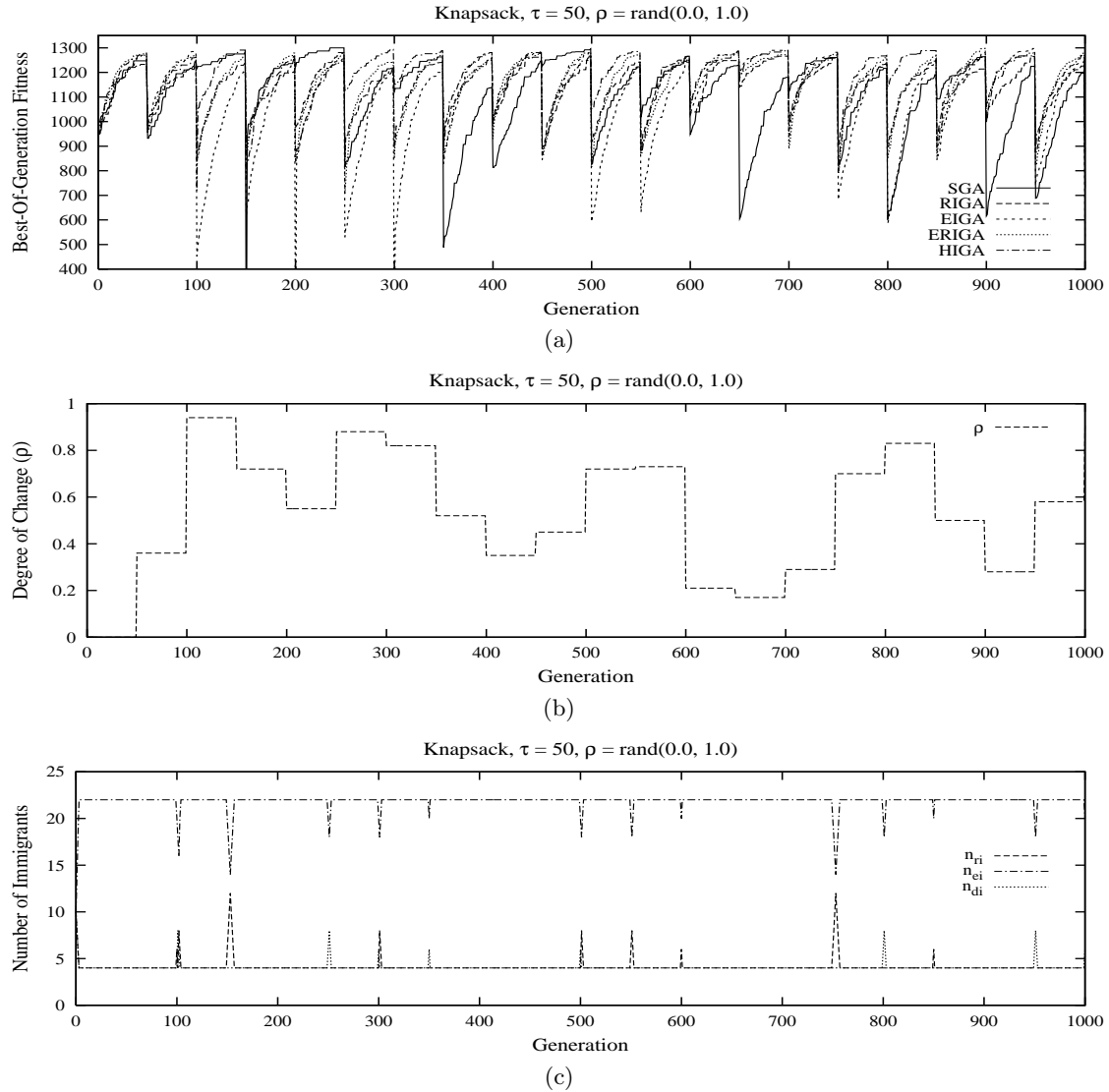


Fig. 6 Dynamic behaviour of a typical run of GAs on the dynamic knapsack problem with  $\tau = 50$  and  $\rho = \text{rand}(0.0, 1.0)$  for the first 20 environments: (a) dynamic performance, (b) the value of  $\rho$  for each environmental change, and (c) the number of immigrants of the three kinds of immigrants schemes within HIGA over generation, i.e.,  $n_{ri}(t)$ ,  $n_{ei}(t)$ , and  $n_{di}(t)$

are shown in Fig. 6(b) and Fig. 6(c) respectively. From Table 2 and Fig. 6, several results regarding the performance of GAs in dynamic environments with random degrees of changes can be observed.

First, the addition of random immigrants significantly improves the performance of RIGA over SG and ERIGA over EIGA respectively, see the corresponding  $t$ -test results. On the contrast, the effect of elitism-based immigrants is mixed, see the  $t$ -test results regarding EIGA - SG and ERIGA - RIGA. And the elitism-based immigrants scheme seems less efficient than the random immigrants scheme, see the  $t$ -test results regarding EIGA - RIGA.

Second, HIGA significantly outperforms all other GAs on the DOPs, see the  $t$ -test results regarding HIGA - SG, HIGA - RIGA, HIGA - EIGA, and HIGA - ERIGA in Table 2. This result further validates the efficiency of the hybrid immigrants scheme for GAs in dynamic environments. The

good performance of HIGA can be further understood by observing the dynamic behaviour of a typical run of GAs shown in Fig. 6.

From Fig. 6, it can be seen that when the environment incurs a significant change, the performance of SG and EIGA usually drops significantly while the performance of RIGA and ERIGA also drops with a great degree though less significantly. On the contrast, the performance of HIGA only drops slightly due to the added dualism-based immigrants scheme.

Third, Fig. 6(c) shows an interesting result regarding the number of immigrants for the three kinds of immigrants schemes within HIGA. The number of elitism-based immigrants  $n_{ei}$  remains at the maximum value 22 while  $n_{ri}$  and  $n_{di}$  remain at the minimum value 4 for each environment in between two changes and when the environment changes slightly. Only when the environment incurs a significant

change can we see the jump up of the value of  $n_{ri}$  and/or  $n_{di}$  for a few generations, which signals the working of the dualism-based immigrants and/or random immigrants. In other words, the dualism-based immigrants and random immigrants in the hybrid immigrants scheme only take effect for a short time when significant changes occur. However, they do improve the performance of HIGA significantly in dynamic environments with random degrees of changes.

## 6 Conclusions

Using immigrants to replace individuals in the population is one of several approaches developed into GAs to address DOPs. This paper proposes a hybrid immigrants scheme for GAs in dynamic environments. In this hybrid scheme, the elite from the previous generation and its dual are used as the bases to create immigrants via a normal bitwise mutation operator. The generated immigrants together with some random immigrants replace the worst individuals in the population. These three kinds of immigrants work together to deal with different degrees of environmental changes and hence efficiently improve the performance of GAs in dynamic environments.

From the experiment results on a series of dynamic problems, the following conclusions can be drawn on the test DOPs. First, traditional random immigrants are beneficial for GAs for many DOPs but may be harmful when the environment changes slightly. Second, the elitism-based immigrants scheme combines the principles of random immigrants and elitism and improves the performance of GAs in slightly changing dynamic environments. Third, the hybrid immigrants scheme combines the principles of random immigrants, elitism, and dualism and the adaptive adjustment of the immigrants size effectively balances the computation effort among the three kinds of immigrants schemes. Finally, a high diversity level of the population does not always lead to a better performance of GAs in dynamic environments. Generally speaking, the experiment results indicate that the hybrid immigrants scheme is a good choice for GAs to address DOPs.

There are several future researches relevant to this paper. First, it is interesting to compare and combine the hybrid immigrants scheme with other approaches, e.g., diversity and memory hybrid schemes<sup>[15, 17]</sup>, for GAs in dynamic environments. Another interesting work is to further integrate the idea of elitism, dualism, and immigrants into other approaches, e.g., multi-population and speciation schemes<sup>[19, 20]</sup>, to develop advanced diversity schemes for GAs in dynamic environments.

## References

- [1] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, USA, 1989.
- [2] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, University of Michigan Press, 1975.
- [3] D. E. Goldberg and R. E. Smith. Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy. In *Proceedings of the 2nd International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Mahwah, NJ, USA, pp. 59–68, 1987.
- [4] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, Boston, MA, 2002.
- [5] R. W. Morrison. *Designing Evolutionary Algorithms for Dynamic Environments*, Springer-Verlag, Berlin Heidelberg, 2004.
- [6] K. Weicker. *Evolutionary Algorithms and Dynamic Optimization Problems*, Osnabrück, Germany, Der andere Verlag, 2003.
- [7] S. Yang, Y.-S. Ong, and Y. Jin (eds.), *Evolutionary Computation in Dynamic and Uncertain Environments*, Springer-Verlag, Berlin Heidelberg, March 2007.
- [8] Y. Jin and J. Branke. Evolutionary Optimization in Uncertain Environments: a Survey. *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
- [9] J. J. Grefenstette. Genetic Algorithms for Changing Environments. *Parallel Problem Solving from Nature II*, Elsevier Science Publishers, The Netherlands, pp. 137–144, 1992.
- [10] F. Vavak and T. C. Fogarty. A Comparative Study of Steady State and Generational Genetic Algorithms for Use in Nonstationary Environments. *AISB Workshop on Evolutionary Computing*, Lecture Notes in Computer Science, vol. 1143, Springer-Verlag, Berlin Heidelberg, pp. 297–304, 1996.
- [11] H. G. Cobb and J. J. Grefenstette. Genetic Algorithms for Tracking Changing Environments. In *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 523–530, 1993.
- [12] R. W. Morrison and K. A. De Jong. Triggered Hypermutation Revisited. In *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, IEEE, pp. 1025–1032, 2000.
- [13] C. N. Bendtsen and T. Krink. Dynamic Memory Model for Non-stationary Optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, IEEE, pp. 145–150, 2002.
- [14] J. Branke. Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems. In *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE, vol. 3, pp. 1875–1882, 1999.
- [15] A. Simões and E. Costa. An Immune System-based Genetic Algorithm to Deal with Dynamic Environments: Diversity and Memory. In *Proceedings of the 6th International Conference on Artificial Neural Networks and Genetic Algorithms*, Springer, pp. 168–174, 2003.
- [16] K. Trojanowski and Z. Michalewicz. Searching for Optima in Non-stationary Environments. In *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE, pp. 1843–1850, 1999.

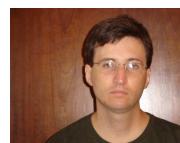
- [17] S. Yang. Memory-based Immigrants for Genetic Algorithms in Dynamic Environments. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1115–1122, 2005.
- [18] S. Yang. Associative Memory Scheme for Genetic Algorithms in Dynamic Environments. *Applications of Evolutionary Computing*, Lecture Notes in Computer Science, vol. 3907, Springer-Verlag, Berlin Heidelberg, pp. 788–799, 2006.
- [19] J. Branke, T. Kaußler, C. Schmidh, and H. Schneck. A Multi-population Approach to Dynamic Optimization Problems. *Proceedings of the Adaptive Computing in Design and Manufacturing*, pp. 299–308, 2000.
- [20] D. Parrott and X. Li. Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation. *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 444–458, 2006.
- [21] S. Yang. Genetic Algorithms with Elitism-based Immigrants for Changing Optimization Problems. *Applications of Evolutionary Computing*, Lecture Notes in Computer Science, vol. 4448, Springer-Verlag, Berlin Heidelberg, pp. 627–636, 2007.
- [22] S. Yang. Non-stationary Problem Optimization Using the Primal-dual Genetic Algorithm. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, IEEE, vol. 3, pp. 2246–2253, 2003.
- [23] 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.
- [24] W. Cedeno and V. R. Vemuri. On the Use of Niching for Dynamic Landscapes. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, IEEE, pp. 361–366, 1997.
- [25] N. Mori, H. Kita and Y. Nishikawa. Adaptation to Changing Environments by Means of the Memory Based Thermodynamical Genetic Algorithm. In T. Bäck (ed.), *Proceedings of the 7th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, pp. 299–306, 1997.
- [26] N. Mori, H. Kita, and Y. Nishikawa. Adaptation to a Changing Environment by Means of the Feedback Thermodynamical Genetic Algorithm. *Parallel Problem Solving from Nature V*, Lecture Notes in Computer Science, vol. 1498, Springer-Verlag Berlin Heidelberg, pp. 149–158, 1998.
- [27] R. W. Morrison and K. A. De Jong. A Test Problem Generator for Non-stationary Environments. In *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE, vol. 3, pp. 2047–2053, 1999.
- [28] M. Mitchell, S. Forrest and J. H. Holland. The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In *Proceedings of the 1st European Conference on Artificial Life*, MIT Press, Cambridge MA, USA, pp. 245–254, 1991.
- [29] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, Boston, MA, 2002.
- [30] L. D. Whitley. Fundamental Principles of Deception in Genetic Search. *Foundations of Genetic Algorithms 1*, Morgan Kaufmann Publishers, pp. 221–241, 1991.



**Shengxiang Yang** received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, P. R. China in 1993, 1996 and 1999 respectively. He was a postdoctoral research associate in the Department of Computer Science, King's College London from October 1999 to October 2000. He is currently a lecturer in the Department of Computer Science at University of Leicester, United Kingdom.

He has published over 50 papers in books, journals and conferences. He has co-guest-edited a special issue for the journal of *Genetic Programming and Evolvable Machines* and has co-edited a book *Evolutionary Computation in Dynamic and Uncertain Environments*, published in March 2007. His current research interests include evolutionary and genetic algorithms, artificial neural networks for combinatorial optimization problems, scheduling problems, dynamic optimization problems, and network flow problems and algorithms.

Dr. Yang serves as the area editor, associate editor and member of editorial board of three international journals. He is a member of the Working Group on Evolutionary Computation in Dynamic and Uncertain Environments, Evolutionary Computation Technical Committee, IEEE Computational Intelligence Society (CIS). He is a member of IEEE and ACM SIGEVO.



**Renato Tinós** received the B.Sc. degree in electrical engineering from State University of São Paulo (UNESP), Brazil in 1994, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of São Paulo (USP) at São Carlos, Brazil in 1999 and 2003 respectively.

He then joined the Department of Computer Science of USP at São Carlos as a research scientist. He is currently an assistant professor at the Department of Physics and Mathematics of USP at Ribeirão Preto.

His research interests include evolutionary algorithms, dynamic optimization, robotics, fault tolerance, and neural networks.