Review article

A comparative review of approaches to prevent premature convergence in GA

Hari Mohan Pandey [a,*], Ankit Chaudhary [b], Deepti Mehrotra [c]

[a] Department of Computer Science & Engineering, Amity University, Noida, Uttar Pradesh, India
[b] Department of Computer Science, MUM, IA, USA
[c] Amity School of Engineering & Technology, Amity University, Noida, Uttar Pradesh, India

ABSTRACT

This paper surveys strategies applied to avoid premature convergence in Genetic Algorithms (GAs). Genetic Algorithm belongs to the set of nature inspired algorithms. The applications of GA cover wide domains such as optimization, pattern recognition, learning, scheduling, economics, bioinformatics, etc. Fitness function is the measure of GA, distributed randomly in the population. Typically, the particular value for each gene start dominating as the search evolves. During the evolutionary search, fitness decreases as the population converges, this leads to the problems of the premature convergence and slow finishing. In this paper, a detailed and comprehensive survey of different approaches implemented to prevent premature convergence with their strengths and weaknesses is presented. This paper also discusses the details about GA, factors affecting the performance during the search for global optima and brief details about the theoretical framework of Genetic algorithm. The surveyed research is organized in a systematic order. A detailed summary and analysis of reviewed literature are given for the quick review. A comparison of reviewed literature has been made based on different parameters. The underlying motivation for this paper is to identify methods that allow the development of new strategies to prevent premature convergence and the effective utilization of genetic algorithms in the different area of research.

© 2014 Elsevier B.V. All rights reserved.

Contents

* Corresponding author. Tel.: +91 9810625304.
  E-mail address: hari04top@yahoo.co.in (H.M. Pandey).

1. Introduction

Genetic Algorithms (GAs) are search and optimization algorithms based on natural selection and genetics. Genetic Algorithms are one of the most popular algorithms in the categories of evolutionary algorithms. The basic principles of GAs were initially developed by Holland [1] and further carried by De Jong and Goldberg. Goldberg and Michalewicz have given the detailed overview and implementation of GAs in various fields [2,3].

GA operates on a population of solutions represented by some encoding. During the implementation process of GAs every solution or individual is assigned a fitness which is the measure of GAs. The fitness of each individual is directly related to the objective function for optimization problem [4]. Then, using the reproduction, crossover and mutation operators the individual population can be modified to a new population [5].

In GAs, the searching is iteratively guided by the fitness of the current parent generation. Whenever, we apply GAs for an optimization problem, it runs over thousands of individual, each represents a solution. The obtained solutions are evaluated and recombined to get offspring. It has been proven in [1–3,6] that the previous generations details are only implicitly and partially preserved in the current generation. Hence, the regeneration can be hard to manage because of the following reasons:

(a) In regeneration an individual population appears that had already been seen in the search space.
(b) Genetic drift can affect the search for optimization in a wrong direction.
(c) As a result, a big part of the search space is left unexplored for the global optimum.

GAs has gained popularity because it can be applied in a wide range of problems, including multimodal function optimization, machine learning, pattern recognition, image processing, natural language processing, grammar inference [4,5,8] and many more. But the behaviors of GAs have not been found as adaptive as expected. A common problem is that of premature convergence [7].

The primary interest of this paper is to show the detailed survey of different approaches proposed and applied for solving the premature convergence problem in GAs.

Section 2 offers a detailed description of different factors affecting GA search. In this section we discussed diversity of population and selective pressure. The aim of Section 3 is to provide the theoretical framework to understand the genetic mechanism of evolutionary search process. This section briefly describes various theories such as Schema theory, Markov chain theory, and Statistical mechanism. This section also discusses and presents

the important literatures to understand the genetic dynamics through Markov chain and statistical mechanics. The major parts of this paper contained in Section 4. This section gives the detailed and comprehensive survey on approaches of handling premature convergence within GAs. In addition, strengths and weaknesses have been provided for each approach. The surveyed research works have been organized according to the publication. Surveyed researches have been summarized in Section 5 using various parameters. The conclusions are drawn in Section 6. Lastly, references are given which covers the important literatures in the area of GA.

2. Factors affecting GAs

A genetic algorithm is a method of optimization based on Darwin's natural selection theory. Darwin's theory was based on survival of the fittest to refine a set of solution in an iterative manner [2]. Darwin's theory says that evolution occurs if and only if three conditions met [9] (see Fig. 1). The conditions shown in Fig. 1 are necessary; it means that if due to any reason any one condition does not meet in the regeneration process, natural selection will cease at that generation.

GA starts with a set of solution which is represented by chromosome. By using an appropriate selection technique a solution from one population is picked depending on the fitness and used to form a new offspring. This process repeated until GA reached to the threshold or the maximum number of generations. Fig. 2 shows the flowchart of the Standard Genetic Algorithms (SGA).

The performance of GA largely depends on the crossover and mutation operations. Hence, adapting the suitable value of crossover and mutation is very important because it guides the search process and maintains the diversity in the population [19].

As discussed, GA is the population based search method. Therefore, selection of the population is the critical step because if
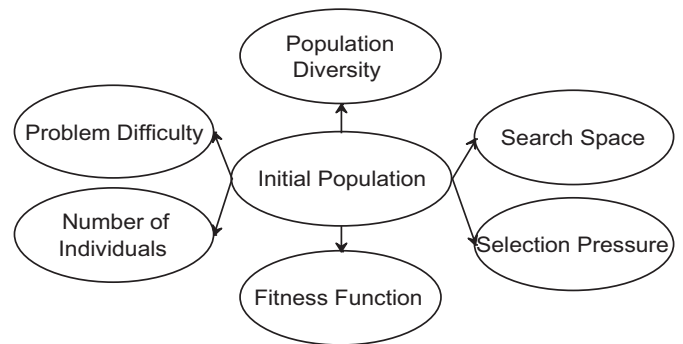

Fig. 1. Necessary and sufficient conditions of Darwin's theory.

Fig. 2. Flowchart of standard genetic algorithm [8].

Fig. 3. Factors affecting GA's behavior.

population has not been selected in an intelligent manner then it will be difficult to get the appropriate answer to the problem. Typically, selection done at two places, one is in the beginning known as *Initial Population* selection or generation of random initial population and the second one is the selection of population for the next generation. Hence, before applying the selection or generation of population one has to take some factors into account as shown in Fig. 3. In this paper, we are considering two important factors, namely *Population Diversity* and *Selection Pressure*.

In order to reach to the global optima and explore the search space adequately, maintaining population diversity is very important. Also, it had been explained clearly that degree of population diversity is one of the main causes of premature convergence [27,28]. It is necessary to select the best solution of the current generation to direct the GA to reach to the global optimum. The tendency to select the best member of the current generation is known as selective pressure. Selection pressure is also a key factor that plays an important role in maintaining genetic diversity. A proper balancing is required between genetic diversity and selection pressure to direct GA search to converge in a time effective manner and achieve global optima. High selective pressure reduces the genetic diversity; hence in this situation premature convergence can occur while the little selective pressure prohibits GA to converge to an optimum in reasonable time.

3. Theoretical framework

This section provides an overview of different theories presented to address theoretical concerns related to evolutionary theory. There are various tools and methods available such as Schema theory, Markov chain theory, Dimensional analysis, Order statistics, Quantitative genetics, Orthogonal functional analysis, Quadratical dynamical systems, and Statistical physics. We have concentrated our study on three methods, namely Schema theory since it is considered as the fundamental theory, Markov chain theory since it is helpful in setting the boundaries for convergence and many more features and Statistical mechanics theory.

3.1. Schema theory

Holland [136] presented Schema theory and it was accepted as fundamental theory to understand GA. It was widely used until the early 1990's. It was claimed that in a population of '*n*' chromosomes, GA process $O(n^3)$ schemata into each generation. This property is known as intrinsic/implicit parallelism [1]. A schema is a similarity template used to describe a subset of string displays similarity at certain string position. It can be formed by ternary alphabet {0, 1, ∗}, where '∗' represents a notation symbol shows the description of all possible similarities among string of a particular length and alphabet [137]. Length and order of schema were used to understand
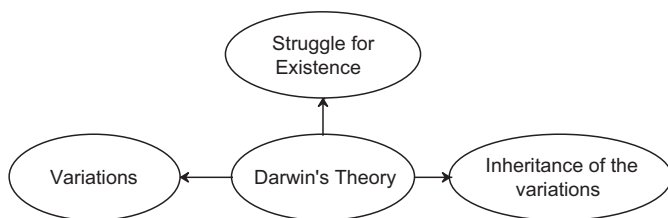
the working of schema theory. Length can be defined as the distance between first and last defined position in the schema while the order of schema is the number of defined positions. The GA modeled in schema theory known as canonical GA, which works on binary strings.

Although the schema theory was considered as fundamental theory, but it was criticized by many researchers. Some of them are pointed out as:

• It is not suitable to explain the dynamical or limit behavior of GA.
• Implicitly separation of problem to some extent.

The ignorance of these assumptions leads to the "Building block hypothesis" which clearly explains the behavior of GA [2]. In addition, Beryer [139] presented an alternative method to show how GA operates. Reeves and Rowe [138] presented the detailing of criticism of schema theory in their book "Genetic Algorithms: Principles and Perspectives (A Guide to GA theory). One more reason of replacement of schema theory is the suitability of Markov chain theory in the field of evolutionary algorithms (EAs).

3.2. Markov chains theory

The term Evolutionary Algorithm (EA) stands for a family of stochastic problem solver based on principles that can be found in biological evolution [140]. A stochastic algorithm can be considered as a random sequence of events $E = \{X_1, X_2, X_3, \ldots\}$ occurring in time, where each $\{X_t\}$ represents a random variable and the possible values that these variables can take are called the states of the system [138]. A Markov chain is a random sequence of variables $X_0, X_1, X_2 \ldots$ satisfies the Markov property, i.e. the simplest kind of dependence within a stochastic process is when the distribution of time t depends only on what happened at the time $t - 1$ [138]. In statistical terms, this is known as "*memory less property*" and can be defined as [142]:

$$P(X_{n+1} = x | X_n = x_n, \ldots, |X_0 = x_0) = P(X_{n+1} = x | X_n = x_n)$$

Rudolph [141] presented the matrix property for Markov chain theory as given below:

A square matrix $P = (p_{ij}) : N \times N$ is said to be

a. Nonnegative ($P \geq 0$), if $p_{ij} \geq 0 \forall i, j \in \{1, \ldots, N\}$
b. Positive ($P > 0$), if $p_{ij} > 0 \forall i, j \in \{1, \ldots, N\}$
   A nonnegative square matrix $P = (p_{ij}) : N \times N$ is said to be
c. Primitive, if $\exists n \in \mathbb{N} : P^n$ is positive
d. Reducible, if P can be brought into the form (with square matrices C and T)

$$\begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$$

By applying the same permutations to rows and columns.
e. Irreducible, if it's not reducible
f. Stochastic, if $\sum_{j=}^{N} p_{ij} = 1 \forall i \in \{1, \ldots, N\}$
   A stochastic matrix $P = (p_{ij}) : N \times N$ is said to be
g. Stable, if it has identical rows
h. Columns allowable if it has at least one positive entry in each column.

Matrices that describe the transition probabilities in a Markov chain are stochastic matrices and will be referred to as a "transition matrix" [142]. In the next paragraph, we have provided a brief review of Markov chain theory which shows the applicability of this model.

Horn [160] used Markov chains to analyze the stochastic effects of the niching operator of a niched GA. It was proved that a Markov chain is an effective tool to study the dynamic nature of a niched GA. The effect of "*Absorbing Markov chain*" and "*Ergodic Markov chain*" was shown to estimate the convergence of a niched GA [160]. Eiben et al. [154] studied classical GA and simulated annealing (SA) and presented Abstract Genetic Algorithm (AGA) which generalize and unifies GA and SA. For the better clarification purpose, similarity and dissimilarity between classical GA and AGA was presented. In addition, Markov chain model was used to show the evolution of AGA. Leung et al. [158] discussed the ability of the GA search by applying zero mutation probability and the relationships between premature convergence and effect of GA parameters were presented. Using Markov chain, the reasons of choosing zero mutation probability when analyzing premature convergence was demonstrated. Jun and Xinghuo [156] showed theoretical analysis to show the condition for the convergence of EA. The necessary and sufficient conditions for the convergence of EA to the global optimum were derived with the help of Markov chain theory to describe the limiting behavior. In addition, the upper and lower bound of the convergence rate required for the EA was explored. Stark and Spall [155] presented computable rate of convergence for GA via Markov chain analysis. Huang [145] extended the work presented in [146] by applying Markov model developed by Nix and Vose [144]. Reeves and Rowe [138] presented the applicability of Markov chain theory to describe the limiting distribution of simple GA in which they explained the effect of selection, mutation and crossover. In addition, modeling GA uses Markov chain theory was presented using suitable example. Auger [143] studied the convergence property of self-adaptive evolutionary strategies via phi-irreducible Markov chain. Basically, he used recurrent Markov chains to prove the existence of critical size of population, which was calculated using different parameters of algorithms and determines the convergence rate of the algorithm. Braak [148] presented Differential Evolution Markov Chain (DE-MC) which combines the essential features of differential evolution (DE) presented in [149–151] for global optimization over real parameter space with Markov chain Monte Carlo (MCMC) [152,153] to solve an important problem of MCMC in real parameter spaces, namely that of choosing an appropriate scale and orientation for jumping distribution. Clark [142] applied this theory for modeling stochastic optimization algorithm. Héder and Bitó [157] implemented a GA to provide special site diversity to simultaneously optimize downlink and uplink the signal to interference and noise ratio (SINR) value in broadband fixed wireless (BFWA) access system. For the convergence analysis, i.e. to determine the maximum number of iterations they used Markov chains where every state represents a possible population containing different terminal station (TS) and base station (BS) assignment set as individuals. Suzuki [147] suggested large deviation principle approach for GA using Markov chains. Poli et al. [159] presented exact schema theory and Markov chain model Genetic Programming (GP) and variable length GA for the homologous crossover. In the absence of mutation the relationship between approximate and exact schema theory for different forms of homologous crossover was shown.

Although Markov chain model was implemented for convergence analysis. But it was found limited by the complexity of calculations and it does not show the predictive description of genetic dynamics [140,161].

3.3. Statistical mechanics

It is a branch of mathematical physics. It can be used to study the state of uncertainty of a mechanical system using probability

theory which works on the average behavior of the system [161,162]. Prügel et al. [163] and Shapiro [164] described the GA using statistical mechanics. Shapiro [164] showed statistical formulation of GA to study the dynamics of GA searching and explained how statistical mechanics theory addresses the limitations of Schema and Markov chain theory. Two sources were identified in studying the dynamics of evolutionary approaches as given below [164]:

- The dynamics are stochastic – selection involves sampling and mutation and crossover involve probabilistic changes to the strings.
- The dynamics acts on the space of possible populations, a space of astronomical dimensionality

It was said that by deriving a deterministic set of equations, one can describe the evolution which must be sufficiently simple to allow for some type of study. There are many previous works available which proves the applicability of this theory [138,165,161,166–168].

4. Approaches for preventing premature convergence

This section gives the detailed explanation of the different approaches given by researchers for managing the risk of premature convergence in GAs. We have reviewed various literatures and presenting twenty-four approaches with their strengths and weakness.

4.1. Crowding method

Crowding was introduced by De Jong [38] to eliminate the most similar individual whenever a new one enters in a subpopulation. The aim of presenting this technique was to maintain the population diversity and to fix the premature convergence. It was applied in the survival solution step of genetic algorithms. The primary objective of introducing it in the survival selection is to decide which individual among those in the current population and their offspring individuals will pass to the next generation.

Mahfoud [12] presented a modified version of the De Jong's strategy of replacing offspring of the most similar parents. The modified version of crowding was found effective in maintaining population diversity and preventing premature convergence. This approach works as follows:

Let, P_i = Next generation population, p_i = First individual, c_i = Second individual (child), $d(x,y)$ = distance between two individuals x and y

Then, $if((d(p_1, c_1) + d(p_2, c_2)) < (d(p_1, c_2) + d(p_2, c_1))$

$P_1 \leftarrow$ Winner between p_1 and c_1

$P_2 \leftarrow$ Winner between p_2 and c_2

Else

$P_1 \leftarrow$ Winner between p_1 and c_2

$P_2 \leftarrow$ Winner between p_2 and c_1

Three variations of crowding approach were presented, namely Standard Crowding [38], Deterministic Crowding [12] and Restricted Tournament Selection (RTS) [84]. Standard crowding was found to be limited in multimodal function optimization. Deterministic crowding (competition between children and parents of identical niches) approach was proposed by Mahfoud [12] to address the limitations [17] of the standard crowding approach suggested by De Jong [38]. The resulting order of complexity of deterministic crowding was O(N). RTS adapts standard tournament selection for multimodal function optimization. In case of RTS two elements from the population was picked to apply reproduction operators and then a random sample of CF (crowding factor)

individual was taken from the population as in standard crowding. This procedure was repeated N/2 times. The order of complexity of RTS is O(CF. N) which varies from O(N) to O(N^2) depending on the value of CF [85].

4.1.1. Strengths and weaknesses

This method was found effective for problems of all levels of difficulties, especially for multimodal function optimization. Using this, one can solve problems that are much more difficult than those solvable by traditional based hill-climbing. It uses a random sample of CF individual where CF is the crowding factor rather than fitness proportionate selection to resolve the selection pressure. There is no need of sharing function which is the key element of the sharing approach proposed by Holland [1] and further expanded by Goldberg and Richardson [16]. Selection of good sharing function needs a-prior knowledge of the objective function characteristics [74].

The problem with this approach is that it may lose lower optima which may lie on a crossover path to higher optima. Maintaining diversity in the population was the main goal of this approach and algorithm was successful to some extent, but stochastic errors by the low value of crowding factor affect the performance. Maintenance of bitwise diversity is the problem. It does not model the method by which a population arrives a stable mixture of species, but instead strives to maintain the diversity of the pre-existing mixture [75]. Handling replacement errors were the main concern with this technique. In addition, writing replacement rules are not an easy task

4.2. Incest prevention algorithm

To avoid premature convergence of GA two approaches were introduced, namely the dynamic application of crossover and mutation operator and the population partial re-initialization [31]. The primary objective of applying the crossover operation is to exchange certain information between two parent chromosomes at the same time one must keep in mind that important information should not be lost. Hence, the selection of mating pair is a very important step in removing the insufficiencies of GAs.

Nicoară [31] and Eshelman and Schaffer [32] used Hamming distance as a measure for selecting the mating pair. The idea is that if Hamming distance is above a threshold then selects the parent chromosome for mating otherwise no. Before starting, the threshold is set to a high value and during running of GA if no pair was chosen with the initialized threshold, the value can be decreased.

A detailed formula (Eq. (1)) was given for crossover operator and mutation operators (Eq. (2)). Eq. (2) is the modified formula as given in [33]. Results of [31] were compared against three other genetic algorithms, namely NSGA-II [34], Canonic genetic algorithm and Elitist genetic algorithm. For the simulation a well-known job shop scheduling problem was implemented and performances of NSGA-DAR (Non-dominated Sorting Genetic Algorithm-Dynamic Application of genetic operators and population partial re-initialization) were tested on the ft10 test instance and the obtained results were found encouraging.

$$\prod(x) \begin{cases} 1, \; if\, D\, dominates\, P1\, and\, P2 \\ \max\left(1 - \dfrac{k1 * t}{G}, 0.5\right), \; if\, D\, dominates\, one\, parent \\ and\, no\, dominance\, relation\, with\, other\, parents \\ \max\left(0.5 - \dfrac{k2 * t}{G}, 0\right), \; if\, D\, is\, dominated\, by\, at\, least\, one\, parent \\ 0.5, \; if\, no\, dominance\, relation\, exist\, between\, D\, and\, P1, P2 \\ 0, \; otherwise \end{cases} \quad (1)$$

Shimodaira [78] employed the concept of incest prevention and proposed a new genetic algorithm Diversity Control-oriented

Genetic Algorithm (DCGA). Incest prevention was applied in survival selection and this increased the diversity by giving individuals with father Hamming distance from the best solution a higher probability of survival [77]. Matsui [79] incorporated two new selections scheme, namely correlative tournament selection (CTS) for reproduction and correlative family based selection (CFBS) for survival. CTS choose the mate with higher fitness and Hamming distance from a set of candidates.

$$\prod(x) \begin{cases} 1, & \text{if } M(s) \text{ dominates } s \\ 0, & \text{if } s \text{ dominates } M(s) \text{ or } M(s) \text{ is not valid} \\ \max\left(1 - \frac{k3*t}{G}, 0.5\right), & \text{if no dominance relation} \\ & \text{exists between } s \text{ and } M(s) \end{cases} \quad (2)$$

A similar approach was proposed by Fernandes and Rosa [80] called as negative assertive mating into GA with varying population size (GAVaPS). They showed that GAVaPS mimic natural species' behavior by selecting parents, according to parenthood or phenotype similarity. A strategy of choosing different crossover methods was proposed in [81]. It was suggested that one can choose the appropriate crossover mechanism to avoid premature convergence on the basis of the couple's Hamming distance. In contrast to incest prevention, three new methods of selection of mating pairs for GA were suggested [82]. It was tried to mimic inbreeding in nature which restricts individuals to mate with one separated by a minimal Hamming distance. Experimental results showed that the proposed approach could enhance exploitation, reduce the disruption of schema and perform better than a conventional genetic algorithm (CGA) and incest prevention in certain problems.

4.2.1. Strengths and weaknesses

This approach works effectively when combined with multiple crossovers on multiple parents. Therefore, it was given a name Multiple Crossover on Multiple Parents with Incest Prevention (MCMPIP) [76]. It was found effective for multimodal function optimization. In addition, it avoids premature convergence and improves the solution at the expense of convergence speed [77]. The approach suggested in [31] is helpful in two fold. First, dynamic application of genetic operators takes the advantages of the identification of the operators to promote the beneficial effects of all the available operators during all evolution. Second, the population partial re-initialization achieves the main goal only in the critical situation, when the risk of premature convergence is big enough which saves the computational cost.

The drawbacks of this approach are: it suffers with the problem of an increase in computation cost for calculating the individual's Hamming distance. Comparison of individual Hamming distance with threshold increases the computational cost.

To overcome the problem (additional cost requires in computing the individual's Hamming distance) of incest prevention, Craighurst and Martine [83] considered the family tree to disallow incest by adhering to the ancestry based incest tree. The level of incest law helps in allowing or disallowing the mating among family members. Experimental results prove its effectiveness on solution quality and diversity maintenance. This approach showed the slow convergence rate and was sensitive to the mutation rate while incest prevention yielded a satisfactory outcome.

4.3. Scheduled sharing approach

Palmer, and Stephen [15] presented a nature inspired approach known as *Scheduled Sharing* to prevent premature convergence by applying an extension called as sharing to standard GA. This approach handles premature convergence by requiring organisms within a niche to compete with their neighbors. Initially, one picks the niche size as large as possible to encourage the organism to spread across the search space. Thereafter, exponentially decays and allow gradual clustering of the organism. This approach is the extension to the standard sharing algorithm [16,17]. The approach given in [15] over [16,17] is the modification of fixed niche sharing to handle more difficult and realistic test function.

Palmer and Stephen [15] not only worked on sharing algorithm, but also they worked on sampled sharing, which removes $O(n^2)$ communication between processors by standard sharing. Experimental results prove that scheduled sharing can delay the convergence as long as require which continues the search process efficiently. To calculate the convergence of a population search space can be divided into subspace or buckets. Suppose, x_i represent the percentage of the population of ith bucket, then the percentage of the total population can be calculated as:

$$\text{Percentage of Total Population} = \sum_i x_i^2 \quad (3)$$

Eq. (3) is an important measure to decide the convergence of a population, higher value represents that the population is more concentrated in one or a few buckets while the lower measure shows that populations are more distributed across the buckets. Two classes of landscapes were used, namely *Easy* and *Difficult* landscapes generated by the weights W_i of the sub-functions. Easy landscapes are those have a small number of local optima with wide basins of attraction while difficult landscapes have just opposite of the easy landscapes. Experimental results prove that both *Scheduled Sharing* and *Non-Sharing* algorithms performed better than the *Random Search* algorithm at all population sizes on both classes of landscapes and prevent premature convergence of a genetic algorithm [15].

4.3.1. Strengths and weaknesses

This approach was found effective on large population genetic optimization of rugged two dimensional landscapes. It is helpful in improving offline performance by requiring organism within a niche to compete with their neighbors. It was shown that it allows the application of sharing for complex problems where there is no definable best niche size, without violating black box principle in the calculation of niche size. This approach takes the advantages of simulated annealing in calculating the niche size by applying an exponential decrease in the schedule. It reduces the computational cost by allowing gradual clustering of organism in areas of greatest interest. It provides flexible control of the convergence of a GA. This approach allows the feedback to define a problem dependent sharing schedule.

Although this technique outperformed over random search and standard sharing techniques. But still there are some limitations.

- The key concern of this approach is finding the appropriate schedule for applying sharing.
- It requires prior knowledge to write the appropriate sharing function which is a difficult task because of a real optimization problem, no information about the search space and distance between the optima is generally available.
- It was found that this approach improves the performance of standard sharing approach by incorporating the concept of cluster analysis and dynamic niching. But still the computational time to obtain the fitness of individuals dominates the computational cost of comparisons.
- It was accepted that to make the schedule sharing more effective, a difficulty detector is needed which might be used to decide upon the schedule [15].

4.4. Migration model and nCUBE based approach

Balakrishnan [24] presented a *Migration Model* to maintain the diversity and variation in the genetic property of individual population which keeps the GA search process alive. The behavior of migration model is very similar to an ordinary communication model, which allows individuals to migrate from one population to another. Therefore, migration model helps in finding global solution and prevent premature convergence. Features of nCUBE model were used to design the migration model because it has several reasons as listed in [24].

Three different scenarios namely *Best Individual*, *Mean Individual* and *Random Individual* were suggested *to* migrate an individual from one population to another. It was observed that once an individual reached to target population the migrated population gets welcomed in a different ways, such as *Warm*, *Neutral* and *Cold* welcome. Cold welcome was used for the simulation in [24].

Two experiments were conducted to ensure the effectiveness. The first experiment tests the suitability of migration of individual based on different migration strategies while the second experiment gives the answer to the scalability issue of the system [24]. The simulation results of both the experiments were found effective to prevent premature convergence. In addition, it improves the performance of GA to solve the complex optimization problem.

4.4.1. Strengths and weaknesses

The present approach was powered by maintaining independent population and allowing inter population migration to maintain the diversity and alleviate premature convergence. nCube model was used for the experiment which improves the performance of GA search. In addition, it was evident that this approach is one of the best approaches for complex optimization problems.

But still there are some key issues in applying this technique. First, setting the migration interval play an important role because both large and small intervals being desirable. If the migration interval is too large then it increases the possibility of premature convergence whereas the small migration interval increases the interaction among the population which results in an increase in the computation time. Second, making decision for an individual to migrate out of a population is a crucial factor. Third, how to choose number of independent populations because solution status will be affected if more independent populations are available.

4.5. Cooperation based approach

Cooperation based genetic algorithm to prevent premature convergence was presented in [18]. The fact which was shown in [18] is that the hardness of a problem inherently related to the way we represent the problem. Therefore, the difficulty level of the problem can increase or decrease based on the representation techniques used. To prove this fact 1's counting problem (max1 problem) was implemented and the difficulty level showed using different representation techniques.

The idea of cooperating genetic algorithm says that instead of using only one representation apply varieties of GA concurrently each optimizing the same object but using its own representation. In other words, after a fixed number of generations, each GA provides other GA's with its best local optima found. Experimental results were found effective for solving those problems which was either difficult or cannot be solvable by sequential GA.

4.5.1. Strengths and weaknesses

Fonlup et al. [18] claimed that the varieties of representation techniques could be helpful in avoiding premature convergence. Experiments were conducted to prove the claim which showed the effectiveness of the presented approach. Various representation techniques are the key concept in removing the correlation between the neighboring individual and maintains the diversity of the population. This approach improves the speed without applying parallelism. An exchange in individuals creates new paths of research and allows GA to choose the new created paths.

The weaknesses with this approach are: (a) selection of the appropriate representation techniques to maintain the diversity in the population because how one can say that the new representation technique will work effectively. (b) In the presented model [18] it was considered that after 20 generations each GA_i broadcasts to every $GA_{j|j \neq i}$ it is 10 individual, but what is the guarantee that same assumption will work for any other problem. (c) Dealing with more than one representation is not an easy task.

4.6. Syntactic analysis of convergence

The average Hamming distance of a population was used as a syntactic metric to prevent premature convergence and predict the convergence time [29,30]. The primary goal of using syntactic metric is to get the probabilistic bound on the time convergence of GAs. It was shown that the analysis of flat function provides worst-case time complexity of static functions which develops a theoretical basis for the premature convergence.

It was observed that mutation is one of the ways of maintaining diversity of the population [29]. But the high mutation rate raises problems. In addition, it also increases the risk to destroy good schemas as bad ones. These observations were worked as motivational factors to come-up with new approaches.

Therefore, it was suggested that in place of increasing mutation rates, pick an individual and add its bit complement to the population [29]. The aim of doing this was to ensure that each allele is present and the population spans the whole encoded space of the problem. The presence of complementary individuals also makes a GA more resistant to deception. Depending on the assumptions about the search space, one can pick individual to be complemented. Various approaches were suggested for doing this such as probabilistic, random selection and others. It was found that randomly selecting the individual to be complemented makes the least number of assumptions and it may be the best strategy in the absence of other information. For conducting the experiment minimum fitness individual were replaced by either the complement of a randomly picked individual in the current population or the complement of the current best individual.

If l_i represent the number of bits needed to represent variable i in a deceptive problem, then the functions used in [29] can be described using Eq. (4).

$$Deceptive(x_i) = \sum_{i=1}^{n} \left\{ \begin{array}{ll} x_i & if \quad x_i \neq 0 \\ 2^{l_i+2} & if \quad x_i = 0 \end{array} \right\} \tag{4}$$

For 10-bit and 20-bit deceptive problem *Dec*1 and *Dec*2 were used which can be represented as:

$$Dec1 : Deceptive(x_1^2, x_2^8) \tag{5}$$

$$Dec2 : Deceptive(x_1^2, x_2^8, x_3^5, x_4^5) \tag{6}$$

In Eqs. (5) and (6) superscripts denote the number of bits needed for each variable in *Dec*1 and *Dec*2 respectively.

The results obtained experimentally were compared with classical GA (CGA) and a GA with compliments (GAC), which proves that GAC easily reached to the optimum whereas CGA fails to find the optimum.

Louis and Gregory [29] accepted that the GAC does not give guarantee for an optimal solution to all problems, whereas Messy GA [45] can make much stronger guarantees. But the drawback of the messy genetic algorithm is its computational complexity. It was

also shown that GAC performed much better than the CGA on Ackley's One Max [46] and no worse on the De Jong test suite [38]. Also, it was agreed that GA hard problems that are both deceptive and epistatic may need masked crossover or other length independent recombinant operators to be solved successfully using complements [47].

A time convergence model was presented to predict the time to convergence [29]. The string similarity assumption was made to develop this model. This model works successfully for unimodal function. The behavior of multimodal function, genetic drift and others can also be predictable using this model. The model starts with the Eq. (7) which computes the change in Hamming average per generation.

$$h_{t+1} = f(h_t) \qquad (7)$$

where h_t = Hamming average in generation t and h_{t+1} = Hamming average in the next generation.

Two observations were made for choosing the function $f(h_t)$. The first observation says that "*the schema theorem along with the similarity assumption indicates $f(h_t)$ is linear* (see Eq. (8))".

$$h_{t+1} = ah_t + b \qquad (8)$$

The second observation says that "*without mutation the final Hamming average is zero*" which implies that $b = 0$. By putting this value in Eq. (8), we will get.

$$h_{t+1} = ah_t \qquad (9)$$

The general solution of the Eq. (9) is given as:

$$h_t = \left\{ \begin{array}{ll} l/2 & t = 0 \\ a^t h_0 & t > 0 \end{array} \right\} \qquad (10)$$

The value of *a* can be obtained by keeping track of the Hamming average, while running a genetic algorithm [29]. It was shown that the model works effectively and a good prediction of time complexity is possible.

4.6.1. Strengths and weaknesses

The primary strength of this approach is that it provides the time limits to the convergence of the GA. In addition, combining fitness prediction with Hamming average prediction provide the details about how much progress is possible with a GA along with bounds on how much remaining work can be done. Syntactic analysis model forms the basis to see the effect of selection and mutation on the behavior of GA. It was evident from the result that good predictions of time complexity are possible, even from a rough model that uses easily computable syntactic information.

The problems with this approach are: mutation rate affects the performance of the GA in the later generation. In addition, computing upper bound and lower bound is not cost effective process. It was accepted that qualitative prediction may not be possible with syntactic information [29,30].

4.7. Pygmy algorithm

Rayan [43] described a new selection scheme to reach to optimal solution. Actually, he suggested *Pygmy* algorithm to deal with multi-objective problems. This algorithm was originally applied to the problem of evolving minimal sorting networks, which must not only be able to sort numbers, but also in as few less number as possible [43]. This approach uses two fitness functions and maintains two lists of parents, one for each criterion of the problem. It was clearly shown that if the population using Pygmy algorithm then the chances to converge to an optimal solution is more compared to other techniques uses a fitness function which is a sum of the assumed criteria. In [44] Pygmy algorithm was applied for handling premature convergence in the case of grammar induction. In order to implement Pygmy algorithm and to select the resulting grammar with minimum number of rule principle was used which was considered as another fitness criterion to create a second parent list.

4.7.1. Strengths and weaknesses

This approach uses a less computational cost compare to the previous approaches such as Crowding [38], Sharing [16], restricted mating, spatial mating and others. It reduces the risk of losing genetic material compared to the methods which reward parsimony [86] or try to combine the two measures [87] run the risk of individuals trading off various parts of the fitness function. Pygmy algorithm maintains two lists of fitness which removes the possibilities of inbreeding between individuals with similar performance. In addition, this approach avoids the chances for individuals to breed with close relatives like siblings or even parents of opposite sex.

The weakness with this approach is that in some situations it is not well suited for multi-objective problems. In order to get the solution of multi-objective problems an individual must solve two or more smaller problems of the same nature to solve the main problem. Pygmy algorithm does not address some of the sub problems simply because individuals in the same group could not mate with each other.

To address the above issue Ryan [43] presented *Racial Harmony*. It was evident in [43] that by applying races among the individual in the same group the performance of Pygmy algorithm was improved. *Racial Preference Factor* (RPF) was used as a measure of probability that an individual will choose an individual from another race when selecting a mate.

4.8. Adaptive probability based approach

In many previous works [68,88–90] the idea of adapting crossover and mutation was applied for the betterment of GA performance. Schaffer et al. [88] implemented a crossover mechanism where the crossover points were chosen depending on the performance of the generated offspring. Davis [89] suggested mechanism of adapting probabilities based on the performance of the operators. He allotted higher probabilities to the operators which create and cause the generation of better strings. The technique presented in [89] was developed in the context of a steady state GA [90]. Fogarty [90] showed the effect of varying mutation rate over generations. In his work he showed that the exponential decrease in mutation rate shows better performance for a single application. Whitley et al. [68] presented that the probability of mutation is a dynamic variable parameter which can be determined by the Hamming distance between the parent solutions. It was evident in [68] that adapting mutation probability dynamically significantly improves the performance.

Adaptive Genetic Algorithm (AGA) (Srinivas and Patnaik) is one of the efficient methods for optimizing the multimodal function [19]. This approach differs from the existing approaches [89,90] as crossover and mutation probability was determined for each individual as a function of its fitness. Whitely et al. [68] implemented adaptive mutation approach in the context of steady state GA, whereas AGA is based on generational replacement strategy. AGA employ best solution protection strategy which resolves the high level disruption while steady state GA applies populationary elitism in which there is no need to protect the best solution.

To prevent premature convergence during the implementation of GA a probability based approach was adopted which varies the probabilities of crossover and mutation depending on the fitness value of the solution. A detailed design was given for suitably choosing the crossover (P_c) and mutation (P_m) probabilities [19]. The

objective of this approach was not only to present a mechanism to choose crossover and mutation probabilities but also present a suitable technique to prevent the GA from getting trapped at a local optimum. It was shown that the adaptive value of crossover and mutation should be from 0.0 to 1.0 and 0.0 to 0.5 respectively. It was proven that the low value of the crossover probability and high value of the mutation probability lead to premature convergence or a shift from genetic algorithms to random search. These facts help in preventing premature convergence of the GA.

Extensive experiments were conducted to show the effectiveness of *AGA*. For the experiment various optimization functions were used such as De Jong's functions known as spiky function (*f*5, *f*6, *f*7), *Order-3 Deceptive* problem, *Traveling Salesman Problem* (*TSP*) and many more. The outcome of the experiment was found to be encouraging.

4.8.1. Strengths and weaknesses

The main strength of this approach is that it provides a basis for adapting the crossover and mutation probability depending on the fitness value of the solution, i.e. for low fitness value assign high value of P_c and P_m and vice versa. AGA protects best solution from the high levels of disruption which receives zero P_c with small amount of P_m this information is helpful in reducing the extra computation cost of applying crossover. In this approach a criteria were set as "all solution with a fitness value less than the average fitness value of the population have P_m = 0.5 which shows that sub-average solutions are completely disrupted and totally new solutions are created" this criteria was also found effective in saving the computational cost and helps GA not to get trapped at a local optimum.

As discussed, this approach gives a concrete basis for selecting the crossover and mutation probability. It was found that most of the individuals were trapped at a local optimum because the exploration effect of mutation was low. To address this issue Jung presented a *Selective Mutation* based approach (discussed in Section 4.21) [35].

4.9. Social disaster technique

Grefenstette [91] attempted Traveling Salesman Problem and suggested the concept of greedy crossover. The crossover operator implemented was called "greedy crossover" because it always prefers locally cheaper paths. Picking a locally cheaper path was based on the knowledge about the problem and showed promising results than more expensive one. It was evident in [91] that crossover based on knowledge, improve convergence, but it might derive to local minima and leads to premature convergence.

To address the above issue Social Disaster Technique (Kureichick et al.) was suggested which considers the whole population [42]. Kureichick et al. [42] directed their efforts to avoid the following two reasons:

- Immoderate crossover greediness
- Low influence of random factors intended to make the population get out of local minima.

The concept of super-individual was suggested which takes over the charge of the population. Catastrophic operator was used in this method to maintain the population diversity. Packing and judgment day operator was used for the experiment. The working of packing operator is to apply separation, i.e. separates all the tours have the same length and only one remains unchanged, whereas in case of judgment day operator only the best instance remains unchanged. In addition, warp operators were used in population, which introduces partial or full randomization.

An experiment was conducted on the Traveling Salesman Problem (TSP) using the benchmarks given by Oliver's 30 [93], Eilon's 50 and Eilon's 75 [92]. It was shown that the thirty town problem was solved easily while 50 and 75 town TSPs was found harder in the presented approach. The result of 5 and 75 TSPs was compared against the results given in [92] and it was observed that the suggested approach showed the shorter tour.

4.9.1. Strengths and weaknesses

The method described above brought significant improvements, some of them are: introduced new operators provide the facility to use the population pool effectively. It is suitable for the small population size problem. Warp operator brought the partial or full randomness which helps the population to get out of the local minima. Decrease in the greediness of the crossover operator increases the variation among the population.

The weakness of this approach is: there was no sufficient information is provided to tell up to what extent greediness of crossover should be decreased. This approach does not fit for complicated problems, i.e. for large population size problems this approach consumes more time to converge.

4.10. The island model genetic algorithm

Vose [94] and Whitley et al. [95] presented exact models of an SGA using infinite population assumption. One derivation of these models was presented in [95] while it was extended to look at finite populations using Markov models [94]. It was found that the weakness of the infinite population model was that one could not take into account the sampling bias introduced by using finite populations, whereas the finite population Markov model was too expensive to execute very small problems. Therefore, it was suggested that using a distribution taken from a finite population one can introduce finite effects into the infinite model.
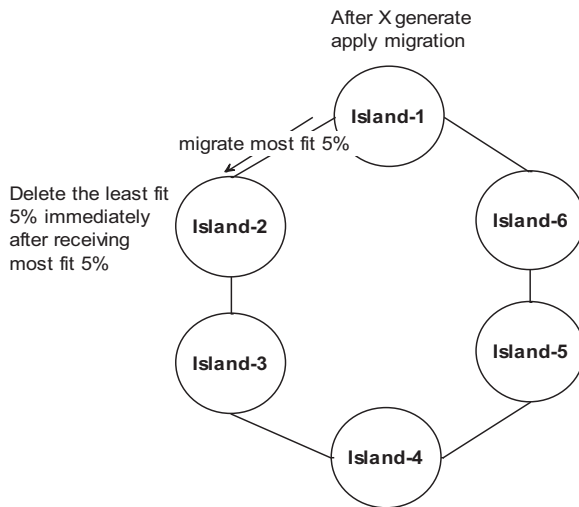
Whitley et al. [69] suggested an application of the infinite population model to see the behavior of multiple SGA running parallel. These SGA was allowed to exchange information time to time. It was shown in previous research that parallel GA yield better performance than serial [70–72]. Island Model was used to implement GA on both serial and parallel machines [65–68]. In an island model each machine executes a GA and maintains its subpopulation for directing the search. It was observed that an island model uses multiple populations for each machine in parallel implementation which helps to preserve genetic diversity, since each island can potentially follow a different search trajectory through the search space [69].

In order to implement island model two parameters were introduced, namely *migration interval* and *migration size*. The number of generations between migrations was represented by a migration interval while migration size is the number of individuals in the population to migrate. Each machine in the island model periodically exchange a portion of their population using a process known as migration.

The function *G* was used to represent trajectory of the infinite population SGA, which works in a vector *p* [94]. p^t represent the population at the time *t* which shows the sampling distribution of a finite or infinite population. The search process enters into new generation using function *G* as depicted in Eq. (11).

$$p^{t+1} = G(p^t) \tag{11}$$

where p^{t+1} represent next generation an infinitely large population. It also represents the expected sampling distribution of a finite population [69]. The construction details of *G* was given by Whitley [96]. It was observed that the behavior of the function *G* was exactly correspond to the SGA provided the infinite large

Fig. 4. Basic organization of Island model using six islands. For first migration after X generation migrate most fit 5% from Island-1 to Island-2 (immediate neighbor on the left). As soon as Island-2 receive the migrated data immediately delete the least fit 5% of the population the same process repeated to every other island excluding itself.

population. The details about the finite and infinite population were covered in varying population size (small and large) [69].

Whitley et al. [69] used function *G* to develop an idealized Island GA. The idea was total population was divided into a number of subpopulations called as "*islands*". These islands are responsible to execute a version of GA on each subpopulation. Each island manages a separate copy of *G* for SGA using 1-point crossover. The crossover rate chosen was 0.6. It was found that migration occurs between subpopulation in a restrictive manner using temporal and spatial considerations.
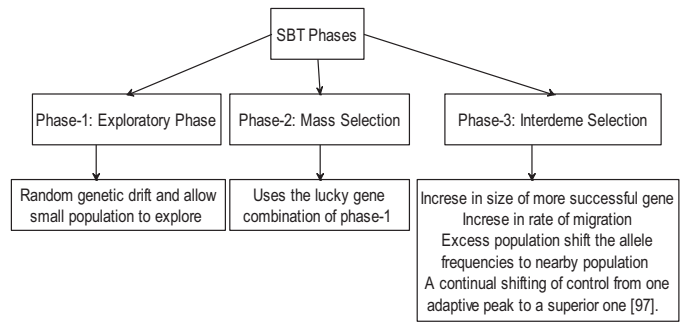
The migration strategy applied in Island model is depicted in Fig. 4. We have used six islands to demonstrate the working. From Fig. 4 it is clear that all the islands are arranged in a ring. For the first migration Island-1 migrate most fit 5% after X generation to its immediate neighbor in the left i.e. Island-2. As soon as Island-2 receives the migrated information it deletes the least fit 5% of its population. For the second migration Island-2 migrate the most fit 5% to its immediate neighbor to its left i.e. Island-3 after X generations. This migration process occurs every X generation until each island has send one set of strings to every other excluding itself then the process repeated [69].

Experiments were conducted using two separable and non-separable problems. Two separable problems were chosen because they represent examples from a well-known set of linearly separable test problem while the non-separable problem was chosen to demonstrate how the added interaction between parameters can affect the GA performance. GENITOR was used as a search engine for the experiment [73]. The experimental results shown in [69] prove that the Island Model genetic algorithms outperform and maintain the population diversity, avoid premature convergence and reach to the global optima.

4.10.1. Strengths and weaknesses

Island GA helps to preserve genetic diversity because every island potentially follows a different search trajectory through the search space [69]. An abstract model was presented which support it to outperform single population models on linearly separable problems. It is suitable for large total population size problems.

This model drawn the attention of researchers since it offers some degree of independence and hence explore different regions of the search space while at the same time sharing of

information allowed. The concept of migration was found effective which helps in maintaining population diversity and the key to resolving premature convergence.

The limitation of this approach is the migration interval, i.e. after how many generations one should apply migration of best fit individual from one island to another. Another factor which was not defined clearly was migration size. For the presented experiments 5% best fit individual was migrated but what is the guarantee that it will be suitable for all types of optimization problems. It was accepted that the presented approach could show very complex and unexpected behavior when applied to separable problem.

4.11. Shifting balance theory in dynamic environment

Wright [20] presented *Shifting Balance Theory* (SBT). The ill behavior of evolutionary system was observed, i.e. getting trapped at a local optimum. It was explained that for a fitness landscape there might be two peaks or even more. In rugged field of this character (very many peaks), selection will easily carry the species to the nearest peak, but there may be innumerable other peaks which are higher but separated by "valleys". Wineberg [97] showed he thought of evolution as a mechanism by which the species may continually find its way from lower to higher peaks. The term "trial and error" was suggested as exploratory mechanisms which take the species to the next higher peak, despite the harsh selection pressures of the valley forcing the species back to the original hill [98].

Hartl and Clark [98] presented a good summary of SBT in their book "*Principles of Population Genetics*". They suggested that subdivision of the population into a set of small semi-isolated demes increases the possibility for demes to explore the full range of the adaptive topography and reach to the highest fitness value on a convoluted adaptive surface. Reduction of large population into smaller demes reduces the possibility of random genetic drift of allele frequencies and provides the flexibility to explore the adaptive topography more or less independently. It was found that the random genetic drift in any subpopulation leads to temporary reduction in fitness. This could be rectified by selection in a large size population which leads the search process to pass through a "valley" of reduced fitness and possibly end up "climbing" a peak of fitness higher than the original. It increases the size if a subpopulation reached to the adaptive peak on the fitted surface. In addition, migration of individual genes will be higher and therefore the favorable gene combinations are gradually spreading throughout the entire set of subpopulations by means of *interdeme selection*.

Hartl and Clark identify three distinct phases and demonstrated the role of each phase [98] as depicted in Fig. 5.

Wright's Demes was one of the significant contributions in justifying the introduction of a novel GA implementation

Fig. 5. Three distinct phases of shifting balance theory.

particularly for parallel GA. It was observed that both fine-grained and coarse grained model (Island model) is presented when creating parallel GA. These techniques use "demes" to indicate a mating pool of individuals locally clustered geographically [97]. We have already discussed Island model (Section 4.10) in which each processor runs an SGA on its own subpopulation or deme.

Cohoon et al. [99] initiated the association of the Wright's SBT with Island model style of parallel GA. Cohoon et al. [100] justified the applicability of island model. He presented the idea of *punctuated equilibria* for parallel GA using Wright's theory [20] and Eldrege and Gould's theory [101] of punctuated equilibria. Similar to island model, many researchers worked on a fine grained model using Wright's theory [102–107]. In this case, each processor is given only a single member of the population. Each member is permitted to mate with its nearest neighbors or possibly within a small neighborhood. The basis of this design was the fact that fast communication may be achieved among neighbor. The neighbors can be arranged using various topologies [97]. Here, diffusion of genetic materials occurs through neighborhood overlap which was identified as demes.

Although Wright's theory was accepted and applied in several applications. But this theory was found controversial. Coyne, Barton and Turelli [108] questioned about the validity and importance of SBT. Coyne et al. [112] studied both theory (Fisher's theory and Wright's theory) and rejected the idea that "*Wright's SBT theory has played a major role in adaptive evolution*". Instead Darwin's and Fisher theory [111] "*adaptation is natural selection acting on differences among individuals without genetic drift, population subdivision and differential migration playing the vital roles hypothesized by the SBT*" was accepted [112]. Peck, Ellner and Gould [109] and Wade and Goodnight [110] presented a framework to support the SBT theory and claim that the dismissal of SBT is premature.

As discussed the SBT theory was controversial still Oppacher and Wineberg [21] presented the modified version of *SBT* in which the original goal was to avoid defects inherent in its original formulation and prevent premature convergence in GAs. The new algorithm implemented is known as *Shifting Balance Genetic Algorithm* (*SBGA*) which improves the performance in a dynamic environment. Two different types of population groups, namely *Core group* and *Colonies* were used in the modified SBT. The core group is the large central population, while a smaller group called as colonies. The main job of colonies is finding the search space of the landscapes that the core group is unable to find. In this way one can prevent premature convergence. To develop a complete system few measures were considered such as population diversity (measured by Eq. (12)), and distance (measured by Eq. (13)). Distance represents the distance between a single chromosome and population in gene space, is a modification of the diversity measure [21].

$$Diversity(P) = \frac{1}{LN(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} HD(p_i, p_j) \qquad (12)$$

$$Dis \tan ce(c, P) = \frac{1}{LN} \sum_{i=1}^{N} HD(c, P_i) \qquad (13)$$

where L represents the length of a chromosome, N is the size of the population P, p_i is the ith chromosome in the population, HD represents the Hamming distance function. The term $HD(c, P_i)$ shows the Hamming distance between a single chromosome c whose distance is being measured and population P_i.
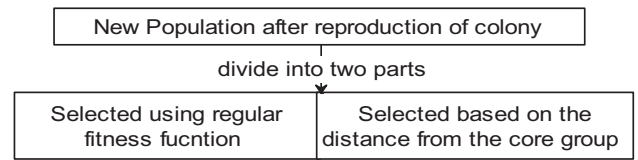

Fig. 6. Structure of new population after reproduction of colony.

One more measure was considered known as containment (Eq. (14)). It was considered as a measure of the extent to which one population is contained within another.

$$Containment(A, B) = \frac{1}{M} \sum_{i=1}^{M} WithinDis \tan ce(a_i, B)$$
$$= \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} d_p(a_i, b_j) \qquad (14)$$

Where, $within \, Distance \, (c, P) = \frac{1}{N} \sum_{i=1}^{N} d_p(c, p_i)$

$$d_p(a, b) = \begin{cases} 1 & if \quad Dis \tan ce(a, P) < Dis \tan ce(b, P) \\ 0 & Otherwise \end{cases}$$

a_i and b_j is the ith and jth member of the population A and B, respectively, and M, N represents the population size of A and B, respectively.

Table 1 presents the state of affairs of the colony and the remedy suggested to keep colony away from the core. The measure of containment (containment function) can be used for this purpose which can be calculated using Eq. (14).

It is depicted in Fig. 6 that when the colony is ready for reproduction the new population was divided into two halves. Member's of the first part was selected using the regular fitness function, whereas the member of the second part was selected based on the distance from the core group using the fact that the greater the distance higher the fitness.

Unlike the SBT theory of immigration of colony members, sending members from the core to the colony, Oppacher and Wineberg [21] suggested no migration from the core to the colony. Although the colony still sends migrants back to the core. It was observed that colony members were selected randomly, stochastically or using elite subgroup for immigration into the core based on their fitness. Adding the colony members into core resulted in a temporary increase in the size which could be reduced to normal size during the reproduction by exerting selection pressure on the core. As the migration of colony members disrupts the core group (see Table 1), hence to rectify this issue, time (migration interval) was given for each colony to evolve potentially useful members.

To ensure the effectiveness of the implemented approach two experiments were conducted using the *F2* function from the De Jong test suite composed with the one dimensional version of the *Griewangk function* known as *F8*. The combined function was used for the experiment called as *F8F2* function. The primary aim of the first experiment was to check the premature convergence while the second experiment was conducted to ensure the dynamic environments. The experimental results were found significantly better and encouraging than the standard GA to prevent premature convergence and also run in a dynamic environment.

4.11.1. Strengths and weaknesses

The main strength of this approach was to resolve premature convergence and up to an extended this approach works successfully. This approach was strengthened by providing the concept of

Table 1
State of affairs of colony and their remedy.

S.No.	State of affairs	Remedy
1	If colony remains completely outside the core.	No action, Freely evolve using regular fitness function, allow looking potentially interesting area ignored by the core.
2	If colony moves or trying to move into area searched by the core group.	Pushed colony away from the core group.
3	If the colony is at the outside boundary of the core	Push colony gently away
4	If the colony is almost entirely contained in the core	Apply strong pressure to push it into novel territory

colonies to search areas of fitness landscapes missed by the core group. The flow of colony member of the core group after a time interval increases the diversity of the core and helps in preventing premature convergence. In addition, it was found suitable in a dynamic environment. In SBGA, if the peak shifts away from the core group the colony are already in an area away from the core.

Oppacher and Wineberg [21] assumed no migration between colony and core and remedial approaches were suggested for the colony member if enters in the core search area still some of the colony member jumps into the core group which might be problematic in managing the diversity. This approach suffers with the problem of migration interval, i.e. when to migrate the colony member. Lastly, it was accepted that there is no dynamic model available of the dynamic environment.

4.12. Random offspring generation approach

Rocha and Jose [22] presented a new approach to remove the insufficiency of GA. This approach is known as *Random Offspring Generation* (*ROG*). By looking at the flowchart shown in Fig. 7 one can easily understand the working of *ROG* process. The flowchart clearly demonstrates the reason of occurring of premature convergence in GA. The answer is for a large amount of individual is using the similar genetic material during the reproduction process. In other words, we can say that for the crossover operation same genetic materials were used. In this case reproduction process will not show any effect because the offspring bred are nothing but simply clone of their parents.

Therefore, to prevent premature convergence in GA, *ROG* process could be applied before applying the reproduction operations.
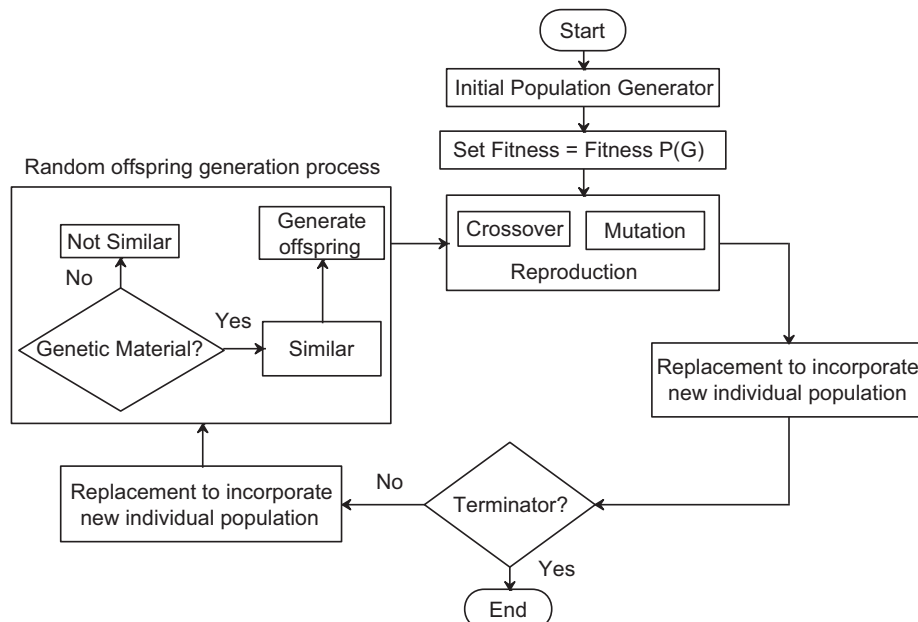
The idea is first test the similarity of genetic material, if genetic materials found similar than generate the random offspring which will code a random solution for the problem otherwise apply reproduction operation. In [22] two different strategies were used in ROG namely *1-RO* (one offspring per two parents) and *2-RO*. In case of *1-RO*, only one offspring will be generated and other one obtained from their parent to reach to the result while in *2-RO* both the offspring will be randomly bred.

ROG strategy was applied to a selected set of Traveling Salesman Problem (TSP) [23]. For the experiment TSP problem was used as a benchmark. The data obtained from the experiments were compared with other approaches such as Adaptive Mutation Rate (AMR) and Social Disaster Techniques (SDT) on various crossover operators. The comparative results prove that the technique presented in [22] is far better to prevent premature convergence to local optima and hence improve the performance of GAs.

4.12.1. Strengths and weaknesses

It was shown experimentally that the presented approach yields the best results compared to adaptive mutation rate (AMR) (Section 4.21) and social disaster technique (SDT) (Section 4.9). As observed AMR showed the solution with a similar degree of quality as obtained by SGA whereas SDT in either of its form didn't show any improvement in the solution. Another important merit of this approach is the simple nature in implementation.

Rocha and Jose [22] accepted that the weakness of this approach is the computational overheads disregarded since there is no change in the selection methodology compared to other approaches such as Crowding [38] (Section 4.1), Sharing [16] and Scheduled Sharing [15] (Section 4.3).

Fig. 7. Genetic algorithm flowchart with a random offspring generation process to prevent premature convergence.
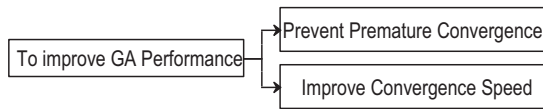
Fig. 8. Key factors play an important role in GAs performance improvement.

4.13. Chaos operator based approach

Juan, Zixing, and Jianqin [26] presented a novel GA (Fig. 9 shows the flow chart) which contains chaos operator within the framework of Markov chain to prevent premature convergence.

Based on the analysis of the previous research it was suggested that two key factors (convergence speed and premature convergence) play a crucial role in improving the performance of GAs as shown in Fig. 8. In addition, it was explained that one can improve GA by the modification of population size; mutation probability and fitness function reduce probability of losing gene and maintain population diversity. But increasing the population size, increase the workload also. Therefore, a dynamic approach for increasing the population size was applied [26]. A dynamic increase in the population size means that applying selections, crossover and mutation up to a certain range of the population and if premature convergence occurs, then increase the population size and then perform the operations again on the modified population size.

It was observed that increasing the population size by generating random individuals affects the performance of the GA searching. Hence, chaotic search was implemented (see Fig. 9) to enlarge the population size dynamically for the GA in which Eq. (15) was used as the chaotic operator.

$$x_{n+1} = ax_n(1 - x_n) \tag{15}$$

The iterator in Eq. (15) may be called as *Logistic* equation. If the value of a = 4 then the system is totally in chaotic state where x might reach to any state in (0, 1). It was found that Eq. (15) is highly sensitive to the initial state. Minute differences say δ may cause divergence after a period of time.

Juan, Zixing, and Jianqin [26] suggested ways to manipulate the chaotic operator which is drawn below:

• Judge the concentrating degree of individuals. For example, compute the difference between the average fitness value and the maximal fitness value as suggested by Srinivas and Patnaik [19] (Section 4.8).
• If the fitness of individual reaches to a certain degree, then use the Eq. (15) to produce a new population together with the current optimal individual as the initial state x_0.
• The genetic operations on the new population to reach to the optimum (threshold) or maximum generation was applied. During this procedure, concentrating degree of the every generation was checked.

Caponetto et al. [113] showed the effect of chaotic sequence for convergence analysis of evolutionary algorithms. The idea presented was based on the substitution of the random number generator with chaotic sequence [113]. Coelho [114] presented a novel Quantum-behaved PSO (QPSO) using chaotic mutation operator. QPSO replaced random sequences and used the application of chaotic sequences based on chaotic Zaslavskii map [114]. The simulation results showed that QPSO was a powerful tool in diversifying the population, which leads in preventing premature convergence to local optima.

Juan et al. [26] implemented the presented approach for multimodal function optimization. It was found that the suggested approach performed better for multi-modal function optimization compared with mono-modal function optimization. The results found to be effective as far as convergence speed is concerned to reach into the optimal state and preventing the premature convergence.

4.13.1. Strengths and weaknesses

This approach works effectively for avoiding premature convergence using chaotic operator. The significance of chaotic operator is given in various literatures [113–117]. As chaos is best suited operator for the highly unstable motion of deterministic system in finite phase space and it disallow the two identical populations even if the initial states obtained by algorithm are very close. These features preserve the population diversity and help GA search process to reach to the global optima. Since, chaos operator ensures the diversity and when combines with crossover and mutation operators (Fig. 9) improve the convergence speed.

Although, it was shown that the present approach works on realizing the relationship between premature convergence and parameters of GA. Chaotic searching increases the population
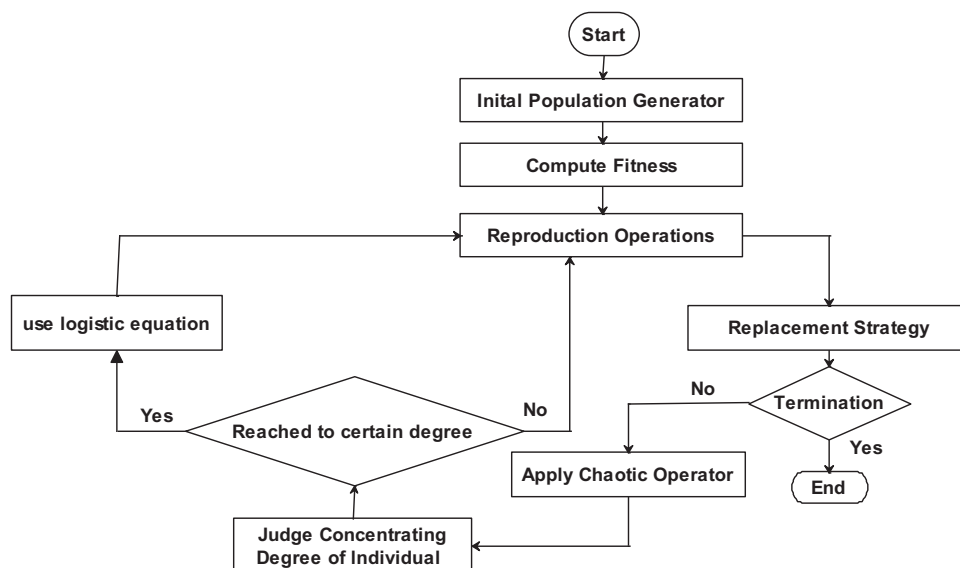

Fig. 9. Genetic algorithm flow chart using the application of chaotic operator.

size dynamically, which introduces the population diversity and prevent premature convergence. But increase in population size brought a tremendous amount of calculation and increases the computation cost. A serious investigation is required to resolve these issues.

4.14. Self-adaptive selection pressure steering approach (SASEGASA)

Affenzeller and Wagner [25] presented a new generic evolutionary algorithm to prevent premature convergence by introducing a new selection scheme. The new selection mechanism maintains the genetic diversity by applying the features of self-adaptive steering of selection pressure. In addition, this selection model not only avoids the premature convergence, but also it enables the intuitive condition to detect premature convergence. The presented selection technique was combined with *Segregative Genetic Algorithm* (SEGA) [55] developed by Michael Affenzeller (2001). SEGA is an advanced GA, which introduces parallelism to improve global solution quality [25].

There is no manageable model exist for handling the selection pressure within the theory of genetic algorithms [56]. Hence, Self-Adaptive Selection Model was introduced in which the concept of virtual population was given to handle the selection pressure (see Fig. 10). The aim of applying this model was to maintain the degree of population diversity. But, to apply this selection model it is necessary to adjust parameters for the actual selection pressure which requires lots of parameter tuning to steer the search process.

The working of selection model can be divided into the following steps:

Algorithm: Self-Adaptive Selection Model

POP: Population, P1 and P2: Parents, FITT_CHLD: Fitness of child population, FITT_PRNT: Fitness of parent population

S1	POP ← population at ith generation
S2	Choose parents P1 and P2 for crossover using appropriate selection mechanisms such as a roulette wheel, linear rank or some kind of tournament selection etc.
S3	Before starting GA search do If FITT_CHLD is better than FITT_PRNT then using Rechenberg's 1/5 success rule for evolution strategies. Filled up the claimed ratio (*SuccRatio*) at $(i+1)$th generation with successful individuals i.e. simply fill $|POP|SuccRatio$. Having filled up the claimed ratio (*SuccRatio*) of the next generation with successful individuals, simply fill up the rest on the next generation ($|POP|(1-SuccRatio)$) with individual arbitrarily chosen from the pool of individuals that were also created by crossover but did not reach the success criterion.
S4	Update the size of the population as: $|POP| = |POP|(SuccRatio) + |POP|(1 - SuccRatio)$

Step3 (S3) was given to check the adoption of Rechenberg's success rule for GA, which says that a child is successful if the fitness of the child (FITT_CHLD) is 'better' than the fitness of its parents (FITT_PRNT). The meaning of better was considered as "if it surpasses the fitness of the weaker, the better, or it's some kind of mean value of both? [25]. Therefore, the concept of simulated annealing (SA) was applied for making appropriate decision. Using this model one can predict the premature convergence as follows: during running GA if one cannot find (*SuccRatio* * |POP|) then in this situation children outperform their own parent which causes the situation of premature convergence. In addition, (*MaxSel Press** |POP|) also shows premature convergence. *MaxSel Press* represents the upper limit of selection pressure.

The algorithm implemented using the above selection mechanism is known as Self Adaptive Segregative Genetic Algorithm with Simulated Annealing aspects (SASEGASA). SASEGASA added two new features in the basic GA.

(a) A new variable was introduced for handling the selection pressure. It also maintains the diversity of the population.
(b) A separation of the population. This concept actually increases the size search space. For getting the final result join the subpopulation at the end. The concept of categorizing the whole population into subpopulation is very similar to divide and conquer [57,58].

By adding these two features the process of SASEGASA will end up with a population, including all the genetic information sufficient for locating a global optimum [25]. In order to test the effectiveness of the method, experiments were conducted on the traveling salesman problem and multimodal combinatorial optimization problems with the aim to see the increase in the number of subpopulation at the start of the evolutionary process allows achieving scalable improvement in terms of global convergence [25].

4.14.1. Strengths and weaknesses
The obtained results showed that SASEGA works effectively in avoiding premature convergence. Adding two new features to SEGA help in maintaining diversity and locating global optima. Virtual population was found effective in handling the selection pressure while the separation of population into subpopulation helped in reducing the toughness of the problem.

It was shown that virtual population and the separation of the population helped GA not trapped at local optima but at the same time these two features could be considered as weakness of this approach. Separation of population increases the population size which leads to a huge increase in computation time. It was suggested that parallel computer architecture will be suitable to apply this approach.

4.15. Multi-combinative strategy

Multi-Combinative strategy was implemented to avoid premature convergence in a *real/binary like coded genetic algorithm* (RBCGA) (Achiche et al.) [59,60]. RBCGA was used in automatic generation of fuzzy knowledge bases (FKBs). This strategy uses varieties of crossover techniques which were applied to the same couple of parents. Two ways were suggested to perform multi-combinative reproduction in GA namely multi-parent recombination i.e. *multi-crossover on multiple parents* (MCMP) and multiple crossovers per couple (MCPC) [61–64]. Both the crossover strategies were implemented to maintain the genetic information from parents within GA search.

Three principle crossover mechanisms such as multi-crossover, premises/conclusion and blended crossover α and uniform mutation were applied. For the selection the following technique was applied: Randomly generate a real value R in the interval [0, 1]. R is then multiplied by S where S is the sum of fitness values of the individual population. Starting with the best individual of the population, the fitness values are summed till the result is higher than RS. The last added individual is assumed as a potential parent.

It was suggested that exploration and exploitation can be used to influence the search process which helps in avoiding the premature convergence and reaching to the global optima. It was shown that the best ways of applying exploration and exploitation is: apply exploration at the early stage of the evolution, relaxing in the middle and exploitation at the end [59].

Multi-combinative approach was applied to 3D surfaces of the type. Three different surfaces such as Sinusoid surface ($z = \sin(x, y)$), Spherical surface ($z = x^2 + y^2$) and Hyper-tangent surface ($z = \tanh(x(x^2 + y^2))$) of different complexities were tested to ensure the effectiveness of this approach. The experimental results showed that MCPC improve the behavior of RBGA in preventing
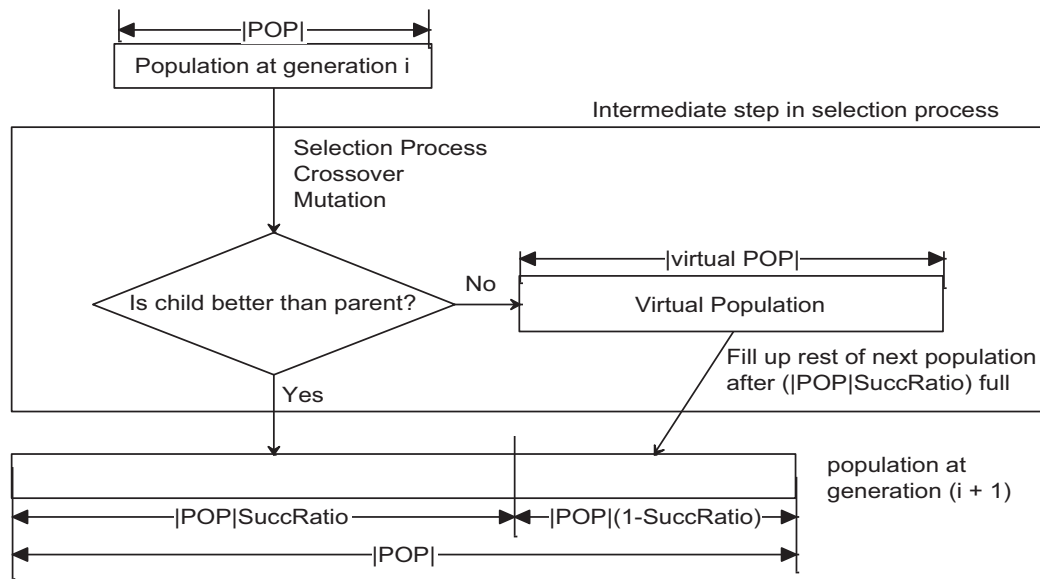
Fig. 10. Flowchart for introducing virtual population in the new selection scheme into a Genetic Algorithm [25].

premature convergence while generating fuzzy knowledge bases using small population size. In addition, it was shown that MCPC resolves the problems of single crossover applications like fast convergence, lack of diversity and reaching a plateau.

4.15.1. Strengths and weaknesses

Different combinations of multi-combinative evolution achieved better results in dealing with RBGA with automatic generation of fuzzy knowledge bases. Creating multiple offspring from the same parents introduces diversity and variation and keeps the search process alive. Also, the exploration/exploitation balance influences the results.

Although this approach showed promising results, we have noticed two issues with this approach. First, increase in the complexity when generating multiple offspring from the same pair of parent's genes. Second, how to deal with too many look-alike individuals at the latter stage of the evolution.

4.16. GA using self-organizing maps

Amor and Achim [6] presented a novel genetic algorithm known as "*Genetic Algorithm using Self-Organizing Maps* (*GASOM*)" to address the problem of premature convergence by intelligent exploration technique. They used self-organizing maps to mine the data from the evolutionary process. *Self-organizing Map* (*SOM*) was used to perform a nonlinear mapping from high dimensional input space onto a two dimensional grid.

Generally speaking, during the implementation of standard GAs only the current population is stored. But *GASOM* says that if we maintain the data of the previous generation, we could gain valuable insight into the way that GA works. By maintaining the historical data we can get the information about the problems encounter in the different generation and then we can come up with the important conclusions in order to guide and improve the search process.

In order to maintain previous generation information an intelligent and efficient technique is required, which can mine and store data. Here, we need to process the incoming data in such a manner by which a new chromosome should not lead to an entire recalculation of previous computed knowledge. In other words, we can say that we want to process high dimensions and independent data entities. Therefore, *SOM* were used to project high dimensional

lattice because this projection does not have to be repeated when new data points are evaluated. Also, *SOM* stores data in a table and uses less memory. The usability of *SOM's* to visualize different entities of evolutionary algorithms was shown in [10].

The training of SOM carried out in the following steps:

(a) Generate the initial weight $m_{i1}, m_{i2}, \ldots m_{in}$ of prototype vector.
(b) Repeat step (c) to (g) until the maximum number of iterations is reached.
(c) Randomly pick the weight vectors m_i of different types.
(d) Ensure that random selection of weight vector should be uniformly distributed; if not, then the learning will be biased.
(e) Evaluate the distance (in Euclidean sense) of all the prototype vectors.
(f) Check for the winning neuron b also known as a *Best Matching Unit* (*BMU*) with the prototype closest to x as:

$$\left\| x - m_b \right\| = \min_i \{ \left\| x - m_i \right\| \} \tag{16}$$

(g) Update the prototype as:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{bi}(t)[x - m_i(t)] \tag{17}$$

where t = current iteration, $\alpha(t)$ = learning rate at t $h_{bi}(t)$ t $h_{bi}(t)$ = neighborhood kernel centered at winning unit and represented as:

$$h_{bi}(t) = \exp\left(-\frac{\left\| r - r' \right\|^2}{2\sigma^2(t)} \right) \tag{18}$$

(h) Ensure the quality of SOM:

$$E(i) = E(j) \quad \forall i, j \in U$$

where $E(i)$ = count function returns the number of times a neuron i was activated.

Each evaluated individual was mapped to the nearest map unit available and store the outcome using the frequency tables, namely *Population Distribution Table* (*PDT*) and *Search History Table* (*SHT*). *PDT* was used to store the information of the activation frequencies w.r.t. the current population represented as $E_p(i)$ while *SHT* keep the track of activation frequencies w.r.t. all individuals computed during evolution represented as $E_h(i)$. Both tables were used to test

the diversity. To test the diversity in the current population *PDT* table was used. Higher the diversity in the population means the higher number of different activated neurons.

On the other hand, *SHT* gives us an insight into the course of evolution. Visual representation techniques were used for interpretation (see Fig. 11).

Fig. 11(a) shows a trained SOM of 10×10 units. Training were done on 16-bits binary chromosomes, represented by a pattern of 16 pixels visualized as 4×4 block which displays the weight vector of that particular unit. Here, black pixel was represented by binary 0 while binary 1 used to represent the white pixel.

If a unit's color is entirely white or black then the corresponding weight vector can be represented as either '1111111111111111' or '0000000000000000' respectively.

Fig. 11(a) also shows that there is a cluster of chromosomes onto neighboring neurons. Fig. 11(b) shows the fitness landscape for the 16bit 1s-counting problem where light color represents the good units, while the dark color represent the worse unit which proves the observation. At this stage a GA was implemented to avoid the dark region and interested in getting the light color i.e. the higher fitness region. Fig. 11(c) shows that the representation of fitness landscape for 1s-counting problem after the implementation of GA where low fitness regions were avoided. Fig. 11(c) also shows that each unit is centered on a high fitness region in the lower right and the upper left corner of the SOM.

GASOM uses two ranks for computing the fitness of a chromosome. One gets the value of the first rank by ordering the population w.r.t. to the fitness values while the second rank results from the novelty of the individual population. Novelty of individual means the total number of similar chromosomes already encountered during the evolution. The novelty factor of current chromosome is represented as. If b_c represent the *BMU* on the trained *SOM* then novelty can be obtained using the function as given in Eq. (19).

$$novelty(c) = 1/E_h(b_c) \tag{19}$$

Individual fitness score (IFS) can be obtained by taking the sum of both the rank.

$$IFS = R1 + R2 \tag{20}$$

where *R1* and *R2* represent the first and second ranks respectively.

Eq. (20) was used to provide two chances to individual population to survive. The first possibility is based on the good performance on the problem under consideration while the second possibility is based on the exploration of new subspace. The activation frequencies of exploited regions were increases drastically during high exploitation. This results in decrease of the novelty factor. Then many individual having low novelty vanishes from the population and provide space to the individual having higher novelty. Hence, an increase or decreases in exploitation also affect an increase of decrease of exploration. In order to regain the lost diversity reseeding operation were applied by introducing individual have high novelty factor. Reseeding process was done in a two-step.

(a) Create a reseeding pool, which consists of a set of buckets distributed uniformly. Each bucket represents a particular neuron on the SOM. Each bucket keeps the information about the newly creating chromosome and represents its BMU.

(b) Then the neurons having lowest activation frequencies are determined. In some situation individual are sampled from respective buckets and inserted into the population. To keep the population size fixed kills the population with lower fitness.

It was shown clearly that the SOM reseeding technique maintains the high exploratory power. But there should be a proper balance between exploration and exploitation to gain the effective and efficient implementation of a genetic algorithm. Therefore,

counting the number of different activated neurons in the population distribution table was suggested to find the amount of exploration in the current state of the GA. Thereafter, a balance function was used to find out the number of neurons that should be activated in the current state of the search process.

If n is the number of fitness evaluation already done, n_{max} represent the maximum number of fitness evaluations and $\|U\|$ is the total number of neurons of the SOM. Then the optimal number of activation units r can be calculated as:

$$r = \left(1 - \frac{n}{n_{max}}\right) * \|U\| \tag{21}$$

To achieve an effective exploration strategy there should be a good combination of the new fitness assignment scheme and the SOM reseeding operator, because if we will apply only reseeding operator then the new individuals would in most cases die immediately and disappear from the population. The reason is we didn't take novelty of an individual into account. Also, if we generate the novel chromosomes then there is no guarantee that it will rely on reproduction operators and only by chance produce new genetic material.

GASOM was implemented on Royal Road function [11], Deceptive F9 function [12] and the generalized Rosenbrock's function [2] and the obtained results shows that the performance of GASOM is far better than other approaches. In addition, experimental results were compared with other approaches such as standard steady-state GA, deterministic crowding [12] and the novel hybrid replacement scheme of Lozano et al. [13]. The results of GASOM were compared against *Steady state*, *Crowding* and *Hybrid* genetic algorithm using average fitness obtained for the test functions. It was observed that GASOM yields the best results out of the tested GA for Deceptive F9 function and the generalized Rosenbrock's function, whereas Hybrid GA performed slightly better than GASOM on the Royal Road function. However, the performance of hybrid method deteriorates on the other two functions.

4.16.1. Strengths and weaknesses

The comparative analysis explains that GASOM is well suited for preventing premature convergence. It performed well on a deceptive problem where other GAs gets trapped at local optima. Also, it maintains good balance between exploration and exploitation. Visualization of GASOM is easier because it offers the use of SOM for the state of evolution.

The fundamental issues one can come across during the implementation of *GASOM* are maintaining the size of *SOM* and the effect of variable length neurons on the performance of *GASOM*.

4.17. Age-layered population structure (ALPS) approach

In the previous literatures several approaches were tried to maintain the genotypic diversity of the population by modifying different parameters during the implementation of GA. As discussed, replacement strategy plays an important role in the population diversity management [12,13,38]. Goldberg and Richardson [16] and Palmer and Smith [15] implemented sharing function and scheduled sharing respectively, which modify the fitness of individuals based on their genotypes similarity. Tanese et al. [67] modified population structure in which individuals have a location and are restricted to interact to interact with their neighbors. The drawbacks with these approaches are that they work on genotypic comparison of an individual; it works effectively with bit strings, then the genetic programs and therefore limited in discovering the solution within the basins of attraction of the initial population. Breaking out the basin of attraction can be achieved by introducing new randomly generated individual in the population. Hence, a chance of a premature convergence problem is fairly common
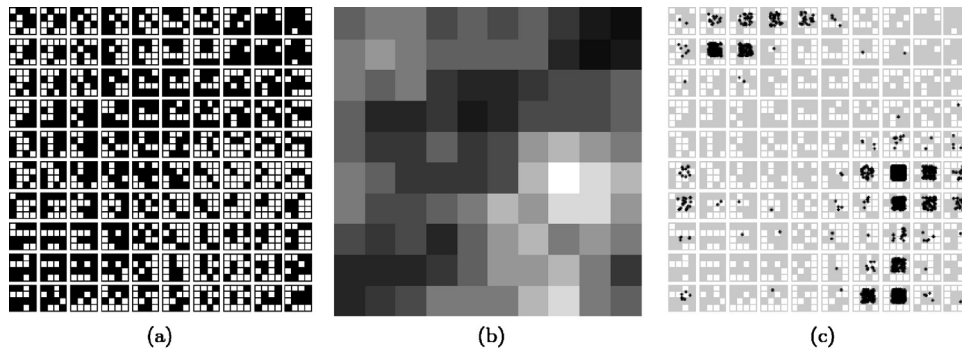
Fig. 11. A trained SOM in prototype representation (a), fitness landscape representation (b) and search history representation(c) [6].

when running GA multiple times with different random number seeds, whereas restarts the search process at least increase the chances of finding the global optimum when the random individual generator can produce every point the fitness landscape, then eventually the global optimal will be found in one of the initial populations, but deciding how long to run the EA before restarting becomes a challenge [41]. If few generations are used then what is the guarantee that the population will get enough time to reach to the global optimum and large number of generations will increase processing time to converge on top of a fitness peak before starting the next generation.

In order to address the discussed issues, Jian et al. [130] presented *Hierarchical Fair Competition* (HFC) model. HFC was based on fitness layer. The idea of HFC was to separate the population into different layers and competition, breeding, selection, and replacement was allowed only within the fitness layer very similar to *Fitness Uniform Selection* (FUS) EA presented by Hutter [131]. But the drawback with this approach was individual, converged to a local optimum near the top of a fitness layer prevent newer individuals in different basins of attraction from climbing through that fitness layer.

To address the insufficiencies of HFC, an adaptive version of HFC was presented known as *Adaptive Hierarchical Fair Competition* (AHCF) [132]. AHCF extends HFC by allowing the admission threshold (determined either initially or adaptively) of fitness levels, which was determined dynamically through evolutionary process. Admission buffer was used to collect qualified candidates synchronously or asynchronously from other subpopulation. In addition, each subpopulation hold export threshold, defined by the admission threshold of the next higher level subpopulation. Another, modified version of HFC was implemented known as *Continuous Hierarchical Fair Competition* (CHFC) [133]. To improve the computational capability of AHCF, Oliveira, A., et al. [134] presented *Parallel Adaptive Hierarchal Fair Competition* (PAHCF) in which some enhancement were suggested in the original HFC such as an adaptive determination of number of levels, more sophisticated admission threshold adaption mechanism and multi-processor parallel implementation. Although, PAHCF showed significantly better result than HFC because of parallel implementation, but other two approaches AHCF and CHCF have not been shown to be significantly better than regular HFC [41].

The *Age-Layered Population Structure* (*ALPS*) was introduced by Hornby [41]. Unlike, HFC and any other evolutionary algorithm, *ALPS* is based on age measure. The key features of ALPS are [135]: it introduces parallel execution among the multiple instances of search algorithm, each age layer assigned a maximum age and not allowed to contain individuals older than that maximum age, age of an individual is based on when the original genetic material was created from random, at regular intervals the search algorithm in the first age layer restarted.
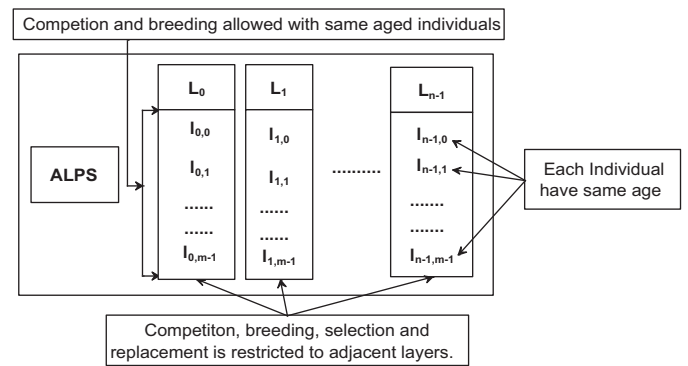

Fig. 12. Age layered population structure demonstrating the layered structure according to the individual's age. Showing the competition and breeding restriction between individuals are based on the age and it is allowed with same aged individuals.

Fig. 12 explains the working of ALPS. It is shown that separation of population has been done according to the individual's age. In addition competition, breeding, selection and replacement is restricted to adjacent layers and allowed only with same aged individuals.

To produce an offspring and individual created through mutation or recombination start with an age of 1 plus the age of its old parents. Each layer has a maximum allowable age for an individual to be in it. This approach checks that how long the genetic materials are participating in the population of the next generation. *Age Measure* can be defined as the total number of generations where an individual genetic material has been evolving in the population. The age of an individual can be computed depending on the situations using Eqs. (22), (23), (24) and (25).

$$Age\ of\ new\ individual = 0 \tag{22}$$

$$Age\ of\ new\ individual = 1 + Age\ of\ older\ parent \tag{23}$$

$$Age\ of\ new\ individual = 1 + Age\ of\ previous\ generation \tag{24}$$

$$Age\ of\ new\ individual = Age\ of\ previous\ generation \tag{25}$$

Eq. (22) is applicable when a new individual was generated randomly, but if we already have the randomly generated population and we have applied the evolutionary operations such as mutation and crossover then Eq. (23) will be applied.

When one has copied an individual for the next generation where the copied offspring is used as a parent, then we use Eq. (24) otherwise Eq. (25). The generalized equation for computing the age of an individual is given in Eq. (26) [135].

$$age = 1 + \frac{evals_{current} - evals_{created}}{popsize} \tag{26}$$
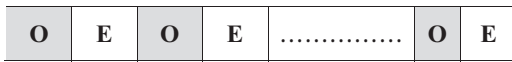
| O | E | O | E | .... | O | E |

Fig. 13. Chromosome structure 'O' and 'E' indicates odd and even, respectively.

The concept of age-layer was introduced to avoid competition and breeding among individuals. The last age layer can have individuals of any age while a maximum age limit was assigned to the other age layer in the population. In order to set the age-limits, different schemes were suggested such as *Linear*, *Fibonacci*, *Polynomial*, and *Exponential*. AGE-GAP parameter was used to keep the population size as small as possible.

To ensure the effectiveness, experiments were conducted and the outcome of the experiment was compared against a handful of other EAs such as *Canonical EAs, Hierarchical Fair Competition Model* and *Multi-start* EA. In all the cases the performance of ALPS-EA was found to be significantly better than others. In addition, hybridizing ALPs was introduced as a future variant to maintain the diversity of the population.

4.18. Strengths and weaknesses

The comparative results of ALPS showed that it outperformed other EA approach. It was observed that ALPS were capable to reduce the premature convergence by continuously generating new sub-population of randomly generated individuals in its bottom layer. It uses individual's age in forming the layered structure of population and restricts competition and breeding, helps in diversity management and give opportunity to new individuals to develop without being dominated by older ones.

The previous work on evolving antenna design optimization for other problems an EA using deterministic crowding had been found best, whereas ALPS showed better than had been produced previously [41]. Although, the author tried to improve the performance of ALPS, but still special treatment is needed to improve this strategy.

4.19. Number structuring approach

Sultan et al. [14] presented odd and even number structuring approach to generate the population to perform GA search. The aim of applying the number structuring scheme was to increase the diversity and preventing the premature convergence. The odd and even number structuring follows the classic binary representation, i.e. 0 and 1. This approach was implemented to solve the timetabling problem. Structuring of the chromosome was done using the sequence of odd and even numbers as shown in Fig. 13.

For the effective implementation of this approach the chromosome should always be in the position of holding odd and even number in each iteration. Therefore, some heuristic was applied to ensure the structure of chromosomes. In other words, a simple adjustment technique (see Fig. 14) was applied to ensure the structure of chromosomes. The OEGA (Odd-Even Genetic Algorithm) which was used for the timetabling problem is shown below (see Fig. 14).

4.19.1. Strengths and weaknesses

It was observed that this approach is comparatively simpler in nature than any other approaches for preventing the premature convergence and to maintain the diversity of the population.

We have observed two issues with this approach. First, maintaining the sequence of odd and even number structure during each generation needs extra processing time. Second, it is not suitable for handling the very complex problems such as multi-modal function optimization or any others.
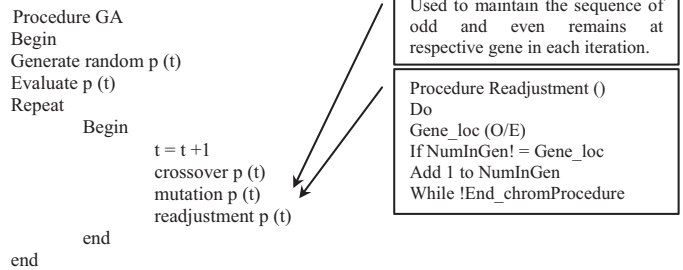


```
Procedure GA
Begin
Generate random p (t)
Evaluate p (t)
Repeat
        Begin
                t = t +1
                crossover p (t)
                mutation p (t)
                readjustment p (t)
        end
end
```

Used to maintain the sequence of odd and even remains at respective gene in each iteration.

```
Procedure Readjustment ()
Do
Gene_loc (O/E)
If NumInGen! = Gene_loc
Add 1 to NumInGen
While !End_chromProcedure
```

Fig. 14. Odd–even genetic algorithm with readjustment procedure used in number structuring.

Table 2
Advantages of *PSO* and *GA*.

Particle Swarm Optimization (PSO)	PSO uses Genetic Algorithm
Algorithmically simpler than GA	Complex approach
Difficult to manage the diversity in the population [40]	Capable to maintain population diversity [39]

4.20. Hybrid particle swarm optimization (PSO) and GA

Eberhart and Kennedy [123] presented *Particle Swarm Optimization* (PSO) and are widely applied to optimization problems due to its high search power in state spaces. Bergh and Engelbrecht [40] showed that PSO is a stochastic optimization technique based on a flock of birds or the sociological behavior of a group of people. PSO is a population based method in which the population is referred as a swarm, consists of a number of individuals known as particles. The gbest was considered as a first PSO model in which each particle i in the swarm keeps the following information such as: current position (X_i), current velocity (V_i), best position (associated with best fitness) of the particle ($pbest_i$) and global best position ($gbest_i$). Although PSO was shown effective for solving many optimization problems such as artificial neural network training [124–126] and function minimization [127,128]. But few limitations were observed during the implementation of PSO:

- The swarm may prematurely converge [40].
- PSO has stochastic approach and performance is largely dependent on the problem.

Hence, to remove the insufficiency of *PSO*, *PSO* using *GA* was suggested which takes the merits of *PSO* and *GA* respectively. *PSO* and *GA* both are used to deal with optimization problems using evolutionary strategy. In [39] modifications were made in *PSO* using *GA* to improve the performance and reach to global maxima. The advantages of *PSO* and GAs are shown in Table 2.

Looking at Table 2, one can easily understand that genetic operators can be used to prevent premature convergence. Using the crossover operation information can be exchanged between two particles which improve the likelihood of searching for the global optimum. Similarly, by applying mutation to *PSO* population diversity can be managed. Three different hybrid approaches were proposed, namely *PSO-GA* (*Type 1*), *PSO-GA* (*Type 2*) and *PSO-GA* (*Type 3*). In *PSO-GA* (*Type 1*) the *gbest* particle position does not change its position over any designated time steps whereas in *PSO-GA* (*Type 2*) the stagnated *pbest* particles are change their position by the mutation operator of *GA*. But in case of *PSO-GA* (*Type 3*) the initial population of *PSO* is assigned by solution of *GA*.

Experiments were conducted on four different benchmark functions: *Shaffer, Rosenbrock, Rastrigin*, and *Ackley*. The outcomes of the experiments were found encouraging for all the three

approaches, but *PSO-GA* (*Type 1*) performed better than the other two approaches and *Standard PSO* for *Rastrigin* function.

4.20.1. Strengths and weaknesses

The experimental results showed that PSO with GA outperform the standard PSO. Hybrid PSO was found effective to find a better solution without trapping in local maximum and achieve fast convergence. Unlike, standard PSO, PSO-GA was found more flexible and robust and produce better quality result in reasonable computational time.

The drawback with this approach is that it sometimes diversifies the particle position without reaching to global optimum or produces a worse solution.

4.21. Selective mutation based approach

Cavicchio [129] presented adaptive search techniques. The purpose was to show the efficiency of adaptive system. The effect of various parameters was tested, including genetic operator probabilities and types and it was accepted that genetic operators probability plays a vital role during the search process. Srinivas and Patnaik [19] suggested adaptive probability based approach for the selection of crossover and mutation probability. In addition, the power of crossover and mutation operations was shown clearly to alleviate the premature convergence. It was observed that most of the individuals get trapped at a local optimum because the exploration effect of mutation was low.

In order to address this issue Jung [35] presented *Selective Mutation* for GA. It was suggested that if the exploratory power of mutation probabilities is high, then the speed of GA to reach to global optimum become slow. Hence, one has to smartly choose the mutation probabilities to address this problem.

Szeto and Zhang [120,121] applied matrix formulation based approach and presented adaptive genetic algorithm using mutation matrix (MOGA) [120,121]. It was shown that mutation matrix can be used in the development of adaptive GA and therefore no external input parameter require during the selection process. MOGA uses mutation probability as a function of time, fitness ranking of the chromosome and locus. The dependence is automatically determined by the statistics of the locus and the fitness ranking of the chromosome at the given time [118]. Looking at the numerical examples on Knapsack problem shown in [118] one can observe the superiority of MOGA over many traditional methods [120–122].

Spears [119] presented the importance of crossover and mutation. Nga and Szeto [118] extended the work done by Szeto and Zhang [120] by including crossover matrix. Two crossover methods were suggested, namely *Long Hamming Distance Crossover* (LHDC) for exploration of the solution space and *Short Hamming Distance Crossover* (SHDC) for the exploitation of the local search process [118]. To ensure the effectiveness of adaptive GA with crossover and mutation matrix, experiments were conducted on one-dimensional Ising model for spin glass and the result suggested the superiority of adaptive GA with crossover and mutation matrix comparing to MOGA.

In selective mutation each individual was assigned a rank as suggested in [118,120–122]. Thereafter, mutated one bit in a part of the individual string which was selected to correspond to their ranks [35]. Few assumptions were made such as low rank individuals are far from the global optimum whereas the high rank individuals are very close to the global optimum. The effect of this assumption was that lower rank individual needs additional mutation to explore the most significant part of the individual's string while high rank individual mutate the least significant part of the individual's string.

To test the effectiveness of GA with selective mutation experiments were conducted on four function optimization problems,

namely, a *Simple Function*, a *Mexican Hat Function*, *Shafer Function 2* and *De Jong Function 2* [36–38]. The obtained result shows that GA with selective mutation performs considerably better than standard GA. In addition, one can easily apply this approach in other optimization problems.

4.21.1. Strengths and weaknesses

Experimental results prove that selective mutation is a powerful technique to alleviate premature convergence. The additional mutation in selective mutation approach (based on rank) enhanced the performance of the GA. In addition, a rule (*if the rank of individuals is high, then a bit in the least significant part is changed, and* vice versa) was provided to apply mutation.

It was observed that this technique showed fast convergence when the population size was large and vice versa. Also, additional processing time require in comparing the rank of an individual and applying mutation twice.

4.22. Elite mating pool (EMP) approach

Elite Mating Pool (*EMP*) was suggested to prevent premature convergence in *Context Free Grammar* (*CFG*) induction problem [49]. In this approach, simple genetic algorithm (SGA) was executed multiple times on different population to choose the best solution. The EMP approach works in the following manner [49]:

S1	Create a mating pool of '*n*' individuals (*n*, very small than population size *N*)
S2	Calculate the fitness of all individuals in the population.
S3	Sort the mating pool on the fitness of the individuals.
S4	Perform crossover of the first parent with the elite member of the mating pool and create two offspring.
S5	Replace old weaker individuals in the mating pool with newly created offspring in sorted order.
S6	Perform mutation on the newly created offspring.
S7	Replace old weaker individuals in the mating pool with newly created offspring in sorted order.
S8	Once repeat steps S4 through S7.

In this approach diversity was maintained by generating elite members capable to satisfy multiple objectives. Here, an entire generation of the chromosome was generated by the application of the elite mating pool until either the population with a new elite chromosome was evolved or a minimum number of generations were completed with converged population.

The collections of *Context Free Language* (*CFL*) as well as *Regular Languages* (*RL*) were used for conducting the experiment [49]. To test the effectiveness of *EMP* approach, GA generations were processed with various probabilities ranging from 0% to 50%. It was observed that *EMP approach can* be used to break the local optimum convergence situation of the GA.

4.22.1. Strengths and weaknesses

EMP approach was found effective in maintaining the population diversity. Hence, reduces the risk of premature convergence, if the population converges before reaching to the threshold value. It was observed that this approach is capable of breaking the local optimum convergence.

The problem with this approach was its execution time because extra processing time requires in maintaining the separate mating pool to get the elite members. Choubey and Kharat [49] accepted that the execution time is assumed to be greater than a normal SGA approach.

In addition, it was shown experimentally that the execution time of *EMP* approach was higher as compared to *Dynamic Application Reproduction Operator* (*DARO*) [49].
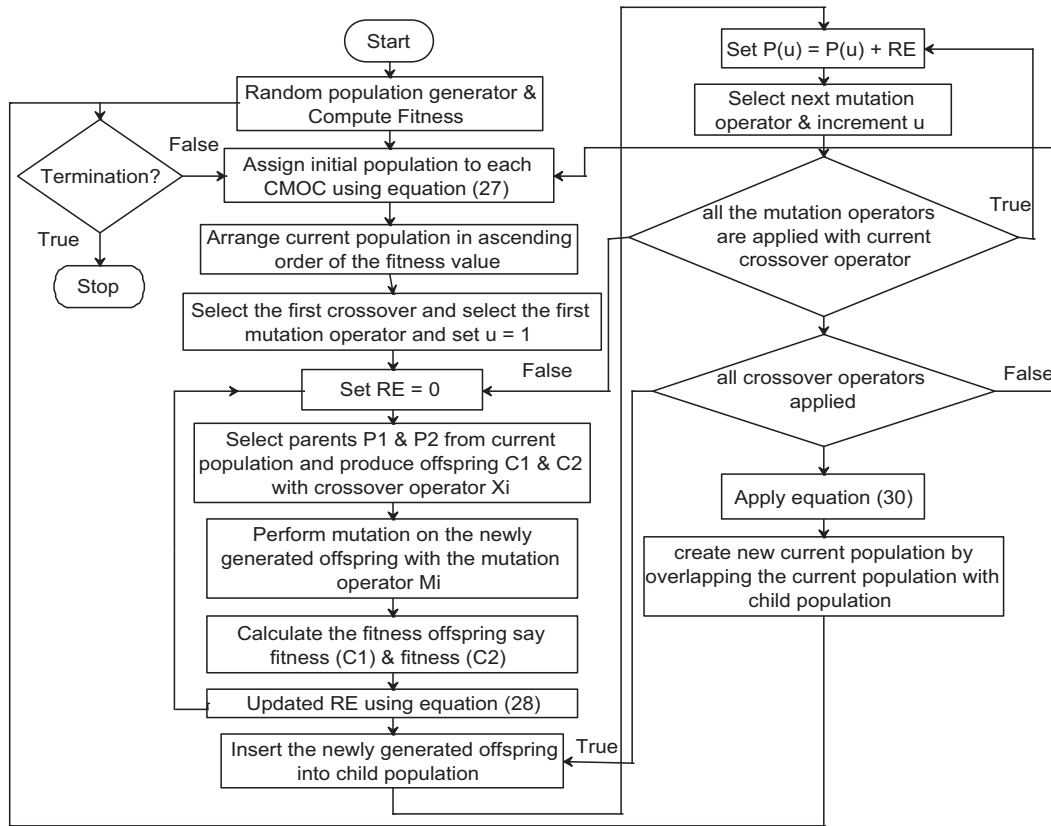
Fig. 15. Flowchart of dynamic application of reproduction operator.

4.23. Dynamic application of reproduction operator

Choubey and Kharat [49] suggested *Dynamic Application of Reproduction Operator* (*DARO*) for handling premature convergence in *the CFG induction problem*. DARO was an extension of the work done by Nicoară [31]. Equal probability was assigned to each reproduction operator combination (crossover-mutation operator combination, CMOC) to begin the genetic algorithm generations. The flowchart for DARO is shown in Fig. 15, which shows the step by step flow of this approach. Three different crossover strategies, namely *two Point Crossover* (*TPC*), *Two-Point Crossover with Internal Swapping* (*TPCIS*) and *Uniform Crossover* (*UC*) and four mutation techniques such as: *Stochastic Mutation, Inverse Mutation, Block Copier with Fixed Length Mutation* and *Block Copier with Random Length Mutation* were implemented in DARO.

The working of this approach was given as: create the initial random population and compute their fitness value. Then assign the initial probability for each CMOC using the heuristic given in Eq. (27).

$$P(u) = \frac{1}{m*n} \text{ and } \sum_{u=1}^{m*n} P(u) = 1 \tag{27}$$

where m and n represents (m crossover operators $\times$ n mutation operators)

Now, select first CMOC and set $u = 1$. A term Resultant Effectivity (RE) was used to ensure the effectiveness of the new offspring. Initially, $RE \leftarrow 0$. Choose two parents say $P1$ and $P2$ and apply the first crossover operation (X_i). This will produce two offspring say $C1$ and $C2$. Perform first mutation (M_i) on $C1$ and

$C2$, compute *fitness*($C1$) and *fitness*($C2$) and then update RE using Eq. (28).

$$RE \leftarrow RE + \frac{\left([fitness(P1) + fitness(P2)]/2\right)}{\left([fitness(C1) + fitness(C2)]/2\right)} \tag{28}$$

Insert the new created offspring in child population. The above procedure will be repeated until $P(u)$ proportion of the population is generated using current CMOC. Then update the value of $P(u)$ using an Eq. (29).

$$P(u) = P(u) + RE \tag{29}$$

Now, select the remaining mutation operators and increment in $P(u)$. If all mutation operators are applied successfully with the current crossover operator (X_i) then, choose the remaining crossover operator one by one and apply each mutation on the resultant offspring (after each crossover operation) and compute RE using Eq. (28) for each offspring created via CMOC. Update $P(u)$ using Eq. (30).

$$P(u)_{next} = \frac{P(u)_{current}}{\sum\limits_{u=1}^{m*n} P(u)_{current}} \text{ and } \sum_{u=1}^{m*n} P(u)_{next} = 1 \tag{30}$$

Finally, create the new current population by overlapping the current population with child population using Eq. (31)

$$POP_{new} = POP_{current} + POP_{child} \tag{31}$$

POP_{new} preserve the diversity and therefore prevent the premature convergence. Same set of languages (languages used for EMP [49]) was considered for the experiment purpose. It was found that this method works effectively for simple languages, but there is scope for applying this method for more complex grammar sets.
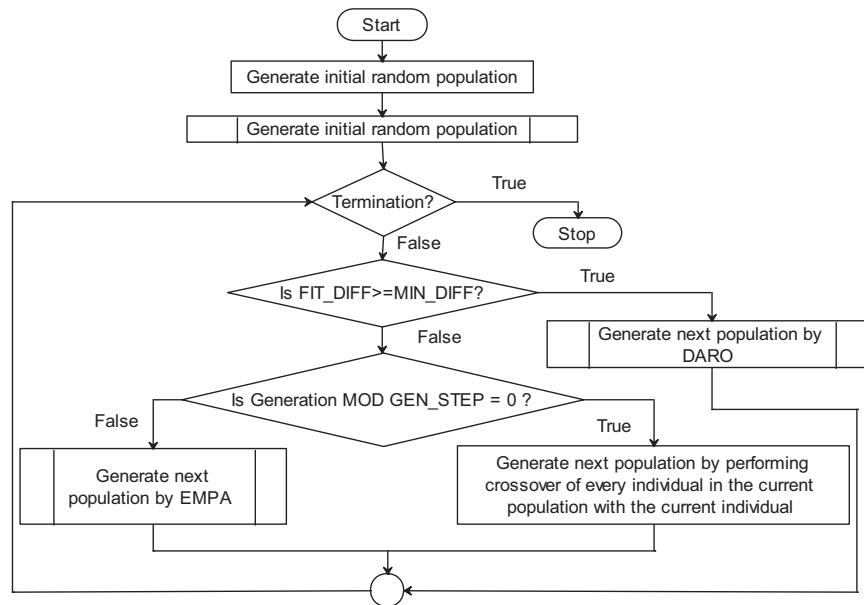
Fig. 16. Hybrid model (EMPA + DARO) for reducing premature convergence.

It was also observed that DARO consume comparatively less time than EMP.

4.23.1. Strengths and weaknesses

DARO was found effective in preventing premature by empowering the GA with dynamic application of crossover and mutation operators. It was observed that this approach consumes comparatively less processing time than EMPA (Section 4.22) and therefore it can be used to overcome the issues found in EMPA.

As discussed, DARO is a time effective algorithm as compare to EMPA but it is not fit for handling the situation of local optimum convergence whereas EMPA handles such situation effectively. Also, there were no fixed rules provided to decide the initial probability to reproduction operator combination while it was already suggested by Srinivas and Patnaik [19] and Jung [35].

4.24. Hybrid strategy using EMPA and DARO

The merits of EMPA and DARO were collected to develop the Hybrid model to avoid premature convergence and maintain the diversity of population [44]. In the hybrid model, the DARO approach leads toward generation of the optimal solution. To prevent premature convergence the appropriate mix of EMPA and random breed was used to create next population [44]. The flowchart for hybrid model is shown in Fig. 16 [44].

Fig. 16 demonstrates the working of hybrid model and show the effective utilization of EMPA and DARO approaches. The hybrid model uses two important elements, namely FIT_DIFF and GEN_STEP. Both the terms are equally important and helpful in making decisions (implement DARO or not). FIT_DIFF represents the fitness difference between the best and worst fitness while the GEN_STEP shows the number of generations to be processed. The second decision box shown in Fig. 16 clearly shows that until the value of FIT_DIFF is equal or more than MIN_DIFF the process of DARO will continue otherwise it predicts the situation of premature convergence. Therefore, to prevent premature convergence, hybrid model makes decision again depending on the value of GEN_STEP. That is, it helps in making the decision to apply EMPA or apply crossover operations to generate population in the next generation (Fig. 16 the last decision box indicates the same) [44].

Orthogonal Array Method [50] was used for parameter selection. It was developed by Dr. Genichi Taguchi for designing experiments to investigate the effect of parameters on the mean and variance of process performance characteristic [44]. Experiments were conducted on the different language set used by Huijsen [51], Keller and Lutz [52], Tomita [53], and Dupont [54]. In addition, other languages were also implemented, namely an odd binary number over $(0+1)$ *, even binary number over $(0+1)$ * and $\{0^n 1^{2n}\}$ over $(0+1)$ *. The results obtained using GA with a hybrid strategy for first ten successful runs on each language set and it was observed that hybrid approach prevent premature convergence effectively. In addition, it was shown that the simple grammar population was converged to the best value in less time in comparison to the complex CFG. It was also seen that both (EMPA and DARO) approaches are significantly more efficient than other approach in order to maintain the performance while the other methods are found to handle the premature convergence more effectively.

4.24.1. Strengths and weaknesses

The convergence analysis of hybrid approach with other approaches such as Simple Genetic Algorithm (SGA), Elite Mating Pool Approach (EMPA) [49], Dynamic Allocation of Reproduction Operator (DARO) [49], Incest Prevention Algorithm (IPA) [32], Pygmy Algorithm (PA) [43], Population Replacement Algorithm (PRA) [38], Random offspring Generation Algorithm (ROGA) [22] and Social Disaster Algorithm (SDA) [42] were shown and it was found that the hybrid model works more effective than other methods. Orthogonal array method was used for the selection of parameters which reduces the issues of selection of parameters as discussed with DARO. In addition, it was also observed that DARO and EMPA showed better convergence as compared to the hybrid method at the cost of convergence at local optima but the hybrid method produces the optimal result of delaying the convergence.

Although using hybrid approach the trade-off between EMPA and DARO for execution time was achieved up to a certain level, but not completely since the weakness of this approach remains the same when applying EMPA because the hybrid approach uses the features of EMPA.

P1	X10	X1	X9	X3	X5	X4	X7	X2	X6	X8
P2	X10	X2	X4	X3	X5	X9	X1	X7	X6	X8

Same gene　　　　　　Same gene　　　　　　Same gene

OS	X10	X7	X1	X3	X5	X2	X9	X4	X6	X8

F=2　　　F = 1　　　F = 2　　　　F = 1　　　F =2

P1 = Parent1, P2 = Parent2, OS = Offspring, F = Frequency Count, X1
X2….. X10 represents the genes.

Fig. 17. Frequency crossover strategy.

4.25. Frequency crossover with nine different mutations

Ramadan [48] implemented a GA to get the solution of the traveling salesman problem (TSP) in which the concept of *Frequency Crossover* (*FC*) along with nine different mutation techniques were used to prevent premature convergence. The idea of FC is: at a time, *Y* different chromosomes were selected from the existing population. All the selected chromosomes were stored in an ascending order so that *Y*/2 chromosomes can form a crossover group for crossover with a 100% crossover rate [48]. The working of FC can be defined as:

Algorithm: Evaluate Frequency

- *Best Chromosome*: Choose the best chromosome in crossover group.
- Produce offspring: apply crossover with every other chromosome.
- Evaluate frequency: The frequency values of the corresponding genes from both parents can be evaluated as:
    If genes carry the same city
- *Frequency ← two*
    Else
- *Frequency ← one*

In order to produce the new offspring first compute the frequency, i.e. look for the genes have frequency of two and copy the value into the offspring while preserving their relative position. Then remaining genes were assigned randomly in the offspring.

Fig. 17 demonstrates the working of frequency crossover strategy. Since, the genes such as X10, X3, X5, X6 and X8 are at the same location in both the parents P1 and P2, therefore the frequency count (*F*) for the gene is 2.

The remaining genes X1, X2, X4, X7 and X9 copied randomly in the offspring because the frequency count is 1. This process preserves the information carried and accumulated over generations, but there is no guarantee that it will maintain the population diversity and hence increases the likelihood of premature convergence. Therefore, frequency crossover was considered as a stability inducer in the GA. The selected parents and the generated offspring were used to replace the existing Y-1 chromosome. The remaining

(since we have Y/2 parents and Y/2-1 offspring) were immigrated [48].

At this stage, subpopulations were formed using the new chromosome Y, which was used in the mutation step. Thereafter, best chromosomes were selected from the subpopulation to apply nine different types of mutation. And then the original subpopulations were replaced by the best chromosome along with the nine offspring generated by mutation which increase the population diversity and guaranteed that the best chromosome found in that particular generation will survive for the next generation. Nine different mutation techniques were applied, which was categorized into two groups, namely *Group1* and *Group2*. Both the group together maintains a high degree of population diversity. *Group1* mutation was used for exploration because it offers a huge change in the chromosome. As *Group2* mutation introduces a minor change, therefore it was used for exploitation. Three different sets of data for TSP were taken from [23] for conducting the experiment. A test of the hypothesis was done to test the effectiveness of the GA [48] and the results showed that the GA with FC and nine different mutations performed significantly better the target value. Author of [48] not only showed the effectiveness of the GA but also the effectiveness of nine different mutations were shown. In addition, it was shown clearly that how many times FC was able to enhance the existing fitness function from one generation to the next one.

4.25.1. Strengths and weaknesses

Overall, it was shown that the frequency crossover with nine different mutations able enough to introduce the diversity in the population and reduce the effect of premature convergence. In addition, it was shown that this approach not only increases the diversity but also guarantee that the best chromosome found in any particular generation will survive for the next generation.

But the demerit of this approach is that it is applicable only for *Order-Based* Chromosome representation. Also, extra processing time requires in computing the frequency and then applying nine mutation strategies.

5. Summary and analyses of reviewed literatures

In this section we have furnished the summary of reviewed literatures and presented in tabulated form for quick review. Table 3 shows the details about the techniques applied in each approach. Tables 4 and 5 show the comparative analysis based on various parameters. Using these tables one can easily and quickly get the answer of questions such as: how diversity management was applied, how the recombination techniques was applied, how to measure the similarity of an individual, what are the key terms used with any particular approach, which selection technique was applied and above all the key concerns related with the approaches. The objective of showing the comparative analysis is to develop in depth understanding of each approach which might be helpful in designing new approaches to address the issue of premature convergence.

Table 3
Summary and analysis of reviewed literatures based on the technique used.

Citation	Approach	Author (s), Year of publication	Technique used
[38]	Crowding Method	De Jong, Kenneth Alan, 1975	Eliminate the most similar individual whenever a new one enters in a subpopulation.
[32]	Incest Prevention Algorithm	Eshelman, Larry J., and David Schaffer, 1991	Used Hamming distance as a measure for selecting the mating pair.
[15]	Scheduled Sharing	Palmer, Michael E., and Stephen J. Smith, 1992	Organism within a niche to compete with their neighbors. To calculate the convergence of a population, the search space can be divided into subspace or buckets.
[24]	Migration Model and nCUBE Based Approach	Balakrishnan, Karthik, 1993	A migration model was introduced to maintain the diversity in the genetic property of individual population to keep the GA search alive.

Table 3 (*Continued*)

Citation	Approach	Author (s), Year of publication	Technique used
[18]	Cooperation Based Approach	Fonlupt, Cyrilm, Philippe Preux, and Denis Robilliard, 1993	Instead of using only one representation, apply varieties of GAs concurrently and each optimizing the same object but using its own representation.
[29]	Syntactic Analysis of Convergence	Louis, Sushil J., and Gregory JE Rawlins, 1993	The average Hamming distance of a population was used as a syntactic metrics to get the probabilistic bound on the time convergence of GAs. A model was presented to predict the time to convergence.
[43]	Pygmy Algorithm	Ryan, Conor, 1994	Uses two fitness functions and maintains two lists of parents, one for each criterion of the problem.
[19]	Adaptive Probability Based Approach	Srinivas, M. and Lalit M. Patnaik, 1994	The probability based approach was adopted which varies the probabilities of crossover and mutation depending on the fitness value of the solution.
[42]	Social Disaster Technique	Kureichick, Victor M., Victor V. Miagkikh, and Alexander P. Topchy, 1996	Works on the whole population to maintain the population diversity. The concept of super individual was suggested which takes over the charge of the population.
[69]	The Island Model Genetic Algorithm	Whitley, Darrell, Soraya Rana, and Robert B. Heckendorn, 1998	Each machine executes a GA and maintains its subpopulation for directing the search which helps to preserve genetic diversity, since each island can potentially follow a different search trajectory through the search space.
[21]	Shifting Balance Theory in Dynamic Environment	Oppacher, Franz, and Mark Wineberg, 1999	Modified version of Shifting Balance Theory. Uses Core group and Colonies to prevent premature convergence.
[22]	Random Offspring Generation Approach	Rocha, Miguel, and José Neves, 1999	Test similarity of genetic material, if found similar than generate the random offspring which will code a random solution for the problem otherwise apply reproduction
[26]	Chaos Operator Based Approach	Juan, Liu, Cai Zixing, and Liu Jianqin, 2000	Novel GA presented which contains chaos operator within the framework of Markov Chain to prevent premature convergence.
[25]	Self-Adaptive Selection Pressure Steering Approach	Affenzeller, Michael, and Stefan Wagner, 2003	Introduced an intermediate step for a new selection scheme (virtual population) to handle the selection pressure. In addition, a separation of population concept was used to increase the search space. In this approach the concept of simulated annealing was also used.
[59]	Multi-Combinative Strategy	Achiche, Sofiane, Marek Balazinski, and Luc Baron, 2004	Works on real/binary coded GA with varieties of crossover techniques.
[6]	GA using Self-Organizing Map	Amor, Heni Ben, and Achim Rettinger, 2005	*Self-organizing Map* was used to address the premature convergence. This approach works by maintaining previous generation information. Two frequency tables were used, namely *Population Distribution Table* and *Search History Table*.
[41]	Age-Layered Population Structure Approach	Hornby, Gregory S, 2006	Based on age measure defined as the total number of generations where an individual genetic material evolving in the population.
[14]	Number Structuring Approach	Sultan, B. M., Ramlan Mahmud, and Muhammad Nasir Sulaiman, 2007	Based on odd and even number representation technique. Introduced a function *readjustment p (t)* to maintain the sequence of odd and even in each generation of GA.
[39]	Hybrid Particle Swarm Optimization and GA	Premalatha, K., and A. M. Natarajan, 2009	Uses the merits of *Particle Swarm Optimization* and *Genetic Algorithms*. Three hybrid approaches were used PSO-GA (type-1), PSO-GA (type-2), and PSO-GA (type-3).
[35]	Selective Mutation Based Approach	Jung, Sung Hoon, 2009	In this approach each individual was assigned a rank. For low rank individuals, apply additional mutation to explore the most significant part of the individual's string while for high rank individual, mutate the least significant part of the individual's string.
[49]	Elite Mating Pool Approach	Choubey, Nitin, and Madan Kharat, 2011	Simple genetic algorithm executed multiple times on different population to choose the best solution.
[49]	Dynamic Application of Reproduction Operator Approach	Choubey, Nitin, and Madan Kharat, 2011	Extension of the work done in [31]. Equal probability was assigned to each reproduction operator (crossover-mutation operator combination, CMOC). Three crossovers and four mutation techniques were used.
[44]	Hybrid Strategy using EMPA and DARO	Choubey, Nitin S., and Madan U. Kharat, 2013	Uses the advantages of EMPA and DARO [49] to maintain the diversity of the population. Orthogonal Array Method [50] was used for parameter selection.
[48]	Frequency Crossover with Nine Different Mutations	Ramadan, Saleem Zeyad, 2013	Based on the concept of Frequency Crossover (FC) along with nine different mutation techniques.

Table 4
Comparison based on key factors, diversity management, similarity measurement and crossover and mutation techniques implemented.

Approach	Key Factors	Diversity Management	Similarity Measurement	Crossover and mutation technique implemented
Crowding Method	Generation Gap (G) and Crowding Factor (CF) had been chosen 2 to 3 by De Jong.	Replacement of similar individual maintains population diversity.	Using genotypic distance.	Form n/2 pairs and then crosses all pairs and mutates the offspring.
Incest Prevention Algorithm	Individual Hamming distance	Avoided mating of pairs showing similarities based on the parent's Hamming distance.	Short Hamming distance indicates that individuals might be related	Uses heterogeneous recombination. Mating is only allowed if the Hamming distance is above the chosen threshold, which decreases as evolution proceeds.
Scheduled Sharing Approach	Sharing schedule, Nice size, Sample sharing.	Punished the members which are similar to each other by being required to share their fitness, whereas the isolated individual retains all the fitness value they achieve.	Using simple two dimensional Cartesian distance function or using Hamming distance between genotypes.	By scaling the fitness in proportion to their similarity which is defined by a sharing function.
Migration Model and nCUBE Based Approach	Sending protocol, migration interval, Number of independent populations	Migration of population from one to another	Spatial segregation approach	Using uniform crossover, mutation and migration of population
Cooperation Based Approach	Appropriate representation techniques and remapping.	By applying remapping, which may increase or decrease of the number of local optima by dynamically changing the neighborhood of individuals.	Using genotypic distance	Crossover and mutation are applied in base 4. After computing fitness chromosome genotype was turned into base 3. Null fitness was assigned to non feasible chromosomes in base 3.
Syntactic Analysis of Convergence	Average Hamming distance, time bound for convergence	Operators not considered which results in less Hamming distance to offspring compared to their parents Hamming distance. The minimum fitness individual was replaced with either the complement of randomly picked individual in the current population or the complement of the current best individual.	Average Hamming distance also known as Hamming average.	N-point crossover and usual mutation operator (pick an individual and add its bits complement to the population). Recombination depended on the individual's fitness.
Pygmy Algorithm	Two lists namely Civil Servants, Pygmy	Maintain two lists namely Civil Servants and Pygmies. For a new offspring one parent was drawn from each list.	Phenotypic distance of parents.	Disassortive matting, breading of phenotypically different parents sometime used by plant breeders to maintain diversity.
Adaptive Probability Based Approach	Crossover Probability (P_C) and Mutation Probability (P_m), new parameters k_1 and k_2 for controlling crossover and mutation	Generational replacement and explicitly protects the best solution from disruption	Spatial distance (based on the fitness value)	P_c and P_m are adapted based on the fitness values of an individual for recombination.
Social Disaster Technique	Fifty-fifty approach, Packing operator, Judgment_Day and wrap operators	By applying crossover operator modification and operators (Packing, Judgment_Day and warp operator) working together on the whole population. In short, determine the suboptimum and then apply either Packing or Judgment_Day operator together with warp operator.	Distance matrix and evaluation function [92]	Modified version of standard greedy crossover called as fifty-fifty approach with a variable probability of bad solution which require proper tuning and usual mutation operators.
The Island Model Genetic Algorithm	Migration interval, migration size, topology used for migration and migration policies adopted.	Migration of highly fitted individuals across subpopulation increase the diversity of the local subpopulations	Some sort of distance in Hamming space	Allows recombination within the island and sometimes across the island
Shifting Balance Theory in Dynamic Environment	Core group, Colonies, diversity measure, distance measure, Containment function, Migration from colony to core and migration interval	A parameter $Diversity(P)$ was suggested to see the diversity of the population. First populate the core group from the member of old core through reproduction. After that form colony and find out up to what extent member's of colony lies in core group. Divide the colony in two groups called as inside and outside. Compute $Containment(A, B)$ and apply immigration of population between core and colony based on fitness. In a nutshell, colony members into the core group given the responsibility to maintain the diversity of the core.	Hamming distance was used. Three measure was identified namely, diversity measurement, distance (distance between a single chromosome and a population in gene-space) measure and containment function.	Recombination was done between colony and core groups. The colony may apply full or partial sharing of its member (selected randomly according to the fitness) with core by

Table 4 (*Continued*)

Approach	Key Factors	Diversity Management	Similarity Measurement	Crossover and mutation technique implemented
Random Offspring Generation Approach	Blind operators, hybrid operators, Random offspring	Concept of Random offspring generation was applied before the recombination. Test the similarity of individual, if individuals are not similar, simply apply crossover and mutation otherwise generate offspring to code a random solution.	Genotype or phenotype distance	Recombination was done after checking the similarity of individual using various crossover operators such as blind operators (UOPX, EDGX, MPX and SCHX) and hybrid operators (GX and HGX). Four different categories of mutation operators (adjacent swap, non-adjacent swap, sub list scramble and partial inversion)
Chaos Operator Based Approach	Chaotic operator, logistic equation,	Chaotic operator ensures that there do not exist two identical populations, even if the two initial states are very close. Such population maintains the diversity and generates optimal solution.	Genotype or phenotype distance	Recombination was done after applying chaotic features. If population reached to certain degree use logistic equation and then apply crossover and mutation otherwise directly apply crossover and mutation.
Self-Adaptive Selection Pressure Steering Approach (SASEGASA)	Simulated annealing, virtual population, success rules, maximum selection pressure and separation of population.	In this method, GA with simulated annealing was applied. During running GA if one cannot find ($SuccRatio * |POP|$) then children outperform their own parent and ($MaxSel Press * |POP|$) both shows premature convergence where $MaxSel Press$ represent the upper limit of selection pressure. This information might be useful in preserving diversity and predicting the premature convergence.	Genotype or phenotype distance	Simulated Annealing (growing selective pressure) as well as a special segregation and reunification strategy with crossover and mutation in the general model of a Genetic Algorithm. Two new features was introduced "virtual population" and "separation of population" when applied recombination operations.
Multi-Combinative Strategy	Multi crossover, fuzzy rule crossover, fuzzy set reducer, Multi parent recombination, Multiple crossover on multi parent (MCMP) and multiple crossover per couple (MCPC)	Multi-parent recombination, MCMP and MCPC were applied with real/binary coded GA to maintain the diversity and keeping the search process alive.	Genotype distance of offspring	Three main crossover techniques (multi-crossover, premises/conclusion and blended crossover α), uniform mutation and selection were applied.
GA using Self-Organizing Maps	Self organizing map, Best matching unit (BMU), Population distribution table (PDT), Search history table (SHT), Novelty of chromosome, Reseeding operator, Reseeding pool, and Activated unit.	SOM was used to visualize various aspects of search. It maintains the search history using two tables such as PDT and SHT. PDT was used to examine the diversity in the current population, while SHT gives an insight into the course of evolution. In addition, a reseeding operator and a control mechanism were applied to balance exploration and exploitation.	A genotype distance of offspring follows a uniform distribution. Distance to all prototype vectors computed using Euclidian distance.	Two point crossover and standard mutation with mutation rate $P_m = 0.5$ and crossover rate $P_c = 0.7$. In addition, novelty factor "novelty (c)" was used to order the population for selection.
Age-Layered Population Structure (ALPS) Approach	Individual's age, Layered structure of populations and Age gap.	Age of individual's was defined and populations were structured into multiple layers with maximum allowable age for an individual and selection, breeding and replacement is restricted to adjacent layers.	Measured by an individual's age.	Standard crossover and mutation was applied. Mutation or recombination starts with an age of 1 plus the age of its old parents.
Number Structuring Approach	Odd and even representation and procedure "readjustment ()"	Using odd-even number side by side along the chromosome. A procedure "*Procedure Readjustment ()*" was used. This procedure consists of simple repair instruction to readjustments back the value inside the gene. It also ensures the sequence of odd-even remains at the respective gene after iterations.	Genotype distance of offspring.	Standard crossover and mutation techniques were used. In addition, before evaluating the quality of individual population "readjustment ()" procedure was applied, which ensures the odd and even sequence along with chromosome and preserve the diversity.

Table 4 (*Continued*)

Approach	Key Factors	Diversity Management	Similarity Measurement	Crossover and mutation technique implemented
Hybrid Particle Swarm Optimization (PSO) and GA	PSO-GA (Type-1), PSO-GA (Type-2) and PSO-GA (Type-3).	Position update of the global best particles was changed. The position update was done through some hybrid mechanism of GA. Three different hybrid approaches were suggested: PSO-GA (Type-1), PSO-GA (Type-2) and PSO-GA (Type-3).	Genotype or phenotype distance	The PSO-GA (Type-1) used to apply crossover operation p on best particle with the chromosome of GA where both PSO and GA runs in parallel. PSO-GA (Type-2) applied to stagnated pbest particle to change the position using mutation operator of GA. In case of PSO-GA (Type-3) total iterations were equally shared by GA and PSO where the first of the iterations was run by GA and the solution was provided as initial population of PSO and remaining iterations was performed by PSO.
Selective Mutation Based Approach	Selective mutation and individual rank.	The selective mutation technique was applied. Each individual was first ranked and then additionally mutated one bit in a part of their strings which was selected to correspond to their rank. The selective mutation helps in diversity management and help GA to reach to global optimum and to quickly escape from local optima.	Genotype or phenotype distance	Standard crossover and mutation techniques were applied. And then selective mutation was applied by sorting the offspring based on fitness and rank. Divide the strings and rank into N parts, then apply selective mutation on the randomly selected bit.
Elite Mating Pool Approach	Elite member, mating pool	Mating pool was created which maintains the diversity by generating the elite members more capable of satisfying the multiple objectives. Crossover and mutation operator was applied to elite members of mating pool and replaces the weaker individual.	Genotype or phenotype distance	The standard crossover operator was applied to elite members of mating pool and replaces weak individual with newly created offspring. Now, mutation operator was applied on newly created offspring and again replaces weak individual with newly generated offspring. Finally, apply selection mechanism.
Dynamic Application of Reproduction Operator Approach	Crossover-Mutation Operator Combination (CMOC), Resultant Effectivity (RE)	Dynamic application of crossovers and mutation operators was applied. Probability value was assigned for each reproduction operator combinations. Resultant Effectivity (RE) was used which was set to zero initially and updated based on the fitness of a pair of parents and created offspring. RE was also used to update the probability of reproduction operator combination. The value of RE helps maintain the diversity of the population and selection of reproduction operator combination.	Genotype or phenotype distance	Three crossovers (Two point crossover, Two point crossovers with internal swapping and uniform crossover) and four mutations (inverse mutation, block copier with fixed length, block copier with random length and stochastic operator) operators were used. Equal probability was assigned to each reproduction operator combination which gets updated in each generation. It is inspired by the approach suggest by Nicoară, Elena Simona (2009) [31].
Hybrid Strategy using EMPA and DARO	Fitness difference (FITT_DIFF), Number of generations (GEN_STEP)	To know the premature convergence, FITT_DIFF was compared with MIN_DIFF. If the FITT_DIFF is greater than or equal to FITT_DIFF then there is no premature convergence and move to next generation otherwise it shows the situation of premature convergence. As this approach uses the merits of EMP and DARO therefore to decide which one will be applied to handle the situation of premature convergence the condition $G\%GEN\_STEP = 0$ was applied if it shows the true result then apply DARO otherwise EMP was applied to maintain the diversity.	Genotype or phenotype distance	Applied the combined mechanism of EMP [49] and DARO [49] approaches
Frequency Crossover with Nine Different Mutations	Order Based Representation (OBR), Frequency Crossover (FC), Ends Exchange mutation (EsEm), Group Insertion mutation (GIm), Reverse Ends mutation (REsm), Two Genes Exchange mutation (TGsEm), Reverse Ends Exchange mutation (REsEm), Reverse End mutation (REm), One Position Swap mutation (OPSm), Middle Reverse mutation (MRm)	FC was applied to maintain diversity. Choose the best chromosome in crossover group and evaluate the frequency. Then nine different mutation strategies were applied. Finally, the original subpopulation was replaced by best chromosome along with the nine offspring which increases the diversity and guaranteed that the best chromosome found in that particular generation will survive for the next generation.	Genotype or phenotype	Order based representation was used to represent the chromosome. The FC crossover strategy was implemented with nine different mutations which were divided into two groups. Group-1 contains five mutation type and remaining four mutation type fall in Group-2. The primary purpose of Group-1 was for exploration purpose, whereas Group-2 was used for exploitation purpose.

Table 5
Analysis of reviewed literatures based on competition technique, key concern, selection techniques and the problem considered for the experiments.

Approach	Competition	Selection Technique	Applied On	Key Concern
Crowding Method	Offspring compete against one of the parents that produce it.	Restricted tournament selection (RTS) adapts standard tournament selection (TS)	Multimodal function optimization	• Stochastic error by low crowding factor (CF), • Maintaining bitwise diversity • Replacement error is the major issue.
Incest Prevention Algorithm		Based on Hamming distance if above a threshold. If in a generation, no pair can be chosen, then decrease the threshold.	Multimodal function optimization	• Additional computation cost to compute individual's similarity which varies the execution time • Difficult to maintain diversity if the member of the new population has common ancestors and consequently due to a finite fixed population size which causes a loss of genetic diversity.
Scheduled Sharing Approach	Organisms presents in a niche compete with their neighbors	Phenotypic fitness by a sharing factor (a number between zero and one)	Rugged two-dimensional landscape	• Finding the appropriate schedule for sharing • Deciding the sharing functions • It is one of the expensive schemes, and • It requires prior knowledge to write sharing function.
Migration Model and nCUBE Based Approach	No (maintained independent populations)	Proportionate fitness and ranking selection	Kookie problem	• Setting migration interval is an important factor • Taking the decision to migrate an individual to migrate, and • Number of independent populations.
Cooperation Based Approach	No correlation between neighboring individual	Using genotype's fitness an individual was selected if their fitness is better.	1's counting problem	• Selection of appropriate representation techniques during the GA search • Remapping of chromosome dynamically, and • Dealing with more than representation is quite difficult.
Syntactic Analysis of Convergence	No correlation between neighboring individual	Fitness proportion selections	Flat function, DeJong's five functions (F1 through F5), and One max problem.	• Mutation rate affects the Hamming convergence. • Finding the upper bound and lower to compute time to converge is difficult and time consuming and • Qualitative prediction may not be possible with pure syntactic information.
Pygmy Algorithm	No attempt was made to measure this.	Selection of parents for breeding was analogous to differing genders. One parent is selected from each list (Pygmy list and Civil Servant list)	Multi-objective function optimization, problem of evolving minimal network sorters.	• Not fit for multi-objective function in some situation because individuals in the same group cannot mate. • The issue was addressed by applying Racial Preference factor for individuals.
Adaptive Probability Based Approach	No correlation between neighboring individual	Fitness proportion selections	Multimodal function optimization, DeJong's functions known as spiky function (f5, f6, f7), Order-3 Deceptive problem, Traveling salesman problem, neural network weight optimization problems and generation of test vectors for VLSI circuits.	• Selection of the Crossover and Mutation Probability • Protecting the best solution explicitly, and • Handling new parameters k_1 and k_2 for controlling the adaptive nature of P_m and P_c.
Social Disaster Technique	Checked using a distance matrix calculation	Roulette wheel selection	Traveling Salesman Problem	• The modified greedy crossover approach was used • Decrease in crossover greediness, increases the variation in the population, • Efficient usage of the population pool, • No measure was provided for the decrement in crossover greediness' • Tuning the probability of bad solutions results in extra computational cost.
The Island Model Genetic Algorithm	No correlation between neighboring individual	Biased rank based selection and fitness proportionate selection	Two linearly separable problems (fully deceptive order-4 function and Rastrigin's function) and two nonseprable problems (Powell's Singular function and Rana's function)	• Maintains diversity by sharing information with different island, • Migration interval and migration size play an important role. • No information is available to decide the migration interval and migration size.
Shifting Balance Theory in Dynamic Environment	Competition among the individual based on the fitness both in the core and the colony.	Based on distance to the core (for inside section), and Fitness proportionate Selection (outside section)	An experiment was done on F8F2 function. F2 function of the De Jong test suite composed with a 1-D version of the Griewangk Function (F8)	• Computation cost increases in dividing the colony and computing the fitness of individual population • This approach also suffers with the problem of migration interval and • It was accepted that no dynamic model available for the dynamic environment.
Random Offspring Generation Approach	No correlation between neighboring individual	Roulette-Wheel scheme	Traveling Salesman Problem (instance taken from [23])	• Computational overhead disregarded compared to Crowding [38], Sharing [16] and Scheduled Sharing [15] approaches.

Table 5 (*Continued*)

Approach	Competition	Selection Technique	Applied On	Key Concern
Chaos Operator Based Approach	No correlation between neighboring individual.	Fitness proportionate selection	Mono-modal function optimization and Multimodal function optimization (performed better compared to mono-modal)	• It helps in faster convergence • Not capable to resolve the premature convergence completely and • The tremendous amount of computation is required.
Self-Adaptive Selection Pressure Steering Approach (SASEGASA)	No correlation between neighboring individual.	Standard selection mechanism such as a roulette wheel, linear rank or some kind of tournament selection with virtual population based on the success ratio (see the algorithm Section 4.14)	Traveling salesman problem and multimodal combinatorial problems.	• Separation of population increases the size of the population; • The computational time for this algorithm is huge, and • Parallel computer architecture will be suitable to apply this approach to enhance the performance
Multi-Combinative Strategy	Competition among the individual population based on their fitness when applying selection mechanism.	Randomly generate a real value R in the interval [0, 1]. Compute S, where S sum of fitness values of the individual population. Evaluate RS. Now, starting with the best individual of the population, the fitness values are summed till the result is higher than RS. The last added individual is assumed as a potential parent.	Applied on 3D surfaces of type $z = f(x, y)$. Three different surfaces of different complexities were used, namely Sinusoid Surface, Spherical Surface and Hyper Tangent Surface.	• Increase in the complexity when creating new offspring using different techniques. • Although, creating new offspring from the same pair of parents' gene was found an effective way to improve the diversity, but at the latter stage of the evolution, the presence of too many look-alike individuals is still a problem.
GA using Self-Organizing Maps	Competition among the individual based on the novelty.	Binary tournament selection	Royal Road function, the Deceptive F9 function from and the generalized Rosenbrock's function.	• Maintaining the size of *the SOM* and • The effect of variable length neurons on the performance of *GASOM*
Age-Layered Population Structure (ALPS) Approach	Based on the age of individual's	Standard selection techniques	Antenna design optimization problem	• EA with deterministic crowding showed the best results for several while ALPS showed better results.
Number Structuring Approach	Competition among individual depends on the fitness value	A standard selection mechanism based on the fitness value.	Timetabling problem	• Suitable only for small sized population. • Not suitable for very complex problems.
Hybrid Particle Swarm Optimization (PSO) and GA	Competition among particle was based on fitness.	Standard selection mechanism	Shaffer's function, Rosenbrock's function, Rastrigin's function, Ackley's function	• Sometimes diversify the particle position without reaching to global optimum or produces a worse solution.
Selective Mutation Based Approach	Based on the fitness and rank assigned to the individual.	Roulette wheel selection.	Simple Function, a Mexican Hat Function, Shafer Function 2 and De Jong Function 2.	• Showed fast convergence when the population size was large whereas slow convergence for small population size. • Extra processing time require in applying mutation twice.
Elite Mating Pool Approach	Competition among elite members of the mating pool based on their fitness	Standard selection mechanism such as Roulette-Wheel scheme, tournament selection or any other.	Collections of Context Free Language (CFL) as well as Regular Languages (RL)	• Consume extra processing time in maintaining the separate pool to get the elite members during offspring creation. • Slower than SGA.
Dynamic Application of Reproduction Operator Approach	Competition among individual depends on the fitness value.	Standard selection mechanism such as Roulette-Wheel scheme, tournament selection or any other.	Collections of Context Free Language (CFL) as well as Regular Languages (RL)	• No fixed rule given to assign the initial probability to reproduction operator combination as suggested by Srinivas and Patnaik (1994) [19] and Jung, Sung Hoon (2009) [35]. • Not suitable for breaking the local optimum convergence.
Hybrid Strategy using EMPA and DARO	Competition among individual depends on the fitness value.	Standard selection mechanism such as Roulette-Wheel scheme, tournament selection or any other.	Applied to different languages set used by Huijsen [51], Keller and Lutz [52], Tomita [53], and Dupont [54]. In addition, odd binary numbers over $(0 + 1)^*$, even binary numbers over $(0 + 1)^*$ and $0^n 1^{2n}$ over $(0 + 1)^*$.	• Again problem remains the same when applying EMP approach since this approach uses the features of EMP. • Orthogonal array method was used for the selection of parameters which reduces the issues of selection of parameters as discussed with DARO.
Frequency Crossover with Nine Different Mutations	Competition among individual largely depends on fitness and frequency.	Standard selection mechanism such as Roulette-Wheel scheme, tournament selection or any other.	Traveling Salesman Problem	• Extra processing time requires in computing the frequency and then applying nine mutations approaches. • For the experimentation order based representation was used, the effect of other representation had not been discussed.

6. Conclusions

In this paper, we have reported the different factors which can affect the working of genetic algorithms. At the early stage of the paper we have discussed the effect of population diversity and selective pressure. In addition, we have given theoretical framework for GA which covers a brief description of three important theories, namely Schema theory, Markov chain theory and Statistical theory and their applicability have been shown by providing important literatures.

The main contribution of this paper is that it presents a review of various approaches used to prevent premature convergence in genetic algorithms. Many representative works were selected from the literature for the review and analysis. The working of each approach has been reported in a comprehensive manner with their merits, demerit and the experimental details. It has been observed that the diversity of the population and selective pressure are the key factor behind the premature convergence. In addition, it has seen that much of the research made use of GA reproduction operators to handle the premature convergence. It was shown in many literatures that by applying the crossover and mutation in an appropriate manner one can avoid premature convergence. A portion of the reviewed research reported where researchers had used other techniques also to control the convergence and keep the search process alive. The reviewed literatures have been summarized for a quick review (Table 3). By looking the summary Table 3 one can easily and quickly understand the workability of different approaches reviewed. The comparative analysis of reviewed approaches has been reported in Tables 4 and 5 based on various aspects. This paper not only covers various approaches to prevent premature convergence, but also it reports the key issues with each approach. The purpose of reporting the issues is to identify new research path to address the problem of premature convergence.

As we have reported, lots of research work had already been done in the area of genetic algorithm. But still there is scope for the development of advance and efficient algorithm to prevent premature convergence within GA search.

References

[1] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, 1992.

[2] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1998.

[4] H.M. Pandey, Natural language grammar induction using genetic and parallel genetic algorithms and load balancing, Int. J. Comput. Sci. Technol. 1 (1) (2012) 28.

[5] H.M. Pandey, Context free grammar induction library using Genetic Algorithms, in: IEEE 2010 International Conference on Computer and Communication Technology (ICCCT), 2010.

[6] H.B. Amor, A. Rettinger, Intelligent exploration for genetic algorithms: using self-organizing maps in evolutionary computation, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, ACM, 2005.

[7] D.B. Fogel, An introduction to simulated evolutionary optimization, IEEE Trans. Neural Netw. 5 (1) (1994) 3–14.

[8] H. Mohan, A. Dixit, D. Mehrotra, Genetic algorithms: concepts, issues and a case study of grammar induction, in: Proceedings of the CUBE International Information Technology Conference, ACM, 2012.

[9] C. Darwin, On the Origins of Species by Means of Natural Selection, Murray, London, 1859.

[10] G. Romero, et al., Genetic algorithm visualization using self-organizing maps, in: Parallel Problem Solving from Nature—PPSN, 2002, pp. 442–451.

[11] M. Mitchell, S. Forrest, J.H. Holland, The royal road for genetic algorithms: fitness landscapes and GA performance, in: Proceedings of the First European Conference on Artificial Life, The MIT Press, Cambridge, 1992.

[12] S.W. Mahfoud, Niching methods for genetic algorithms, Urbana 51 (1995) 61801.

[13] M. Lozano, F. Herrera, C.J. Ramón, Replacement strategies to preserve useful diversity in steady-state genetic algorithms, Inf. Sci. 178 (23) (2008) 4421–4433.

[14] B.M. Sultan, R. Mahmud, M.N. Sulaiman, Reducing premature convergence problem through numbers structuring in genetic algorithm, Int. J. Comput. Sci. Netw. Secur. 7 (2007) 215–217.

[15] M.E. Palmer, S.J. Smith, Improved evolutionary optimization of difficult landscapes: control of premature convergence through scheduled sharing, Complex Syst. 5 (5) (1992) 443–458.

[16] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, L. Erlbaum Associates Inc., 1987.

[17] K. Deb, D.E. Goldberg, An investigation of niche and species formation in genetic function optimization, in: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., 1989.

[18] C. Fonlupt, P. Preux, D. Robilliard, Preventing premature convergence via cooperating genetic algorithms, in: Proceedings of the Workshop on Foundations of Genetic Algorithms, Vail, CO, USA, 1993.

[19] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Syst. Man Cybern. 24 (4) (1994) 656–667.

[20] S. Wright, The roles of mutation, inbreeding, crossbreeding and selection in evolution, in: Proceedings of the Sixth International Congress on Genetics, 1932.

[21] F. Oppacher, M. Wineberg, The shifting balance genetic algorithm: improving the GA in a dynamic environment, in: Proceedings of the Genetic and Evolutionary Computation Conference, 1999.

[22] M. Rocha, J. Neves, Preventing premature convergence to local optima in genetic algorithms via random offspring generation, Multiple Approache to Intel. Syst. (1999) 127–136.

[23] G. Reinelt, "Tsplib95." Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR). Heidelberg, 1995.

[24] K. Balakrishnan, Parallel Genetic Algorithms, Premature Convergence and the nCUBE, 1993.

[25] M. Affenzeller, S. Wagner, SASEGASA: an evolutionary algorithm for retarding premature convergence by self-adaptive selection pressure steering, Comput. Method Neural Model. (2003) 1042.

[26] L. Juan, C. Zixing, L. Jianqin, Premature convergence in genetic algorithm: Analysis and prevention based on chaos operator, in: Proceedings of the 3rd World Congress on Intelligent Control and Automation, IEEE, 2000.

[27] T. Friedrich, et al., Analysis of diversity-preserving mechanisms for global exploration*, Evol. Comput. 17 (4) (2009) 455–476.

[28] D. Whitley, An overview of evolutionary algorithms: practical issues and common pitfalls, Inf. Softw. Technol. 43 (14) (2001) 817–831.

[29] S.J. Louis, G.J.E. Rawlins, Syntactic analysis of convergence in genetic algorithms, Found. Genet. Algorithms 2 (1993) 141–151.

[30] S.J. Louis, G.J.E. Rawlins, Predicting convergence time for genetic algorithms, Found. Genet. Algorithms 2 (1993) 141–161.

[31] E.S. Nicoară, Mechanisms to avoid the premature convergence of genetic algorithms, Petrol. Gas Univ. Ploieşti Bull. Math. Inf. Phys. Ser. 61 (2009) 87–96.

[32] L.J. Eshelman, J. David Schaffer, Preventing premature convergence in genetic algorithms by preventing incest, in: Proceedings of the Fourth International Conference on Genetic Algorithms, 1991.

[33] M. Basseur, F. Seynhaeve, E.-G. Talbi, Design of multi-objective evolutionary algorithms: application to the flow-shop scheduling problem, in: Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02), IEEE, 2002.

[34] K. Deb, et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[35] S.H. Jung, Selective mutation for genetic algorithms, World Acad. Sci. Eng. Technol. 56 (2009) 478–481.

[36] J. Andre, P. Siarry, T. Dognon, An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization, Adv. Eng. Softw. 32 (1) (2001) 49–60.

[37] S.H. Jung, Queen-bee evolution for genetic algorithms, Electron. Lett. 39 (6) (2003) 575–576.

[38] K.A. De Jong, Analysis of the behavior of a class of genetic adaptive systems, 1975.

[39] K. Premalatha, A.M. Natarajan, Hybrid PSO and GA for global maximization, Int. J. Open Problems Compt. Math. 2 (4) (2009) 597–608.

[40] F. Van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 225–239.

[41] G.S. Hornby, ALPS: the age-layered population structure for reducing the problem of premature convergence, in: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, 2006.

[42] V.M. Kureichick, V.V. Miagkikh, A.P. Topchy, Genetic algorithm for solution of the traveling salesman problem with new features against premature convergence, in: Proc. of GA+ SE 96, 1996.

[43] C. Ryan, Racial harmony in genetic algorithms, in: KI'94 Workshop on Genetic Algorithms, 1994.

[44] N.S. Choubey, M.U. Kharat, Hybrid system for handling premature convergence in GA-case of grammar induction, Appl. Soft Comput. (2012).

[45] D.E. Goldberg, K. Bradley, D. Kalyanmoy, Messy Genetic Algorithms: Motivation, Analysis, and First Results, University of Alabama, the Clearinghouse for Genetic Algorithms, Tuscaloosa, 1989, TCGA Report No. 89002.

[46] D.A. Ackley, A Connectionist Machine for Genetic Hill-climbing, Kluwer Academic Publishers, 1987.

[47] S.J. Louis, G.J. Rawlins, Designer genetic algorithms: genetic algorithms in structures design, in: Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kauffman, 1991, pp. 53–60.

[48] S.Z. Ramadan, Reducing premature convergence problem in genetic algorithm: application on travel salesman problem, Comput. Inf. Sci. 6 (1) (2012) 47.

[49] N. Choubey, M. Kharat, Approaches for handling premature convergence in CFG induction using GA, Soft Comput. Ind. Appl. (2011) 55–66.

[50] T.P. Bagchi, K. Deb, Calibration of GA parameters: the design of experiments approach, Comput. Sci. Inf. 26 (1996) 46–56.

[51] W.-O. Huijsen, Genetic grammatical inference, in: CLIN IV: Papers from the Fourth CLIN Meeting, 1993.

[52] B. Keller, L. Rudi, Evolving stochastic context-free grammars from examples using a minimum description length principle, in: Workshop on Automatic Induction, Grammatical Inference and Language Acquisition, 1997.

[53] M. Tomita, Dynamic construction of finite-state automata from examples using hill-climbing, in: Proceedings of the Fourth Annual Cognitive Science Conference, 1982.

[54] P. Dupont, Regular grammatical inference from positive and negative samples by genetic search: the GIG method, Gramm. Inference Appl. (1994) 236–245.

[55] M. Affenzeller, Segregative genetic algorithms (SEGA): a hybrid superstructure upwards compatible to genetic algorithms for retarding premature convergence, Int. J. Comput. Syst. Signal. 2 (1) (2001) 16–30.

[56] E. Schöneburg, F. Heinzmann, S. Feddersen, Genetische Algorithmen und Evolutionsstrategien, vol. 1, Addison-Wesley, Reading, MA, 1994.

[57] H.M. Pandey, Design Analysis and Algorithm, Firewall Media, 2008.

[58] H.M. Pandey, Data Structure and Algorithm, Laxmi Publications, Ltd., 2009.

[59] S. Achiche, M. Balazinski, L. Baron, Multi-combinative strategy to avoid premature convergence in genetically-generated fuzzy knowledge bases, J. Theor. Appl. Mech. Warsaw 42 (2004) 417–444.

[60] S. Achiche, L. Baron, M. Balazinski, Real/binary-like coded genetic algorithm to automatically generate fuzzy knowledge bases, in: Proceedings. 4th International Conference on Control and Automation (ICCA'03), IEEE, 2003.

[61] A.E. Eiben, T. Bäck, Empirical investigation of multiparent recombination operators in evolution strategies, Evol. Comput. 5 (3) (1997) 347–365.

[62] H. Kita, I. Ono, S. Kobayashi, Theoretical analysis of the unimodal normal distribution crossover for real-coded genetic algorithms, in: IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE, 1998.

[63] S. Tsutsui, Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring, in: Parallel Problem Solving from Nature—PPSN V, Springer, Berlin/Heidelberg, 1998.

[64] S. Tsutsui, A. Ghosh, A study on the effect of multi-parent recombination in real coded genetic algorithms, in: IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE, 1998.

[65] T.C. Belding, The Distributed Genetic Algorithm Revisited, 1995.

[66] M. Gorges-Schleuter, Explicit parallelism of genetic algorithms through population structures, in: Parallel Problem Solving from Nature, 1991, pp. 150–159.

[67] R. Tanese, J.H. Co-Chairman-Holland, Q.F. Co-Chairman-Stout, Distributed Genetic Algorithms for Function Optimization, 1989.

[68] D. Whitley, T. Starkweather, Genitor II: a distributed genetic algorithm, J. Exp. Theor. Artif. Intell. 2 (3) (1990) 189–214.

[69] D. Whitley, S. Rana, R.B. Heckendorn, The island model genetic algorithm: on separability, population size and convergence, J. Comput. Inf. Technol. 7 (1999) 33–48.

[70] V.S. Gordon, D. Whitley, Serial and parallel genetic algorithms as function optimizers, Colorado State University, 1993.

[71] E. Cantú-Paz, A survey of parallel genetic algorithms, Calculateurs Paralleles, Reseaux et Systems Repartis 10 (2) (1998) 141–171.

[72] E. Cantú-Paz, A summary of research on parallel genetic algorithms, 1995.

[73] Da. Whitley, The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best, in: Proceedings of the Third International Conference on Genetic Algorithms, 1989.

[74] M. Hall, A Cumulative Multi-Niching Genetic Algorithm for Multimodal Function Optimization, 2013.

[75] Samir W. Mahfoud, Crowding and preselection revisited, Urbana 51 (1992) 61801.

[76] G.F. Minetti, et al., Combining incest prevention and multiplicity in evolutionary algorithms, V Congreso Argentino de Ciencias de la Computación, 1999.

[77] C.-K. Ting, S.-T. Li, C. Lee, On the harmonious mating strategy through tabu search, Inf. Sci. 156 (3) (2003) 189–214.

[78] H. Shimodaira, DCGA: a diversity control oriented genetic algorithm, in: Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence, IEEE, 1997.

[79] K. Matsui, New selection method to improve the population diversity in genetic algorithms, in: 1999 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC'99), IEEE, 1999.

[80] C. Fernandes, A. Rosa, A study on non-random mating and varying population size in genetic algorithms using a royal road function, in: Proceedings of the 2001 Congress on Evolutionary Computation, IEEE, 2001.

[81] B. Runqiang, C. Zengqiang, Y. Zhuzhi, Improved crossover strategy of genetic algorithms and analysis of its performance, in: Proceedings of the 3rd World Congress on Intelligent Control and Automation, IEEE, 2000.

[82] S. De, S.K. Pal, A. Ghosh, Genotypic and phenotypic assortative mating in genetic algorithm, Inf. Sci. 105 (1) (1998) 209–226.

[83] R. Craighurst, W.N. Martin, Enhancing GA performance through crossover prohibitions based on ancestry, in: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., 1995.

[84] G.R. Harik, Finding Multimodal Solutions Using Restricted Tournament Selection, ICGA, 1995.

[85] B. Sareni, L. Krahenbuhl, Fitness sharing and niching methods revisited, IEEE Trans. Evol. Comput. 2 (3) (1998) 97–106.

[86] J.R. Koza, Genetic Programming: vol. 1, On the programming of computers by means of natural selection, vol. 1, MIT Press, 1992.

[87] J. Horn, N. Nafpliotis, D.E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on, IEEE, 1994.

[88] J.D. Schaffer, A. Morishima, An adaptive crossover distribution mechanism for genetic algorithms, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application Evolutionary Computation, L. Erlbaum Associates Inc., 1987.

[89] L. Davis, Adapting operator probabilities in genetic algorithms, in: International Conference on {G} enetic {A} lgorithms\'89, 1989.

[90] T.C. Fogarty, Varying the probability of mutation in the genetic algorithm, in: Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., 1989.

[91] J.J. Grefenstette, et al., Genetic algorithms for the traveling salesman problem, in: Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc., 1985.

[92] L.D. Whitley, T. Starkweather, D. Fuquay, Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator, vol. 89, ICGA, 1989.

[93] I.M. Oliver, D.J. Smith, J.R.C. Holland, A study of permutation crossover operators on the traveling salesman problem, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, L. Erlbaum Associates Inc., 1987.

[94] M.D. Vose, G.E. Liepinsl, Punctuated equilibria in genetic search, Complex Syst. 5 (1991) 31–44.

[95] D. Whitley, R. Das, C. Crabb, Tracking primary hyperplane competitors during genetic search, Ann. Math. Artif. Intell. 6 (4) (1992) 367–388.

[96] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85.

[97] M. Wineberg, Improving the behavior of the genetic algorithm in a dynamic environment, Dissertation, Carleton University, 2000.

[98] D.L. Hartl, A.G. Clark, Principles of Population Genetics, vol. 116, Sinauer Associates, Sunderland, 1997.

[99] J.P. Cohoon, W.N. Martin, D.S. Richards, A Multi-Population Genetic Algorithm for Solving the K-Partition Problem on Hyper-Cubes, vol. 91, ICGA, 1991.

[100] J.P. Cohoon, et al., Punctuated equilibria: a parallel genetic algorithm, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application, L. Erlbaum Associates Inc., 1987.

[101] S.J. Gould, N. Eldredge, Punctuated equilibrium comes of age, Nature (1993) 223–227.

[102] M. Gorges-Schleuter, ASPARAGOS an asynchronous parallel genetic optimization strategy, in: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., 1989.

[103] P. Spiessens, B. Manderick, A massively parallel genetic algorithm: implementation and first analysis, in: Proceedings of the Fourth International Conference on Genetic Algorithms, 1991.

[104] H. Mühlenbein, Parallel genetic algorithms, population genetics and combinatorial optimization, in: Parallelism, Learning, Evolution, Springer, Berlin/Heidelberg, 1991, pp. 398–406.

[105] W.D. Hillis, Co-evolving parasites improve simulated evolution as an optimization procedure, Phys. D 42 (1) (1990) 228–234.

[106] R.J. Collins, D.R. Jefferson, Selection in massively parallel genetic algorithms, Computer Science Department, University of California, 1991.

[107] Y. Davidor, A Naturally Occurring Niche and Species Phenomenon: The Model and First Results, ICGA, 1991.

[108] J.A. Coyne, N.H. Barton, M. Turelli, Perspective: a critique of Sewall Wright's shifting balance theory of evolution, Evolution (1997) 643–671.

[109] S.L. Peck, S.P. Ellner, F. Gould, A spatially explicit stochastic model demonstrates the feasibility of Wright's shifting balance theory, Evolution 52 (6) (1998) 1834–1839.

[110] M.J. Wade, C.J. Goodnight, Perspective: the theories of Fisher and Wright in the context of metapopulations: when nature does many small experiments, Evolution 52 (6) (1998) 1537–1553.

[111] R.A. Fisher, The Genetical Theory of Natural Selection, 1958.

[112] J.A. Coyne, N.H. Barton, M. Turelli, Is Wright's shifting balance process important in evolution? Evolution 54 (1) (2000) 306–317.

[113] R. Caponetto, et al., Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (3) (2003) 289–304.

[114] L.S. Coelho, A quantum particle swarm optimizer with chaotic mutation operator, Chaos Solitons Fract. 37 (5) (2008) 1409–1418.

[115] D. Davendra, Chaos driven evolutionary algorithm for the traveling salesman problem, Travel. Salesman Probl. Theor. Appl. (2010) 55–70.

[116] C.-T. Cheng, et al., Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos, Water Resour. Manag. 22 (7) (2008) 895–909.

[117] B.B. Firouzi, et al., A novel multi-objective Chaotic Crazy PSO algorithm for optimal operation management of distribution network with regard to fuel cell power plants, Int. J. Innov. Comput. I 7 (2011) 6395–6409.

[118] N.L. Law, K.Y. Szeto, Adaptive Genetic Algorithm with Mutation and Crossover Matrices, IJCAI, 2007.

[119] W.M. Spears, Crossover or mutation? FOGA, 1992.

[120] K.Y. Szeto, J. Zhang, Adaptive genetic algorithm and quasi-parallel genetic algorithm: application to knapsack problem, in: Ivan Lirkov, Svetozar Margenov, Jerzy Waśniewski (Eds.), Proceedings of the 5th International Conference on Large-Scale Scientific Computing (LSSC'05), Springer-Verlag, Berlin/Heidelberg, 2005, pp. 189–196.

[121] J. Zhang, K.Y. Szeto, Mutation matrix in evolutionary computation: An application to resource allocation problem, in: Advances in Natural Computation, Springer, Berlin/Heidelberg, 2005, pp. 112–119.

[122] C.W. Ma, K.-Y. Szeto, Locus oriented adaptive genetic algorithm: application to the zero/one knapsack problem, 2004.

[123] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95), 4–6 October, 1995, p. 39.

[124] R.C. Eberhart, X. Hu, Human tremor analysis using particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999.

[125] A.P. Engelbrecht, A. Engelbrecht, A. Ismail, Training product unit neural networks, 1999.

[126] F. van den Bergh, A.P. Engelbrecht, Cooperative learning in neural networks using particle swarm optimizers S. Afr. Comput. J. 26 (84–90) (2000) 21.

[127] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 4–9 May, 1998, p. 69.

[128] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999, p. 1950.

[129] D.J. Cavicchio, Adaptive search using simulated evolution, 1970.

[130] J.J. Hu, E.D. Goodman, The hierarchical fair competition (HFC) model for parallel evolutionary algorithms, in: Proceedings of the 2002 Congress on Evolutionary Computation, 12–17 May, 2002, p. 54.

[131] M. Hutter, Fitness uniform selection to preserve genetic diversity, in: Proceedings of the 2002 Congress on Evolutionary Computation, 12–17 May, 2002, p. 783.

[132] H. Jianjun, et al., Adaptive hierarchical fair competition (AHFC) model for parallel evolutionary algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, New York, 2002.

[133] J. Hu, E.D. Goodman, K. Seo, Continuous hierarchical fair competition model for sustainable innovation in genetic programming, in: Genetic Programming Theory and Practice, Springer, USA, 2003, pp. 81–98.

[134] A. Oliveira, et al., An adaptive hierarchical fair competition genetic algorithm for large-scale numerical optimization, 2004.

[135] G.S. Hornby, The age-layered population structure (ALPS) evolutionary algorithm, 2009.

[136] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[137] S.N. Sivanandam, S.N. Deepa, Introduction to Genetic Algorithms, Springer, 2007.

[138] C.R. Reeves, J.E. Rowe, Genetic Algorithms-Principles and Perspectives: A Guide to GA Theory, vol. 20, Springer, 2002.

[139] H.-G. Beyer, An alternative explanation for the manner in which genetic algorithms operate, Biosystems 41 (1) (1997) 1–15.

[140] A.E. Eiben, G.r Rudolph, Theory of evolutionary algorithms: a bird's eye view, Theor. Comput. Sci. 229 (1) (1999) 3–9.

[141] G. Rudolph, Convergence analysis of canonical genetic algorithms, IEEE Trans. Neural Netw. 5 (1) (1994) 96–101.

[142] E.B. Clark, A Framework for Modelling Stochastic Optimisation Algorithms with Markov Chains, Dissertation, University of York, 2008.

[143] A. Auger, Convergence results for the $(1,\lambda)$-SA-ES using the theory of ϕ-irreducible Markov chains, Theor. Comput. Sci. 334 (1) (2005) 35–69.

[144] A.E. Nix, M.D. Vose, Modeling genetic algorithms with Markov chains, Ann. Math. Artif. Intell. 5 (1) (1992) 79–88.

[145] C.-F. Huang, A Markov Chain Analysis of Fitness Proportional Mate Selection Schemes in Genetic Algorithm, vol. 2, GECCO, 2002.

[146] C.-F. Huang, A study of Fitness Proportional Mate selection schemes in Genetic Algorithms, Population 30 (35) (2002) 40.

[147] J. Suzuki, A Markov chain analysis of genetic algorithms: large deviation principle approach, J. Appl. Prob. 47 (4) (2010) 967–975.

[148] T. Braak, J.F. Cajo, A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces, Stat. Comput. 16 (3) (2006) 239–249.

[149] K.V. Price, An introduction to differential evolution, in: New Ideas in Optimization, McGraw-Hill Ltd., UK, 1999.

[150] K. Price, R. Storn, Differential evolution, Dobb's J. 264 (1997) 18–24.

[151] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.

[152] W.R. Gilks, S. Richardson, D.J. Spiegelhalter (Eds.), Markov chain Monte Carlo in Practice, vol. 2, CRC Press, 1996.

[153] W.R. Gilks, G.O. Roberts, Strategies for improving MCMC, Markov Chain Monte Carlo Pract. (1996) 89–114.

[154] A.E. Eiben, E.H.L. Aarts, K.M. Van Hee, Global convergence of genetic algorithms: a Markov chain analysis, in: Parallel Problem Solving From Nature, Springer, Berlin/Heidelberg, 1991, pp. 3–12.

[155] D.R. Stark, J.C. Spall, Computable rate of convergence in evolutionary computation, in: Proceedings of the Fifth International Conference on Information Fusion, IEEE, 2002.

[156] J. He, X. Yu, Conditions for the convergence of evolutionary algorithms, J. Syst. Archit. 47 (7) (2001) 601–612.

[157] B. Héder, J. Bitó, Convergence analysis of genetic algorithm applied for dynamic optimization of terminal to base station assignment in satellite fed BFWA systems, in: IEEE International Workshop on Satellite and Space Communications (IWSSC 2008), IEEE, 2008.

[158] Y. Leung, Y. Gao, Z.-B. Xu, Degree of population diversity – a perspective on premature convergence in genetic algorithms and its markov chain analysis, IEEE Trans. Neural Netw. 8 (5) (1997) 1165–1176.

[159] R. Poli, N.F. McPhee, J.E. Rowe, Exact schema theory and Markov chain models for genetic programming and variable-length genetic algorithms with homologous crossover, Genet. Program. Evol. Mach. 5 (1) (2004) 31–70.

[160] R.H. Je, Finite Markov chain analysis of genetic algorithms with niching, Forrest 727 (1993) 110–117.

[161] A. Prügel-Bennett, J.L. Shapiro, Analysis of genetic algorithms using statistical mechanics, Phys. Rev. Lett. (US) 72 (9) (1994).

[162] R. Balescu, Equilibrium and nonequilibrium statistical mechanics, NASA STI/Recon Tech. Rep. A 76 (1975) 32809.

[163] A. Prügel-Bennett, A. Rogers, Modelling genetic algorithm dynamics, in: Theoretical Aspects of Evolutionary Computing, Springer, Berlin/Heidelberg, 2001, pp. 59–85.

[164] J.L. Shapiro, Statistical mechanics theory of genetic algorithms, in: Theoretical Aspects of Evolutionary Computing, Springer, Berlin/Heidelberg, 2001, pp. 87–108.

[165] A. Kuri, A statistical genetic algorithm, in: National Computation Meeting, Hidalgo, México, 1999.

[166] D. Shilane, et al., A general framework for statistical performance comparison of evolutionary computation algorithms, Inf. Sci. 178 (14) (2008) 2870–2879.

[167] A. Petrovski, A. Wilson, J. McCall, Statistical analysis of genetic algorithms and inference about optimal factors, School of Mathematics in Computer Science, Faculty of Science and Technology, The Robert Gordon University, Aberdeen, UK, 1998.

[168] J.L. Shapiro, A. Prügel-Bennett, M. Rattray, A statistical mechanics analysis of genetic algorithms for search and learning, in: Mathematics of Neural Networks, Springer, USA, 1997, pp. 318–322.

Hari Mohan Pandey obtained M.Tech degree in Computer Engineering from Mukesh Patel School of Technology Management & Engineering. He is persuing Ph.D. programming in Computer Science & Engineering. He has published research papers in various journals. He has written many books in the field of Computer Science & Engineering for McGraw-Hill, Pearson Education, and University Science Press. He is associated with various International Journals as reviewer and editorial board member. His area of interest Machine Learning Computer, Artificial Intelligence, Soft Computing etc.

Ankit Chaudhary is major in Computer Science & Engineering and received his PhD in Computer Vision. His current research interests are in Vision based applications, Intelligent Systems and Graph Algorithms. He carried out his research work at Machine Vision Lab, CEERI-CSIR with Dr.-Ing. J. L. Raheja. Later, he was a Post-Doc at Department of Electrical and Computer Engineering, The University of Iowa, with Prof. Milan Sonka. Currently he is Assistant Professor at Department of Computer Science, MUM. Prior to this, he has been a faculty at Department of Computer Science, BITS Pilani. He has also worked with CITRIX R&D and AVAYA INC in past. He is on the Editorial Board of several International Journals and serves as TPC in many Conferences. He is also reviewer for Journals including IEEE Transactions on Image Processing, Machine Vision and Applications, ACM Transactions on Interactive Intelligent Systems, Signal Image and Video Processing, Expert Systems with Applications, Robotics and Autonomous Systems and others.

Deepti Mehrotra is gold medalist of Lucknow University and had received Ph.D. from Lucknow University. Currently she is Director of Amity School of Computer Science, Noida, India. She has more than 18 year of experience in teaching, research and content writing. She had published more than 50 papers in international refereed Journals and conference Proceedings. Dr. Mehrotra is member of many committees like member of Board of Study of various Universities, member of DRC of Amity University and reviewer for many referred journal and conferences. She is regularly invited as resource person for FDPs and invited talk in national and international conference. She is currently guiding 7 Ph.D. and four M.Tech. scholars.