Genetic Algorithms with Adaptive Immigrants for Dynamic Environments

Michalis Mavrovouniotis
Centre for Computational Intelligence (CCI),
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester, LE1 9BH, U.K.
Email: mmavrovouniotis@dmu.ac.uk

Shengxiang Yang
Centre for Computational Intelligence (CCI),
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester, LE1 9BH, U.K.
Email: syang@dmu.ac.uk

***Abstract*—One approach integrated with genetic algorithms (GAs) to address dynamic optimization problems (DOPs) is to maintain diversity of the population via introducing immigrants. Many immigrants schemes have been proposed that differ on the way new individuals are generated, e.g., mutating the best individual of the previous environment to generate elitism-based immigrants. This paper examines the performance of elitism-based immigrants GA (EIGA) with different immigrant mutation probabilities and proposes an adaptive mechanism that tends to improve the performance in DOPs. Our experimental study shows that the proposed adaptive immigrants GA outperforms EIGA in almost all dynamic test cases and avoids the tedious work of fine-tuning the immigrant mutation probability parameter.**

I. Introduction

Genetic algorithms (GAs) have proved that they are powerful techniques for solving different optimization problems in real-world applications with stationary environments [7], [21]. However, in many real-world applications, we have to deal with dynamic environments in which the objective function, decision variables, problem instance, constraints, and so on, may vary over time [16].

When a dynamic change occurs, it may take some time for a GA to adapt to the new environment due to premature convergence. A direct solution is to restart the GA whenever a dynamic change occurs and consider every dynamic change as the arrival of a new problem instance that needs to be solved from scratch [6]. However, this solution is not suitable because it generally requires substantial computational effort. Therefore, it could be useful, in terms of computational effort, to utilize the information obtained in the previous environment to find new good solutions in the newly changed environment. This is usually true if the new environment is closely related to the old one [5].

Different strategies have been proposed to address DOPs, such as maintaining diversity via immigrants [4], [15], [22], increase diversity after a change [3], [10], memory schemes [1], [19], multi-population schemes [2], [9], and memetic/hybrid algorithms [11], [12]. Among these approaches, maintaining diversity via immigrants schemes are simple to implement, less computationally expensive, and they are validated to be effective in DOPs [23], [24]. Generally, different immigrants schemes are suitable for different types of DOPs. For example, the elitism-based immigrants GA (EIGA) [15] has shown good performance in slightly and slowly changing DOPs whereas random immigrants GA (RIGA) has shown good performance in severely and rapidly changing DOPs.

In this paper, an adaptive immigrants GA (AIGA) is proposed, which is an extension of the EIGA, to achieve better performance on more types of DOPs. More precisely, in order to generate elitism-based immigrants, the elite of the previous environment is mutated with a fixed probability in EIGA, whereas in order to generate adaptive immigrants the elite of the previous environment is mutated with an adaptive probability in AIGA.

The rest of the paper is outlined as follows. Section II describes the concept of immigrants schemes in GAs. Section III describes the proposed AIGA. Section IV describes the dynamic test environment constructed for the experiments. Section V presents the experimental results and analysis. Section VI concludes this paper.

II. GAs with Immigrants Schemes for DOPs

Conventional GAs cannot adapt well in dynamic environments once they converge. This is because when a dynamic change occurs it may take some time for the GA to escape from the optimum that has currently converged to the previous environment and search for the optimum of the newly generated environment.

Immigrants schemes have proved that they are good methods to integrate with GAs in order to enhance their performance in DOPs [15], [22]–[24]. The basic principle of immigrants schemes is to introduce new individuals into the evolving population that replace a predefined portion of the population (usually the worst individuals). In this way, the diversity of the individuals within the population can be maintained throughout the run.

The traditional way to generate immigrants is randomly. RIGA has shown good performance in environments where the frequency is fast and the magnitude is severe since it enhances the diversity of the GA and addresses the premature convergence problem [4]. However, the continuous adaptation of GAs in dynamic optimization makes sense only when the environments are similar, i.e., when the magnitude of change is small to medium [1]. Therefore, random immigrants may not be suitable in such dynamic cases because they may generate high diversity levels that may lead to randomization and disturb the optimization process.

To address these challenges, EIGA [13] and memory-based immigrants GA (MIGA) [12] were proposed that generate elitism-based and memory-based immigrants, respectively, considering individual-information from previous environments. Their only difference is that MIGA uses information from the best individual of a memory, whereas EIGA from the best individual of the previous generation. Within EIGA, for each generation t, after normal genetic operations (i.e., selection and recombination), the elite $E(t-1)$ from previous generation is used as the base to create immigrants. By mutating $E(t-1)$ bitwise with a probability p_m^i, a set of $n^i = r_i \times n$ individuals are iteratively generated, where n is the population size and r_i the immigrants replacement rate. Then, the worst individuals in the current population are replaced with these newly introduced immigrants. EIGA has shown good performance in environments where the frequency is slow and the magnitude is slight since it uses the idea of elitism to guide the immigrants towards the optimum of environment. Moreover, hybrid immigrants GA (HIGA) was proposed to improve the performance of EIGA in severe environments [17]. Every generation, within HIGA, apart from elitism-based immigrants generated, random immigrants and dualism-based immigrants are also generated. Dualism-based immigrants are generated from mutating the dual of the elite $E(t-1)$. The dual of an individual is the one that is symmetric to it with respect to the central point of the search space.

Another way to maintain diversity via transferring knowledge is the environmental-information-based immigrants GA (EIIGA) [22], which considers environmental-information from the previous environment. Within EIIGA, the allele distribution in the population is calculated at first and then acts as the base to generate immigrants. For generation t, after normal genetic operations, the allele distribution vector is extracted from the current population. For binary encoding, the frequency of ones over the population in a gene locus can be regarded as the allele distribution for that locus. Then a set of n^i environmental-information individuals are generated by sampling the allele distribution vector. Moreover, environmental-information hybrid immigrants (EIHIGA) was proposed [22]. Every generation, within EIHIGA, in addition to the environmental-information immigrants generated via sampling the allele distribution, more immigrants are generated via sampling the complementary allele distribution.

III. Adaptive Immigrants Scheme

A. The Role of p_m^i in Immigrants

The mutation probability of elitism-based immigrants, i.e., p_m^i, is responsible for the diversity generated in the population. If $p_m^i = 0$, then EIGA is the same as a GA with elitism of size r_i because all the generated immigrants will be the same with the elite of the previous environment. If $p_m^i = 1$, then it is the same as a GA with random immigrants because all the generated individuals will be completely random from the elite of the previous generation. Usually p_m^i has a very small value, i.e., $p_m^i = 0.01$, and it is fixed in EIGA.

Considering the performance of RIGA and EIGA on previous experimental studies [15], [23], it is natural to expect that higher p_m^i probabilities may work in favour of environments with severe changes, whereas lower p_m^i probabilities may work in favour of environment with slight changes. In [15], it has been shown that the effect of p_m^i on the performance of the algorithm depends on the DOP.

Moreover, we believe that a fixed p_m^i may not be the best choice for EIGA because the algorithm may need different levels of diversity at different stages of the evolutionary process (generation). For example, in EIGA with the typical $p_m^i = 0.01$, the immigrants generated are more likely to have similar fitness with the best individual of the previous environment. Apart from that these individuals may not be suitable in cases where the environments are not similar; they may also transfer high levels of knowledge and start the optimization in the new environment from a (or near) local optimum solution. Therefore, increasing the mutation probability may address this issue, e.g., escaping from the local optimum, and improve the performance.

B. Evaluating the Effect of Immigrants

In order to evaluate the effect of the newly generate individuals, their fitness is compared with the average fitness of the actual population in every generation. Let $\xi(t)$ denote the effect of the immigrants generated at iteration t, and $\xi(t)$ can be defined as follows:

$$\xi(t) = \frac{n^i\{F_I^k(t)) \geq F_{Avg}(t))\}}{n^i} \tag{1}$$

where n^i is the number of immigrants generated, $F_I^k(t)$ is the fitness of the $k-$th immigrant and $F_{Avg}(t)$ is the average fitness of the current population. A similar method to measure the effect has been proposed in [24] to adapt the immigrants replacement rate, i.e., r_i, using the median of the population, whereas in [8], [18] the average fitness of the population is considered to adapt the mutation probability of each gene within an individual.

C. Adapting the Mutation Probability of Immigrants

In case adaptive immigrants have a positive effect, using Equation (1), the mutation probability increases in order generate higher level of guided diversity; otherwise the mutation probability decreases to avoid randomization.

Given the effect on the previous generation, i.e., $t-1$, the $p_m^i(t)$ parameter is adapted as follows:

$$p_m^i(t) = \begin{cases} p_m^i(t-1) + \sigma, & \text{if } \xi(t-1) > \theta, \\ p_m^i(t-1) - \sigma, & \text{if } \xi(t-1) < \theta, \\ p_m^i(t-1), & \text{otherwise.} \end{cases} \tag{2}$$

where $p_m^i(t)$ is bounded in the interval of $[0, 1]$, σ is a constant value that defines the step size of the mutation

Algorithm 1 AIGA

$t := 0$
initialize population $P(0)$ randomly
evaluate the initial population $P(0)$
while termination condition not satisfied **do**
  $P'(t)$:= selectForReproduction($P(t)$)
  crossover($P'(t)$, p_c)
  mutation($P'(t)$, p_m)
  evaluate interim population $P'(t)$
  $E(t-1)$:= the elite in $P(t-1)$
  generate n^i immigrants by mutating $E(t-1)$ with $p_m^i(t)$
  evaluate these adaptive immigrants
  replace the worst individuals in $P'(t)$ with the generated immigrants
  update $p_m^i(t)$ using Equation (2)
  $P(t+1) := P'(t)$
end while

probability and θ is a threshold that defines whether the effect of the immigrants generated is negative or positive. It can be observed that a new parameter is introduced in the proposed adaptive scheme. However in Section V-C, it can be observed that the θ parameter is not as sensitive as the p_m^i parameter to the performance of the GA.

D. Adaptive Immigrants GA (AIGA)

Different immigrants schemes perform better on different conditions for DOPs [15], [23]. For example, RIGA performs well on rapidly and severely changing environments and EIGA (or MIGA) performs well on slowly and slightly changing environments.

The proposed AIGA aims to perform well across different DOPs due to its adaptive characteristics. Within AIGA, for each generation t, after normal genetic operations, the elite $E(t-1)$ from the previous generation is used as the base to create immigrants as in EIGA. By mutating $E(t-1)$ bitwise with a probability $p_m^i(t)$, a set of $n^i = r_i \times n$ individuals are iteratively generated, where n is the population size and r_i is the immigrants replacement rate. Every generation, $p_m^i(t)$ is adapted using Equation (2) according to the effect of immigrants, calculated in Equation (1), of the previous generation. Then, the worst individuals in the current population are replaced with these newly introduced adaptive immigrants. The pseudocode of AIGA is presented in Algorithm 1.

IV. Dynamic Test Environments

The DOP generator 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 [14], [20]. Suppose the environment changes in every τ algorithmic generations, the dynamics can be formulated as follows:

$$f(\vec{x}, t) = f(\vec{x} \oplus \vec{M}(k)), \tag{3}$$

where $\oplus$ is the XOR operator (i.e., $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 0 = 0$), $k = \lceil t/\tau \rceil$ is the index of the period and $\vec{M}(k)$ is the XORing mask that occurs incrementally and it is defined as follows:

$$\vec{M}(k) = \vec{M}(k-1) \oplus \vec{T}(k), \tag{4}$$

where $\vec{T}(k)$ is an intermediate binary template randomly created with $\rho \times l$ ones. Parameters $\rho \in (0.0, 1.0)$ and τ control the magnitude and frequency of change of a DOP, respectively. Higher value of ρ means severer dynamic changes, whereas a lower value of τ means faster dynamic changes.

In this paper, three 100-bit binary encoded problems are selected as the stationary problems to generate DOPs. Each problem consists of 25 copies of 4-bit building blocks and have optimum of 100. The first one is the OneMax function, which aims to maximize the number of ones in a chromosome. The second one is the Plateau function, where each building block contributes four (or two) to the total fitness if its unitation (i.e., the number of ones inside the building block) is four (or three); otherwise, it contributes zero. The third one is the Deceptive function, where the building block is a fully deceptive sub-function. Generally, the difficulty of the three functions for GAs is increasing in the order from OneMax to Plateau to Deceptive.

Dynamic test environments are generated from the three aforementioned binary-encoded function using the XOR DOP generator with τ set to 10 and 50, indicating fast and slowly changing environments, respectively, and ρ set to 0.1, 0.25, 0.5 and 0.75, indicating slowly,to medium, to severe changing environments, respectively. Totally, a series of 8 DOPs are constructed from each stationary function.

V. Experimental Study

A. Experimental Setup

In the experiments, we investigate the EIGA with different fixed p_m^i values and the proposed AIGA with an adaptive $p_m^i(t)$ value. All GAs were 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. The population size n was set to 100 and r_i was set to 0.3. Hence, $n^i = 30$ for EIGA and AIGA. The initial p_m^i value and σ parameters in AIGA were set to 0.01.

For each GA on a DOP, 30 independent runs were executed on the same set of random seeds. For each run 1000 generations were allowed and the best-of-generation fitness was recorded every generation. The overall offline performance of a GA on DOP is defined as:

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

where G is the total number generations, N is the total number of runs and $F_{BOG_{ij}}$ is the fitness of the best-of-generation individual at generation i of run j. Moreover, the diversity of the population was recorded every generation. The overall diversity of a GA on a DOP is defined as:

$$\bar{T}_{DIV} = \frac{1}{G}\sum_{i=1}^{G}\left(\frac{1}{N}\sum_{j=1}^{N} Div_{ij}\right) \tag{6}$$

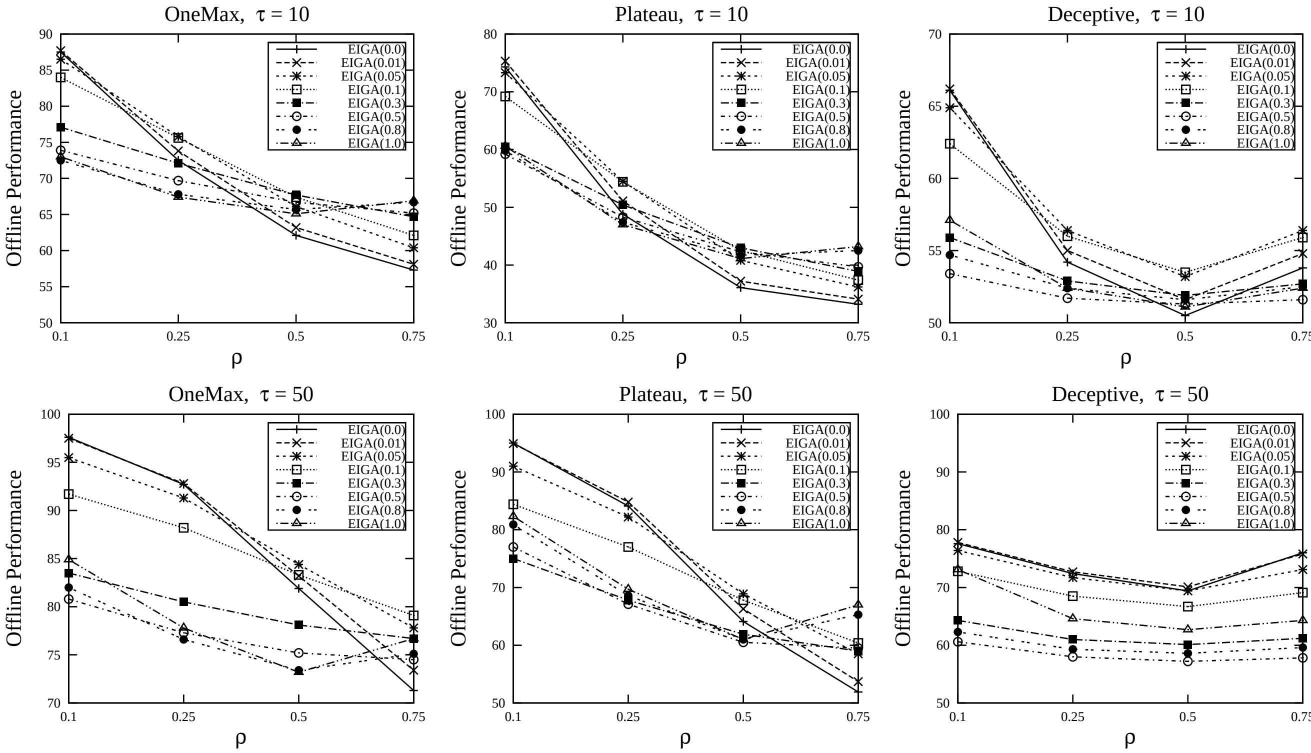

Fig. 1. Offline performance of EIGA with different mutation probabilities p_m^i on different DOPs.

where G and N are defined as in Equation (5) and Div_{ij} is the diversity at generation i of run j, which is defined as:

$$Div_{ij} = \frac{1}{ln(n-1)} \sum_{p=1}^{n} \sum_{q \neq p}^{n} HD(p,q) \tag{7}$$

where l is the encoding length, n is the population size and $HD(p,q)$ is the hamming distance between the p-th individual and q-th individual.

B. Experimental Analysis on the Impact of p_m^i in EIGA's Performance

The offline performance of EIGA for all dynamic test cases with different immigrants mutation probabilities is plotted in Fig. 1, where the EIGA with the immigrants mutation probability p_m^i is denoted as EIGA(p_m^i).

It can be observed that the performance of EIGA(0.0) is degraded as ρ increases for all OneMax and Plateau DOPs. For example, when $\rho = 0.1$ it performs better or has similar performance from other GAs, whereas when $\rho = 0.75$ it has the worst performance clearly. This is because the elitism mechanism works only when the environments are similar.

The performance of EIGA(0.8) and EIGA(1.0) is improved as ρ increases for all OneMax and Plateau DOPs when $\tau = 10$. This is because the environment changes rapidly and there is not enough time to transfer knowledge. Therefore, the generation of random diversity via immigrants is more suitable. There is a similar observation in the case of $\tau = 50$. However there is no need to maintain too much diversity because the environment changes slowly and, thus, EIGA(0.1) may perform well when $\tau = 50$ and $\rho = 0.75$. The low and high levels of diversity of varying p_m^i from a smaller to a bigger value can be also observed in Fig. 5, e.g., for EIGA(0.01) and EIGA(1.0).

In contrast, the performance of EIGA(1.0) is not improved as ρ increases in Deceptive DOPs, either when $\tau = 10$ or $\tau = 50$. It can be observed that the elitism mechanism performs better in all dynamic cases of the Deceptive function because a value $0.0 \leq p_m^i \leq 0.1$ always improves the performance of EIGA.

The above observations support our claim in Section III-A that the value of p_m^i depends on ρ, e.g., as ρ increases a higher p_m^i achieves better performance for EIGA. Moreover, the observations support the claim in [15] that the value of p_m^i also depends on the DOP, e.g., the same p_m^i has a different impact on the performance of EIGA between the Deceptive function and the remaining functions.

C. Experimental Analysis on the Impact of θ in AIGA's Performance

The offline performance of AIGA for all dynamic test cases with different threshold values, used in Equation (2) is plotted in Fig. 2. An AIGA with different threshold values is denoted as AIGA(θ).

It can be observed that AIGA(0.9) achieves better performance in almost all cases. This is natural because the fitness of each elitism-based immigrant generated is more likely to be

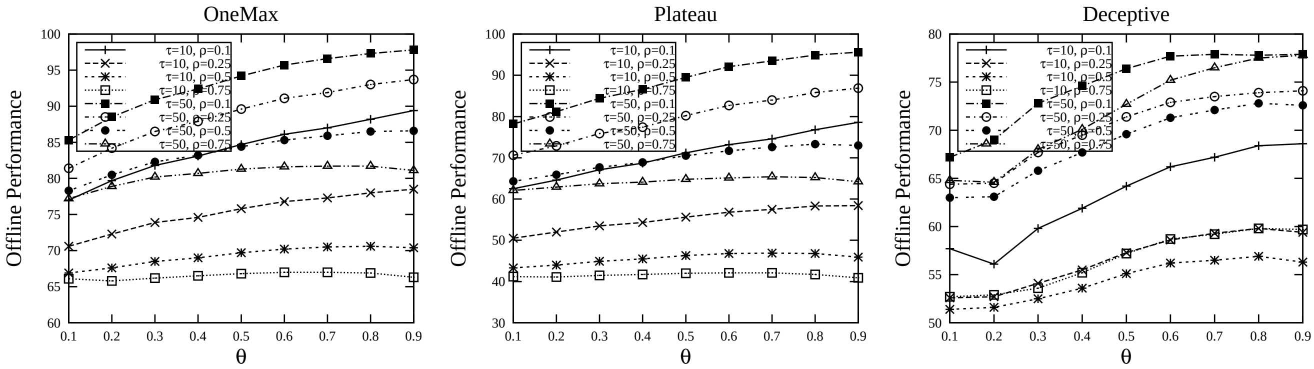

Fig. 2. The impact of the θ value on the performance of AIGA on different DOPs.

TABLE I

EXPERIMENTAL RESULTS AND STATISTICAL TESTS REGARDING THE OFFLINE PERFORMANCE OF AIGA AGAINST EIGA* WITH AN OPTIMIZED p_m^i VALUE IN DOPS. BOLD VALUES ARE SIGNIFICANT AT 0.05 LEVEL OF SIGNIFICANCE BY WILCOXON RANK-SUM TEST

GAs & Functions	OneMax				Plateau				Deceptive			
$\tau = 10$, $\rho \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AIGA	**89.4**	**78.5**	**70.4**	66.3	**78.6**	**58.4**	**45.9**	42.1	**68.6**	**59.4**	**56.3**	**59.7**
EIGA*	87.7	75.8	67.7	**66.9**	75.3	54.5	43.0	**43.2**	66.2	56.4	53.5	56.4
$\tau = 50$, $\rho \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AIGA	**97.8**	**93.7**	**86.6**	**81.1**	**95.6**	**86.9**	**73.0**	65.4	**77.9**	**74.1**	**72.6**	**77.8**
EIGA*	97.6	92.8	84.4	79.1	95.0	84.8	68.9	**67.0**	**77.8**	72.7	70.1	76.0

better than the average fitness of the whole population. Therefore, the effect of the immigrants, as measured in Equation (1), is more likely to be high, and, thus, the immigrants generated have positive effect. This observation supports the claim that the performance of AIGA does not rely on θ.

There are some cases, e.g., DOPs with $\tau = 10$ and $\rho = 0.75$, that AIGA(0.8) performs slightly better than AIGA(0.9) on the Plateau function. However, this shows that θ is generally not sensitive because it does not affect the performance of the algorithm. Probably, a larger step size, i.e., $\sigma = 0.05$, may improve the performance, or even self-adapt the step size may improve the performance.

A value $0.8 \leq \theta \leq 0.9$ can achieve satisfactory performance for all DOPs. Therefore, the tedious work of fine-tuning the value of the immigrants mutation probability in algorithms without adaptation in order to improve the performance slightly for a certain DOP can be avoided using AIGA.

D. Experimental Analysis of Adaptive versus Non-Adaptive Immigrants

The experimental results regarding the overall offline performance of AIGA with the adaptive p_m^i against EIGA with the best fixed p_m^i value for each dynamic case found in Fig. 1, denoted as EIGA*, are presented in Table I with the corresponding statistical results of Wilcoxon rank-sum test, at the 0.05 level of significance. Moreover, the dynamic behaviour of the algorithms regarding overall performance and diversity are presented in Fig. 4 and Fig. 5, respectively. The dynamic behaviour of the adapted p_m^i value of AIGA are presented in Fig. 3. From the experimental results, several observations can be made by comparing the behaviour of the algorithms.

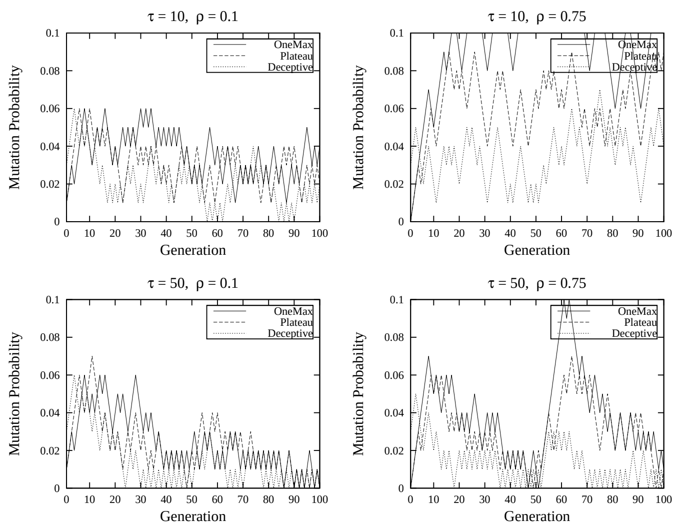

Fig. 3. Dynamic behaviour of the p_m^i value of AIGA on DOPs with $\tau = 10$ and $\rho = 0.1$ and $\rho = 0.75$, respectively, and on DOPs $\tau = 50$ and $\rho = 0.1$ and $\rho = 0.75$, respectively, for the first 100 generations.

First, AIGA performs significantly better than EIGA* in almost all dynamic cases; see the comparisons in Table I. In

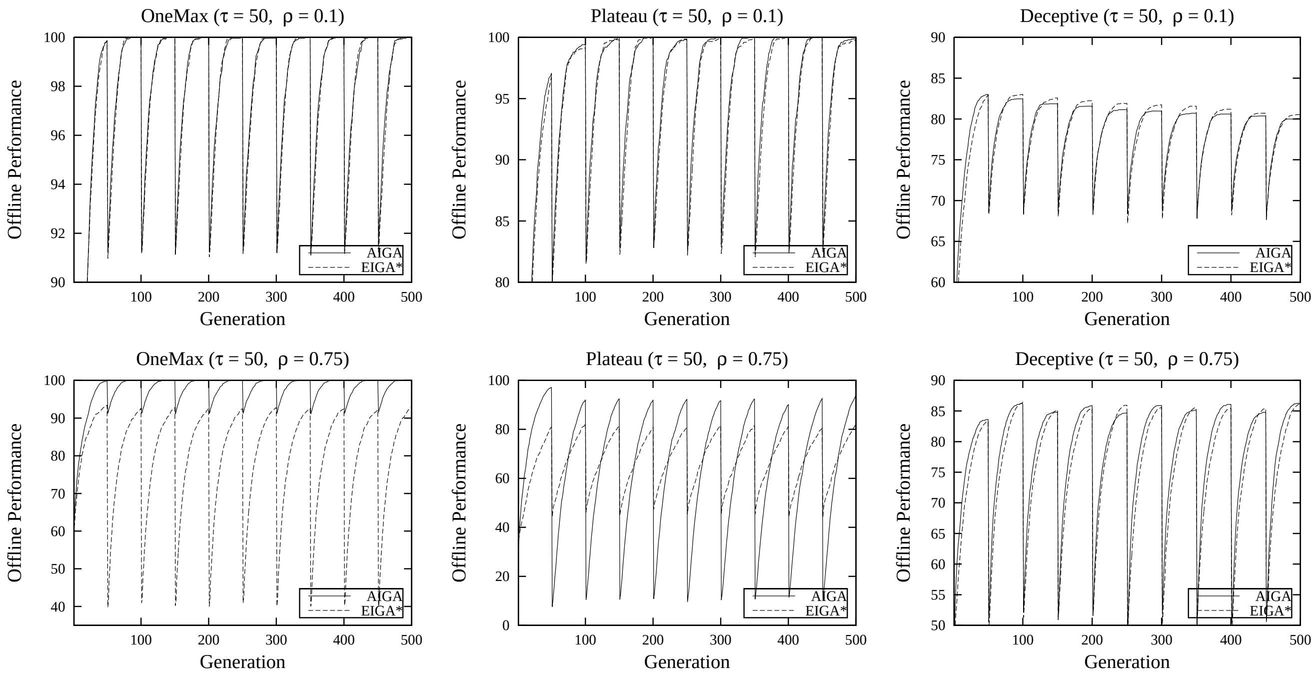

Fig. 4. Dynamic overall offline performance of GAs on DOPs with $\tau = 50$ and $\rho = 0.1$ and $\rho = 0.75$, respectively, for the first 500 generations.

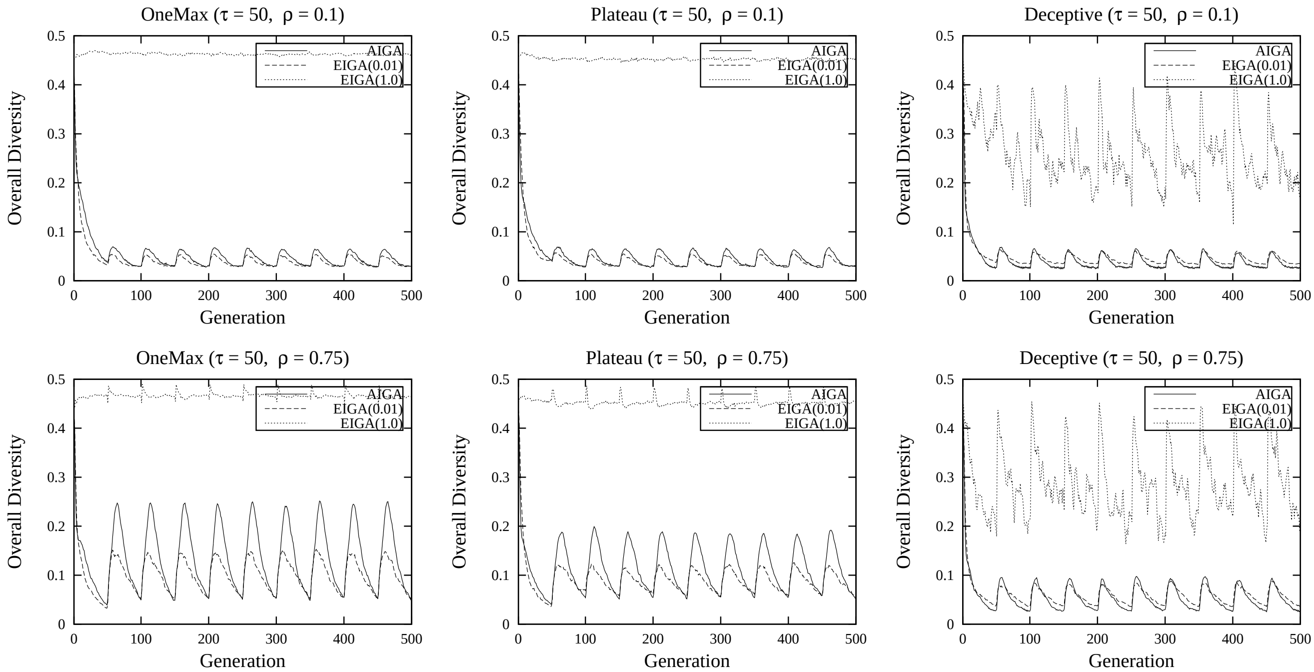

Fig. 5. Dynamic overall diversity of GAs on DOPs with $\tau = 50$ and $\rho = 0.1$ and $\rho = 0.75$, respectively, for the first 500 generations.

some cases of OneMax and Plateau DOPs with $\rho = 0.75$ EIGA* performs significantly better than AIGA. This is because the adaptive mechanism in AIGA may not be able to increase the value of p_m^i as high as in EIGA*, in which $p_m^i = 1.0$ performs better in most severe DOPs; see Fig. 1. This behaviour can be observed from Fig. 3 where the values of the adapted parameter in AIGA is $0.0 \leq p_m^i(t) \leq 0.2$ in different DOPs.

Second, from Fig. 5 it can be observed that AIGA maintains higher diversity than EIGA* throughout the execution for OneMax and Plateau functions. In fact, the difference between the diversity of AIGA and EIGA* is much higher when $\rho = 0.75$ than when $\rho = 0.1$. This shows that the adaptive immigrants in AIGA are able to maintain the diversity in the

TABLE II

EXPERIMENTAL RESULTS AND STATISTICAL TESTS REGARDING THE OFFLINE PERFORMANCE OF AIGA AGAINST OTHER PEER GAS IN DOPS. BOLD VALUES ARE SIGNIFICANT AT 0.05 LEVEL OF SIGNIFICANCE BY THE WILCOXON RANK-SUM TEST.

GAs & Functions	OneMax				Plateau				Deceptive			
$\tau = 10$, $\rho \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
SGA	73.9	67.9	64.5	63.0	58.8	47.0	39.7	36.9	55.3	52.0	51.1	52.0
RIGA	73.9	69.7	66.7	65.3	59.2	48.5	42.2	39.8	53.3	51.7	51.2	51.6
EIGA	87.7	73.8	63.2	58.1	75.3	51.1	37.2	34.1	66.2	55.0	51.6	54.8
AIGA	**89.4**	**78.5**	**70.4**	**66.3**	**78.6**	**58.4**	**45.9**	**42.1**	**68.6**	**59.4**	**56.3**	**59.7**
$\tau = 50$, $\rho \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
SGA	82.7	77.8	72.4	68.5	75.5	66.4	56.3	49.9	64.5	60.3	58.7	61.1
RIGA	80.9	77.3	75.1	74.4	77.1	67.3	60.5	59.5	60.3	57.5	57.5	57.9
EIGA	97.5	92.8	83.2	73.4	94.9	84.8	66.3	53.7	**77.8**	72.7	70.1	75.8
AIGA	**97.8**	**93.7**	**86.6**	**81.1**	**95.6**	**86.9**	**73.0**	**65.4**	**77.9**	**74.1**	**72.6**	**77.8**

population depending on the DOP which can be observed from Fig. 3, where the mutation probability increases between the DOPs with $\rho = 0.1$ and $\rho = 0.75$, especially when $\tau = 10$.

Third, AIGA maintains lower diversity in the Deceptive function before the change occurs and higher after a change in all DOPs; see Fig. 5. This is probably because GAs benefit from the elitism mechanism as shown in the experiments in Section V-B and Fig. 1. Therefore, the adaptive mechanism in AIGA may decrease the p_m^i since it works in favour for the Deceptive function. This can be supported from Fig. 3 because the adapted p_m^i reaches 0 level in many stages of the evolutionary process only in the Deceptive function. Moreover, this behaviour supports our claim in Section III-A that a fixed value of p_m^i may not be the best choice because the algorithm may need different levels of diversity at different stages of the evolutionary process.

E. Experimental Analysis of the Performance between AIGA and Other Peer GAs

The experimental results regarding the overall offline performance of AIGA against other peer GAs are presented in Table II with the corresponding statistical results of the Wilcoxon rank-sum test at the 0.05 level of significance. In this section, EIGA is applied with its traditional p_m^i value, i.e., 0.01. In order to have fair comparisons among GAs, the population size and ratios of immigrants were set such that each GA has 130 fitness evaluations per generation as follows: the population size n was set to 130 for standard GA (SGA) and 100 for RIGA, EIGA and AIGA and the ratio r_i was set to 0.3 for EIGA and RIGA. The rest of the parameters are the same as in the experiments above.

From Table II, it can be clearly observed that the proposed AIGA outperforms its competitors in all DOPs. This confirms our expectation that AIGA may perform well on DOPs of different dynamics. For example, AIGA outperforms RIGA in DOPs with rapidly and severely changing environments, in which RIGA performs usually better than other GAs, and AIGA outperforms EIGA in DOPs with slowly and changing environment, in which EIGA performs usually better than other GAs. This is because AIGA maintains the appropriate level of diversity during different stages of the evolutionary process, which is supported in the above experiments.

VI. CONCLUSIONS AND FUTURE WORK

Immigrants schemes have been successfully applied in GAs to address DOPs. The performance of different immigrants schemes depends on the characteristics of the DOP. In this paper, we propose an adaptive immigrants scheme for GAs in dynamic environments in which the elite of the previous environment is used as the base to generated immigrants via mutation. The immigrants mutation probability is adapted in every generation according to the effect of the immigrants to the population.

From the experimental results on a series of DOPs, the following conclusions can be drawn. First, the immigrant mutation probability is an important parameter, in terms of the performance for GAs, and depends on the DOP. Second, the best immigrant mutation probability varies at different stages of the evolutionary process. Third, AIGA outperforms other GAs in all dynamic test cases. Finally, higher levels of guided diversity does not always achieve better performance for GAs in DOPs.

For future work, it will be interesting to self-adapt the step size of the immigrant mutation rate, which might further improve the performance of AIGA for DOPs. Moreover, other immigrants schemes also have the p_m^i parameter, e.g., memory-based immigrants [12], in which the proposed adaptive mechanism can be applied. Another future work is to furthermore investigate the impact of p_m^i on the performance of GAs with different immigrants replacement rates, i.e., n^i [24]. Probably an adapted p_m^i value may require fewer immigrants and avoid the waste of function evaluations in every generation.

ACKNOWLEDGMENT

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant EP/K001310/1.

REFERENCES

[1] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. 1999 IEEE Congr. on Evol. Comput.*, vol. 3, 1999, pp. 1875–1882.

[2] J. Branke, T. Kauβler, C. Schmidth, H.Schmeck, "A multi-population approach to dynamic optimization problems," in *Proc. of the Adaptive Comput. in Design and Manufacturing*, pp. 299–308, 2000.

[3] H.G. Cobb, "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time- dependent nonstationary environments,", Naval Research Laboratories, Washington, DC, Tech. Rep. AIC-90-001, 1990.

[4] J.J. Grefenstette, "Genetic algorithms for changing environments," *Parallel Problem Solving from Nature*, vol. 2, pp. 137–144, 1992.

[5] Y. Jin and J. Branke. "Evolutionary optimization in uncertain environments - a survey," *IEEE Trans. on Evol. Comput.*, vol. 9, no. 3, pp. 303–317, 2005.

[6] N. Raman and F.B, Talbot, "The job shop tardiness problem: a decomposition approach," *European Journal Operational Research* vol. 69, pp. 187–199, 1993.

[7] M. Tang and X. Yao, "A memetic algorithm for VLSI floor-planning," *IEEE Trans. Syst. Man Cybern. Part B: Cybern.*, vol. 37, no. 1 pp. 62–69, 2007.

[8] S. Uyar, S. Sariel and G. Eryigit, "A gene based adaptive mutation strategy for genetic algorithms," *Proc. 2004 Genetic and Evol. Comput. Conf.*, LNCS, vol. 3103, pp. 271–281, 2004

[9] R.K. Ursem RK, "Multinational GAs: multimodal optimization techniques in dynamic environments," in *Proc. 2000 Genetic and Evol. Comput. Conf.*, pp. 19–26, 2000.

[10] F. Vavak F, T.C. Fogarty, K. Jukes, "A genetic algorithm with variable range of local search for tracking changing environments.," in *Proc. 4th Intern. Parallel Problem Solcing From Nature*, LNCS, vol. 1141, pp. 376–385, 1996.

[11] H. Wang, D. Wang and S. Yang, "A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems," *Soft Computing*, vol. 13, no. 8–9, pp. 763-780, 2009.

[12] S. Yang, "Memory-based immigrants for genetic algorithms in dynamic environments," in *Proc. 2005 Genetic and Evol. Comput. Conf.*, vol. 2, pp. 1115–1122, 2005.

[13] S. Yang, "Genetic algorithms with elitism-based immigrants for changing optimization problems" in *Applications of Evolu. Comput.*, LNCS, vol. 4448, pp. 627–636, 2007.

[14] S. Yang, "Non-stationary problem optimization using the primal-dual genetic algorithm", in *Proc. 2003 IEEE Congress on Evol. Comput.*, vol. 3, pp. 2246–2253, 2003.

[15] S. Yang, "Genetic algorithms with memory and elitism based immigrants in dynamic environments," *Evol. Comput.*, vol. 16, no. 3, pp. 385–416, 2008.

[16] S. Yang, H. Cheng and F. Wang, "Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks," *IEEE Trans. Syst., Man, and Cybern. Part C: Appl. and Rev.*, vol. 40, no. 1, pp. 52-63, 2010.

[17] S Yang and R. Tinos, "A hybrid immigrants scheme for genetic algorithms in dynamic environments," *International Journal Automated Computing*, vol. 4, no. 3, pp. 243–254, 2007.

[18] S. Yang and S. Uyar, "Adaptive mutation with fitness and allele distribution correlation for genetic algorithms", *Proc. 21st ACM Symposium on Applied Comput.*, pp. 940–944, 2006.

[19] S. Yang and X. Yao, "Population-based incremental learning with associative memory for dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 542–561, 2008.

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

[21] J.X. Yu, X. Yao, C-.H Choi and G. Gou, "Materialized view selection as constrained evolutionary optimization," *IEEE Trans. Syst. Man Cybern. Part C: Appl. and Rev.*, vol 33, no. 4, pp. 458–467, 2003.

[22] X. Yu, K. Tang and X. Yao, "An immigrants scheme based on environmental information for genetic algorithms in changing environments," in *Proc. 2008 IEEE Congress on Evolut. Comput.*, pp 1141–1147, 2008.

[23] X. Yu, K. Tang, T. Chen and X. Yao, "Empirical Analysis of Evolutionary Algorithms with Immigrants Schemes for Dynamic Optimization,"*Memetic Computing*, vol. 1, no. 1, pp. 3–24, 2009.

[24] X. Yu, K. Tang and X. Yao, "Immigrant schemes for evolutionary algorithms in dynamic environments: Adapting the replacement rate," *Science China Information Sciences*, vol. 54, no. 7, pp. 1352–1364, 2011.