



# A review on genetic algorithm: past, present, and future

Sourabh Katoch<sup>1</sup> · Sumit Singh Chauhan<sup>1</sup> · Vijay Kumar<sup>1</sup> 

Received: 27 July 2020 / Revised: 12 October 2020 / Accepted: 23 October 2020 /

Published online: 31 October 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

In this paper, the analysis of recent advances in genetic algorithms is discussed. The genetic algorithms of great interest in research community are selected for analysis. This review will help the new and demanding researchers to provide the wider vision of genetic algorithms. The well-known algorithms and their implementation are presented with their pros and cons. The genetic operators and their usages are discussed with the aim of facilitating new researchers. The different research domains involved in genetic algorithms are covered. The future research directions in the area of genetic operators, fitness function and hybrid algorithms are discussed. This structured review will be helpful for research and graduate teaching.

**Keywords** Optimization · Metaheuristic · Genetic algorithm · Crossover · Mutation · Selection · Evolution

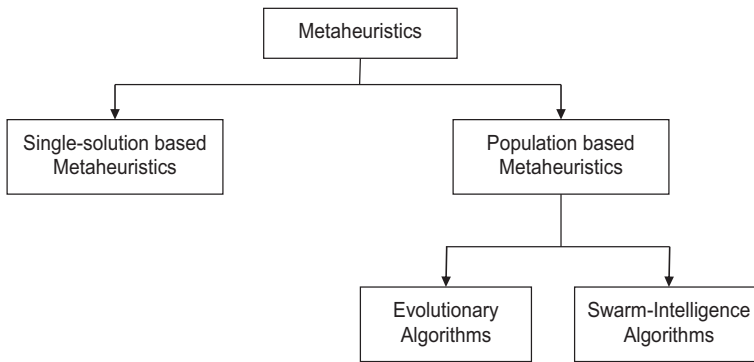
## 1 Introduction

In the recent years, metaheuristic algorithms are used to solve real-life complex problems arising from different fields such as economics, engineering, politics, management, and engineering [113]. Intensification and diversification are the key elements of metaheuristic algorithm. The proper balance between these elements are required to solve the real-life problem in an effective manner. Most of metaheuristic algorithms are inspired from biological evolution process, swarm behavior, and physics' law [17]. These algorithms are broadly classified into two categories namely single solution and population based metaheuristic algorithm (Fig. 1). Single-solution based metaheuristic algorithms utilize single candidate solution and improve this solution by using local search. However, the solution obtained from single-solution based metaheuristics may stuck in local optima [112]. The well-known single-solution based metaheuristics are

---

✉ Vijay Kumar  
vijaykumarchahar@gmail.com

<sup>1</sup> Computer Science and Engineering Department, National Institute of Technology, Hamirpur, India



**Fig. 1** Classification of metaheuristic Algorithms

simulated annealing, tabu search (TS), microcanonical annealing (MA), and guided local search (GLS). Population-based metaheuristics utilizes multiple candidate solutions during the search process. These metaheuristics maintain the diversity in population and avoid the solutions are being stuck in local optima. Some of well-known population-based metaheuristic algorithms are genetic algorithm (GA) [135], particle swarm optimization (PSO) [101], ant colony optimization (ACO) [47], spotted hyena optimizer (SHO) [41], emperor penguin optimizer (EPO) [42], and seagull optimization (SOA) [43].

Among the metaheuristic algorithms, Genetic algorithm (GA) is a well-known algorithm, which is inspired from biological evolution process [136]. GA mimics the Darwinian theory of survival of fittest in nature. GA was proposed by J.H. Holland in 1992. The basic elements of GA are chromosome representation, fitness selection, and biological-inspired operators. Holland also introduced a novel element namely, Inversion that is generally used in implementations of GA [77]. Typically, the chromosomes take the binary string format. In chromosomes, each locus (specific position on chromosome) has two possible alleles (variant forms of genes) - 0 and 1. Chromosomes are considered as points in the solution space. These are processed using genetic operators by iteratively replacing its population. The fitness function is used to assign a value for all the chromosomes in the population [136]. The biological-inspired operators are selection, mutation, and crossover. In selection, the chromosomes are selected on the basis of its fitness value for further processing. In crossover operator, a random locus is chosen and it changes the subsequences between chromosomes to create off-springs. In mutation, some bits of the chromosomes will be randomly flipped on the basis of probability [77, 135, 136]. The further development of GA based on operators, representation, and fitness has diminished. Therefore, these elements of GA are focused in this paper.

The main contribution of this paper are as follows:

1. The general framework of GA and hybrid GA are elaborated with mathematical formulation.
2. The various types of genetic operators are discussed with their pros and cons.
3. The variants of GA with their pros and cons are discussed.
4. The applicability of GA in multimedia fields is discussed.

The main aim of this paper is two folds. First, it presents the variants of GA and their applicability in various fields. Second, it broadens the area of possible users in various fields. The various types of crossover, mutation, selection, and encoding techniques are discussed. The single-objective, multi-objective, parallel, and hybrid GAs are deliberated with their advantages and disadvantages. The multimedia applications of GAs are elaborated.

The remainder of this paper is organized as follows: Section 2 presents the methodology used to carry out the research. The classical genetic algorithm and genetic operators are discussed in Section 3. The variants of genetic algorithm with pros and cons are presented in Section 4. Section 5 describes the applications of genetic algorithm. Section 6 presents the challenges and future research directions. The concluding remarks are drawn in Section 7.

## 2 Research methodology

PRISMA's guidelines were used to conduct the review of GA [138]. A detailed search has been done on Google scholar and PubMed for identification of research papers related to GA. The important research works found during the manual search were also added in this paper. During search, some keywords such as “Genetic Algorithm” or “Application of GA” or “operators of GA” or “representation of GA” or “variants of GA” were used. The selection and rejection of explored research papers are based on the principles, which is mentioned in Table 1.

Total 27,64,792 research papers were explored on Google Scholar, PubMed and manual search. The research work related to genetic algorithm for multimedia applications were also included. During the screening of research papers, all the duplicate papers and papers published before 2007 were discarded. 4340 research papers were selected based on 2007 and duplicate entries. Thereafter, 4050 research papers were eliminated based on titles. 220 research papers were eliminated after reading of abstract. 70 research papers were left after third round of screening. 40 more research papers were discarded after full paper reading and facts found in the papers. After the fourth round of screening, final 30 research papers are selected for review.

Based on the relevance and quality of research, 30 papers were selected for evaluation. The relevance of research is decided through some criteria, which is mentioned in Table 1. The

**Table 1** Selection criterion for shortlisted research papers

| Sr. No. | Parameters   | Selection criteria  | Elimination criteria  |
|---------|--------------|---|---|
| 1       | Duration     | Research papers published from 2007 to 2020                                 | Research papers published before 2007   |
| 2       | Analysis     | Research includes various operators and modification in GA                  | Research includes operators of other metaheuristics   |
| 3       | Comparison   | Research focuses on variants of GA  | Research focuses on variants of other metaheuristics. GA included in some part of research              |
| 4       | Applications | Research involves on multimedia, operation management and wireless networks | Research involves on engineering design, data mining, software applications, and astronomy applications |
| 5       | Study        | Research includes mathematical foundation and experimental results          | Research includes patent, case study, papers having language other than English                         |

selected research papers comprise of genetic algorithm for multimedia applications, advancement of their genetic operators, and hybridization of genetic algorithm with other well-established metaheuristic algorithms. The pros and cons of genetic operators are shown in preceding section.

### 3 Background

In this section, the basic structure of GA and its genetic operators are discussed with pros and cons.

#### 3.1 Classical GA

Genetic algorithm (GA) is an optimization algorithm that is inspired from the natural selection. It is a population based search algorithm, which utilizes the concept of survival of fittest [135]. The new populations are produced by iterative use of genetic operators on individuals present in the population. The chromosome representation, selection, crossover, mutation, and fitness function computation are the key elements of GA. The procedure of GA is as follows. A population ( $Y$ ) of  $n$  chromosomes are initialized randomly. The fitness of each chromosome in  $Y$  is computed. Two chromosomes say  $C1$  and  $C2$  are selected from the population  $Y$  according to the fitness value. The single-point crossover operator with crossover probability ( $C_p$ ) is applied on  $C1$  and  $C2$  to produce an offspring say  $O$ . Thereafter, uniform mutation operator is applied on produced offspring ( $O$ ) with mutation probability ( $M_p$ ) to generate  $O'$ . The new offspring  $O'$  is placed in new population. The selection, crossover, and mutation operations will be repeated on current population until the new population is complete. The mathematical analysis of GA is as follows [126]:

GA dynamically change the search process through the probabilities of crossover and mutation and reached to optimal solution. GA can modify the encoded genes. GA can evaluate multiple individuals and produce multiple optimal solutions. Hence, GA has better global search capability. The offspring produced from crossover of parent chromosomes is probable to abolish the admirable genetic schemas parent chromosomes and crossover formula is defined as [126]:

$$R = (G + 2\sqrt{g})/3G \quad (1)$$

where  $g$  is the number of generations, and  $G$  is the total number of evolutionary generation set by population. It is observed from Eq.(1) that  $R$  is dynamically changed and increase with increase in number of evolutionary generation. In initial stage of GA, the similarity between individuals is very low. The value of  $R$  should be low to ensure that the new population will not destroy the excellent genetic schema of individuals. At the end of evolution, the similarity between individuals is very high as well as the value of  $R$  should be high.

According to Schema theorem, the original schema has to be replaced with modified schema. To maintain the diversity in population, the new schema keep the initial population during the early stage of evolution. At the end of evolution, the appropriate schema will be produced to prevent any distortion of excellent genetic schema [65, 75]. Algorithm 1 shows the pseudocode of classical genetic algorithm.

---

**Algorithm 1: Classical Genetic Algorithm (GA)**

---

**Input:***Population Size,  $n$* *Maximum number of iterations,  $MAX$* **Output:***Global best solution,  $Y_{bt}$* 

---

**begin***Generate initial population of  $n$  chromosomes  $Y_i$  ( $i = 1, 2, \dots, n$ )**Set iteration counter  $t = 0$* *Compute the fitness value of each chromosomes***while** ( $t < MAX$ )*Select a pair of chromosomes from initial population based on fitness**Apply crossover operation on selected pair with crossover probability**Apply mutation on the offspring with mutation probability**Replace old population with newly generated population**Increment the current iteration  $t$  by 1.***end while***return the best solution,  $Y_{bt}$* **end**

---

## 3.2 Genetic operators

GAs used a variety of operators during the search process. These operators are encoding schemes, crossover, mutation, and selection. Figure 2 depicts the operators used in GAs.

### 3.2.1 Encoding schemes

For most of the computational problems, the encoding scheme (i.e., to convert in particular form) plays an important role. The given information has to be encoded in a particular bit string [121, 183]. The encoding schemes are differentiated according to the problem domain. The well-known encoding schemes are binary, octal, hexadecimal, permutation, value-based, and tree.

Binary encoding is the commonly used encoding scheme. Each gene or chromosome is represented as a string of 1 or 0 [187]. In binary encoding, each bit represents the characteristics of the solution. It provides faster implementation of crossover and mutation operators. However, it requires extra effort to convert into binary form and accuracy of algorithm depends upon the binary conversion. The bit stream is changed according the problem. Binary encoding scheme is not appropriate for some engineering design problems due to epistasis and natural representation.

In octal encoding scheme, the gene or chromosome is represented in the form of octal numbers (0–7). In hexadecimal encoding scheme, the gene or chromosome is represented in the form of hexadecimal numbers (0–9, A-F) [111, 125, 187]. The permutation encoding

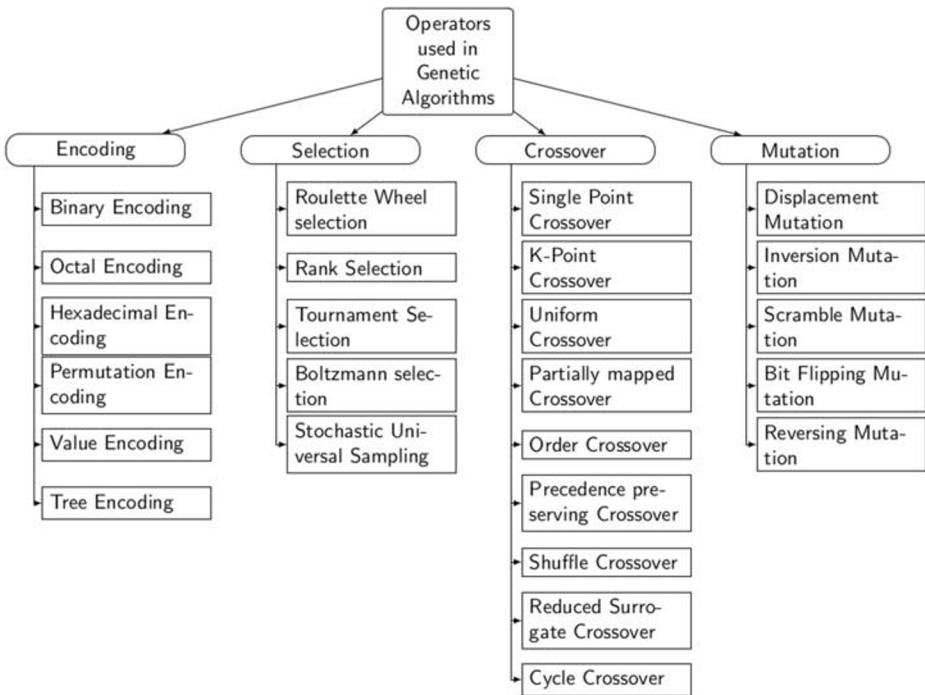


Fig. 2 Operators used in GA

scheme is generally used in ordering problems. In this encoding scheme, the gene or chromosome is represented by the string of numbers that represents the position in a sequence. In value encoding scheme, the gene or chromosome is represented using string of some values. These values can be real, integer number, or character [57]. This encoding scheme can be helpful in solving the problems in which more complicated values are used. As binary encoding may fail in such problems. It is mainly used in neural networks for finding the optimal weights.

In tree encoding, the gene or chromosome is represented by a tree of functions or commands. These functions and commands can be related to any programming language. This is very much similar to the representation of repression in tree format [88]. This type of encoding is generally used in evolving programs or expressions. Table 2 shows the comparison of different encoding schemes of GA.

### 3.2.2 Selection techniques

Selection is an important step in genetic algorithms that determines whether the particular string will participate in the reproduction process or not. The selection step is sometimes also known as the reproduction operator [57, 88]. The convergence rate of GA depends upon the selection pressure. The well-known selection techniques are roulette wheel, rank, tournament, boltzmann, and stochastic universal sampling.

Roulette wheel selection maps all the possible strings onto a wheel with a portion of the wheel allocated to them according to their fitness value. This wheel is then rotated randomly to select specific solutions that will participate in formation of the next generation [88]. However,

**Table 2** Comparison of different encoding schemes

| Encoding Scheme | Pros                                  | Cons                                       | Application                           |
|-----------------|---------------------------------------|--|---------------------------------------|
| Binary          | Easy to implement<br>Faster Execution | No support for inversion operator          | Problems that support binary encoding |
| Octal           | Easy to implement                     | No support for inversion operator          | Limited use                           |
| Hexadecimal     | Easy to implement                     | No support for inversion operator          | Limited use                           |
| Permutation     | Support inversion operator            | No support for binary operators            | Task ordering Problem                 |
| Value           | No need of value conversion           | Requires specific crossover and mutation   | Neural Network Problems               |
| Tree            | Operator can easily applied           | Difficult to design tree for some problems | Evolving Programs                     |

it suffers from many problems such as errors introduced by its stochastic nature. De Jong and Brindle modified the roulette wheel selection method to remove errors by introducing the concept of determinism in selection procedure. Rank selection is the modified form of Roulette wheel selection. It utilizes the ranks instead of fitness value. Ranks are given to them according to their fitness value so that each individual gets a chance of getting selected according to their ranks. Rank selection method reduces the chances of prematurely converging the solution to a local minima [88].

Tournament selection technique was first proposed by Brindle in 1983. The individuals are selected according to their fitness values from a stochastic roulette wheel in pairs. After selection, the individuals with higher fitness value are added to the pool of next generation [88]. In this method of selection, each individual is compared with all  $n-1$  other individuals if it reaches the final population of solutions [88]. Stochastic universal sampling (SUS) is an extension to the existing roulette wheel selection method. It uses a random starting point in the list of individuals from a generation and selects the new individual at evenly spaced intervals [3]. It gives equal chance to all the individuals in getting selected for participating in crossover for the next generation. Although in case of Travelling Salesman Problem, SUS performs well but as the problem size increases, the traditional Roulette wheel selection performs relatively well [180].

Boltzmann selection is based on entropy and sampling methods, which are used in Monte Carlo Simulation. It helps in solving the problem of premature convergence [118]. The probability is very high for selecting the best string, while it executes in very less time. However, there is a possibility of information loss. It can be managed through elitism [175]. Elitism selection was proposed by K. D. Jong (1975) for improving the performance of Roulette wheel selection. It ensures the elitist individual in a generation is always propagated to the next generation. If the individual having the highest fitness value is not present in the next generation after normal selection procedure, then the elitist one is also included in the next generation automatically [88]. The comparison of above-mentioned selection techniques are depicted in Table 3.

### 3.2.3 Crossover operators

Crossover operators are used to generate the offspring by combining the genetic information of two or more parents. The well-known crossover operators are single-

**Table 3** Comparison of different selection techniques

| Selection Techniques                          | Pros   | Cons   |
|---|--|--|
| Roulette wheel                                | Easy to implement<br>Simple  | Risk of Premature convergence<br>Depends upon variance present in the fitness function |
| Rank  | Free from Bias<br>Preserve diversity<br>Free from Bias               | Slow convergence<br>Sorting required<br>Computationally Expensive                      |
| Tournament                                    | Preserve diversity<br>Parallel Implementation<br>No sorting required | Loss of diversity when the tournament size is large                                    |
| Boltzmann<br>Stochastic Universal<br>Sampling | Global optimum achieved<br>Fast Method<br>Free from Bias             | Computationally Expensive<br>Premature convergence                                     |
| Elitism                                       | Preserve best Individual in population                               | Best individual can be lost due to crossover and mutation operators                    |

point, two-point, k-point, uniform, partially matched, order, precedence preserving crossover, shuffle, reduced surrogate and cycle.

In a single point crossover, a random crossover point is selected. The genetic information of two parents which is beyond that point will be swapped with each other [190]. Figure 3 shows the genetic information after swapping. It replaced the tail array bits of both the parents to get the new offspring.

In a two point and k-point crossover, two or more random crossover points are selected and the genetic information of parents will be swapped as per the segments that have been created [190]. Figure 4 shows the swapping of genetic information between crossover points. The middle segment of the parents is replaced to generate the new offspring.

In a uniform crossover, parent cannot be decomposed into segments. The parent can be treated as each gene separately. We randomly decide whether we need to swap the gene with the same location of another chromosome [190]. Figure 5 depicts the swapping of individuals under uniform crossover operation.

Partially matched crossover (PMX) is the most frequently used crossover operator. It is an operator that performs better than most of the other crossover operators. The partially matched (mapped) crossover was proposed by D. Goldberg and R. Lingle [66]. Two parents are choose for mating. One parent donates some part of genetic material and the corresponding part of other parent participates in the child. Once this process is completed, the left out alleles are copied from the second parent [83]. Figure 6 depicts the example of PMX.

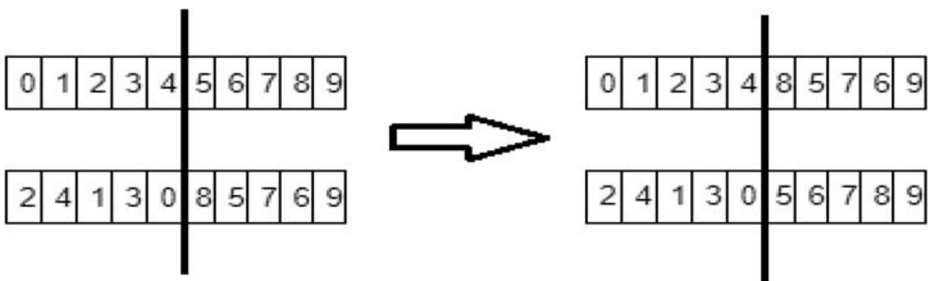
**Fig. 3** Swapping genetic information after a crossover point





Fig. 4 Swapping genetic information between crossover points

Order crossover (OX) was proposed by Davis in 1985. OX copies one (or more) parts of parent to the offspring from the selected cut-points and fills the remaining space with values other than the ones included in the copied section. The variants of OX are proposed by different researchers for different type of problems. OX is useful for ordering problems [166]. However, it is found that OX is less efficient in case of Travelling Salesman Problem [140]. Precedence preserving crossover (PPX) preserves the ordering of individual solutions as present in the parent of offspring before the application of crossover. The offspring is initialized to a string of random 1’s and 0’s that decides whether the individuals from both parents are to be selected or not. In [169], authors proposed a modified version of PPX for multi-objective scheduling problems.

Shuffle crossover was proposed by Eshelman et al. [20] to reduce the bias introduced by other crossover techniques. It shuffles the values of an individual solution before the crossover and unshuffles them after crossover operation is performed so that the crossover point does not introduce any bias in crossover. However, the utilization of this crossover is very limited in the recent years. Reduced surrogate crossover (RCX) reduces the unnecessary crossovers if the parents have the same gene sequence for solution representations [20, 139]. RCX is based on the assumption that GA produces better individuals if the parents are sufficiently diverse in their genetic composition. However, RCX cannot produce better individuals for those parents that have same composition. Cycle crossover was proposed by Oliver [140]. It attempts to generate an offspring using parents where each element occupies the position by referring to the position of their parents [140]. In the first cycle, it takes some elements from the first parent. In the second cycle, it takes the remaining elements from the second parent as shown in Fig. 7.

Table 4 shows the comparison of crossover techniques. It is observed from Table 4 that single and k-point crossover techniques are easy to implement. Uniform crossover is suitable for large subsets. Order and cycle crossovers provide better exploration than the other crossover techniques. Partially matched crossover provides better exploration. The performance of partially matched crossover is better than the other crossover techniques. Reduced surrogate and cycle crossovers suffer from premature convergence.



Fig. 5 Swapping individual genes

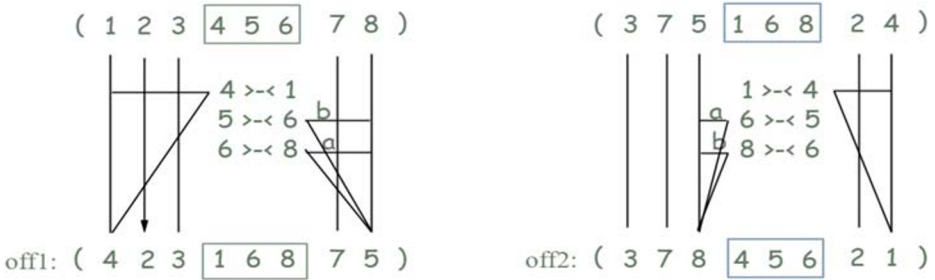


Fig. 6 Partially matched crossover (PMX) [117]

### 3.2.4 Mutation operators

Mutation is an operator that maintains the genetic diversity from one population to the next population. The well-known mutation operators are displacement, simple inversion, and scramble mutation. Displacement mutation (DM) operator displaces a substring of a given individual solution within itself. The place is randomly chosen from the given substring for displacement such that the resulting solution is valid as well as a random displacement mutation. There are variants of DM are exchange mutation and insertion mutation. In Exchange mutation and insertion mutation operators, a part of an individual solution is either exchanged with another part or inserted in another location, respectively [88].

The simple inversion mutation operator (SIM) reverses the substring between any two specified locations in an individual solution. SIM is an inversion operator that reverses the randomly selected string and places it at a random location [88]. The scramble mutation (SM) operator places the elements in a specified range of the individual solution in a random order and checks whether the fitness value of the recently generated solution is improved or not [88]. Table 5 shows the comparison of different mutation techniques.

Table 6 shows the best combination of encoding scheme, mutation, and crossover techniques. It is observed from Table 6 that uniform and single-point crossovers can be used with most of encoding and mutation operators. Partially matched crossover is used with inversion mutation and permutation encoding scheme provides the optimal solution.

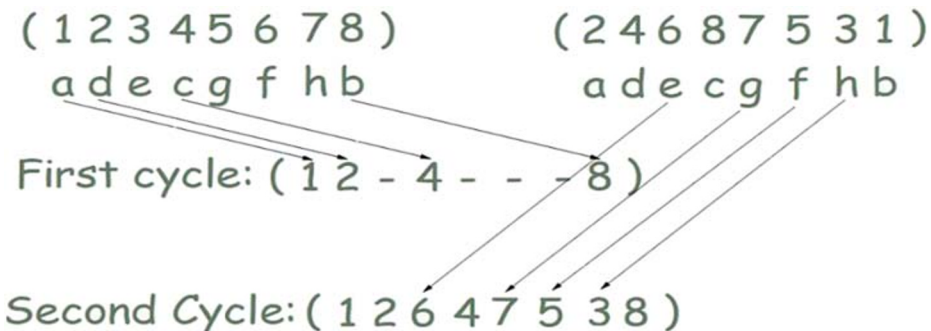


Fig. 7 Cycle Crossover (CX) [140]

**Table 4** Comparison of different crossover techniques

| Technique                     | Pros  | Cons  |
|-------------------------------|---|---|
| Single point                  | Easy to implement<br>Simple   | Less diverse solutions                                |
| Two and K-point               | Easy to implement   | Less diverse solutions<br>Applicable on small subsets |
| Reduced Surrogate             | Better performance over small optimization problems                                   | Premature convergence                                 |
| Uniform                       | Unbiased Exploration<br>Applicable on large subsets<br>Better recombination potential | Less diverse solutions                                |
| Precedence Preservative (PPX) | Better offspring generation   | Redundancy Problem                                    |
| Order Crossover (OX)          | Better Exploration  | Loss of information from previous individual          |
| Cycle Crossover               | Unbiased Exploration  | Premature convergence                                 |
| Partially Mapped (PMX)        | Better Convergence rate<br>Superior than the other crossovers                         | NA  |

## 4 Variants of GA

Various variants of GA's have been proposed by researchers. The variants of GA are broadly classified into five main categories namely, real and binary coded, multiobjective, parallel, chaotic, and hybrid GAs. The pros and cons of these algorithms with their application has been discussed in the preceding subsections.

### 4.1 Real and binary coded GAs

Based on the representation of chromosomes, GAs are categorized in two classes, namely binary and real coded GAs.

#### 4.1.1 Binary coded GAs

The binary representation was used to encode GA and known as binary GA. The genetic operators were also modified to carry out the search process. Payne and Glen [153] developed a binary GA to identify the similarity among molecules. They used binary representation for position of molecule and their conformations. However, this method has high computational complexity. Longyan et al. [203] investigated three different method for wind farm design

**Table 5** Comparison of different mutation operators

| Operator                  | Pros   | Cons  |
|---------------------------|--|---|
| Displacement Mutation     | Easy to implement<br>Applicable on small problem instances             | Risk of Premature convergence   |
| Simple-Inversion Mutation | Easy to implement  | Premature convergence   |
| Scramble Mutation         | Affects large number of genes<br>Applicable on large problem instances | Disturbance in the population<br>Deterioration of solution quality in some problems |

**Table 6** Best combination of various operators under optimal Environment

| Encoding Scheme | Mutation     | Crossover  |
|-----------------|--------------|--|
| Binary Encoding | Inversion    | Uniform, Arithmetic, 1-Point, N-Point                            |
| Permutation     | Inversion    | Partially Matched Crossover,<br>Cycle Crossover, Order Crossover |
| Value           | Displacement | Uniform, Arithmetic, 1-Point, N-Point                            |
| Tree            | Scramble     | Uniform, 1-Point   |

using binary GA (BGA). Their method produced better fitness value and farm efficiency. Shukla et al. [185] utilized BGA for feature subset selection. They used mutual information maximization concept for selecting the significant features. BGAs suffer from Hamming cliffs, uneven schema, and difficulty in achieving precision [116, 199].

#### 4.1.2 Real-coded GAs

Real-coded GAs (RGAs) have been widely used in various real-life applications. The representation of chromosomes is closely associated with real-life problems. The main advantages of RGAs are robust, efficient, and accurate. However, RGAs suffer from premature convergence. Researchers are working on RGAs to improve their performance. Most of RGAs are developed by modifying the crossover, mutation and selection operators.

**Crossover operators** The searching capability of crossover operators are not satisfactory for continuous search space. The developments in crossover operators have been done to enhance their performance in real environment. Wright [210] presented a heuristics crossover that was applied on parents to produce off-spring. Michalewicz [135] proposed arithmetical crossover operators for RGAs. Deb and Agrawal [34] developed a real-coded crossover operator, which is based on characteristics of single-point crossover in BGA. The developed crossover operator named as simulated binary crossover (SBX). SBX is able to overcome the Hamming cliff, precision, and fixed mapping problem. The performance of SBX is not satisfactory in two-variable blocked function. Eshelman et al. [53] utilized the schemata concept to design the blend crossover for RGAs. The unimodal normal distribution crossover operator (UNDX) was developed by Ono et al. [144]. They used ellipsoidal probability distribution to generate the offspring. Kita et al. [106] presented a multi-parent UNDX (MP-UNDX), which is the extension of [144]. However, the performance of RGA with MP-UNDX is much similar to UNDX. Deep and Thakur [39] presented a Laplace crossover for RGAs, which is based on Laplacian distribution. Chuang et al. [27] developed a direction based crossover to further explore the all possible search directions. However, the search directions are limited. The heuristic normal distribution crossover operator was developed by Wang et al. [207]. It generates the cross-generated offspring for better search operation. However, the better individuals are not considered in this approach. Subbaraj et al. [192] proposed Taguchi self-adaptive RCGA. They used Taguchi method and simulated binary crossover to exploit the capable offspring.

**Mutation operators** Mutation operators generate diversity in the population. The two main challenges have to tackle during the application of mutation. First, the probability of mutation operator that was applied on population. Second, the outlier produced in chromosome after mutation process. Michalewicz [135] presented uniform and non-uniform mutation operators for RGAs. Michalewicz and Schoenauer [136]

developed a special case of uniform mutation. They developed boundary mutation. Deep and Thakur [38] presented a novel mutation operator based on power law and named as power mutation. Das and Pratihari [30] presented direction-based exponential mutation operator. They used direction information of variables. Tang and Tseng [196] presented a novel mutation operator for enhancing the performance of RCGA. Their approach was fast and reliable. However, it stuck in local optima for some applications. Deb et al. [35] developed polynomial mutation that was used in RCGA. It provides better exploration. However, the convergence speed is slow and stuck in local optima. Lucasius et al. [129] proposed a real-coded genetic algorithm (RCGA). It is simple and easy to implement. However, it suffers from local optima problem. Wang et al. [205] developed multi-offspring GA and investigated their performance over single point crossover. Wang et al. [206] stated the theoretical basis of multi-offspring GA. The performance of this method is better than non-multi-offspring GA. Pattanaik et al. [152] presented an improvement in the RCGA. Their method has

**Table 7** Mathematical formulation of genetic operators in RGAs

| Ref.  | Operator  | Mathematical Formulation  |
|-------|---|---|
| [34]  | Simulated Binary crossover                      | $p_i = \frac{1}{2}[(1-\beta)x_i + (1 + \beta)y_i]$ $q_i = \frac{1}{2}[(1 + \beta)x_i + (1-\beta)y_i]$ <p>Here, two off-springs (<i>P</i> and <i>Q</i>) are generated. <i>X</i> and <i>Y</i> are individuals. <math>\beta</math> is a variable whose value lies in the interval of <math>[0, \infty)</math></p>  |
| [53]  | Blend crossover                                 | <p>Offspring <i>P</i> is generated from parents <i>X</i> and <i>Y</i> from interval <math>[Min - ((Max - Min)\delta), Max + ((Max - Min)\delta)]</math> where <math>Min = \min(x_i, y_i)</math> and <math>Max = \max(x_i, y_i)</math>. <math>\delta</math> is a variable whose value lies in the interval of <math>[0, 1]</math></p>  |
| [135] | Arithmetic crossover<br>Geometric crossover     | <p>Arithmetic crossover</p> $p_i = \delta x_i + (1-\delta)y_i$ $q_i = \delta y_i + (1-\delta)x_i$ <p>Geometric crossover</p> $p_i = x_i^\delta \cdot y_i^{(1-\delta)}$ $q_i = y_i^\delta \cdot x_i^{(1-\delta)}$  |
| [144] | Unimodal normal distribution crossover operator | $p_i = x_p + \mu d + \sum_{k=1}^{n-1} \psi_k D e_k$ $q_i = x_p - \mu d - \sum_{k=1}^{n-1} \psi_k D e_k$ <p>where <math>e_k, k = 1, \dots, n - 1</math> are orthogonal bases that perpendicular to <math>d</math>. <math>x_p</math> is the midpoint and <math>d</math> is difference vector. <math>\mu</math> is a random value taken from normal distribution and <math>\psi_k</math> are <math>n-1</math> random values follows a normal distribution. <math>D</math> is the length from parent 3 to perpendicular line.</p> |
| [39]  | Laplace crossover                               | $p_i = x_i + \beta x_i - y_i $ $q_i = y_i + \beta x_i - y_i $ <p>Here,</p> $\beta = \begin{cases} a - b \log_e(u), & u \leq \frac{1}{2} \\ a + b \log_e(u), & u > \frac{1}{2} \end{cases}$ <p>Where <math>a</math> and <math>b</math> are variables. The default values of <math>a</math> and <math>b</math> are 0 and 1, respectively. <math>u</math> is random variable.</p>  |

better convergence speed and quality of solution. Wang et al. [208] proposed multi-offspring RCGA with direction based crossover for solving constrained problems.

Table 7 shows the mathematical formulation of genetic operators in RGAs.

## 4.2 Multiobjective GAs

Multiobjective GA (MOGA) is the modified version of simple GA. MOGA differ from GA in terms of fitness function assignment. The remaining steps are similar to GA. The main motive of multiobjective GA is to generate the optimal Pareto Front in the objective space in such a way that no further enhancement in any fitness function without disturbing the other fitness functions [123]. Convergence, diversity, and coverage are main goal of multiobjective GAs. The multiobjective GAs are broadly categorized into two categories namely, Pareto-based, and decomposition-based multiobjective GAs [52]. These techniques are discussed in the preceding subsections.

### 4.2.1 Pareto-based multi-objective GA

The concept of Pareto dominance was introduced in multiobjective GAs. Fonseca and Fleming [56] developed first multiobjective GA (MOGA). The niche and decision maker concepts were proposed to tackle the multimodal problems. However, MOGA suffers from parameter tuning problem and degree of selection pressure. Horn et al. [80] proposed a niched Pareto genetic algorithm (NPGA) that utilized the concept of tournament selection and Pareto dominance. Srinivas and Deb [191] developed a non-dominated sorting genetic algorithm (NSGA). However, it suffers from lack of elitism, need of sharing parameter, and high computation complexity. To alleviate these problems, Deb et al. [36] developed a fast elitist non-dominated sorting genetic algorithm (NSGA-II). The performance of NSGA-II may be deteriorated for many objective problems. NSGA-II was unable to maintain the diversity in Pareto-front. To alleviate this problem, Luo et al. [130] introduced a dynamic crowding distance in NSGA-II. Coello and Pulido [28] developed a multiobjective micro GA. They used an archive for storing the non-dominated solutions. The performance of Pareto-based approaches may be deteriorated in many objective problems [52].

### 4.2.2 Decomposition-based multiobjective GA

Decomposition-based MOGAs decompose the given problem into multiple subproblems. These subproblems are solved simultaneously and exchange the solutions among neighboring subproblems [52]. Ishibuchi and Murata [84] developed a multiobjective genetic local search (MOGLS). In MOGLS, the random weights were used to select the parents and local search for their offspring. They used generation replacement and roulette wheel selection method. Jaszkievicz [86] modified the MOGLS by utilizing different selection mechanisms for parents. Murata and Gen [141] proposed a cellular genetic algorithm for multiobjective optimization (C-MOGA) that was an extension of MOGA. They added cellular structure in MOGA. In C-MOGA, the selection operator was performed on the neighboring of each cell. C-MOGA was further extended by introducing an immigration procedure and known as CI-MOGA. Alves and Almeida [11] developed a multiobjective Tchebycheffs-based genetic algorithm (MOTGA) that ensures convergence and diversity. Tchebycheff scalar function was used to generate non-dominated solution set. Patel et al. [151] proposed a decomposition based MOGA (D-MOGA). They integrated opposition based

learning in D-MOGA for weight vector generation. D-MOGA is able to maintain the balance between diversity of solutions and exploration of search space.

### 4.3 Parallel GAs

The motivation behind the parallel GAs is to improve the computational time and quality of solutions through distributed individuals. Parallel GAs are categorized into three broad categories such as master-slave parallel GAs, fine grained parallel GAs, and multi-population coarse grained parallel Gas [70]. In master-slave parallel GA, the computation of fitness functions is distributed over the several processors. In fine grained GA, parallel computers are used to solve the real-life problems. The genetic operators are bounded to their neighborhood. However, the interaction is allowed among the individuals. In coarse grained GA, the exchange of individuals among sub-populations is performed. The control parameters are also transferred during migration. The main challenges in parallel GAs are to maximize memory bandwidth and arrange threads for utilizing the power of GPUs [23]. Table 8 shows the comparative analysis of parallel GAs in terms of hardware and software. The well-known parallel GAs are studied in the preceding subsections.

#### 4.3.1 Master slave parallel GA

The large number of processors are utilized in master-slave parallel GA (MS-PGA) as compared to other approaches. The computation of fitness functions may be increased by increasing the number of processors. Hong et al. [79] used MS-PGA for solving data mining problems. Fuzzy rules are used with parallel GA. The evaluation of fitness function was performed on slave machines. However, it suffers from high computational time. Sahingzo [174] implemented MS-PGA for UAV path finding problem. The genetic operators were executed on processors. They used multicore CPU with four cores. Selection and fitness evaluation was done on slave machines. MS-PGA was applied on traffic assignment problem in [127]. They used thirty processors to solve this problem at National University of Singapore. Yang et al. [213] developed a web-based parallel GA. They implemented the master slave version of NSGA-II in distributed environment. However, the system is complex in nature.

**Table 8** Analysis of parallel GAs in terms of hardware and software

| Ref.  | Hardware      | No. of processors | Language used | API                 | Application        |
|-------|---------------|-------------------|---------------|---------------------|--------------------|
| [79]  | Cluster       | 130               | JAVA          | –                   | Data Mining        |
| [174] | Multicore CPU | 8                 | JAVA          | –                   | Path Finding       |
| [127] | Cluster       | 30                | Fortran       | –                   | Road Traffic       |
| [213] | Cluster       | 48                | JavaScript    | MPI                 | Building Structure |
| [161] | Multicore CPU | 8                 | JAVA          | java.util.component | Land Planning      |
| [115] | Multicore CPU | 3                 | –             | –                   | Job Scheduling     |
| [209] | Cloud         | 300               | –             | MPI                 | Internet of Things |
| [220] | Cluster       | 100               | –             | MPI                 | Wireless Network   |
| [158] | GPU           | 448               | –             | CUDA                | Scheduling         |
| [182] | –             | 240               | –             | –                   | Nanoscience        |
| [170] | GPU           | 512               | –             | CUDA                | Electronics        |



### 4.3.2 Fine grained parallel GA

In last few decades, researchers are working on migration policies of fine grained parallel GA (FG-PGA). Porta et al. [161] utilized clock-time for migration frequency, which is independent of generations. They used non-uniform structure and static configuration. The best solution was selected for migration and worst solution was replaced with migrant solution. Kurdi [115] used adaptive migration frequency. The migration procedure starts until there is no change in the obtained solutions after ten successive generations. The non-uniform and dynamic structure was used. In [209], local best solutions were synchronized and formed a global best solutions. The global best solutions were transferred to all processors for father execution. The migration frequency depends upon the number of generation. They used uniform structure with fixed configuration. Zhang et al. [220] used parallel GA to solve the set cover problem of wireless networks. They used divide-and-conquer strategy to decompose the population into sub-populations. Thereafter, the genetic operators were applied on local solutions and Kuhn-Munkres was used to merge the local solutions.

### 4.3.3 Coarse grained parallel GA

Pinel et al. [158] proposed a GraphCell. The population was initialized with random values and one solution was initialized with Min-min heuristic technique. 448 processors were used to implement the proposed approach. However, coarse grained parallel GAs are less used due to complex in nature. The hybrid parallel GAs are widely used in various applications. Shayeghi et al. [182] proposed a pool-based Birmingham cluster GA. Master node was responsible for managing global population. Slave node selected the solutions from global population and executed it. 240 processors are used for computation. Roberge et al. [170] used hybrid approach to optimize switching angle of inverters. They used four different strategies for fitness function computation. Nowadays, GPU, cloud, and grid are most popular hardware for parallel GAs [198].

## 4.4 Chaotic GAs

The main drawback of GAs is premature convergence. The chaotic systems are incorporated into GAs to alleviate this problem. The diversity of chaos genetic algorithm removes premature convergence. Crossover and mutation operators can be replaced with chaotic maps. Tiong et al. [197] integrated the chaotic maps into GA for further improvement in accuracy. They used six different chaotic maps. The performance of Logistic, Henon and Ikeda chaotic GA performed better than the classical GA. However, these techniques suffer from high computational complexity. Ebrahimzadeh and Jampour [48] used Lorenz chaotic for genetic operators of GA to eliminate the local optima problem. However, the proposed approach was unable to find relationship between entropy and chaotic map. Javidi and Hosseinpoufard [87] utilized two chaotic maps namely logistic map and tent map for generating chaotic values instead of random selection of initial population. The proposed chaotic GA performs better than the GA. However, this method suffers from high computational complexity. Fuertes et al. [60] integrated the entropy into chaotic GA. The control parameters are modified through chaotic maps. They investigated the relationship between entropy and performance optimization.

Chaotic systems have also used in multiobjective and hybrid GAs. Abo-Elnaga and Nasr [5] integrated chaotic system into modified GA for solving Bi-level programming problems. Chaotic helps the proposed algorithm to alleviate local optima and enhance the convergence. Tahir et al. [193] presented a binary chaotic GA for feature selection in healthcare. The chaotic maps were used



to initialize the population and modified reproduction operators were applied on population. Xu et al. [115] proposed a chaotic hybrid immune GA for spectrum allocation. The proposed approach utilizes the advantages of both chaotic and immune operator. However, this method suffers from parameter initialization problem.

## 4.5 Hybrid GAs

Genetic Algorithms can be easily hybridized with other optimization methods for improving their performance such as image denoising methods, chemical reaction optimization, and many more. The main advantages of hybridized GA with other methods are better solution quality, better efficiency, guarantee of feasible solutions, and optimized control parameters [51]. It is observed from literature that the sampling capability of GAs is greatly affected from population size. To resolve this problem, local search algorithms such as memetic algorithm, Baldwinian, Lamarckian, and local search have been integrated with GAs. This integration provides proper balance between intensification and diversification. Another problem in GA is parameter setting. Finding appropriate control parameters is a tedious task. The other metaheuristic techniques can be used with GA to resolve this problem. Hybrid GAs have been used to solve the issues mentioned in the preceding subsections [29, 137, 186].

### 4.5.1 Enhance search capability

GAs have been integrated with local search algorithms to reduce the genetic drift. The explicit refinement operator was introduced in local search for producing better solutions. El-Mihoub et al. [54] established the effect of probability of local search on the population size of GA. Espinoza et al. [50] investigated the effect of local search for reducing the population size of GA. Different search algorithms have been integrated with GAs for solving real-life applications.

### 4.5.2 Generate feasible solutions

In complex and high-dimensional problems, the genetic operators of GA generate infeasible solutions. PMX crossover generates the infeasible solutions for order-based problems. The distance preserving crossover operator was developed to generate feasible solutions for travelling salesman problem [58]. The gene pooling operator instead of crossover was used to generate feasible solution for data clustering [19]. Konak and Smith [108] integrated a cut-saturation algorithm with GA for designing the communication networks. They used uniform crossover to produce feasible solutions.

### 4.5.3 Replacement of genetic operators

There is a possibility to replace the genetic operators which are mentioned in Section 3.2 with other search techniques. Leng [122] developed a guided GA that utilizes the penalties from guided local search. These penalties were used in fitness function to improve the performance of GA. Headar and Fukushima [74] used simplex crossover instead of standard crossover. The standard mutation operator was replaced with simulated annealing in [195]. The basic concepts of quantum computing are used to improve the performance of GAs. The heuristic crossover and hill-climbing operators can be integrated into GA for solving three-matching problem.

#### 4.5.4 Optimize control parameters

The control parameters of GA play a crucial role in maintaining the balance between intensification and diversification. Fuzzy logic has an ability to estimate the appropriate control parameters of GA [167]. Beside this, GA can be used to optimize the control parameters of other techniques. GAs have been used to optimize the learning rate, weights, and topology of neural networks [21]. GAs can be used to estimate the optimal value of fuzzy membership in controller. It was also used to optimize the control parameters of ACO, PSO, and other metaheuristic techniques [156]. The comparative analysis of well-known GAs are mentioned in Table 9.

## 5 Applications

Genetic Algorithms have been applied in various NP-hard problems with high accuracy rates. There are a few application areas in which GAs have been successfully applied.

### 5.1 Operation management

GA is an efficient metaheuristic for solving operation management (OM) problems such as facility layout problem (FLP), supply network design, scheduling, forecasting, and inventory control.

#### 5.1.1 Facility layout

Datta et al. [32] utilized GA for solving single row facility layout problem (SRFLP). For SRFLP, the modified crossover and mutation operators of GA produce valid solutions. They applied GA to large sized problems that consists of 60–80 instances. However, it suffers from parameter dependency problem. Sadrzadeh [173] proposed GA for multi-line FLP have multi products. The facilities were clustered using mutation and heuristic operators. The total cost obtained from the proposed GA was decreased by 7.2% as compared to the other algorithms. Wu et al. [211] implemented hierarchical GA to find out the layout of cellular manufacturing system. However, the performance of GA is greatly affected from the genetic operators. Aiello et al. [7] proposed MOGA for FLP. They used MOGA on the layout of twenty different departments. Palomo-Romero et al. [148] proposed an island model GA to solve the FLP. The proposed technique maintains the population diversity and generates better solutions than the existing techniques. However, this technique suffers from improper migration strategy that can be utilized for improving the population. GA and its variants has been successfully applied on FLP [103, 119, 133, 201].

#### 5.1.2 Scheduling

GA shows the superior performance for solving the scheduling problems such as job-shop scheduling (JSS), integrated process planning and scheduling (IPPS), etc. [119]. To improve the performance in the above-mentioned areas of scheduling, researchers developed various genetic representation [12, 159, 215], genetic operators, and hybridized GA with other methods [2, 67, 147, 219].

**Table 9** Comparative study of GA's variants in terms of pros and cons

| Reference             | Year | Pros   | Cons   | Application                                |
|-----------------------|------|--|--|--|
| <b>Real-Coded GAs</b> |      |  |  |  |
| [129]                 | 1989 | No encoding required<br>Simple   | Trapped in Local optima  | Chemo-metrics                              |
| [144]                 | 1997 | Fast Convergence   | Trapped in Local optima  | Optimization Problems                      |
| [39]                  | 2007 | Better Performance   | Less success rate  | Optimization Problems                      |
| [192]                 | 2011 | Fast convergence   | Premature convergence  | Economic dispatch                          |
| [196]                 | 2013 | Better convergence speed<br>Less computational cost  | Premature convergence for some applications                            | Optimization Problems                      |
| [35]                  | 2014 | Better search capability<br>Fast convergence   |  |  |
| [27]                  | 2016 | Better Exploration   | Stuck in Local optima  | Optimization Problems                      |
| [205]                 | 2016 | Better offspring generation<br>Fast convergence  | Slow convergence speed<br>Limited search directions                    | Optimization Problems                      |
| [207]                 | 2018 | Better population diversity  | Expensive computational cost   | Traveling Salesman Problem                 |
| [152]                 | 2018 | Guarantee cross-generated offspring<br>Fast convergence  | Better individuals not considered<br>Premature convergence             | Optimization Problems<br>Economic dispatch |
| [208]                 | 2019 | Better solution quality  | Computationally expensive  | Optimization Problems                      |
| [132]                 | 2020 | Better convergence<br>Not stuck in local optima<br>Better performance<br>Suitable for constrained search space | Tuning of crossover operator required                                  | Optimization Problems                      |
| <b>Chaotic GAs</b>    |      |  |  |  |
| [197]                 | 2012 | Superior performance over standard GA  | More computational time required                                       | Optimization Problems                      |
| [48]                  | 2013 | Improved performance due to chaotic process  | Unable to classify the chaotic map and its relationship with entropy   | Optimization Problems                      |
| [87]                  | 2015 | Better convergence<br>Enhance diversity of population<br>Avoid local optima                                    | More computational time required                                       | Optimization Problems                      |
| [60]                  | 2019 | Ability to establish relationship between chaotic map and entropy<br>Better solution quality                   | Influence of multifractals in initial population for some applications | Optimization Problems                      |
| [5]                   | 2020 | Fast convergence   | More computational time required                                       | Bi-level Programming                       |
| [193]                 | 2020 | Better performance   |  | Healthcare                                 |
| [51]                  | 2020 | Better classification accuracy<br>Better performance   | Parameter initialization   | Spectrum Allocation                        |
| [90]                  | 2018 | Fast Execution   | Unable to utilize the full power of machine                            | Text Feature Clustering                    |
| <b>Parallel GAs</b>   |      |  |  |  |

Table 9 (continued)

| Reference | Year | Pros   | Cons   | Application                 |
|-----------|------|--|--|-----------------------------|
| [10]      | 2018 | Better convergence<br>Better optimization accuracy<br>Low computational time | Require optimize instruction execution                           | Optimization Problems       |
| [63]      | 2019 | Fast convergence   | Inferior solution quality  | Community Detection         |
| [13]      | 2020 | Easy to implement<br>Faster execution  | More improvement in GPU utilization required                     | Logistics Management        |
| [23]      | 2020 | Optimize memory access<br>Optimize instruction execution                     | Need of GPU accelerated libraries                                | Railway Scheduling          |
| [1]       | 2020 | Better performance<br>Highest Speedup  | Low utilization of GPU cluster                                   | Transportation System       |
| [153]     | 1993 | Fast convergence   | Binary Coded GAs   | Molecular Recognition       |
| [203]     | 2014 | Superior Performance<br>Flexible   | More computational time required<br>Tuning of control parameters | Wind Farm Design            |
| [185]     | 2019 | Fast<br>Efficient searching capability                                       | Influence from setting of control parameters                     | Feature Selection           |
| [217]     | 2020 | Faster convergence rate<br>Better distribution                               | Hybrid GAs   | Routing                     |
| [131]     | 2020 | Better Line of Code decline rate<br>Improve performance of GA                | Parameter tuning is required for better result                   | Program Analysis            |
| [181]     | 2020 | Better accuracy score  | Slow convergence   | Accidental Death Record     |
| [85]      | 2020 | Robust<br>Efficient<br>Accurate  | Stuck in local optima<br>Premature convergence                   | Stock Market Prediction     |
| [162]     | 2020 | Improve local search capability  | Premature convergence  | Job Scheduling              |
| [68]      | 2020 | Improve local search capability<br>Better solution quality                   | Slow convergence   | Travelling Salesman Problem |
| [165]     | 2020 | Better solution quality  | Premature convergence  | Feature Selection           |
| [163]     | 2018 | Better search space  | Unable to capture quantitative information                       | Agriculture                 |
| [164]     | 2018 | Better classification accuracy   | Slow convergence   | Agriculture                 |
| [37]      | 2018 | Superior performance   | High computational time  | Function Approximation      |

### 5.1.3 Inventory control

Besides the scheduling, inventory control plays an important role in OM. Backordering and lost sales are two main approaches for inventory control [119]. Hiassat et al. [76] utilized the location-inventory model to find out the number and location of warehouses. Various design constraints have been added in the objective functions of GA and its variants for solving inventory control problem [].

### 5.1.4 Forecasting and network design

Forecasting is an important component for OM. Researchers are working on forecasting of financial trading, logistics demand, and tourist arrivals. GA has been hybridized with support vector regression, fuzzy set, and neural network (NN) to improve their forecasting capability [22, 78, 89, 178, 214]. Supply network design greatly affect the operations planning and scheduling. Most of the research articles are focused on capacity constraints of facilities [45, 184]. Multi-product multi-period problems increases the complexity of supply networks. To resolve the above-mentioned problem, GA has been hybridized with other techniques [6, 45, 55, 188, 189]. Multi-objective GAs are also used to optimize the cost, profit, carbon emissions, etc. [184, 189].

## 5.2 Multimedia

GAs have been applied in various fields of multimedia. Some of well-known multimedia fields are encryption, image processing, video processing, medical imaging, and gaming.

### 5.2.1 Information security

Due to development in multimedia applications, images, videos and audios are transferred from one place to another over Internet. It has been found in literature that the images are more error prone during the transmission. Therefore, image protection techniques such as encryption, watermarking and cryptography are required. The classical image encryption techniques require the input parameters for encryption. The wrong selection of input parameters will generate inadequate encryption results. GA and its variants have been used to select the appropriate control parameters. Kaur and Kumar [96] developed a multi-objective genetic algorithm to optimize the control parameters of chaotic map. The secret key was generated using beta chaotic map. The generated key was use to encrypt the image. Parallel GAs were also used to encrypt the image [97].

### 5.2.2 Image processing

The main image processing tasks are preprocessing, segmentation, object detection, denoising, and recognition. Image segmentation is an important step to solve the image processing problems. Decomposing/partitioning an image requires high computational time. To resolve this problem, GA is used due to their better search capability [26, 102]. Enhancement is a technique to improve the quality and contrast of an image. The better image quality is required to analyze the given image. GAs have been used to enhance natural contrast and magnify image [40, 64, 99]. Some researchers are working on hybridization of rough set with adaptive

genetic algorithm to merge the noise and color attributes. GAs have been used to remove the noise from the given image. GA can be hybridized with fuzzy logic to denoise the noisy image. GA based restoration technique can be used to remove haze, fog and smog from the given image [8, 110, 146, 200]. Object detection and recognition is a challenging issue in real-world problem. Gaussian mixture model provides better performance during detection and recognition process. The control parameters are optimized through GA [93].

### 5.2.3 Video processing

Video segmentation has been widely used in pattern recognition, and computer vision. There are some critical issues that are associated with video segmentation. These are distinguishing object from the background and determine accurate boundaries. GA can be used to resolve these issues [9, 105]. GAs have been implemented for gesture recognition successfully by Chao et al. [81] used GA for gesture recognition. They applied GAs and found an accuracy of 95% in robot vision. Kaluri and Reddy [91] proposed an adaptive genetic algorithm based method along with fuzzy classifiers for sign gesture recognition. They reported an improved recognition rate of 85% as compared to the existing method that provides 79% accuracy. Beside the gesture recognition, face recognition play an important role in criminal identification, unmanned vehicles, surveillance, and robots. GA is able to tackle the occlusion, orientations, expressions, pose, and lighting condition [69, 95, 109].

### 5.2.4 Medical imaging

Genetic algorithms have been applied in medical imaging such as edge detection in MRI and pulmonary nodules detection in CT scan images [100, 179]. In [120], authors used a template matching technique with GA for detecting nodules in CT images. Kavitha and Chellamuthu [179] used GA based region growing method for detecting the brain tumor. GAs have been applied on medical prediction problems captured from pathological subjects. Sari and Tuna [176] used GA used to solve issues arises in biomechanics. It is used to predict pathologies during examination. Ghosh and Bhattachrya [62] implemented sequential GA with cellular automata for modelling the coronavirus disease 19 (COVID-19) data. GAs can be applied in parallel mode to find rules in biological datasets [31]. The authors proposed a parallel GA that runs by dividing the process into small sub-generations and evaluating the fitness of each individual solution in parallel. Genetic algorithms are used in medicine and other related fields. Koh et al. [61] proposed a genetic algorithm based method for evaluation of adverse effects of a given drug.

### 5.2.5 Precision agriculture

GAs have been applied on various problems that are related to precision agriculture. The main issues are crop yield, weed detection, and improvement in farming equipment. Pachepsky and Acock [145] implemented GA to analyze the water capacity in soil using remote sensing images. The crop yield can be predicted through the capacity of water present in soil. The weed identification was done through GA in [142]. They used aerial image for classification of plants. In [124], color image segmentation was used to discriminate the weed and plant. Peerlink et al. [154] determined the appropriate rate of fertilizer for various portions of agriculture field. They GA for determining the nitrogen in wheat field. The energy

requirements in water irrigation systems can be optimized by viewing it as a multi-objective optimization problem. The amount of irrigation required and thus power requirements change continuously in a SMART farm. Therefore, GA can be applied in irrigation systems to reduce the power requirements [33].

### 5.2.6 Gaming

GAs have been successfully used in games such as gomoku. In [202], the authors shown that the GA based approach finds the solution having the highest fitness than the normal tree based methods. However, in real-time strategy based games, GA based solutions become less practical to implement [82]. GAs have been implemented for path planning problems considering the environment constraints as well as avoiding the obstacles to reach the given destination. Burchardt and Salomon [18] described an implementation for path planning for soccer games. GA can encode the path planning problems via the coordinate points of a two-dimensional playing field, hence resulting in a variable length solution. The fitness function in path planning considers length of path as well as the collision avoiding terms for soccer players.

## 5.3 Wireless networking

Due to adaptive, scalable, and easy implementation of GA, it has been used to solve the various issues of wireless networking. The main issues of wireless networking are routing, quality of service, load balancing, localization, bandwidth allocation and channel assignment [128, 134]. GA has been hybridized with other metaheuristics for solving the routing problems. Hybrid GA not only producing the efficient routes among pair of nodes, but also used for load balancing [24, 212].

### 5.3.1 Load balancing

Nowadays, multimedia applications require Quality-of-Service (QoS) demand for delay and bandwidth. Various researchers are working on GAs for QoS based solutions. GA produces optimal solutions for complex networks [49]. Roy et al. [172] proposed a multi-objective GA for multicast QoS routing problem. GA was used with ACO and other search algorithms for finding optimal routes with desired QoS metrics. Load balancing is another issue in wireless networks. Scully and Brown [177] used MicroGAs and MacroGAs to distribute the load among various components of networks. He et al. [73] implemented GA to determine the balance load in wireless sensor networks. Cheng et al. [25] utilized distributed GA with multi-population scheme for load balancing. They used load balancing metric as a fitness function in GA.

### 5.3.2 Localization

The process of determining the location of wireless nodes is called as localization. It plays an important role in disaster management and military services. Yun et al. [216] used GA with fuzzy logic to find out the weights, which are assigned according to the signal strength. Zhang et al. [218] hybridized GA with simulated annealing (SA) to determine the position of wireless nodes. SA is used as local search to eliminate the premature convergence.

### 5.3.3 Bandwidth and channel allocation

The appropriate bandwidth allocation is a complex task. GAs and its variants have been developed to solve the bandwidth allocation problem [92, 94, 107]. GAs were used to investigate the allocation of bandwidth with QoS constraints. The fitness function of GAs may consists of resource utilization, bandwidth distribution, and computation time [168]. The channel allocation is an important issue in wireless networks. The main objective of channel allocation is to simultaneously optimize the number of channels and reuse of allocated frequency. Friend et al. [59] used distributed island GA to resolve the channel allocation problem in cognitive radio networks. Zhenhua et al. [221] implemented a modified immune GA for channel assignment. They used different encoding scheme and immune operators. Pinagapany and Kulkarni [157] developed a parallel GA to solve both static and dynamic channel allocation problem. They used decimal encoding scheme. Table 10 summarizes the applications of GA and its variants.

## 6 Challenges and future possibilities

In this section, the main challenges faced during the implementation of GAs are discussed followed by the possible research directions.

### 6.1 Challenges

Despite the several advantages, there are some challenges that need to be resolved for future advancements and further evolution of genetic algorithms. Some major challenges are given below:

#### 6.1.1 Selection of initial population

Initial population is always considered as an important factor for the performance of genetic algorithms. The size of population also affects the quality of solution [160]. The researchers argue that if a large population is considered, then the algorithm takes more computation time. However, the small population may lead to poor solution [155]. Therefore, finding the appropriate population size is always a challenging issue. Harik and Lobo [71] investigated the population using self-adaption method. They used two approaches such as (1) use of self-adaption prior to execution of algorithm, in which the size of population remains the same and (2) in which the self-adaption used during the algorithm execution where the population size is affected by fitness function.

#### 6.1.2 Premature convergence

Premature convergence is a common issue for GA. It can lead to the loss of alleles that makes it difficult to identify a gene [15]. Premature convergence states that the result will be suboptimal if the optimization problem coincides too early. To avoid this issue, some researchers suggested that the diversity should be used. The selection pressure should be used to increase the diversity. Selection pressure is a degree which favors the better individuals in the initial population of GA's. If selection pressure (SP1) is greater than some selection



**Table 10** Applications of GA

| Broad Area           | Sub-domain               | Target Problems   | Variants of GA   | Ref.   |  |
|----------------------|--------------------------|---|--|--|--|
| Operation Management | Facility layout Design   | Static facility layout problem<br>Dynamic facility layout problem<br>Flexible bay structure | GA, MOGA, Parallel GA, Hierarchical GA                                   | [7, 32, 103, 119, 133, 148, 173, 201, 211]                   |  |
|                      | Supply network design    | Multi-product, multi-period<br>Multi-product, single-period                                 | GA, NSGA-II, GA + PSO, MOGA, GA + Fuzzy                                  | [6, 45, 55, 184, 188, 189]                                   |  |
|                      | Scheduling               | Single-product, single-period<br>Vehicle routing<br>Resource sharing and scheduling         | GA, GA + BB, GA + ABC, GA + Local search, MOGA, NSGA-II, Hierarchical GA | [A132–138]   |  |
|                      | Forecasting              | Machine scheduling<br>Airline flight scheduling<br>Financial trading<br>Tourism demand      | GA, Chaotic GA, Adaptive GA, GA + NN                                     | [22, 78, 89, 178, 214]                                       |  |
|                      | Inventory control        | Healthcare demand<br>Inventory-routing<br>Lot sizing  | GA, NSGA-II  | [14, 76, 150]  |  |
|                      | Multimedia               | Information Security  | Location-inventory routing<br>Encryption<br>Watermarking                 | GA, Parallel GA, NSGA-II, NSGA                               | [96–98, 114]                                     |
|                      |                          | Image Processing  | Segmentation<br>Enhancement<br>Object detection<br>De-noising            | GA, NSGA-II, Parallel GA, Hybrid GA, Adaptive GA, Chaotic GA | [8, 26, 40, 64, 93, 99, 102, 110, 146, 200, 204] |
|                      |                          | Video Processing  | Recognition<br>Video segmentation<br>Gesture recognition                 | GA, NSGA, Adaptive GA, Hybrid GA                             | [9, 69, 81, 91, 95, 100, 104, 105, 109]          |
|                      |                          | Medical Imaging   | Face recognition<br>Tumor diagnosis<br>COVID-19 diagnosis                | GA, Hybrid GA, Parallel GA, Sequential GA                    | [31, 61, 62, 120, 176, 179]                      |
|                      |                          | Precision Agriculture   | Bioinformatics<br>Weed detection<br>Crop management                      | GA, Hybrid GA, NSGA  | [16, 33, 124, 142, 145, 154]                     |
| Wireless Networking  | Gaming                   | Water irrigation<br>Google Chrome dinosaur<br>Chess   | GA, Coevolutionary GA, NSGA  | [18, 82, 202]  |  |
|                      | Wireless mesh networks   | Strategic games   | GA, Sequential GA, MOGA  | [24, 128, 212]   |  |
|                      |                          | Mobile Ad-hoc networks  | MOGA, GA + Fuzzy Logic, NSGA, GA + ACO, NSGA-II                          | [4, 44, 49, 143, 172]  |  |
|                      |                          | Wireless sensor networks  | MicroGA, MacroGA, Distributed GA   | [25, 73, 177]  |  |
|                      | Wireless sensor networks | Load balancing  | MicroGA, GA + SA, GA + Fuzzy Logic                                       | [194, 216, 218]  |  |
|                      |                          | Localization  | GA, Distributed GA, GA + Local search, GA + Greedy Algorithms, MOGA      | [92, 94, 107, 168]   |  |
|                      |                          | Bandwidth allocation  | MOGA, Parallel GA, Distributed Island GA                                 | [59, 157, 221]   |  |
|                      |                          | Channel assignment  |  |  |  |

pressure (SP2), then population using SP1 should be larger than the population using SP2. The higher selection pressure can decrease the population diversity that may lead to premature convergence [71].

Convergence property has to be handled properly so that the algorithm finds global optimal solution instead of local optimal solution (see Fig. 8). If the optimal solution lies in the vicinity of an infeasible solution, then the global nature of GA can be combined with local nature of other algorithms such as Tabu search and local search. The global nature of genetic algorithms and local nature of Tabu search provide the proper balance between intensification and diversification.

### 6.1.3 Selection of efficient fitness functions

Fitness function is the driving force, which plays an important role in selecting the fittest individual in every iteration of an algorithm. If the number of iterations are small, then a costly fitness function can be adjusted. The number of iterations increases may increase the computational cost. The selection of fitness function depends upon the computational cost as well as their suitability. In [46], the authors used Davies-Bouldin index for classification of documents.

### 6.1.4 Degree of mutation and crossover

Crossover and mutation operators are the integral part of GAs. If the mutation is not considered during evolution, then there will be no new information available for evolution. If crossover is not considered during evolution, then the algorithm can result in local optima. The degree of these operators greatly affect the performance of GAs [72]. The proper balance between these operators are required to ensure the global optima. The probabilistic nature cannot determine the exact degree for an effective and optimal solution.

### 6.1.5 Selection of encoding schemes

GAs require a particular encoding scheme for a specific problem. There is no general methodology for deciding whether the particular encoding scheme is suitable for any type of

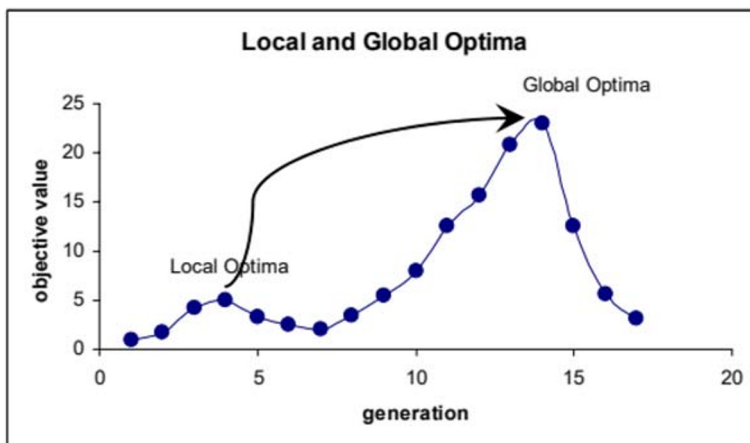


Fig. 8 Local and global optima [149]

real-life problem. If there are two different problems, then two different encoding schemes are required. Ronald [171] suggested that the encoding schemes should be designed to overwhelm the redundant forms. The genetic operators should be implemented in a manner that they are not biased towards the redundant forms.

## 6.2 Future research directions

GAs have been applied in different fields by modifying the basic structure of GA. The optimality of a solution obtained from GA can be made better by overcoming the current challenges. Some future possibilities for GA are as follows:

- 1) There should be some way to choose the appropriate degree of crossover and mutation operators. For example Self-Organizing GA adapt the crossover and mutation operators according to the given problem. It can save computation time that make it faster.
- 2) Future work can also be considered for reducing premature convergence problem. Some researchers are working in this direction. However, it is suggested that new methods of crossover and mutation techniques are required to tackle the premature convergence problem.
- 3) Genetic algorithms mimic the natural evolution process. There can be a possible scope for simulating the natural evolution process such as the responses of human immune system and the mutations in viruses.
- 4) In real-life problems, the mapping from genotype to phenotype is complex. In this situation, the problem has no obvious building blocks or building blocks are not adjacent groups of genes. Hence, there is a possibility to develop novel encoding schemes to different problems that does not exhibit same degree of difficulty.

## 7 Conclusions

This paper presents the structured and explained view of genetic algorithms. GA and its variants have been discussed with application. Application specific genetic operators are discussed. Some genetic operators are designed for representation. However, they are not applicable to research domains. The role of genetic operators such as crossover, mutation, and selection in alleviating the premature convergence is studied extensively. The applicability of GA and its variants in various research domain has been discussed. Multimedia and wireless network applications were the main attention of this paper. The challenges and issues mentioned in this paper will help the practitioners to carry out their research. There are many advantages of using GAs in other research domains and metaheuristic algorithms.

The intention of this paper is not only provide the source of recent research in GAs, but also provide the information about each component of GA. It will encourage the researchers to understand the fundamentals of GA and use the knowledge in their research problems.

## References

1. Abbasi M, Rafiee M, Khosravi MR, Jolfaei A, Menon VG, Koushyar JM (2020) An efficient parallel genetic algorithm solution for vehicle routing problem in cloud implementation of the intelligent transportation systems. *Journal of cloud Computing* 9(6)

2. Abdelghany A, Abdelghany K, Azadian F (2017) Airline flight schedule planning under competition. *Comput Oper Res* 87:20–39
3. Abdulal W, Ramachandram S (2011) Reliability-aware genetic scheduling algorithm in grid environment. *International Conference on Communication Systems and Network Technologies*, Katra, Jammu, pp 673–677
4. Abdullah J (2010) Multiobjectives ga-based QoS routing protocol for mobile ad hoc network. *Int J Grid Distrib Comput* 3(4):57–68
5. Abo-Elnaga Y, Nasr S (2020) Modified evolutionary algorithm and chaotic search for Bilevel programming problems. *Symmetry* 12:767
6. Afrouzy ZA, Nasser SH, Mahdavi I (2016) A genetic algorithm for supply chain configuration with new product development. *Comput Ind Eng* 101:440–454
7. Aiello G, Scalia G (2012) La, Enea M. A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding *Expert Syst Appl* 39(12):10352–10358
8. Alaoui A, Adamou-Mitiche ABH, Mitiche L (2020) Effective hybrid genetic algorithm for removing salt and pepper noise. *IET Image Process* 14(2):289–296
9. Alkhafaji BJ, Salih MA, Nabat ZM, Shnain SA (2020) Segmenting video frame images using genetic algorithms. *Periodicals of Engineering and Natural Sciences* 8(2):1106–1114
10. Al-Oqaily AT, Shakah G (2018) Solving non-linear optimization problems using parallel genetic algorithm. *International Conference on Computer Science and Information Technology (CSIT)*, Amman, pp. 103–106
11. Alvesa MJ, Almeida M (2007) MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Comput Oper Res* 34:3458–3470
12. Arakaki RK, Usberti FL (2018) Hybrid genetic algorithm for the open capacitated arc routing problem. *Comput Oper Res* 90:221–231
13. Arkhipov DI, Wu D, Wu T, Regan AC (2020) A parallel genetic algorithm framework for transportation planning and logistics management. *IEEE Access* 8:106506–106515
14. Azadeh A, Elahi S, Farahani MH, Nasirian B (2017) A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Comput Ind Eng* 104:124–133
15. Baker JE, Grefenstette J (2014) *Proceedings of the first international conference on genetic algorithms and their applications*. Taylor and Francis, Hoboken, pp 101–105
16. Bolboca SD, JAntschli L, Balan MC, Diudea MV, Sestras RE (2010) State of art in genetic algorithms for agricultural systems. *Not Bot Hort Agrobot Cluj* 38(3):51–63
17. Bonabeau E, Dorigo M, Theraulaz G (1999) *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Inc
18. Burchardt H, Salomon R (2006) Implementation of path planning using genetic algorithms on Mobile robots. *IEEE International Conference on Evolutionary Computation*, Vancouver, BC, pp 1831–1836
19. Burdshall B, Giraud-Carrier C (1997) Evolving fuzzy prototypes for efficient data clustering," in second international ICSC symposium on fuzzy logic and applications. Zurich, Switzerland, pp. 217–223.
20. Burkowski FJ (1999) Shuffle crossover and mutual information. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406), Washington, DC, USA, 1999, pp. 1574–1580
21. Chaiyaratana N, Zalzal AM (2000) "Hybridisation of neural networks and a genetic algorithm for friction compensation," in the 2000 congress on evolutionary computation, vol 1. San Diego, USA, pp 22–29
22. Chen R, Liang C-Y, Hong W-C, Gu D-X (2015) Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. *Appl Soft Comput* 26:434–443
23. J.R. Cheng and M. Gen (2020) Parallel genetic algorithms with GPU computing. *Impact on Intelligent Logistics and Manufacturing*.
24. Cheng H, Yang S (2010) Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. *Applications of evolutionary computation*. Springer, In, pp 562–571
25. Cheng H, Yang S, Cao J (2013) Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc net-works. *Expert Syst Appl* 40(4):1381–1392
26. Chouhan SS, Kaul A, Singh UP (2018) Soft computing approaches for image segmentation: a survey. *Multimed Tools Appl* 77(21):28483–28537
27. Chuang YC, Chen CT, Hwang C (2016) A simple and efficient real-coded genetic algorithm for constrained optimization. *Appl Soft Comput* 38:87–105
28. Coello CAC, Pulido GT (2001) A micro-genetic algorithm for multiobjective optimization. In: *EMO*, volume 1993 of lecture notes in computer science, pp 126–140. Springer
29. Das, K. N. (2014). Hybrid genetic algorithm: an optimization tool. In *global trends in intelligent computing Research and Development* (pp. 268-305). IGI global.

30. Das AK, Pratihari DK (2018) A direction-based exponential mutation operator for real-coded genetic algorithm. *IEEE International Conference on Emerging Applications of Information Technology*.
31. Dash SR, Dehuri S, Rayaguru S (2013) Discovering interesting rules from biological data using parallel genetic algorithm, 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad,, pp. 631–636.
32. Datta D, Amaral ARS, Figueira JR (2011) Single row facility layout problem using a permutation-based genetic algorithm. *European J Oper Res* 213(2):388–394
33. de Ocampo ALP, Dadios EP (2017) "Energy cost optimization in irrigation system of smart farm by using genetic algorithm," 2017IEEE 9th international conference on humanoid. Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, pp 1–7
34. Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. *Complex Systems* 9: 115–148
35. Deb K, Deb D (2014) Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing* 4(1):1–28
36. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
37. Deep K, Das KN (2008) Quadratic approximation based hybrid genetic algorithm for function optimization. *Appl Math Comput* 203(1):86–98
38. Deep K, Thakur M (2007) A new mutation operator for real coded genetic algorithms. *Appl Math Comput* 193:211–230
39. Deep K, Thakur M (2007) A new crossover operator for real coded genetic algorithms. *Appl Math Comput* 188:895–911
40. Dhal KP, Ray S, Das A, Das S (2018) A survey on nature-inspired optimization algorithms and their application in image enhancement domain. *Archives of Computational Methods in Engineering* 5:1607–1638
41. Dhiman G, Kumar V (2017) Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70
42. Dhiman G, Kumar V (2018) Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowl-Based Syst* 159:20–50
43. Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl-Based Syst* 165:169–196
44. Di Fatta G, Hoffmann F, Lo Re G, Urso A (2003) A genetic algorithm for the design of a fuzzy controller for active queue management. *IEEE Trans Syst Man Cybern Part C Appl Rev* 33(3):313–324
45. Diabat A, Deskoors R (2016) A hybrid genetic algorithm based heuristic for an integrated supply chain problem. *J Manuf Syst* 38:172–180
46. Diaz-Manriquez A, Ríos-Alvarado AB, Barrón-Zambrano JH, Guerrero-Melendez TY, Elizondo-Leal JC (2018) An automatic document classifier system based on genetic algorithm and taxonomy. *IEEE Access* 6:21552–21559. <https://doi.org/10.1109/ACCESS.2018.2815992>
47. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization - artificial ants as a computational intelligence technique. *IEEE Comput Intell Mag* 1(2006):28–39
48. Ebrahimpour R, Jampour M (2013) Chaotic genetic algorithm based on Lorenz chaotic system for optimization problems. *I.J. Intelligent Systems and Applications Intelligent Systems and Applications* 05(05):19–24
49. EkbataniFard GH, Monsefi R, Akbarzadeh-T M-R, Yaghmaee M et al. (2010) A multi-objective genetic algorithm based approach for energy efficient qos-routing in two-tiered wireless sensor net-works. In: *wireless pervasive computing (ISWPC), 2010 5th IEEE international symposium on*. IEEE, pp 80–85
50. El-Mihoub T, Hopgood A, Nolle L, Battersby A (2004) Performance of hybrid genetic algorithms incorporating local search. In: Horton G (ed) 18th European simulation multi-conference (ESM2004). Germany, Magdeburg, pp 154–160
51. El-Mihoub TA, Hopgood AA, Lars N, Battersby A (2006) Hybrid genetic algorithms: A review. *Eng Lett* 13:2
52. Emmerich MTM, Deutz AH (2018) A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Nat Comput* 17(3):585–609
53. Eshelman LJ, Caruana RA, Schaffer JD (1997) Biases in the crossover landscape.
54. Espinoza FB, Minsker B, Goldberg D (2003) Performance evaluation and population size reduction for self adaptive hybrid genetic algorithm (SAHGA), in the Genetic and Evolutionary Computation Conference, vol. 2723, Lecture Notes in Computer Science San Francisco, USA: Springer, pp. 922–933.
55. Farahani RZ, Elahipannah M (2008) A genetic algorithm to optimize the total cost and service level for just-in-time distribution in a supply chain. *Int J Prod Econ* 111(2):229–243

56. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: ICGA, pp 416–423. Morgan Kaufmann
57. Fox B, McMahon M (1991) Genetic operators for sequencing problems, in Foundations of Genetic Algorithms, G. Rawlins, Ed. Morgan Kaufmann Publishers, San Mateo, CA, Ed. 1991, pp. 284–300.
58. Freisleben B, Merz P (1996) New genetic local search operators for the traveling salesman problem," in the Fourth Conference on Parallel Problem Solving from Nature vol. 1141, Lectures Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, pp. 890–899.
59. Friend DH, El Nainay, M, Shi Y, MacKenzie AB (2008) Architecture and performance of an island genetic algorithm-based cognitive network. In: Consumer communications and networking conference, 2008. CCNC 2008. 5th IEEE. IEEE, pp 993–997
60. Fuertes G, Vargas M, Alfaro M, Soto-Garrido R, Sabattin J, Peralta M-A (2019) Chaotic genetic algorithm and the effects of entropy in performance optimization.
61. Ghaheeri A, Shoar S, Naderan M, Hoseini SS (2015) The applications of genetic algorithms in medicine. *CJ* 30:406–416
62. Ghosh S, Bhattacharya S (2020) A data-driven understanding of COVID-19 dynamics using sequential genetic algorithm based probabilistic cellular automata. *Applied Soft Computing*. 96
63. Ghoshal AK, Das N, Bhattacharjee S, Chakraborty G (2019) A fast parallel genetic algorithm based approach for community detection in large networks. International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, pp. 95–101.
64. Gogna A, Tayal A (2012) Comparative analysis of evolutionary algorithms for image enhancement. *Int J Met* 2(1)
65. Goldberg D (1989) Genetic algorithm in search. Optimization and Machine Learning, Addison -Wesley, Reading, MA 1989
66. Goldberg D, Lingle R (1985) Alleles, loci and the traveling salesman problem. In: Proceedings of the 1st international conference on genetic algorithms and their applications, vol. 1985. Los Angeles, USA, pp 154–159
67. Guido R, Conforti D (2017) A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Comput Oper Res* 87:270–282
68. Ha QM, Deville Y, Pham QD, Ha MH (2020) A hybrid genetic algorithm for the traveling salesman problem with drone. *J Heuristics* 26:219–247
69. HajiRassouliha A, Gamage TPB, Parker MD, Nash MP, Taberner AJ, Nielsen, PM (2013) FPGA implementation of 2D cross-correlation for real-time 3D tracking of deformable surfaces. In Proceedings of the 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), Wellington, New Zealand, 27–29 November 2013; IEEE: Piscataway, NJ, USA; pp. 352–357
70. Harada T, Alba E (2020) Parallel genetic algorithms: a useful survey. *ACM Computing Survey* 53(4):1–39
71. Harik GR, Lobo FG (1999) A parameter-less genetic algorithm, in Proceedings of the Genetic and Evolutionary Computation Conference, pp. 258–265.
72. Hassanat A, Almohammadi K, Alkafaween E, Abunawas E, Hammouri A, Prasath VBS (December 2019) Choosing mutation and crossover ratios for genetic algorithms—A review with a new dynamic approach. *Information* 10:390. <https://doi.org/10.3390/info10120390>
73. He J, Ji S, Yan M, Pan Y, Li Y (2012) Load-balanced CDS construction in wireless sensor networks via genetic algorithm. *Int J Sens Netw* 11(3):166–178
74. Hedar A, Fukushima M (2003) Simplex coding genetic algorithm for the global optimization of nonlinear functions, in Multi-Objective Programming and Goal Programming, Advances in Soft Computing, T. Tanino, T. Tanaka, and M. Inuiguchi, Eds.: Springer-Verlag, pp. 135–140.
75. Helal MHS, Fan C, Liu D, Yuan S (2017) Peer-to-peer based parallel genetic algorithm. International Conference on Information, Communication and Engineering (ICICE), Xiamen, pp 535–538
76. Hiassat A, Diabat A, Rahwan I (2017) A genetic algorithm approach for location-inventory-routing problem with perishable products. *J Manuf Syst* 42:93–103
77. Holland JH (1975) Adaptation in natural and artificial systems. The U. of Michigan Press
78. Hong W-C, Dong Y, Chen L-Y, Wei S-Y (2011) SVR with hybrid chaotic genetic algorithms for tourism demand forecasting. *Appl Soft Comput* 11(2):1881–1890
79. Hong T-P, Lee Y-C, Min-Thai W (2014) An effective parallel approach for genetic-fuzzy data mining. *Exp Syst Applic* 41(2):655–662
80. Horn J, Nafpliotis N, Goldberg DE. (1994) A niched Pareto genetic algorithm for multiobjective optimization. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, vol. 1, Piscataway, NJ: IEEE Service Center, p. 67–72.



81. Hu C, Wang X, Mandal MK, Meng M, Li D (2003) Efficient face and gesture recognition techniques for robot control. Department of Electrical and Computer Engineering University of Alberta, Edmonton, AB, T6G 2V4, Canada. CCECE2003 - CCGEI 2003, Montreal, May/mai 2003 IEEE, pp 1757–1762.
82. Peng Huo, Simon C. K. Shiu, Haibo Wang, Ben Niu (2009) Application and Comparison of Particle Swarm Optimization and Genetic Algorithm in Strategy Defense Game. Fifth International Conference on Natural Computation, pp 387–392.
83. Hussain A, Muhammad YS, Nauman Sajid M, Hussain I, Mohamd Shoukry A, Gani S (2017) Genetic algorithm for traveling salesman problem with modified cycle crossover operator. *Computational intelligence and neuroscience* 2017:1–7
84. Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans Syst Man Cybern Part C Appl Rev* 28(3):392–403
85. Jafari A, Khalili T, Babaei E, Bidram A (2020) Hybrid optimization technique using exchange market and GA. *IEEE Access* 8:2417–2427
86. Jaskiewicz A (February 2002) Genetic local search for multi-objective combinatorial optimization. *Eur J Oper Res* 137(1):50–71
87. Javidi M, Hosseinpourfard R (2015) Chaos genetic algorithm instead genetic algorithm. *Int J Inf Tech* 12(2):163–168
88. Jebari K (2013) Selection methods for genetic algorithms. Abdelmalek Essaâdi University. *International Journal of Emerging Sciences* 3(4):333–344
89. Jiang S, Chin K-S, Wang L, Qu G, Tsui KL (2017) Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. *Expert Syst Appl* 82:216–230
90. Jiang M, Fan X, Pei Z, Zhang Z (2018) Research on text feature clustering based on improved parallel genetic algorithm. Tenth International Conference on Advanced Computational Intelligence (ICACI), Xiamen, pp. 235–238
91. Kaluri R, Reddy P (2016) Sign gesture recognition using modified region growing algorithm and adaptive genetic fuzzy classifier. *International Journal of Intelligent Engineering and Systems* 9(4):225–233
92. Kandavanam G, Botvich D, Balasubramaniam S, Jennings B (2010) A hybrid genetic algorithm/variable neighborhood search approach to maximizing residual bandwidth of links for route planning. *Artificial evolution*. Springer, In, pp 49–60
93. Kannan S (2020) Intelligent object recognition in underwater images using evolutionary-based Gaussian mixture model and shape matching. *SIViP* 14:877–885
94. Karabudak D, Hung C-C, Bing B (2004) A call admission control scheme using genetic algorithms. In: *Proceedings of the 2004ACM symposium on applied computing*. ACM, pp 1151–1158
95. Katz P, Aron M, Alfalou A (2001) A face-tracking system to detect falls in the elderly; SPIE newsroom. SPIE, Bellingham, WA, USA, p 201
96. Kaur M, Kumar V (2018) Beta chaotic map based image encryption using genetic algorithm. *Int J Bifurcation Chaos* 28(11):1850132
97. Kaur M, Kumar V (2018) Parallel non-dominated sorting genetic algorithm-II-based image encryption technique. *The Imaging Science Journal*. 66(8):453–462
98. Kaur M, Kumar V (2018) Fourier–Mellin moment-based intertwining map for image encryption. *Modern Physics Letters B* 32(9):1850115
99. Kaur G, Bhardwaj N, Singh PK (2018) An analytic review on image enhancement techniques based on soft computing approach. *Sensors and Image Processing, Advances in Intelligent Systems and Computing* 651:255–266
100. Kavitha AR, Chellamuthu C (2016) Brain tumour segmentation from MRI image using genetic algorithm with fuzzy initialisation and seeded modified region growing (GFSMRG) method. *The Imaging Science Journal* 64(5):285–297
101. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks (1995)*, pp 1942–1948
102. Khan, A., ur Rehman, Z., Jaffar, M.A., Ullah, J., Din, A., Ali, A., Ullah, N. (2019) Color image segmentation using genetic algorithm with aggregation-based clustering validity index (CVI). *SIViP* 13(5), 833–841
103. Kia R, Khaksar-Haghani F, Javadian N, Tavakkoli-Moghaddam R (2014) Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. *J Manuf Syst* 33(1):218–232
104. Kim EY, Jung K (2006) Genetic algorithms for video segmentation. *Pattern Recogn* 38(1):59–73
105. Kim EY, Park SH (2006) Automatic video segmentation using genetic algorithms. *Pattern Recogn Lett* 27(11):1252–1265

106. Kita H, Ono I, Kobayashi S (1999). The multi-parent unimodal normal distribution crossover for real-coded genetic algorithms. Proceedings of the 1999 Congress on Evolutionary Computation, vol. 2, IEEE (1999), pp. 1588–1595
107. Kobayashi H, Munetomo M, Akama K, Sato Y (2004) Designing a distributed algorithm for bandwidth allocation with a genetic algorithm. *Syst Comput Jpn* 35(3):37–45
108. Konak A, Smith AE (1999) A hybrid genetic algorithm approach for backbone design of communication networks, in the 1999 Congress on Evolutionary Computation. Washington D.C, USA: IEEE, pp. 1817–1823.
109. Kortil Y, Jridi M, Falou AA, Atri M (2020) Face recognition systems: A survey. *Sensors*. 20:1–34
110. Krishnan N, Muthukumar S, Ravi S, Shashikala D, Pasupathi P (2013) Image restoration by using evolutionary technique to Denoise Gaussian and impulse noise. In: Prasath R., Kathirvalvakumar T. (eds) mining intelligence and knowledge exploration. Lecture notes in computer science, vol 8284. Springer, Cham.
111. Kumar A (2013) Encoding schemes in genetic algorithm. *Int J Adv Res IT Eng* 2(3):1–7
112. Kumar V, Kumar D (2017) An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems. *Adv Eng Softw* 112:231–254
113. Kumar V, Chhabra JK, Kumar D (2014) Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. *J Comput Sci* 5(2):144–155
114. Kumar C, Singh AK, Kumar P (2017) A recent survey on image watermarking techniques and its application in e-governance. *MultiMed Tools Appl*.
115. Kurdi M (2016) An effective new island model genetic algorithm for job shop scheduling problem. *Comput Oper Res* 67(2016):132–142
116. Larranaga P, Kuijpers CMH, Murga RH, Yurramendi Y (July 1996) Learning Bayesian network structures by searching for the best ordering with genetic algorithms. in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 26(4):487–493
117. Larranaga P, Kuijpers C, Murga R, Inza I, Dizdarevic S (1999) Genetic algorithms for the travelling salesman problem: a review of representations and operators. *Artificial Intelligence Review* 13:129–170
118. Chang-Yong Lee (2003) Entropy-Boltzmann selection in the genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 1, pp. 138–149, Feb. 2003.
119. Lee CKH (2018) A review of applications of genetic algorithms in operations management. *Eng Appl Artif Intell* 76:1–12
120. Lee Y, Hara T, Fujita H, Itoh S, Ishigaki T (July 2001) Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique. in *IEEE Transactions on Medical Imaging* 20(7):595–604
121. Joon-Yong Lee, Min-Soeng Kim, Cheol-Taek Kim and Ju-Jang Lee (2007) Study on encoding schemes in compact genetic algorithm for the continuous numerical problems, *SICE Annual Conference 2007*, Takamatsu, pp. 2694–2699.
122. Leng LT (1999) Guided genetic algorithm. University of Essex, Doctoral Dissertation
123. Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: A survey. *ACM Computing surveys*
124. Lie Tang L (2000) Tian and Brian L steward, "color image segmentation with genetic algorithm for in-field weed sensing". *Transactions of the ASAE* 43(4):1019–1027
125. Lima S.J.A., de Araújo S.A. (2018) A new binary encoding scheme in genetic algorithm for solving the capacitated vehicle routing problem. In: Korošec P., Melab N., Talbi EG. (eds) *Bioinspired Optimization Methods and Their Applications*. BIOMA 2018. Lecture notes in computer science, vol 10835. Springer, Cham
126. Liu D (2019) Mathematical modeling analysis of genetic algorithms under schema theorem. *Journal of Computational Methods in Sciences and Engineering* 19:S131–S137
127. Liu Z, Meng Q, Wang S (2013) Speed-based toll design for cordon-based congestion pricing scheme. *Transport Res Part C: Emerg Technol* 31(2013):83–98
128. Lorenzo B, Glisic S (2013) Optimal routing and traffic scheduling for multihop cellular networks using genetic algorithm. *IEEE Trans Mob Comput* 12(11):2274–2288
129. Lucasius CB, Kateman G (1989) Applications of genetic algorithms in chemometrics. In: Proceedings of the 3rd international conference on genetic algorithms. Morgan Kaufmann, Los Altos, CA, USA, pp 170–176
130. Luo B, Jinhua Zheng, Jiongliang Xie, Jun Wu. Dynamic crowding distance – a new diversity maintenance strategy for MOEAs. *ICNC '08, Fourth Int. Conf. on Natural Comp.*, vol. 1 (2008), pp. 580–585
131. Maghawry A, Kholief M, Omar Y, Hodhod R (2020) An approach for evolving transformation sequences using hybrid genetic algorithms. *Int J Intell Syst* 13(1):223–233



132. Manzoni L, Mariot L, Tuba E (2020) Balanced crossover operators in genetic algorithms. *Swarm and Evolutionary Computation* 54:100646
133. Mazinani M, Abedzadeh M, Mohebbi N (2013) Dynamic facility layout problem based on flexible bay structure and solving by genetic algorithm. *Int J Adv Manuf Technol* 65(5–8):929–943
134. Mehboob U, Qadir J, Ali S, Vasilakos A (2016) Genetic algorithms in wireless networking: techniques, applications, and issues. *Soft Comput* 20:2467–2501
135. Michalewicz Z (1992) *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, New York
136. Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. *Evol Comput* 4(1):1–32
137. Mishra R, Das KN (2017). A novel hybrid genetic algorithm for unconstrained and constrained function optimization. In *bio-inspired computing for information retrieval applications* (pp. 230–268). IGI global
138. Moher D, Liberati A, Tetzlaff J, Altman DG, The PRISMA Group (2009) Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS Med* 6(7):e1000097
139. Mooi S, Lim S, Sultan M, Bakar A, Sulaiman M, Mustapha A, Leong KY (2017) Crossover and mutation operators of genetic algorithms. *International Journal of Machine Learning and Computing* 7:9–12
140. Mudaliar DN, Modi NK (2013) Unraveling travelling salesman problem by genetic algorithm using m-crossover operator. *International Conference on Signal Processing, Image Processing & Pattern Recognition, Coimbatore*, pp 127–130
141. T. Murata and M. Gen (2000) Cellular genetic algorithm for multi-objective optimization, in *Proceedings of the Fourth Asian Fuzzy System Symposium*, pp. 538–542
142. Neto JC, Meyer GE, Jones DD (2006) Individual leaf extractions from young canopy images using gustafssonkessel clustering and a genetic algorithm. *Comput Electron Agric* 51(1):66–85
143. NKFC, Viswanatha SDK (2009) Routing algorithm using mobile agents and genetic algorithm. *Int J Comput Electr Eng*, vol 1, no 3
144. Ono I, Kobayashi S (1997) A real-coded genetic algorithm for functional optimization using unimodal normal distribution crossover. In: Back T (ed) *Proceedings of the 7th international conference on genetic algorithms, ICGA-7*. Morgan Kaufmann, East Lansing, MI, USA, pp 246–253
145. Pachevsky Y, Acock B (1998) Stochastic imaging of soil parameters to assess variability and uncertainty of crop yield estimates. *Geoderma* 85(2):213–229
146. Paiva JPD, Toledo CFM, Pedrini H (2016) An approach based on hybrid genetic algorithm applied to image denoising problem. *Appl Soft Comput* 46:778–791
147. Palencia AER, Delgado GEM (2012) A computer application for a bus body assembly line using genetic algorithms. *Int J Prod Econ* 140(1):431–438
148. Palomo-Romero JM, Salas-Morera L, García-Hernández L (2017) An island model genetic algorithm for unequal area facility layout problems. *Expert Syst Appl* 68:151–162
149. Pandian S, Modrák V (December 2009) "possibilities, obstacles and challenges of genetic algorithm in manufacturing cell formation," advanced logistic systems, University of Miskolc. Department of Material Handling and Logistics 3(1):63–70
150. Park Y-B, Yoo J-S, Park H-S (2016) A genetic algorithm for the vendor-managed inventory routing problem with lost sales. *Expert Syst Appl* 53:149–159
151. Patel R, Raghuvanshi MM, Malik LG (2012) Decomposition based multi-objective genetic algorithm (DMOGA) with opposition based learning
152. Pattanaik JK, Basu M, Dash DP (2018) Improved real coded genetic algorithm for dynamic economic dispatch. *Journal of electrical systems and information technology*. Vol. 5(3):349–362
153. Payne AW, Glen RC (1993) Molecular recognition using a binary genetic system. *J Mol Graph* 11(2):74–91
154. Peerlinck A, Sheppard J, Pastorino J, Maxwell B (2019) Optimal Design of Experiments for precision agriculture using a genetic algorithm. *IEEE Congress on Evolutionary Computation*.
155. Pelikan M, Goldberg DE, Cantu-Paz E (2000) Bayesian optimization algorithm, population sizing, and time to convergence, Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep
156. Pilat ML, White T (2002) Using genetic algorithms to optimize ACS-TSP, in the Third International Workshop on Ant Algorithms, vol. *Lecture Notes In Computer Science* 2463. Berlin, Germany: Springer-Verlag, pp. 282–287.
157. Pinagapany S, Kulkarni A (2008) Solving channel allocation problem in cellular radio networks using genetic algorithm. In: *Communication Systems software and middleware and workshops, 2008.COMSWARE 2008*. 3rd International Conference on. IEEE, pp239–244
158. Pinel F, Dorronsoro B, Bouvry P (2013) Solving very large instances of the scheduling of independent tasks problem on the GPU. *J Parallel Distrib. Comput* 73(1):101–110

159. Pinto G, Ainbinder I, Rabinowitz G (2009) A genetic algorithm-based approach for solving the resource-sharing and scheduling problem. *Comput Ind Eng* 57(3):1131–1143
160. Piszcz A, Soule T (2006) Genetic programming: optimal population sizes for varying complexity problems, in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 953–954.
161. Porta J, Parapar R, Doallo F, Rivera F, Santé I, Crecente R (2013) High performance genetic algorithm for land use planning. *Comput Environ Urb Syst* 37(2013):45–58
162. Rafsanjani MK, Riyahi M (2020) A new hybrid genetic algorithm for job shop scheduling problem. *International Journal of Advanced Intelligence Paradigms* 16(2):157–171
163. Rathi R, Acharjya DP (2018) A framework for prediction using rough set and real coded genetic algorithm. *Arab J Sci Eng* 43(8):4215–4227
164. Rathi R, Acharjya DP (2018) A rule based classification for vegetable production using rough set and genetic algorithm. *International Journal of Fuzzy System Applications (IJFSA)* 7(1):74–100
165. Rathi R, Acharjya DP (2020) A comparative study of genetic algorithm and neural network computing techniques over feature selection, In *advances in distributed computing and machine learning* (pp. 491–500). Springer, Singapore
166. Ray SS, Bandyopadhyay S, Pal SK (2004) New operators of genetic algorithms for traveling salesman problem," *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Cambridge pp 497-500
167. Richter JN, Peak D (2002) Fuzzy evolutionary cellular automata, in *international conference on artificial neural networks in engineering*, vol 12. USA, Saint Louis pp. 185-191
168. Riedl A (2002) A hybrid genetic algorithm for routing optimization in ip networks utilizing bandwidth and delay metrics. In: *IP operations and management, 2002 IEEE Workshop on*. IEEE, pp 166–170
169. Ripon KSN, Siddique N, Torresen J (2011) Improved precedence preservation crossover for multi-objective job shop scheduling problem. *Evolving Systems* 2:119–129
170. Roberge V, Tarbouchi M, Okou F (2014) Strategies to accelerate harmonic minimization in multilevel inverters using a parallel genetic algorithm on graphical processing unit. *IEEE Trans Power Electron* 29(10):5087–5090
171. Ronald S (1997) Robust encoding in genetic algorithms: a survey of encoding issues. *IEEE international conference on evolutionary computation*, pp. 43-48
172. Roy A, Banerjee N, Das SK (2002) An efficient multi-objective qos-routing algorithm for wireless multicasting. In: *Vehicle technology conference, 2002. VTC Spring 2002*. IEEE 55th, vol 3., pp 1160–1164
173. Sadrzadeh A (2012) A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Comput Ind Eng* 62(4):1055–1064
174. Sahingoz OK (2014) Generation of Bezier curve-based flyable trajectories for multi-UAV systems with parallel genetic algorithm. *J Intell Robot Syst* 74(1):499–511
175. Saini N (2017) Review of selection methods in genetic algorithms. *International Journal of Engineering and Computer Science* 6(12):22261–22263
176. Sari M, Can T (2018) Prediction of pathological subjects using genetic algorithms. *Computational and Mathematical Methods in Medicine* 2018:1–9
177. Scully T, Brown KN (2009) Wireless LAN load balancing with genetic algorithms. *Knowl Based Syst* 22(7):529–534
178. Sermpinis G, Stasinakis C, Theofilatos K, Karathanasopoulos A (2015) Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms—support vector regression forecast combinations. *European J. Oper. Res.* 247(3):831–846
179. Shabankareh SG, Shabankareh SG (2019) Improvement of edge-tracking methods using genetic algorithm and neural network, 2019 5th Iranian conference on signal processing and intelligent systems (ICSPIS). Shahrood, Iran, pp 1–7. <https://doi.org/10.1109/ICSPIS48872.2019.9066026>
180. Sharma S, Gupta K (2011) Solving the traveling salesman problem through genetic algorithm with new variation order crossover. *International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Udaipur, pp. 274–276
181. Sharma N, Kaushik I, Rathi, R, Kumar S (2020) Evaluation of accidental death records using hybrid genetic algorithm. Available at SSRN: <https://ssrn.com/abstract=3563084> or <https://doi.org/10.2139/ssrn.3563084>
182. Shayeghi A, Gotz D, Davis JBA, Schafer R, Johnston RL (2015) Pool-BCGA: A parallelised generation-free genetic algorithm for the ab initio global optimisation of nano alloy clusters. *Phys Chem Chem Phys* 17(3):2104–2112
183. Guoyong Shi, H. Iima and N. Sannomiya (1996) A new encoding scheme for solving job shop problems by genetic algorithm, *Proceedings of 35th IEEE Conference on Decision and Control, Kobe, Japan, 1996*, pp. 4395–4400 vol.4.

184. Shi J, Liu Z, Tang L, Xiong J (2017) Multi-objective optimization for a closed-loop network design problem using an improved genetic algorithm. *Appl Math Model* 45:14–30
185. Shukla AK, Singh P, Vardhan M (2019) A new hybrid feature subset selection framework based on binary genetic algorithm and information theory. *International Journal of Computational Intelligence and Applications* 18(3):1950020(1–10)
186. Singh A, Deep K (2015) Real coded genetic algorithm operators embedded in gravitational search algorithm for continuous optimization. *Int J Intell Syst Appl* 7(12):1
187. Sivanandam SN, Deepa SN (2008) *Introduction to genetic algorithm*, 1st edn. Springer-Verlag, Berlin Heidelberg
188. Soleimani H, Kannan G (2015) A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks. *Appl Math Model* 39(14):3990–4012
189. Soleimani H, Govindan K, Saghafi H, Jafari H (2017) Fuzzy multi-objective sustainable and green closed-loop supply chain network design. *Comput Ind Eng* 109:191–203
190. Soon GK, Guan TT, On CK, Alfred R, Anthony P (2013) "A comparison on the performance of crossover techniques in video game," 2013 IEEE international conference on control system. *Computing and Engineering*, Mindeh, pp 493–498
191. Srinivas N, Deb K (1995) Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evol Comput* 2(3):221–248
192. Subbaraj P, Rengaraj R, Salivahanan S (2011) Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem. *Appl Soft Comput* 11(1):83–92
193. Tahir M, Tubaishat A, Al-Obeidat F, et al. (2020) A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare. *Neural Comput & Appl*
194. Tam V, Cheng K-Y, Lui K-S (2006) Using micro-genetic algorithms to improve localization in wireless sensor networks. *J Commun* 1(4):1–10
195. Tan KC, Li Y, Murray-Smith DJ, Sharman KC (1995) System identification and linearisation using genetic algorithms with simulated annealing, in *First IEE/IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl.* Sheffield, UK, pp. 164–169.
196. Tang PH, Tseng MH (2013) Adaptive directed mutation for real-coded genetic algorithms. *Appl Soft Comput* 13(1):600–614
197. Tiong SK, Yap DFW, Koh SP (2012) A comparative analysis of various chaotic genetic algorithms for multimodal function optimization. *Trends in Applied Sciences Research* 7:785–791
198. Toutouh J, Alba E (2017) Parallel multi-objective metaheuristics for smart communications in vehicular networks. *Soft Comput* 21(8):1949–1961
199. Umbarkar A, Sheth P (2015) Crossover operators in genetic algorithms: a review. *Journal on Soft Computing* 6(1)
200. Verma D, Vishwakarma VP, Dalal S (2020) A hybrid self-constrained genetic algorithm (HSGA) for digital image Denoising based on PSNR improvement. *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, In, pp 135–153
201. Vitayasak S, Pongcharoen P, Hicks C (2016) A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a genetic algorithm or modified backtracking search algorithm. *Int J Prod Econ*
202. Junru Wang and Lan Huang (2014) Evolving gomoku Solver by Genetic Algorithm. *IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA)* pp 1064–1067.
203. Wang L, Kan MS, Shahriar Md R, Tan ACC (2014) Different approaches of applying single-objective binary genetic algorithm on the wind farm design. In *World Congress on Engineering Asset Management*.
204. Wang N, Li Q, Abd El-Latif AA, Zhang T, Niu X (2014) Toward accurate localization and high recognition performance for noisy iris images. *Multimed Tools Appl* 71(3):1411–1430
205. Wang JQ, Ersoy OK, He MY et al (2016) Multi-offspring genetic algorithm and its application to the traveling salesman problem. *Appl Soft Comput* 43:415–423
206. Wang FL, Fu XM, Zhu HX et al (2016) Multi-child genetic algorithm based on two-point crossover. *J Northeast Agric Univ* 47(3):72–79
207. Wang JQ, Cheng ZW, Ersoy OK et al (2018) Improvement analysis and application of real-coded genetic algorithm for solving constrained optimization problems. *Math Probl Eng* 2018:1–16
208. Wang J, Zhang M, Ersoy OK, Sun K, Bi Y (2019) An improved real-coded genetic algorithm using the Heuristical Normal distribution and direction-based crossover. *Computational Intelligence and Neuroscience* 2019:1–17
209. Wen Z, Yang R, Garraghan P, Lin T, Xu J, Rovatsos M (2017) Fog orchestration for internet of things services. *IEEE Internet Comput* 21(2) (Mar. 2017):16–24
210. Wright AH (1991) *Genetic algorithms for real parameter optimization*. In *Foundations of genetic algorithms I*, G. J. E. Rawlins, Ed., Morgan Kaufmann, San Mateo, CA, USA

211. Wu X, Chu C-H, Wang Y, Yan W (2007) A genetic algorithm for cellular manufacturing design and layout. *European J Oper Res* 181(1):156–167
212. Yang S, Cheng H, Wang F (2010) Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Trans Syst Man Cybern Part C Appl Rev* 40(1):52–63
213. Yang C, Li H, Rezgui Y, Petri I, Yuce B, Chen B, Jayan B (2014) High throughput computing based distributed genetic algorithm for building energy consumption optimization. *Energy Build* 76(2014):92–101
214. Yu F, Xu X (2014) A short-term load forecasting model of natural gas based on optimized genetic algorithm and improve BR neural network. *Appl Energy* 134:102–113
215. Yuce B, Fruggiero F, Packianather MS, Pham DT, Mastrocinque E, Lambiase A, Fera M (2017) Hybrid genetic bees algorithm applied to single machine scheduling with earliness and tardiness penalties. *Comput Ind Eng* 113:842–858
216. Yun S, Lee J, Chung W, Kim E, Kim S (2009) A soft computing approach to localization in wireless sensor networks. *Expert Syst Appl* 36(4):7552–7561
217. Zhai R (2020) Solving the optimization of physical distribution routing problem with hybrid genetic algorithm. *J Phys Conf Ser* 1550:1–6
218. Zhang Q, Wang J, Jin C, Zeng Q (2008) Localization algorithm for wireless sensor network based on genetic simulated annealing algorithm. In: 4th IEEE International Conference on Wireless communications, networking and mobile computing. Pp 1–5
219. Zhang R, Ong SK, Nee AYC (2015) A simulation-based genetic algorithm approach for remanufacturing process planning and scheduling. *Appl Soft Comput* 37:521–532
220. Zhang X-Y, Zhang J, Gong Y-J, Zhan Z-H, Chen W-N, Li Y (2016) Kuhn-Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks. *IEEE Trans Evol Comput* 20(5):695–710
221. Zhenhua Y, Guangwen Y, Shanwei L, Qishan Z (2010) A modified immune genetic algorithm for channel assignment problems in cellular radio networks. In: Intelligent system design and engineering application (ISDEA), 2010 International Conference on, vol 2. , pp 823–826

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.