

# Multi-Population Methods with Adaptive Mutation for Multi-Modal Optimization Problems

Imtiaz Korejo<sup>1</sup>, Shengxiang Yang<sup>2</sup>(IEEE Member), Kamran Brohi<sup>3</sup>, Z.U.A. Khuhro<sup>4</sup>

<sup>1</sup> Institute of Mathematics & Computer Science, University of Sindh, Jamshoro, Pakistan

cskorejo@gmail.com

<sup>2</sup>School of Computer Science and Informatics, De Montfort University, Leicester, United Kingdom

syang@dmu.ac.uk

## ABSTRACT

*This paper presents an efficient scheme to locate multiple peaks on multi-modal optimization problems by using genetic algorithms (GAs). The premature convergence problem shows due to the loss of diversity, the multi-population technique can be applied to maintain the diversity in the population and the convergence capacity of GAs. The proposed scheme is the combination of multi-population with adaptive mutation operator, which determines two different mutation probabilities for different sites of the solutions. The probabilities are updated by the fitness and distribution of solutions in the search space during the evolution process. The experimental results demonstrate the performance of the proposed algorithm based on a set of benchmark problems in comparison with relevant algorithms.*

## KEYWORDS

*Multi-population approaches, adaptive mutation operator, multi-modal function optimization, genetic algorithms.*

## 1. INTRODUCTION

Genetic algorithms (GAs) are adaptive heuristic search techniques inspired by genetic inheritance and natural evolution of species. Over the years, GAs have been successfully used for solving many optimization problems due to the properties of easy-to-use and robustness for determining candidate solution for ambitious problems. There are a lot of applications of GAs in different areas of real world problems, such as science, engineering, business, and social science. The interested reader may wish to consult [11]. The basic concept of GAs was introduced by John Holland in the 1960s in the USA. There are different components of GAs, such as the encoding scheme, the population size, the selection approaches, the crossover and mutation operators; the performance of GAs depends on good choice of operators and relevant parameters. However, it is difficult to choose proper operators and relevant parameters for the optimal performance. A limited amount of research has been done on GAs to find multiple optima for multi-modal optimization problems. However, the implicit parallelism of GAs is able to find several optima (or multiple optima including global and local optima) in the search space instead of just a single solution. Recently, researchers have shown a great interest in optimizing multimodal functions with GAs. Several GA techniques have been proposed to optimize multi-modal problems.

In recent years, various techniques have been suggested into standard evolutionary algorithms (EAs) to divide the population proposed to optimize multi-modal problems. In recent years, various techniques have been suggested into standard evolutionary algorithms (EAs) to divide the population into different sub-populations, including k-means clustering algorithm [16], self-organizing scouts (SOS) [5], multinational GAs [38], clustering scheme [27], niching

method [28], and a hierarchical clustering [46]. These techniques have considered sub-populations as a means of increasing the diversity of GAs. The main motivation behind these schemes is to maintain multiple populations on different peaks, the highest peak in the search space is the global optimum and other peaks with a lower height than the global optimum are local optima.

There are different schemes for controlling the values of relevant parameters and genetic operators of an EA. Parameter control or adaptive adaptation techniques can be classified into three categories: deterministic adaptation, adaptive adaptation, and self-adaptive adaptation approaches. Deterministic adaptation adjusts the values of parameter according to some deterministic rule without using any feedback information from the search space. The strategy variables are modified by using the feedback information from the search space. This is called adaptive adaptation. The best example of this type of adaptive adaptations is Rechenberg's "success rule" in evolutionary strategies (ESs) [31] (p. 110). In the case of self-adaptive adaptation, parameters are altered by EAs themselves. Strategy parameters to be altered are encoded into chromosomes and undergo variation with the rest of the chromosome. The self-adaptive adaptation was proposed by Back and Schlutz [3] into GAs.

In this paper, a new multi-population with adaptive mutation operator is proposed for GAs. Two different multi-population approaches are considered in this study, one is species with adaptive mutation (SWAM) and another is basin with adaptive mutation (BWAM). The proposed technique can maintain the exploration and exploitation in the evolution process, which can be considered into the category of adaptive adaptation EAs. This paper gives the comparison of several adaptive mutation operators, including adaptive GA, self-adaptive GA, and site-specific rate GA with our proposed technique. The experimental results demonstrate, the proposed algorithm exhibits better performance in terms of the quality of candidate solutions found on the benchmark problems.

The rest of this paper is outlined as follows. Section 2 explains related work, the classification of adaptation in EAs, and some sub-population approaches. Section 3 presents the SWAM and BWAM adaptive mutation operators in detail. Section 4 describes the experimental study of comparing our proposed techniques with other algorithms. Finally, conclusion and discussions on further directions of research are stated in Section 5.

## 2. RELATED RESEARCH

Generally speaking, a lot of research has been done on selecting appropriate genetic operators and relevant parameters for GAs to find optimal or near-optimal solutions. Usually, these genetic operators and relevant parameters are fixed before the execution of GAs. For example, some constant mutation probabilities are derived from experience or by trial-and-error. It is very difficult, if not impossible, to find suitable parameter setting for the optimal performance of GAs. Different values of parameters and operators might be optimal at different levels of the evolutionary process. Due to this reason, researchers are turning to adaptive operators and parameters for GAs to find the optimal results.

### 2.1. Adaptation in GAs

The main motivation of adapting a GA is to adjust genetic operators and relevant parameters in order to find better solutions of a problem. Adaptation in GAs is one of the most important and promising research areas. The efficiency of GAs is dependent on not only the algorithms, representation, and operators for the problem, but the set of the parameter values for GAs to find a good solution. There are different methods for controlling the values of various parameters and genetic operators of GAs. Generally, there are two classification schemes [1], [8], [9]. The classification of adaptation based on these two distinct schemes are explained as follows:

- 1) Type of adaptation: the type of adaptation means how a parameter is modified.

2) Level of adaptation: the level of adaptation means where changes occur.

The above classification schemes were first introduced by Angeline in [1]. According to Angeline, the type of adaptation is further divided into absolute update rules and empirical update rules. The level of adaptation is also further classified into three levels, i.e., population level, individual level, and component level. In [8], [9], the authors extended the classification schemes and broadened the idea introduced by Angeline in [1]. They further classified the type of adaptation in two subfields: parameter tuning and parameter control. This is based on the technique of adaptation applied in the process. It is explained as follows

1) Parameter Tuning or Static Adaptation: Parameter tuning (or static adaptation) means to set the suitable parameters before the run of EAs and they remain constant during the execution of EAs. In other words, no change in the parameters occurs. Static adaptation is done by an external process (e.g., a person or a program) for selecting appropriate values. Many researchers have suggested different static appropriate values for key parameters of GAs. These values are derived from experiences or by trial-and-error methods. In [18], De Jong proposed static single-point crossover and mutation probabilities for finding a good solution for problems using a traditional GA. Some other researchers also suggested well known heuristics with other static parameters for finding a good solution for problems [2], [11], [30], [25], [23], [24]. The different values of parameters and operators might be optimal at different levels of the evolutionary process. Due to this reason, static adaptation (parameter tuning) is gradually discouraged in the EA community. So, there is no common optimal parameter setting that can be found a priori (consult to No Free Lunch Theorem [42]). According to this problem, researchers diverted their attention towards adaptive adaptation (parameter control) methods.

2) Parameter Control or Adaptive Adaptation: Strategy parameters are adjusted on the basis of various methods during the execution of EAs. This class of adaptation can be further divided into three classes, described as deterministic adaptation, adaptive adaptation, and self-adaptive adaptation. Deterministic adaptation adjusts the values of parameter according to some deterministic rule without using any feedback information from the search space. Adaptive adaptation modifies strategy variables by using feedback information from the search space. The best example of adaptive adaptations is Rechenberg's "1/5 success rule" in ESs [31] (p. 110). In the case of self-adaptive adaptation, the parameters are altered by EAs themselves. Strategy parameters to be altered are encoded into chromosomes and undergo variation with the rest of the chromosome. The self-adaptive adaptation was first proposed by Schwefel [32] into ESs.

There are three distinct levels at which adaptation can take place in adaptive EAs [34]. For the population-level adaptation, strategy parameters are modified globally for the whole population. Various examples are available for population-level adaptation in ESs and GAs [43], [44]. For the individual level adaptation, an EA modifies the strategy parameters of an individual and these modifications occur independently with each individual. In [41], the crossover probability in a GA is changed at the individual level. For the component level adaptation, parameters are changed for some genes of an individual in the population. Self-adaptation is a well-known example of component-level adaptation.

Fogarty [10] proposed a dynamic mutation rate control scheme for GAs, in which the mutation rate decreases exponentially over the number of generations. In papers [6], [7], Devis proposed an effective algorithm that updates operator probabilities according to the performance of the operators. This scheme specifies for the modification of operator rates in associated to the fitness of solutions generated by the operators. It has been introduced for the steady state GA. Srinivas and Patnaik [37] suggested an adaptive GA that uses an adaptive crossover and mutation probability scheme. The rates of crossover and mutation are modified according to the fitness of the solution. Uyar et al. proposed an asymmetric gene based adaptive mutation (GBAM) technique [39]. In GBAM, each gene locus has two different mutation probabilities:  $pm^1$  is used for those loci that have the value of "1" and  $pm^0$  is used for those loci that have the value of "0". The probabilities of  $pm^1$  and  $pm^0$  are automatically updated based on the

feedback information from the search space, according to the relative success or failure of those chromosomes having a “1” or “0” at that locus for each generation. Yang and Uyar [45] suggested another gene based adaptive mutation with fitness and allele distribution correlation (GBAM FAD). The mutation rates of each gene locus are adaptively modified based on the correlated feedback information from the search process, according to the relative success or failure of solutions.

A dynamic mutation GA (DMGA) was proposed by [15]. This approach simultaneously uses four mutation operators in solving optimization problems. These mutation operators are adaptively applied in the GA. In [19], the authors introduced an adaptive mutation technique, which is closely related to Hong’s method, simultaneously applying multiple mutation operators in GAs [15]. The authors proposed a directed mutation (DM) operator for real coded GAs [17], which is to explore the promising area in the search space by using the feedback information from the current population.

This is an altered version of the standard mutation. In the DM scheme, solution moving is not only based on the feedback information of average fitness of intervals, but also on the population distribution. By taking into account the information of population distribution, DM avoids the premature convergence problem. an adaptive mutation technique, which is closely related to Hong’s method, simultaneously applying multiple mutation operators in GAs [15]. The authors proposed a directed mutation (DM) operator for real coded GAs [17], which is to explore the promising area in the search space by using the feedback information from the current population. This is an altered version of the standard mutation. In the DM scheme, solution moving is not only based on the feedback information of average fitness of intervals, but also on the population distribution. By taking into account the information of population distribution, DM avoids the premature convergence problem.

## 2.2. Multi-population Approaches

Several researchers have considered multi-population schemes for locating multiple peaks in a multi-modal fitness landscape. These approaches can be applied to cover solutions on different peaks with different multi-populations. In this paper, we focus on more than one peak in the fitness landscape. A global optimum is the best peak in the fitness landscape and local optima are referred to the rest of the lower height peaks. Some other techniques have been proposed to maintain the diversity of the population. These approaches work just like multi-population schemes, which are crowding [21], fitness sharing [12], island and parallel GAs [13], [4]. Multi-population techniques can improve the diversity in the GA and avoid the premature convergence problem.

Branke et al. [5] introduced a self-organizing scout (SOS) algorithm, which has shown promising results on evaluated problems. In this scheme, the population consists of a parent population that explores through the whole search space and child populations that locate previously detected optima. The parent population is continuously examined. If the condition is satisfied, then a child population is split off from the parent population. The size of each child population is updated regularly, although the total number of solutions are fixed and no new solutions are proposed.

Ursem [38] proposed a multi-population scheme referred to as the multinational GA, which employs a multinational GA technique to find multiple peaks of multi-modal problems in dynamic environments. Each nation corresponds to the policy and best representatives. In multinational GAs, a “hill-valley detection” approach is applied to sample points on a line marked between policies. This procedure uses migration of solutions from one nation to another, to merge nations, and to create a new nation in a newly located region.

A clustering approach for multi-modal functions was proposed in [27]. This approach determines the number of clusters within the k-means particle swarm optimization (PSO) algorithm. It has been applied for the optimization of a criterion function in a probabilistic mixture-model framework. In this algorithm, particles are supposed to be generated by a mix of

several probabilistic distributions. Each different cluster is associated to a different distribution. Then, locating the optimum number  $k$  is equivalent to adapting the model with the determined data while optimizing some criterion.

The niching method [28] and the species conserving GA (SCGA) [20] have been used for multi-modal optimization problems. Both the clearing (niching) procedure and SCGA adapted a scheme for splitting population according to the nation of species, which is incorporated under the evolution process of a standard GA. The two methods have shown to provide best results on evaluated multi-modal problems.

Yang and Li [46] developed a hierarchical clustering based PSO algorithm which is specially introduced for finding and tracking multiple optima in dynamic environments. This approach can distribute particles in separate promising subregions and the main contribution of this approach is adaptively modifying the number of sub-swarms required, and automatically compute the search region for each sub-swarm. A fast local search method is incorporated in this approach, which specifies the optimal solutions in promising subregion detected by the clustering scheme.

### **3. MULTI-POPULATION SCHEMES WITH ADAPTIVE MUTATION OPERATOR**

#### **3.1 Motivation**

In order to explain the problem of premature convergence is due to the loss of diversity, the multi-population approach can be applied to maintain the diversity and retain the convergence capacity of GAs. The suggested technique is composed of multi-population with adaptive mutation operator, which identifies two different mutation rate vectors for each subpopulation. These vectors are updated by using the fitness and distribution of individuals in the fitness landscape. Each site of solution having two different mutation rates. This approach can be able to improve the performance of the algorithm and to avoid the premature convergence to multi-modal optimization problems.

#### **3.2 Multi-population Schemes**

1) Species Based Multi-population approach: A species can be a collection of individuals sharing with common characteristic regarding to the similarity of the individuals. The measurement of two individuals similarity is calculated by the hamming distance. The smaller hamming distance between two individuals, the more similar they are. Species-based particle swarm optimization (SPSO) scheme was proposed by Parrot and Li [26].

2) An Other Multi-population approach: In this paper [40], the authors has recently proposed a partitioned technique, which divides the current population into different subpopulation according to the fitness and distribution of individuals in the landscape. Each subpopulation can be able to ideally incorporate the basins of attraction of similar optima. The basic idea of proposed partition scheme is to distribute best individuals away from each other (according to the distance measure) under the different peak in the search space.

#### **3.3 Adaptive Mutation Operator with Multi-population approaches**

Traditionally, mutation operator considered as a background operator, it affects the alteration of the value of each bit of individual with small mutation probability  $pm \in [0,1]$ . The main motivation of the mutation in the GAs has been that of insurance policy against permanent loss or unexplored genetic material in the population to avoid the premature convergence of the GAs to local optima.

The adaptive techniques are used to accelerate the convergence speed and preserving GAs from being trapped into local optima. The important contribution of adaptive mutation operator to improve the performance of GAs. All above explained adaptive schemes are used to

a single population in the process of evolution, either using steady state GA or generational GAs.

In this paper, multi-population with adaptive mutation scheme is proposed. Basically adaptive mutation operator is combining with two different multi-population approaches, one is species based multi-population and the other one is basin (fitness of solution and distribution of individual in the search space) based approach. BWAM and SWAM represents basin with adaptive mutation and species with adaptive mutation respectively. GAs has two properties exploration and exploitation but the problem is that how to balance these properties during the evolution. In order to address this problem, the researchers have been used separately multi-population approaches and adaptive probabilities of genetic operators to increase the diversity during the evolution. Our proposed scheme is going to combine both of them for finding the better solution in the multi-modal problems. This technique is applied to maintain the balance of explorative and exploitative properties of genetic operator.

The framework of the multi-population GAs with the adaptive mutation scheme is represented in Algorithm 1. This algorithm starts from randomly generated population which is called  $p$  population. Then, sub-populations are created either using BWGA or SWGA scheme. When sub-populations have been generated then, calculate mutation probability, apply mutation and selection approaches on them. At the end, overlapping and convergence techniques are applied on each sub-population before starts the next generation. Some of the task in this algorithm are similar to the classical SGA but here we are going to focus on the main operations of this approach, including partition, Statistics, Mutation, overlapping and convergence methods. These methods determine the general theme of technique. The detail description of the these functions are given in the following:

**Algorithm 1:** Multi-population scheme with adaptive mutation operator

```

1: Randomly create an initial population  $p$ ;
2: Evaluate the fitness of each solution of  $p$ ;
3:  $t := 0$ ;
4  $mlst_t := \text{partition}(p_t)$ ;
5: while ( $t < \text{max gen}$ ) do
6:   for ( $i := \text{to } mlst_t$ ) do
7:     Statistics( $mlst_t^i$ );
8:     Mutation( $mlst_t^i$ );
9:     Selection( $mlst_t^i$ );
10:  end for
11:  Overlapping(  $mlst_t$ );
12:  Convergence(  $mlst_t$ );
13:  if ( $p < \text{MinInds}$ ) then
14:    regenerate:=Re-initialization(removedSols);
15:     $tlst_t := \text{partition}(\text{regenerate}p_t)$ ;
16:    Merge  $mlst_t$  and  $tlst_t$  into  $mlst_t$ 
17:  end if
18:   $t := t + 1$ ;
19: end while

```

1) partition(pt): There are several techniques available in the literature to generate multi-population, two approaches are considered in this paper, detail explanation of these techniques are mentioned in the section 3.2.  $p_t$  represents the current population, which is passed to the partition function as a parameter. The functionality of this method is to divide the current population into different subpopulation. The different sub-population have different position in

the fitness landscape. The multi-population scheme can be applied to maintain the diversity according to the convergence. This approach can be helpful towards getting the better results on multi-modal problems. Each sub-population has assigned and update probabilities separately. In the following section will compute the probability of each site of individual of each sub-population.

2)Statistics: In the statistics method, the mutation rates are computed by using the feedback information of current sub-population. Each sub-population has defined two different probability vectors, where  $M_{rt}^{0i}$  and  $M_{rt}^{1i}$  are the mutation rates  $M_r^0$  and  $M_r^1$  of subpopulation  $i$  at generation  $t$ . Each vector contains set of element, which is associating to the probabilities of keeping a particular allele at each site. Initially each element of vector is set to an initial value within its determined boundaries. These vectors are generated real-valued probabilities of the each sub-population, which determines high quality sub-population with high probability. For each generation vectors are updated based on the fitness and distribution of solutions in the specified sub-population, it means that each sub-population has altered own vectors independently. The two different mutation rates are specified for each site of solution of each sub-population. The estimation of both vectors are updated on the explication shown in equation 2 and 3.

$$M_{rt}^{0i}(j, t + 1) = \begin{cases} M_{rt}^{0i}(j, t) + \gamma, & \text{if } \bar{G}_{jt}^{1i} > \bar{P}_t^i \\ M_{rt}^{0i}(j, t) - \gamma, & \text{otherwise} \end{cases} \quad (1)$$

$$M_{rt}^{1i}(j, t + 1) = \begin{cases} M_{rt}^{1i}(j, t) - \gamma, & \text{if } \bar{G}_{jt}^{1i} > \bar{P}_t^i \\ M_{rt}^{1i}(j, t) + \gamma, & \text{otherwise} \end{cases} \quad (2)$$

where  $\gamma$  is the updated value of mutation probabilities of the vectors,  $\bar{G}_{jt}^{1i}$  is the average fitness with allele "1" of those individuals which are contained in subpopulation  $i$  for site  $j$  at generation  $t$ , and  $\bar{P}_t^i$  is the average fitness of the subpopulation  $i$  at iteration  $t$ .

3) Mutation( $p_t^i$ ): This function is performed mutation on each sub-population at every iteration. It is distinctly employed on each subpopulation using the associated vectors. Suppose, an offspring ( $\hat{x} = \{\hat{x}_1, \dots, \hat{x}_l\}$ ) generated by mutation of parent ( $x = \{x_1, \dots, x_l\}$ ) according to corresponding vectors, which is the member of subpopulation. The mutation is performed on whole solution site by site, which is mentioned in the Algorithm 2, where Rnd is the uniform random number, which is generated within the interval(0,1).

**Algorithm 2** Mutated each Individual by using adaptive mutation

```

1: for (j := 0 to 1) do
2:   if (x[j] == 0) then
3:     if (Rand(0,1) < Mrt0i[j]) then
4:       x̂[j] = 1;
5:     end if
6:   else
7:     if (Rand(0,1) < Mrt1i[j]) then
8:       x̂[j] = 0;
9:     end if
10:  end if
11: end for

```

The proposed mutation scheme is applied with two different partition approaches (See Section 3.2). The contribution of the adaptive mutation with multiple population is summarized in the following distinct characteristics:

- 1) It describes the two different vectors for each subpopulation.
- 2) Each site of subpopulation having two different mutation rates and different mutation probabilities for the different locus of the encoded individual.
- 3) The rate of each locus are updated iteration-by-iteration to modify the current state of the evolution.
- 4) This is the one of the important property of this approach, which investigates the undetected regions around the search space.
- 5) The suggested method is focused on the locating multiple optima of the multi-modal optimization problems.

### 3.4 Overlapping and Convergence of sub-populations

After mutation and selection operations, all the subpopulations are checked regarding overlapping and convergence. Normally, the overlapping check between two subpopulations can be determined by comparing the distance of the best solutions of the sub-populations. If the search radius of best individuals of sub-populations is less than threshold value *minDist*, then delete the sub-population, which is related to less fitness best solution and keep the total individuals, which are associated to the removed sub-population. Algorithm 3 is shown the overlapping search of sub-populations.

**Algorithm 3:** The algorithm for overlapping search

```

1: while ((mlst.size() > 0) AND (outer < mlst.size())) do
2:   b1 is the best solution of mlst(outer);
3:   inner := 0;
4:   inner := outer + 1;
5:   while ((mlst.size() > 0) AND (inner < mlst.size())) do
6:     if (distance(b1, b2) < minDist) then
7:       if (b1 < b2) then
8:         removedSols := removedSols + mlst(outer).size();
9:         delete mlst(outer);
10:      else
11:        removedSols := removedSols + mlst(inner).size();
12:        delete mlst(inner);
13:      end if
14:    end if
15:  end while
16: end while

```

After overlapping search, all sub-populations will undergo the convergence process, which is determined to see whether sub-population has converged. If the total similar solutions of associated sub-population is greater than threshold value *Lmt\_conv*, the value of threshold (0.95\**size* of subpopulation) is fixed in this paper. If sub-population is converged on one of the available peaks then sub-population is removed from *mlst*, which is converged. Convergence of subpopulation is given in the algorithm 4.

**Algorithm 4** Algorithm for determine convergence of subpopulation

```

1: for each sub_population s mlst do
2:   if (totSimillnds > (0.95 * s.size())) then

```



```
3:   removedSols := removedSols + s.size();
4:   delete s from mlst;
5:   end if
6: end for
```

## 4. EXPERIMENTAL STUDY

In this section, there are three different group of experiments were conducted on well-known decision problems, which has been evaluated in the popular area of maximum satisfiability (MAX-SAT) problems. Boolean satisfiability problem involves finding an assignment of variables that maximizes the number of satisfied clauses of a given constraints. The general form of these constraints are presented in conjunctive normal form (CNF) or product-of-sum form. MAX-SAT is also well known NP-hard optimization problems. In addition, Boolean satisfiability expression has applied to introduced a approach for generating problems with controllable degree of "multimodality" and "epistasis".

In the rest of this section, first introduce the mapping between GAs to MAX-SAT problems, spear [36] for generating constraints with controllable degree of multi-modality and epistasis and then investigate the result of experiments.

### 4.1 Boolean Satisfiability and Genetic Algorithms

In order to employ GAs to any particular problem, that problem consider two critical components: 1) specify the appropriate representation for the solution space (search space) and 2) define an external evolution function which determines utility of individuals (fitness of individuals). MAX-SAT has a simple string representation which is highly compatible to the GAs. Each individual shows binary strings of length  $l$  in which  $i^{th}$  bit determines the truth value of the  $i^{th}$  boolean variable of the  $l$  boolean variables shown in the boolean expression.

Any candidate solution is evaluated by the fitness function, which delegates fitness value 1 to that individual (string) boolean values of that string by which the boolean expression is evaluated to 1 and 0 to other solutions.

Spear proposed another scheme [36] that has been assigned fitness to individual subexpressions in the original expression and aggregate them in some way to produce a complete individual fitness value. According to this context the general and natural way is to specify the value of TRUE to be 1 and the value of FALSE to be 0 by Smith [35].

### 4.3 Comparison of proposed scheme with other approaches

The performance of proposed algorithm is compared with some existing GAs techniques, which are chosen from the GAs literature. According to the performance, these algorithms were tested on different multi-modal boolean satisfiability problems, which are widely applied in the literature. Initially our scheme has been compared to standard GAs with random fixed mutation and crossover probabilities. In addition, the proposed technique compared with some other parameter control approaches such as adaptive [37], self-adaptive [3] and SSRGA [40] schemes. These all comparable approaches are considered, which briefly described as follows:

The simple genetic algorithm (SGA) is collection of individuals, these solutions of SGA are represented as a bit string of length  $l$ , mutation and crossover operators are used in the SGA as a variation operator. The probabilities of these variation operators are initially assigned by random values (0,1) in our experiments. The initial values are fixed during the whole evolution process.

Srinivas and Patnaik introduced a adaptive genetic algorithm (AGA) [37] which is efficient adaptive algorithm for multimodal optimization problems. The main motivation of this approach is to maintain the diversity in the population and retain the convergence capacity of

the GA by using adaptive probabilities of crossover and mutation. The rate of mutation and crossover are updated depending on the fitness values of the solutions. In order to address the convergence property, to determines  $p_m$  and  $p_c$  for each solution to protect highly-fit solutions from high disruption. The mutation and crossover rates are updated applying following equations.

$$P_c = \begin{cases} K_1 (f_{max} - f') / (f_{max} - \bar{f}), & \text{if } f' > \bar{f} \\ K_3, & \text{otherwise} \end{cases} \quad (3)$$

$$P_m = \begin{cases} K_2 (f_{max} - f) / (f_{max} - \bar{f}), & \text{if } f > \bar{f} \\ K_4, & \text{otherwise} \end{cases} \quad (4)$$

where  $f_{max}$  and  $\bar{f}$  represents the maximum fitness and average fitness of the population,  $f'$  is the larger of the fitness values of the individual to be crossed and  $f$  is the fitness of solution which is being mutated.  $K_1, K_2, K_3, K_4 \in [0,1]$  are predefined constants. These predefined constant values are taken from [37], same values are considered for our experiments.

Self-adaptive genetic algorithm (SAGA) was developed by Back and Schutz [3]. which uses single mutation rate separately per solution. For example an individual composes of a bit binary string of length  $l$  and associated mutation rate  $\mu \in [0, 1]$ . The new mutation probability is adjusted according to the following equation.

$$\hat{\mu} = \left( 1 + \frac{1 - \mu}{\mu} \exp(\gamma N(0,1)) \right) \quad (5)$$

where  $\gamma = 0.22$  is the learning rate and  $N(0, 1)$  is the uniform distributed value with mean 0.0 and standard deviation 1.0. The new solution is specified through bit wise mutation of 1 bits applying the mutated mutation rate value  $\hat{\mu}$ . The mutation rate is not less than  $1/l$ .

Vafae and Peter recently proposed a site-specific rate genetic algorithm (SSRGA) [40] scheme. Which introduced mutation scheme to determines different mutation rate for different sites of the individuals. The main motivation of this approach to face both explorative and exploitative responsibilities of variation operators. This scheme starts from partitions of population into number of sub-populations is defined based on fitness and distribution of solutions contained in the search space. Then, in order to identify the representative individual associating to each sub-population. Finally, the result of this procedure is a probability vector (SSR vector) with elements associating to the rate of possessing a specific allele at each locus of every solution included in a sub-population.

#### 4.4 Experimental setting

In order to investigate our proposed scheme with other four approaches on problems with different level of difficulty, the difficulty of the problem means to modify the length of individuals  $l$  and population size  $n$ . Generally, the problem becomes more difficult if the length of individuals would be increase. To avoid the premature convergence due to increase population size and may also be the algorithms have a ability to find a better solutions during the evolution process. We have been considered different levels of problem difficulty by using the parameter pairs  $(l, n)$  with (20,30), (30,50), (50,50) and (60,70). These parameter setting is used to multimodal landscape with different degree of multi-modality by assigning number of peaks 1,5,10. Each peak is shown by a randomly created solution of length  $l$ , which identifies the specific position of peak in the search space. The default setting of parameters applied in the experiments of this paper can be observed in the original papers [37], [3], [40]. To compare the performance of multi-population with adaptive approach with other described above algorithms in section 4.3. All algorithms were allowed same population size, length of individual, stop criteria and maximum number of generations for each run.

#### 4.4.1. Analysis of Parameter Sensitivity

The performance of included approaches depends on the operator and parameters applied. In this paper, the key parameters are considered for the sensitivity analysis such as population size, individual length

and radius(*minDist*). To locate the multiple optima by using the small population size, it is not always good idea. Increasing the population size, however, to reduce the probability of being trapped into local optimum and increasing the probability of locating better solution during the evolution process. Solution is consist of length of  $l$  bit binary string, if length of string increases the problem become more challenging since the state space is exponentially expended according to the length of solution increases.

Table 1  
Parameter setting for all compared algorithms

Parameter	Setting			
$l$	20	30	50	60
$n$	30	50	50	70
$r$	0.5	0.6		0.7
Peaks	01	05		10

$r(\text{minDist})$  is used to determine the species and species seed in the SPSO [26] and to identify the representative individual from population [40]. The radius ( $r$ ) is the key parameter to both approaches. If the radius is too small, there is potential problem that few isolated solutions species may tarp into local optimum due to very small size of radius. In this context, the loss of diversity then the algorithms can't contribute further more during the process. If radius is too large, there may be more than one optimum within the radius. It is very difficult to choose the suitable value of radius. In this paper, we have considered different values of radius to investigate the performance of algorithms. The different parameters and their settings of different level of multi-modality problems applied in our experiments can be seen in Table 1. In order to determine that which parameter setting gives the great performance among the compared algorithms.

#### 4.4.2 Experimental Results and Analysis

In order to investigate that multi-population with adaptive mutation schemes can improve the performance of GAs on solution quality and convergence rate for multi-modality problems. All algorithms were run independently 50 times on the different degree of multi-modality problems. From the Table 2, It can be determined that the same value of  $r$  is considered for first three algorithms because which is related to these approaches; the  $r$  is applied with distinctly in different algorithms, first one is used a  $r$  as threshold (minimum distance between two solutions), value in BWAM, another one is determined as a radius of the each species SWAM and last one is considered of  $r$  values with same meaning as in first one.

The experimental results are shown in Table 2, which gives detail result of average and standard deviation of the best-found fitness values during the evolution. Both values are computed when the maximum number of iteration is completed or the global optimum is found. The algorithm determine better results among the other algorithms which presents bold average best fitness. In addition, those average best fitness values are underlined which is associated with smallest standard deviation among the other approaches. The effect of smaller standard deviation determine to produced reliable solutions.

1) Effect of varying the length of solution and population size: The algorithms have been investigated on different problems with different level of difficulty. The difficulty means to change the length of individual and population size. The problem would become more difficult if increasing the length of solution. The size of population is increasing, however, less probability of premature convergence to local optimum and increase probability to locate the

better solutions at the termination of optimization process [36]. The proposed algorithm have been tested on different length of solution and population size.

**Table 2**  
Comparison results of different algorithms with different parameter setting with different problems

<b>Peak 1</b>													
(l,n)		(20,30)			(30,50)			(50,50)			(60,70)		
r*1		(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)
BWAM	avg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SWAM	avg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SSRGA	avg	1.0	1.0	1.0	0.998	1.0	0.998	0.922	0.916	0.921	0.891	0.893	0.897
	std	0.0	0.0	0.0	0.007	0.0	0.006	0.016	0.019	0.022	0.016	0.014	0.016
SAGA	avg		0.877			0.824			0.750			0.740	
	std		0.033			0.024			0.022			0.020	
AGA	avg		0.93			0.872			0.794			0.774	
	std		0.026			0.023			0.020			0.015	
SGA	avg		0.897			0.876			0.817			0.810	
	std		0.051			0.047			0.041			0.034	
<b>Peak 5</b>													
(l,n)		(20,30)			(30,50)			(50,50)			(60,70)		
r*1		(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)
BWAM	avg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SWAM	avg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SSRGA	avg	1.0	1.0	1.0	0.991	0.995	0.944	0.906	0.906	0.911	0.879	0.874	0.878
	std	0.0	0.0	0.0	0.014	0.011	0.012	0.017	0.017	0.016	0.026	0.026	0.024
SAGA	avg		0.909			0.850			0.779			0.760	
	std		0.022			0.021			0.018			0.012	
AGA	avg		0.943			0.878			0.802			0.779	
	std		0.026			0.022			0.017			0.014	
SGA	avg		0.905			0.886			0.806			0.807	
	std		0.038			0.039			0.034			0.028	
<b>Peak 10</b>													
(l,n)		(20,30)			(30,50)			(50,50)			(60,70)		
r*1		(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)	(0.5*1)	(0.6*1)	(0.7*1)
BWAM	avg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SWAM	avg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SSRGA	avg	1.0	1.0	1.0	0.987	0.988	0.987	0.876	0.882	0.884	0.831	0.837	0.833
	std	0.0	0.0	0.0	0.017	0.015	0.016	0.040	0.027	0.030	0.031	0.030	0.034
SAGA	avg		0.923			0.866			0.788			0.774	
	std		0.030			0.021			0.018			0.021	
AGA	avg		0.952			0.890			0.810			0.791	
	std		0.030			0.021			0.020			0.019	
SGA	avg		0.920			0.893			0.820			0.805	
	std		0.036			0.032			0.034			0.027	

2) Effect of varying the peaks: The performance of BWAM and SWAM is equal on specific setting of ( $l, n$ ) on the different number of peaks, but efficiency of SSRGA slightly declines as number of peaks increases. However, the rest of the compared algorithms are reversed.

3) Effect of varying the ( $r/minDist$ ): Niche radius( $r$ ) is a very important role regarding the performance of both approaches (BWAM and SWAM), which is determine subpopulations. The many species could be generated if the value of  $r$  is very small. it is a big problem due to this reason few isolated solutions of sup-population trap into local optimum very quickly. Setting  $r$  to a too large will cause more than one optimum could be contained within the circle ( $r$ ). The experiments determine the effect of the different values on the performance of those algorithms which are associated with the  $r$  in this paper.

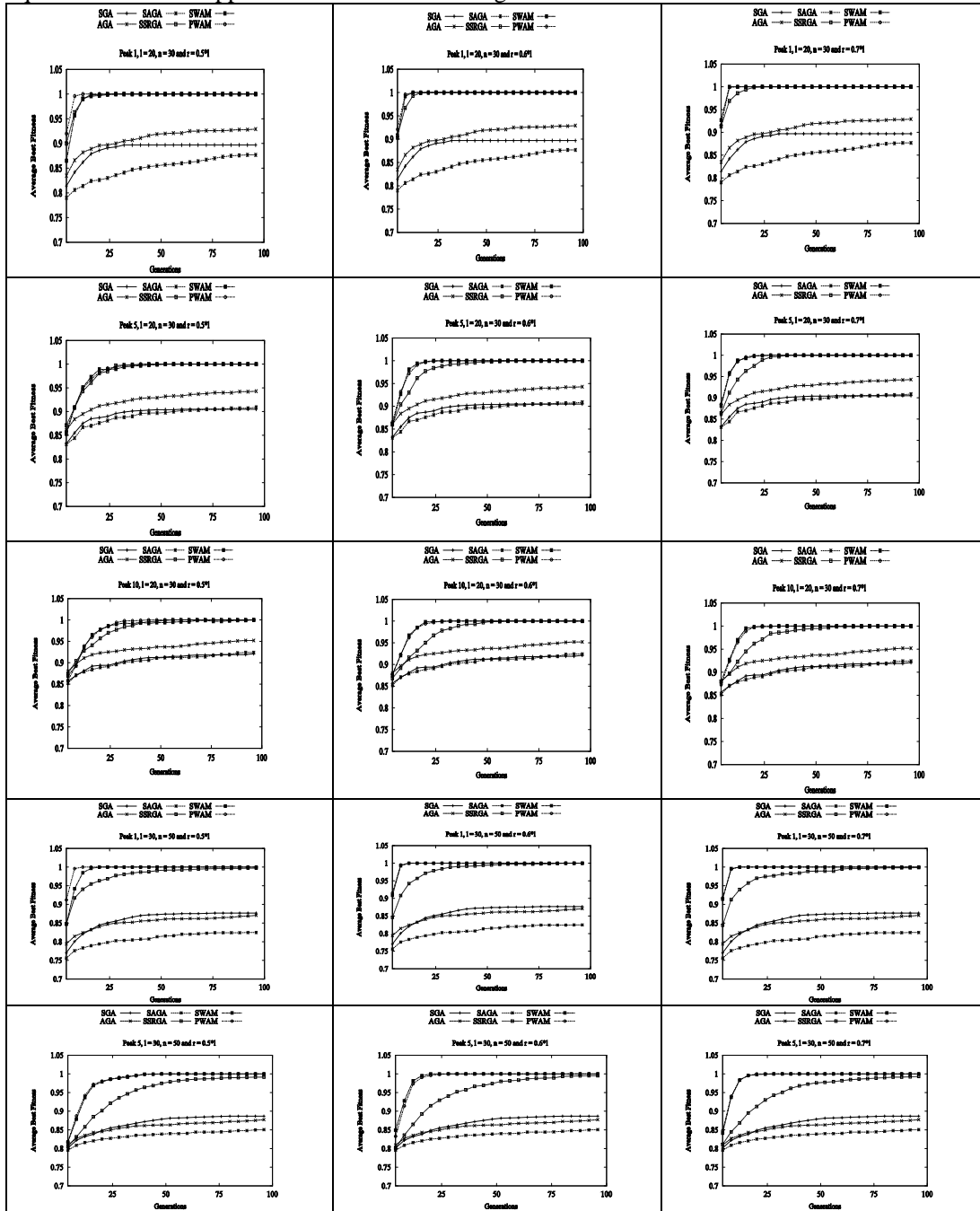
The statistical results of comparing algorithms is investigated applying two-tailed t-test with a 98 degree of freedom at 0.05 level of significance. Table 3 presents t-test results of comparing pair of algorithms, where the result is determined as “s+”, “s-”, “+”, “-”, and “\_” if the first algorithm in pairs is significantly better than, significantly worse than the insignificantly better than, insignificantly worse than or equivalent to the second algorithm, respectively. The t-test result of comparing algorithms shows on peak 10 and  $minDist$  is chosen  $0.6 * l$  with different level of problem settings. From Table 3 it can be observed that the BWAM and SWAM both approaches are statistically better than other comparing schemes to locating the optimum value in the fitness landscape. Here we presents only statistical results of compared techniques with different level of difficulty of different problems on peak 10.

According to the parameter setting of different level of multi-modality problems are divided into three categories such as simple problems, bit harder problems, and difficult problems. The performance of all algorithms varies on different categories. From Fig.1 it can be determined that the convergence speed of involved algorithms, which were tested on different level of difficulty in simple multi-modal problems. In Fig.1 BWAM, SWAM, and SSRGA have a higher convergence rate than self-adaptive GA, adaptive GA, and Standard GA on simple problems. The overall the convergence speed of BWAM and SWAM is initially better than SSRGA on simple problems but with time going the performance of all three algorithms are same except last two rows in Fig.1.

Table 3  
t-test values of comparing algorithms on different problems with different level of difficulty

Problems	(20,30)	(30,50)	(50,50)	(60,70)
BWAM-SWAM	~	~	~	~
BWAM-SSRGA	~	~	s+	s+
BWAM-SAGA	s+	s+	s+	s+
BWAM-AGA	s+	s+	s+	s+
BWAM-SGA	s+	s+	s+	s+
SWAM-SSRGA	~	~	s+	s+
SWAM-SAGA	s+	s+	s+	s+
SWAM-AGA	s+	s+	s+	s+
SWAM-SGA	s+	s+	s+	s+
SSRGA-SAGA	s+	s+	s+	s+
SSRGA-AGA	s+	s+	s+	s+
SSRGA-SGA	s+	s+	s+	s+
SAGA-AGA	s+	s-	s+	s-
SAGA-SGA	s+	s+	s-	s-
AGA-SGA	s+	s+	-	-

Fig 2. presents the results of the evolutionary process of compared algorithms on bit harder problems. It can be clearly noticed that the convergence speed of BWAM and SWAM are quicker than other approaches but it is interesting to observe that different scheme have different



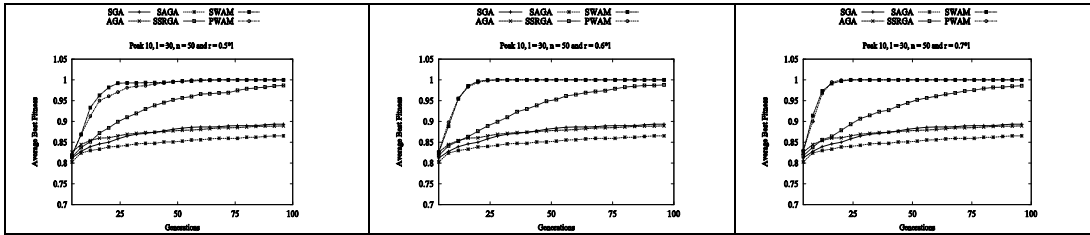


Fig 1. Evolutionary process of the comparison algorithms on simple problems

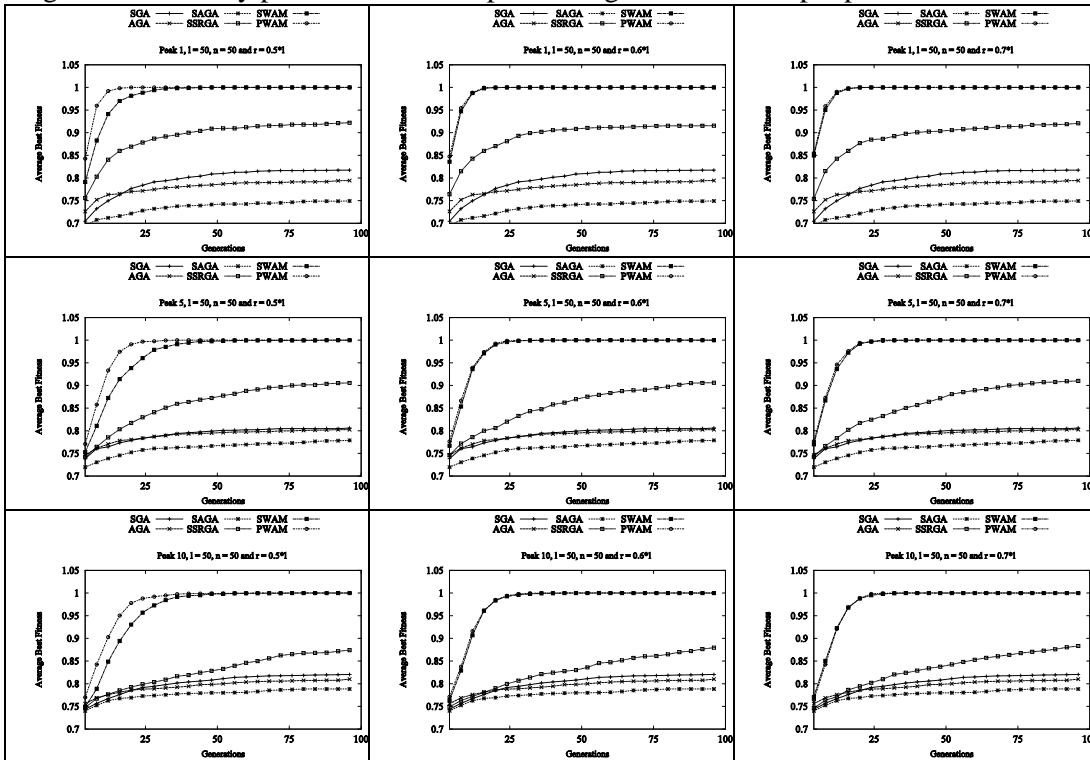


Fig. 2. Evolutionary process of comparison algorithms on bit harder problems.

convergence rate on different (bit harder) problems with different peaks. On few problems, the BWAM is initially better than SWAM regarding convergence speed.

From Fig3. it can be seen that the convergence rate of BWAM initially is faster than SWAM from beginning to 50 generations on first column, afterwards the both schemes have same convergence curve. In this figure, the convergence curves are quiet clear among the compared algorithms.

Analyzing the convergence rate of six comparing schemes. The very interesting things is that the convergence speed of both (BWAM, SWAM) approaches is initially quick convergence rate than compared techniques even on simple problems where SSRGA is going to compete to the BWAM and SWAM. It can also be seen two properties of proposed approach from above figures. The first one is the quick convergence rate against with other comparable algorithms except simple problems. BWAM and SWAM have quick convergence speed than other compared approaches on different levels of problem difficulty by setting the parameters. Another goal of suggested scheme is the ability to explore the prominent area by avoiding the premature convergence in the fitness landscape on multimodality problems. The proposed technique reduce the less probability of trapping into local optima and increase the probability of locating the better results than the other algorithms.

## 5. CONCLUSION AND FUTURE WORK

From the beginning, community of evolutionary algorithm has been done lot of work regarding to increase the performance and locating multiple optima in multi-modal optimization problems of GAs. The exploration and exploitation are becoming two main characteristic in the case of performance. The researchers have applied multi-population scheme to increase the efficiency of GAs in this context.

This paper proposed a new mutation approach, which is the combination of multi-population and adaptive mutation genetic operator. It is determine two different probability

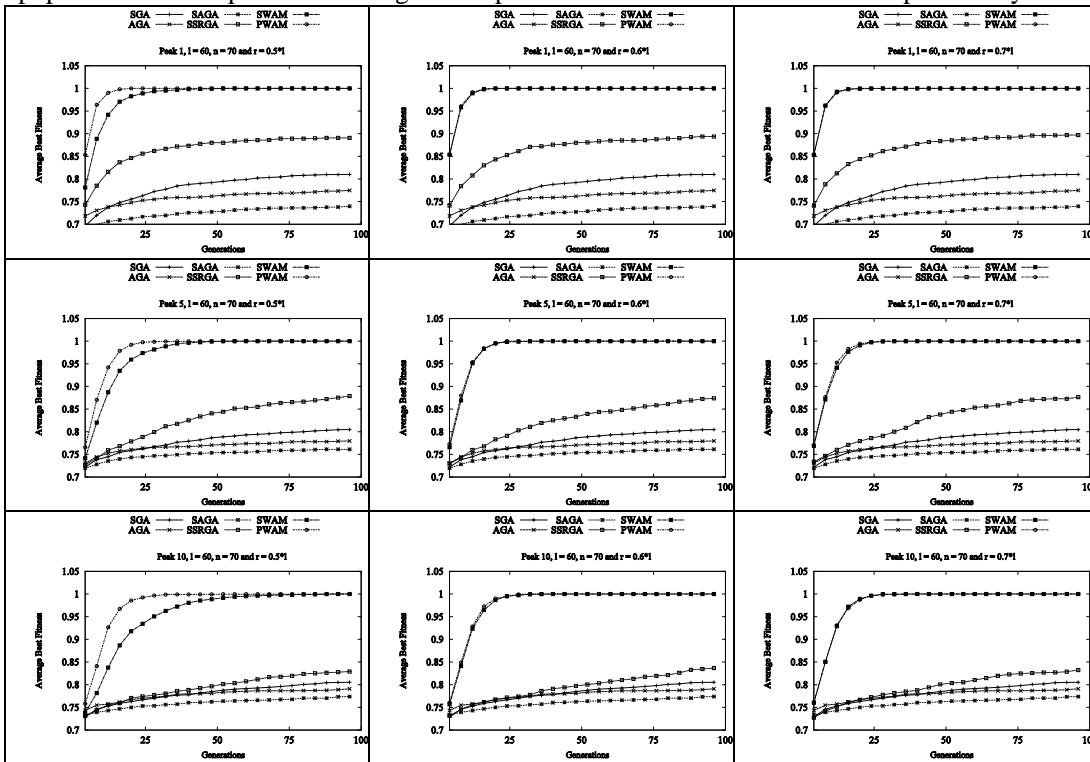


Fig. 3. Evolutionary process of comparison algorithms on difficult problems.

vectors for each sub-population. Each vector contains elements associating to the probabilities of possessing a specific allele at each site. These vectors are updated during the evolution process. The suggested mutation scheme has been considered two different multi-population techniques in this paper.

In order to analysis the performance of the our proposed algorithms, experimental results were conducted on different benchmark problems and to compare the results to standard GA, and some other approaches which belongs to parameter control approaches. From the experiments, it can be seen that suggested scheme is greatly improve the performance of GA regarding to the locating and tracking the multiple optima in the fitness landscape by using the multi-population with adaptive mutation operator.

In the future, we will use proposed mutation scheme to large and complex real-world problems because GAs have a lot of applications in different area of real-world problems such as science, engineering, business and social science, the interested reader may wish to refer [11] p. 125. In addition, it would be interesting to combine clustering partition technique with adaptive mutation operator to improve the performance of GAs. Finally, we will conduct further experiments on epistatic [36] and deceptive[11] benchmark problems, which are similar to the multi-modal problems.



## REFERENCES

- [1] P.J. Angeline, (1995) "Adaptive and Self-adaptive evolutionary computations," In M. Palaniswami and Y. Attikiouzel, editors, *Computational Intelligence: A dynamic system perspective*, pp. 152-163. IEEE Press.
- [2] T. Back, (1992) "Self-Adaptation in Genetic Algorithms," In *Proc. of the 1st European Conf. on Artificial Life*, pp. 263-271.
- [3] T. Back and M. Schütz, (1996) "Intelligent mutation rate control in canonical genetic algorithms," In *Foundation of Intelligent Systems*, Springer, pp. 158-167.
- [4] M. Bessaou, A. Paetrowski, and P. Siarry (2000) "Island model cooperating with speciation for multimodal optimization," In *Proc. of the 6th Int.Conf. on Parallel Problem Solving from Nature*, pp. 16-20.
- [5] J. Branke, T. Kaubler, C. Schmidt, and H. Schmeck, (2000) "A multi-population approach to dynamic optimization problems," In *Proc. of the 4th Int. Conf. on Adaptive Comput. Des Manuf*, pp. 299-308.
- [6] L. Davis, (1989) "Adapting operator probabilities in genetic algorithms," *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, pp. 61-69.
- [7] L. Davis, (1991) "Handbook of Genetic Algorithms," Van Nostrand Reinhold.
- [8] A.E Eiben, R. Hinterding, and Z. Michalewicz, (1999) "Parameter Control in Evolutionary Algorithms," *IEEE Trans. on Evol. Comput.*, 3(2): pp. 124- 141.
- [9] A.E. Eiben, Z. Michalewicz, M. Schoenauer, and J.E. Smith. (2007) *Parameter Control in Evolutionary Algorithms, Parameter setting in Evolutionary Algorithms*, Chapter 2, Springer Verlag, pp. 19-46.
- [10] T.C. Fogarty, (1989) "Varying the probability of mutation in genetic algorithms," In *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, pp. 104-109.
- [11] D. E. Goldberg, (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, New York: Addison-Wesley.
- [12] D. E. Goldberg, and J. Richardson, (1987) "Genetic algorithms with sharing for multimodal function optimization," In *Proc. of the 2nd Int. Conf.on Genetic Algorithms*, pp. 41-49.
- [13] V.S. Gordon, D. Whitley, and A. Bohn, (1992) "Dataflow parallelism in genetic algorithms," In *parallel Problem Solving from Nature*, pp. 533-542.
- [14] J. J. Greffentette, (1986) "Optimization of Control Parameters for Genetic Algorithms," *IEEE Trans. on Sys. Man and Cyber.*, 16(1): pp. 122-128.
- [15] T. P. Hong, H. S. Wang, and W. C. Chen. (2000) "Simultaneously applying multiple mutation operators in genetic algorithms," *Journal of Heuristics*, 6: pp. 439-455.
- [16] J. Kennedy. (2000) "Stereotyping: Improving particle swarm performance with cluster analysis," In *Proc. 2000 IEEE Congr. Evol. Comput*, pp. 1507- 1512.
- [17] I. Korejo, S. Yang and C. Li, (2010) "A Directed Mutation Operator for Real Coded Genetic Algorithms," *EvoApplications 2010, Part I, LNCS 6024*,pp. 491-500.
- [18] K. A. De Jong, (1975) "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," PhD Thesis, Department of Computer and Communication Science, University of Michigan, Ann Arbor.
- [19] C. Li, S. Yang, and I. Korejo, (2008) "An Adaptive Mutation Operator for particle swarm optimization," In *Proc. of the 2008 UK Workshop on Computational Intelligence*, pp. 165-170.
- [20] J. P. Li, M. E. Balazs, G. Parks, and P. J. Clarkson, (2002) "A species Conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207-234.

- [21] S. W. Mahfound, (1992) "Crowding and preselection revisited," In Proc. of the 2nd Int. Conf. on Parallel Problem Solving from Nature, vol 2, pp. 27-36.
- [22] H. Mühlenbein, (1992) "How Genetic Algorithms Really Work I. Mutation and Hillclimbing," In Proc. of the 2nd Int. Conf. on Parallel Problem Solving from Nature, pp. 15-29.
- [23] G. Ochoa, (2002) "Setting the mutation rate: Scope and limitations of the 1/l heuristics," Proc. of the 2002 Genetic and Evol. Comput. Conf., pp. 315-322.
- [24] G. Ochoa, (2006) "Error thresholds in genetic algorithms," Evol. Comput., vol. 14, no. 2, pp. 157-182.
- [25] G. Ochoa, (1999) "Error thresholds and their relation to optimal mutation rates," Springer-Verlag, Berlin, Germany.
- [26] D. Parrott and X. Li, (2006) "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," IEEE Trans. Evol. Comput., vol. 10, no. 4, pp. 440-458, Aug.
- [27] A. Passaro and A. Starita, (2008) "Particle swarm optimization for multimodal functions: A clustering approach," J. Artif. Evol. Appl., vol. 2008, pp. 1-15.
- [28] A. Petrowski, (1996) "A clearing procedure as a niching method for genetic algorithms," In Proc. 1996 IEEE Int. Conf. Evol. Comput., pp. 798-803.
- [29] R. Rechenberg, (1996) Evolutionary Algorithms in Theory and Practice. New York: Oxford University, Press.
- [30] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das, (1989) "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization," Proc. of the 3rd Int. Conf. on Genetic Algorithms, pp. 51-60.
- [31] H. P. Schwefel, (1995) Evolution and Optimum Seeking, Chichester, UK: John Wiley.
- [32] H. P. Schwefel, (1981) "Numerical optimization computer models," Chichester, Wiley.
- [33] J.E. Smith and T.C. Fogarty, (1996) "Self-adaptation of mutation rates in a steady state genetic algorithm," In Proc. of the 3rd IEEE Conf. on Evol. Comput., pp. 318-323.
- [34] J.E. Smith and T.C. Fogarty, (1997) "Operator and parameter Adaptation in genetic algorithms," Soft Comput., vol. 1, no. 2, pp. 81-87.
- [35] G. Smith, (1979) "Adaptive genetic algorithms and boolean satisfiability problem," Technical Report, University of Pittsburgh, Pittsburgh.
- [36] W. M. Spears, (2000) Evolutionary algorithms, the role of mutation and recombination, Natural Computing, Springer-Verlag, Berlin, Germany.
- [37] M. Srinivas and L.M. Patnaik, (1994) "Adaptive probabilities of crossover and mutation in genetic algorithms," IEEE Trans. on System, Man and Cybern., vol. 24, no. 4, pp. 656-666.
- [38] R. K. Ursem, (2000) "Multinational GAs: Multimodal Optimization Techniques in Dynamic Environments," In Proc. 2nd Genetic Evol. Comput. Conf., pp. 19-26.
- [39] S. Uyar, S. Sariel, and G. Eryigit, (2004) "A gene based adaptive mutation strategy for genetic algorithms," Proc. of the 2004 Genetic and Evol. Comput. Conf., pp. 271-281.
- [40] F. Vafee and P. C. Nelson (2010) "An Explorative and Exploitative Mutation Scheme," Proc. 2010 IEEE World Congr. on Comput. Intell., pp. 18-23.
- [41] T. White and F. Oppacher, (1994) "Adaptive crossover using automata," Proc. of the 3rd Conf. on PPSN, pp. 229-238.
- [42] D. H. Wolpert and W. G. Macready, (1997) "No free lunch theorems for optimization," IEEE Trans. on Evol. Comput., vol. 1, no. 1, pp. 67- 82.
- [43] S. Yang, (2002) "Adaptive non-uniform crossover based on statistics for genetic algorithms," Proc. of the 2002 Genetic and Evol. Comput. Conf., pp. 650-657.

- [44] S. Yang, (2003) "Adaptive mutation using statistics mechanism for genetic algorithms," In F. Coenen, A. Preece, and A. Macintosh (editors), *Research and Development in Intelligent System XX*, pp. 19-32.
- [45] S. Yang and S. Uyar, (2006) "Adaptive mutation with fitness and allele distribution correlation for genetic algorithms," *Proc. of the 21st ACM Symp. on Applied Computing*, pp. 940-944.
- [46] S. Yang and C. Li, (2010) "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments," *IEEE Trans. on Evol. Comput.* vol. 14, no. 6, pp. 959-974, December.

### Authors

Imtiaz Ali Korejo received his B.Sc.(Hons) and M.Sc.(Hons) in Computer Science from University of Sindh, Jamshoro, Pakistan, in 1999, and 2000, respectively. He worked as a research associate in the Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Pakistan from 2001 to April 2003, as a Lecturer in the same institute from April 2003 to 2012; He is currently working in the same institute as an Assistant Professor since April 2012. He received his Ph.D. from the Department of Computer Science, University of Leicester, United Kingdom in 2012.



His research interests are evolutionary algorithms, genetic algorithms and adaptive approaches.

Shengxiang Yang received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, China in 1993, 1996, and 1999, respectively. From October 1999 to June 2012, he worked as a Postdoctoral Research Associate in the Department of Computer Science, King's College London, U.K., a Lecturer in the Department of Computer Science, University of Leicester, U.K., and a Senior Lecturer in the Department of Information Systems and Computing, Brunel University, U.K., respectively. Since July 2012, he has been appointed as a Professor at the Center for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, U.K.



His major research interests include evolutionary and genetic algorithms, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling and relevant real-world applications. He has over 150 publications. He has given invited keynote speeches in several international conferences and co-organized several symposiums, workshops and special sessions in conferences. He serves as the area editor, associate editor or editorial board member for four international journals. He has co-edited several books and conference proceedings and co-guest-edited several journal special issues. He is the chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments, Evolutionary Computation Technical Committee, IEEE Computational Intelligence Society, and the founding chair of the Task Force on Intelligent Network Systems, Intelligent Systems Applications Technical Committee, IEEE Computational Intelligence Society.