Genetic Algorithms with Elitism-Based Immigrants for Changing Optimization Problems

Shengxiang Yang

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

Abstract. Addressing dynamic optimization problems has been a challenging task for the genetic algorithm community. Over the years, several approaches have been developed into genetic algorithms to enhance their performance in dynamic environments. One major approach is to maintain the diversity of the population, e.g., via random immigrants. This paper proposes an elitism-based immigrants scheme for genetic algorithms in dynamic environments. In the scheme, the elite from previous generation is used as the base to create immigrants via mutation to replace the worst individuals in the current population. This way, the introduced immigrants are more adapted to the changing environment. This paper also proposes a hybrid scheme that combines the elitism-based immigrants scheme with traditional random immigrants scheme to deal with significant changes. The experimental results show that the proposed elitism-based and hybrid immigrants schemes efficiently improve the performance of genetic algorithms in dynamic environments.

1 Introduction

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 design variables and environmental conditions, may change over time. Addressing DOPs has been a challenging task for the genetic algorithm (GA) community due to their dynamic characteristics [6,11]. For stationary optimization problems, our goal is to develop GAs that can quickly and precisely locate the optima 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. Over the years, several approaches have been developed into GAs to address DOPs [3], such as diversity schemes [5,7,12], memory schemes [1,2,10,14], and multi-population and species approaches [4,9].

Among the approaches developed for GAs for DOPs, the random immigrants scheme has proved to be beneficial for many DOPs. It works by maintaining the diversity of the population by replacing individuals from the population with randomly created individuals. In this paper, an *elitism-based immigrants* scheme is proposed and investigated for GAs in dynamic environments. In this scheme, the elite from previous generation is used as the base to create immigrants via mutation to replace the worst individuals in the current population. This way, the introduced immigrants are more adapted to the current environment than random immigrants. This paper also proposes a hybrid immigrants scheme that combines the elitism-based immigrants scheme and traditional random immigrants scheme in order to deal with significant changes.

Based on the dynamic problem generator proposed in [13,15], a series of dynamic test problems are constructed from several stationary functions and experimental study is 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 immigrants schemes for GAs in dynamic environments. The experiment results show that the proposed elitism-based immigrants scheme and the hybrid immigrants scheme efficiently improves the performance of GAs in dynamic environments.

The rest of this paper is outlined as follows. The next section briefly reviews random immigrants for GAs in dynamic environments. Section 3 presents the proposed elitism-based and hybrid immigrants schemes for GAs in dynamic environments. Section 4 describes the dynamic test environments for this study. The experimental results and analysis are presented in Section 5. Finally, Section 6 concludes this paper with discussions on future work.

2 Random Immigrants for GAs in Dynamic Environments

The standard GA (SGA) 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. Usually, with the iteration of the GA, the population will eventually converge to the optimum solution(s) due to the pressure of selection.

In stationary environments, convergence at a proper pace is really what we expect for GAs to locate the optimum solution(s) for many optimization problems. However, for DOPs, convergence usually becomes a big problem for GAs because changing environments usually require GAs to keep a certain population diversity level to maintain their adaptability. To address this problem, the random immigrants approach is a quite natural and simple way [5,7]. 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. It maintains the diversity level of the population through replacing some individuals of the current population with random individuals, called *random immigrants*, every generation. As to which individuals in the population should be replaced, usually there are two strategies: replacing random individuals or replacing the worst ones [12]. In

```
begin
  $t := 0$ and initialize population $P(0)$ randomly
  evaluate population $P(0)$
  repeat
    $P'(t)$ := selectForReproduction($P(t)$)
    crossover($P'(t), p_c$)      // $p_c$ is the crossover probability
    mutate($P'(t), p_m$)         // $p_m$ is the mutation probability
    evaluate the interim population $P'(t)$

    // perform elitism-based immigration
    denote the elite in $P(t-1)$ by $E(t-1)$
    generate $r_{ei} \times n$ immigrants by mutating $E(t-1)$ with $p_m^i$
    evaluate these elitism-based immigrants

    if the hybrid scheme is used then        // for HIGA
      generate $r_{ri} \times n$ random immigrants
      evaluate these random immigrants

    replace the worst individuals in $P'(t)$ with the generated immigrants
    $P(t+1) := P'(t)$
  until the termination condition is met      // e.g., $t > t_{max}$
end
```

Fig. 1. Pseudo-code for the elitism-based immigrants GA (EIGA) and the hybrid immigrants GA (HIGA)

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 to the population size is usually set to a small value, e.g., 0.2.

3 The Elitism-Based Immigrants Scheme

As discussed above, traditional random immigrants approach works by replacing random individuals into the population. This may increase the population diversity level and hence may benefit GA's performance in dynamic environments, especially when a change occurs. However, in a slowly changing environment, the introduced random immigrants may divert the searching force of the GA 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 severity of changes, random immigrants may not have any actual effect even when a change occurs because individuals in the previous environment may still be quite fit in the new environment.

Based on the above consideration, this paper proposes an immigrants approach, called *elitism-based immigrants*, for GAs to address DOPs. Fig. 1 shows the pseudo-code for the GA with the proposed elitism-based immigrants scheme, denoted *EIGA* in this paper. Within EIGA, for each generation t, after 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 of $r_{ei} \times n$ individuals are iteratively generated by mutating $E(t-1)$ bitwise with a probability p_m^i, where n is the population size and r_{ei} is the ratio of the number of elitism-based immigrants to the population size. 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 traditional random immigrants scheme. It uses the elite from previous population to guide the immigrants toward the current environment, which is expected to improve GA's performance in dynamic environments.

In order to address significant changes that a DOP may suffer, the elitism-based immigrants can be hybridized with traditional random immigrants scheme. The pseudo-code for the GA with the hybrid immigrants scheme, denoted *HIGA* in this paper, is also shown in Fig. 1. Within HIGA, in addition to the $r_{ei} \times n$ immigrants created from the elite of previous generation, $r_{ri} \times n$ immigrants are also randomly created, where r_{ri} is the ratio of the number of random immigrants to the population size. These two sets of immigrants will then replace the worst individuals in the current population.

4 Dynamic Test Environments

The DOP generator proposed in [13,15] can construct dynamic environments from any binary-encoded stationary function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \{0,1\}^l$) by a bitwise exclusive-or (XOR) operator. The environment is changed every τ generations. For each environmental period k, an XORing mask $\boldsymbol{M}(k)$ is incrementally generated as follows:

$$\boldsymbol{M}(k) = \boldsymbol{M}(k-1) \oplus \boldsymbol{T}(k) \tag{1}$$

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

$$f(\boldsymbol{x}, t) = f(\boldsymbol{x} \oplus \boldsymbol{M}(k)) \tag{2}$$

where $k = \lceil t/\tau \rceil$ is the environmental index. With this generator, the parameter τ controls the change speed while $\rho \in (0.0, 1.0)$ controls the severity of changes. Bigger ρ means severer changes while smaller τ means faster changes.

In this paper, three 100-bit binary-encoded problems are selected as the stationary functions. The first one is the *OneMax* function, which aims to maximize the number of ones in a chromosome. The second one, denoted *Royal Road* due to its similarity to the Royal Road function by Mitchell et. al [8], consists of 25 contiguous 4-bit building blocks. Each building block contributes 4 to the total fitness if all its four bits are set to one; otherwise, it contributes 0. The third problem is a 100-item 0-1 knapsack problem with the weight and profit of each item randomly created in the range of $[1, 30]$ and the capacity of the knapsack set to half of the total weight of all items. The fitness of a feasible solution is the sum of the profits of the selected items. If a solution overfills the knapsack, its fitness is set to the difference between the total weight of all items and the weight of selected items, multiplied by a small factor 10^{-5} to make it in-competitive with those solutions that do not overfill the knapsack.

Dynamic test environments are constructed from the three stationary functions using the aforementioned XOR DOP generator with τ set to 10 and 50 and ρ set to 0.1, 0.2, 0.5, and 1.0 respectively. Totally, a series of 8 DOPs are constructed from each stationary function.

5 Experimental Study

5.1 Experimental Design

In the experiments, four GAs were investigated on the above constructed DOPs. They are the standard GA (SGA), traditional random immigrants GA (denoted RIGA), EIGA and HIGA. All GAs are set as follows: generational, uniform crossover with $p_c = 0.6$, flip mutation with $p_m = 0.01$, and fitness proportionate selection with elitism of size 1. In order to have fair comparisons among GAs, the population size and ratios of immigrants are set such that each GA has 120 fitness evaluations per generation as follows: the population size n is set to 120 for SGA and 100 for RIGA, EIGA and HIGA, the ratio r_{ei} is set to 0.2 for EIGA and 0.1 for HIGA, and r_{ri} is set to 0.2 for RIGA and 0.1 for HIGA. For EIGA and HIGA, p_m^i of bitwise mutating the elite for immigrants is set to 0.01.

For each GA on a DOP, 50 independent runs were executed with the same set of random seeds. For each run of a GA on a DOP, 200 environmental changes were allowed and the best-of-generation fitness was recorded every generation. The overall offline performance of a GA on a DOP is defined as the best-of-generation fitness averaged over the 50 runs and over the data gathering period, as formulated below:

$$\overline{F}_{BOG} = \frac{1}{G} \sum_{i=1}^{G} \left(\frac{1}{N} \sum_{j=1}^{N} F_{BOG_{ij}} \right) \tag{3}$$

where $G = 200 * \tau$ is the total number of 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.

5.2 Experimental Results and Analysis

The experimental results of GAs on the DOPs are presented in Table 1. 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 2. In Table 2, the t-test result regarding Alg. 1 − Alg. 2 is shown as "$s+$", "$s-$", "$+$", or "$-$" when Alg. 1 is significantly better than, significantly worse than, insignificantly better than,

Table 1. Experimental results with respect to overall performance of GAs

Performance	*OneMax*				*Royal Road*				*Knapsack*			
$\tau = 10$, $\rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
SGA	74.0	69.5	64.6	62.0	45.5	36.0	27.1	40.1	1020.4	979.5	933.9	895.1
RIGA	74.4	71.0	66.5	63.8	45.4	36.5	28.3	39.1	1042.3	1000.1	945.6	908.8
EIGA	86.9	77.0	63.7	55.9	53.7	37.5	25.7	46.7	1110.2	1028.3	921.6	871.5
HIGA	82.7	75.3	67.3	63.5	48.3	37.3	28.2	43.4	1054.8	1007.5	946.8	907.7
$\tau = 50$, $\rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
SGA	83.2	79.4	72.4	65.3	67.7	58.7	44.8	41.5	1110.3	1077.4	1011.1	929.7
RIGA	81.9	78.9	75.2	73.8	69.0	59.2	47.0	40.8	1125.4	1095.2	1040.8	1007.6
EIGA	97.6	94.2	81.9	63.6	85.9	72.0	48.5	43.4	1228.6	1183.6	1068.9	923.9
HIGA	94.7	90.2	82.6	80.9	76.1	64.3	49.4	42.7	1149.7	1114.0	1049.4	1013.0

Table 2. The t-test results of comparing GAs on dynamic test problems

t-test Result	*OneMax*				*Royal Road*				*Knapsack*			
$\tau = 10$, $\rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
$RIGA - SGA$	$s+$	$s+$	$s+$	$s+$	$-$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s+$
$EIGA - SGA$	$s+$	$s+$	$s-$	$s-$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s-$	$s-$
$EIGA - RIGA$	$s+$	$s+$	$s-$	$s-$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s-$	$s-$
$HIGA - SGA$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$
$HIGA - RIGA$	$s+$	$s+$	$s+$	$s-$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s+$	$s-$
$HIGA - EIGA$	$s-$	$s-$	$s+$	$s+$	$s-$	$s-$	$s+$	$s-$	$s-$	$s-$	$s+$	$s+$
$\tau = 50$, $\rho \Rightarrow$	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
$RIGA - SGA$	$s-$	$s-$	$s+$	$s+$	$s+$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s+$
$EIGA - SGA$	$s+$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s-$
$EIGA - RIGA$	$s+$	$s+$	$s+$	$s-$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s-$
$HIGA - SGA$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$
$HIGA - RIGA$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$
$HIGA - EIGA$	$s-$	$s-$	$s+$	$s+$	$s-$	$s-$	$s+$	$s-$	$s-$	$s-$	$s-$	$s+$

or insignificantly worse than Alg. 2 respectively. The results are also plotted in Fig. 2. The dynamic behaviour of GAs for the first 10 environments is plotted with respect to best-of-generation fitness against generation on the DOPs with $\tau = 50$ and $\rho = 0.1$ and $\rho = 1.0$ in Fig. 3, where the data were averaged over 50 runs. From the tables and figures several results can be observed.

First, RIGA does significantly outperform SGA on most dynamic test problems, see the t-test results regarding $RIGA-SGA$ in Table 2. This result validates the benefit of introducing random immigrants for the GA for DOPs. However, on the *OneMax* problems with $\tau = 50$ and $\rho = 0.1$ and 0.2, RIGA is beaten by SGA. This confirms our prediction made in Section 3: when the environment changes slowly and slightly, random immigrants may not be beneficial.

Second, EIGA outperforms SGA and RIGA on most DOPs with $\tau = 50$ and on DOPs with $\tau = 10$ and $\rho = 0.1$ and 0.2, see the t-test results regarding $EIGA - SGA$ and $EIGA - RIGA$ in Table 2. This result confirms our expectation of

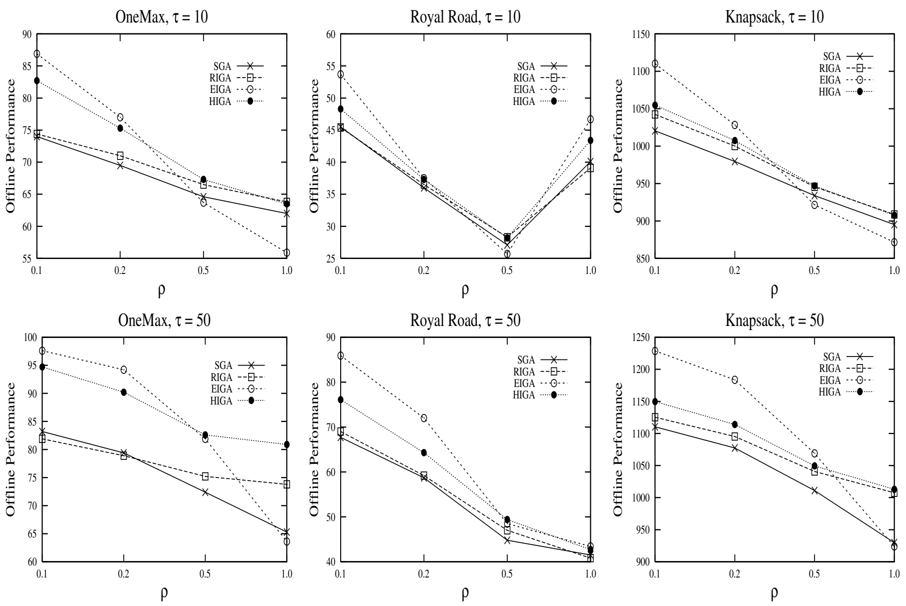

Fig. 2. Experimental results of GAs on the dynamic test problems

the elitism-based immigrants scheme for GAs in dynamic environments. When the environment changes slowly or slightly, it would be better to introduce immigrants guided toward the current environment via the elite. For example, see Fig. 3 for the dynamic behaviour of GAs on DOPs with $\tau = 50$ and $\rho = 0.1$. For each environment EIGA manages to maintain a much higher fitness level than SGA and RIGA.

When the environment changes significantly, e.g., $\rho = 1.0$, EIGA is beaten by SGA and RIGA on dynamic $OneMax$ and $Knapsack$ problems. The reason lies in that each time when the environment changes significantly, the elite from the previous generation may become significantly unfit in the newly changed 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 $Knapsack$ with $\rho = 1.0$ in Fig. 3.

Third, regarding the effect of the hybrid immigrants scheme for GAs, it can be seen that HIGA now outperforms SGA and RIGA on almost all DOPs, see the t-test results regarding $HIGA - SGA$ and $HIGA - RIGA$ in Table 2. This result shows the advantage of the hybrid immigrants scheme over no immigrants and random immigrants schemes. When comparing the performance of HIGA over EIGA, it can be seen that HIGA beats EIGA on DOPs with ρ set to bigger value 0.5 and 1.0 while is beaten by EIGA on DOPs with $\rho = 0.1$ and 0.2. The hybrid immigrants scheme improves the performance of HIGA over EIGA in significantly changing environments at the price of degrading the performance

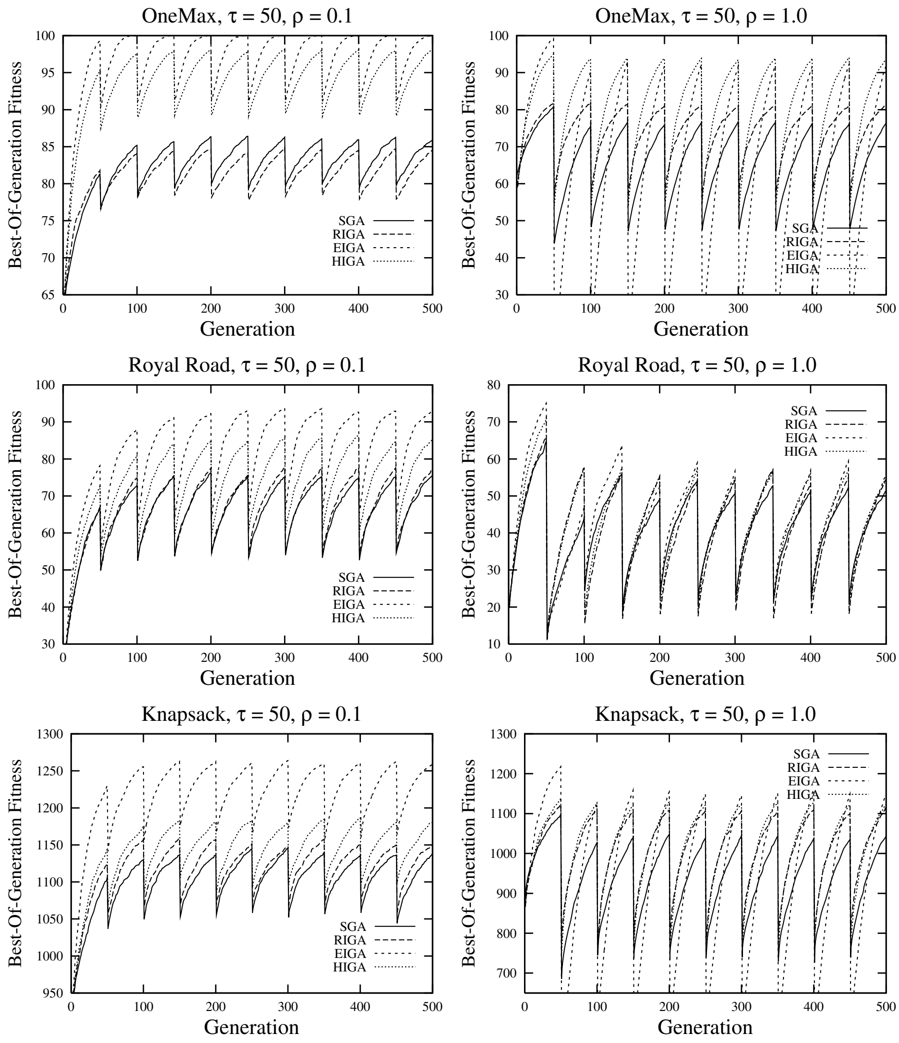

Fig. 3. Dynamic behaviour of GAs on DOPs with $\tau = 50$ for the first 10 environments

in slightly changing environments. This result can be more clearly observed from the dynamic performance of HIGA and EIGA in Fig. 3. The random immigrants added in HIGA prevent HIGA from climbing to the fitness level as high as EIGA does when the environment slightly changes with $\rho = 0.1$ while they also prevent the performance of HIGA from a sharp drop when the environment significantly changes with $\rho = 1.0$.

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

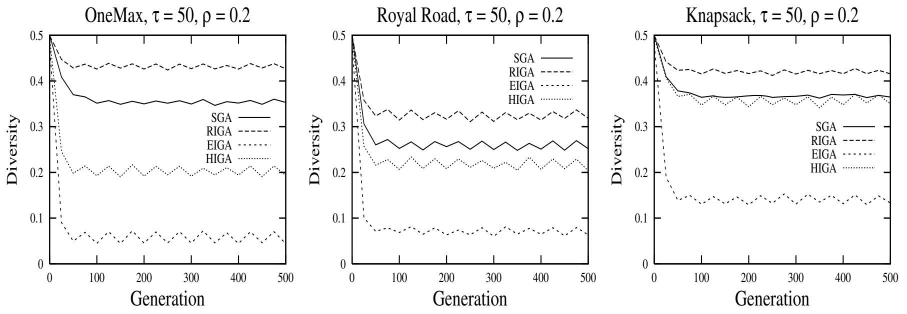

Fig. 4. Diversity dynamics of GAs on DOPs with $\tau = 50$ and $\rho = 0.2$ for the first 10 environments

a GA on a DOP at generation t over 50 runs is calculated according to the following formula:

$$\overline{Div}(t) = \frac{1}{50} \sum_{k=1}^{50} \left(\frac{1}{ln(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} HD_{ij}(k,t) \right), \tag{4}$$

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 generations for GAs on DOPs with $\tau = 50$ and $\rho = 0.2$ is shown in Fig. 4. From Fig. 4, 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.

6 Conclusions

The random immigrants scheme is one of several approaches developed into GAs to address DOPs. This paper proposes an elitism-based immigrants scheme for GAs in dynamic environments, where the elite from last generation is used as the base to create immigrants into the population via a normal bit flip mutation. This way, the introduced immigrants become more adapted to the current environment and hence more efficient in improving GA's performance. The elitism-based immigrants scheme can be combined with traditional random immigrants scheme to further improve the performance of GAs in dynamic environments.

From the experiment results on a series of dynamic problems, the following conclusions can be drawn. First, random immigrants are beneficial for most dynamic environments. Second, the proposed elitism-based immigrants scheme combines the working principles of random immigrants and elitism approaches and improves GA's performance for DOPs, especially in slowly or slightly changing environments. Third, the hybrid immigrants scheme seems a good choice for

GAs for DOPs. Finally, a high diversity level of the population does not always mean better performance of GAs in dynamic environments.

As relevant future work, it is interesting to compare the elitism-based and hybrid immigrants schems with other advanced diversity schemes, e.g., diversity and memory hybrid schemes [10,14], for GAs in dynamic environments. Another interesting work is to further integrate the idea of elitism and immigrants into other approaches, e.g., multi-population and speciation schemes [4,9], to develop advanced diversity schemes for GAs in dynamic environments.

References

1. C. N. Bendtsen and T. Krink. Dynamic memory model for non-stationary optimization. *Proc. of the 2002 Congress on Evol. Comput.*, pp. 145–150, 2002.
2. J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. *Proc. of the 1999 Congr. on Evol. Comput.*, vol. 3, pp. 1875–1882, 1999.
3. J. Branke. *Evolutionary Optimization in Dynamic Environments.* Kluwer Academic Publishers, 2002.
4. J. Branke, T. Kaußler, C. Schmidth, and H. Schmeck. A multi-population approach to dynamic optimization problems. *Proc. of the Adaptive Computing in Design and Manufacturing*, pp. 299–308, 2000.
5. 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.
6. D. E. Goldberg and R. E. Smith. Nonstationary function optimization using genetic algorithms with dominance and diploidy. *Proc. of the 2nd Int. Conf. on Genetic Algorithms*, pp. 59–68, 1987.
7. J. J. Grefenstette. Genetic algorithms for changing environments. *Parallel Problem Solving from Nature II*, pp. 137–144, 1992.
8. M. Mitchell, S. Forrest and J. H. Holland. The royal road for genetic algorithms: fitness landscapes and GA performance. *Proc. of the 1st European Conf. on Artificial Life*, pp. 245–254, 1992.
9. D. Parrott and X. Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Trans. on Evol. Comput.*, **10**(4): 444-458, 2006.
10. 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 Algorithms*, pp. 168-174, 2003.
11. K. Trojanowski and Z. Michalewicz. Searching for optima in non-stationary environments. *Proc. of the 1999 Congress on Evol. Comput.*, pp. 1843–1850, 1999.
12. 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, LNCS*, vol. 1143, pp. 297–304, 1996.
13. S. Yang. Non-stationary problem optimization using the primal-dual genetic algorithm. *Proc. of the 2003 Congr. on Evol. Comput.*, vol. 3, pp. 2246–2253, 2003.
14. S. Yang. Memory-based immigrants for genetic algorithms in dynamic environments. *Proc. of the 2005 Genetic and Evol. Comput. Conf.*, vol. 2, pp. 1115–1122, 2005.
15. S. Yang and X. Yao. Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft Computing*, **9**(11): 815-834, 2005.