# Genetic Algorithm

Priyadarshi Roy
St. Xavier's College (Autonomous), Kolkata

***Abstract*—Genetic Algorithm is a population based metaheuristic algorithm. It mimics Darwinian theory of survival of fittest in nature. This paper is a study of this Genetic Algorithm. The steps and procedures have been studied and further advancements of the algorithm have been mentioned. Finally, the positives and negatives of the algorithm have been explored.**

***Keywords—Optimization, Metaheuristic, Genetic algorithm, Fitness, Selection, Crossover, Mutation, Evolution***

## I. Introduction

In recent years, metaheuristic algorithms are used to solve many real-life problems arising from different fields such as economics, engineering, politics, management and engineering [1]. Most metaheuristic algorithms are inspired from biological evolution process, swarm behaviour and physics laws.

Metaheuristic algorithms are broadly classified into two categories: single solution and population based metaheuristic algorithms. 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 get stuck in local optima.The well known single-solution based metaheuristics are simulated annealing, tabu search, microcanonical annealing and guided local search. Population-based metaheuristics utilize multiple candidate solutions during the search process. These metaheuristics maintain the diversity in the population and hence can avoid solutions getting stuck in local optima. Few well known population-based metaheuristics include genetic algorithm (GA)[2,3], particle swarm optimizer [12], ant colony optimization, spotted hyena optimizer, emperor penguin optimizer and seagull optimization. As the names indicate, the inspiration for most of these metaheuristics are based on a metaphor of some natural or man-made process. [4]

One such population-based metaheuristic, genetic algorithm (GA) is explored in this paper.

## II. Background

Among the metaheuristic algorithms, Genetic algorithm (GA) is a well-known algorithm, which is inspired from the biological evolution process [5]. GA simulates the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to the next generation. GA thus 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.

## III. Terminology

- ***Population***: It is a subset of all the possible (encoded) solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have Candidate Solutions representing human beings. (Fig. 1)
- ***Chromosome***: A chromosome is one such solution to the given problem. (Fig. 1)
- ***Gene***: A gene is one element of a position of a chromosome. (Fig. 1)
- ***Allele***: It is the value a gene takes for a particular chromosome. (Fig. 1)
- ***Genotype***: Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system. (Fig. 2)
- ***Phenotype***: Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations. (Fig. 2)
- ***Decoding and Encoding***: For simple problems, the phenotype and genotype spaces are the same. However, in most of the cases, the phenotype and genotype

- spaces are different. Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space. (Fig. 2)
- ***Fitness Function***: A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. It takes a candidate solution to the problem as input and produces as output how "fit" our how "good" the solution is with respect to the problem in consideration. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem. Calculation of fitness value is done repeatedly in a GA and therefore it should be sufficiently fast. A slow computation of the fitness value can adversely affect a GA and make it exceptionally slow.
- ***Genetic Operators***: These alter the genetic composition of the offspring. These include encoding, selection, crossover and mutation. (Fig.3)

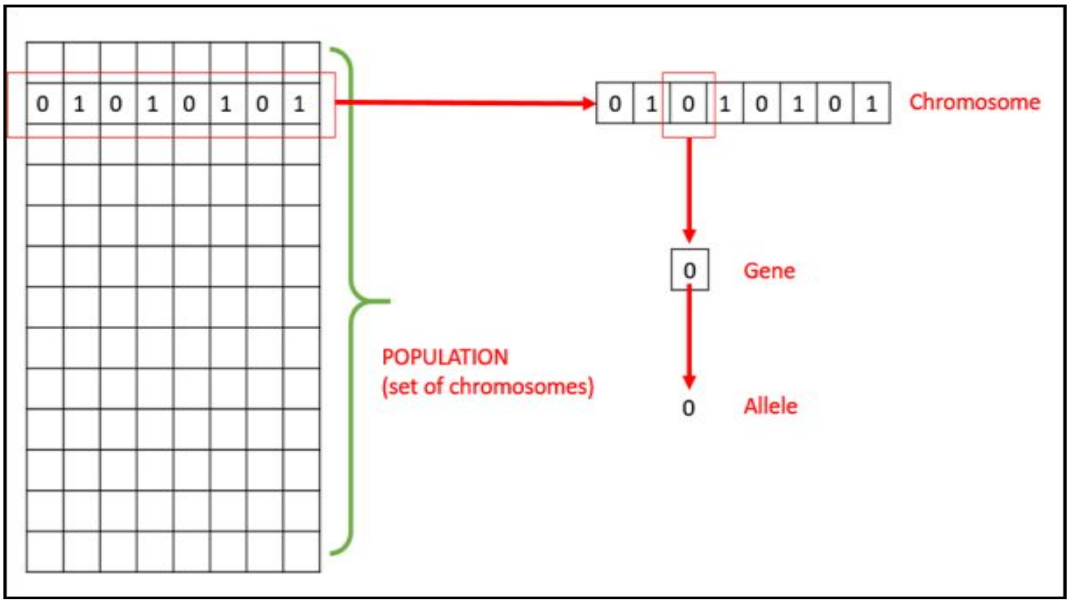


Fig.1. Population, chromosome, gene and allele

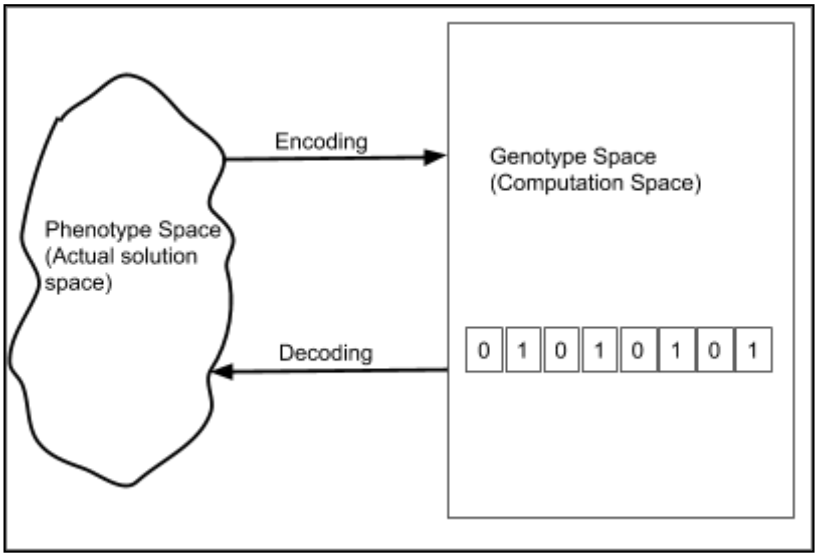


Fig.2. Genotype, phenotype, encoding and decoding

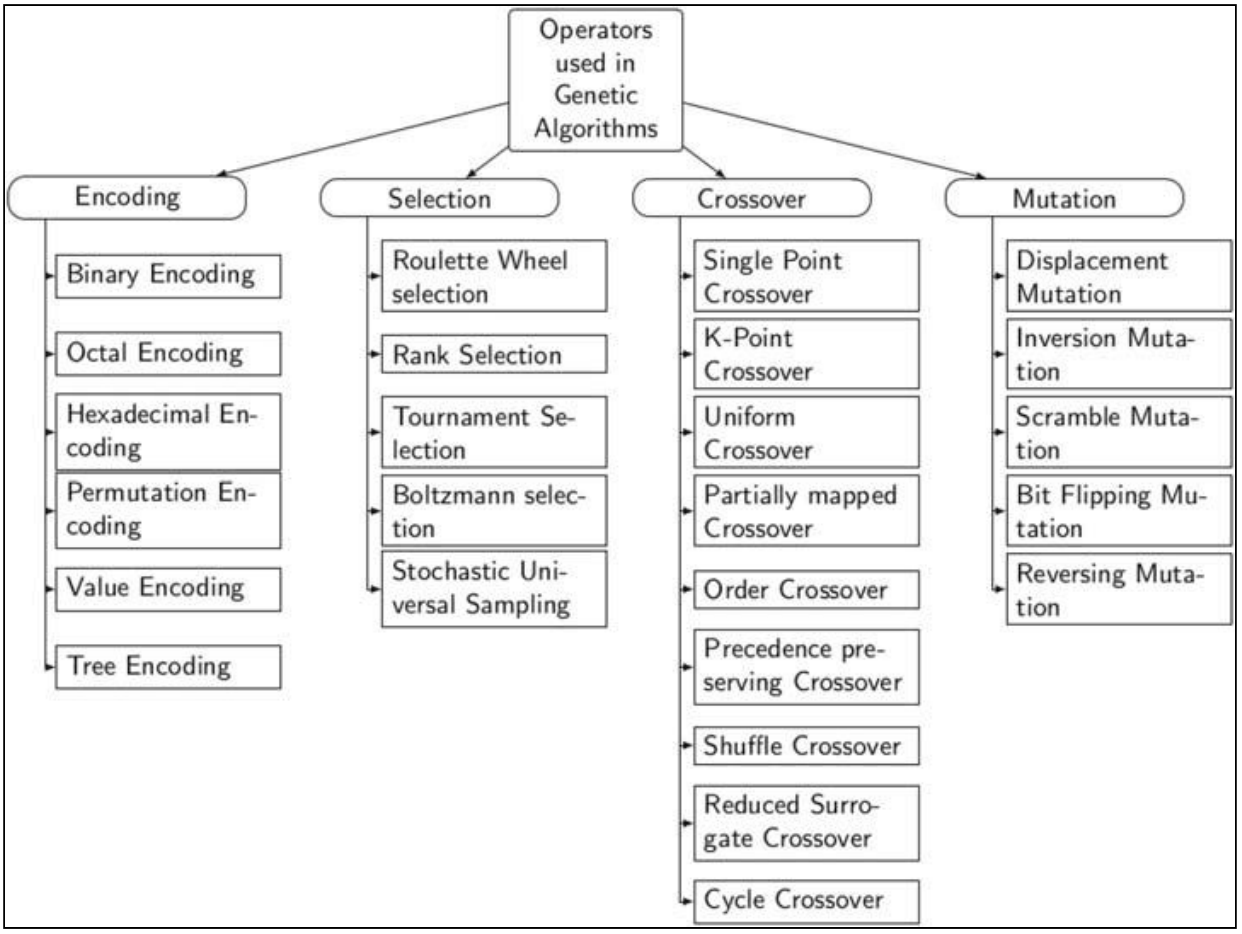


Fig.3. Genetic Operators

## IV. ALGORITHM

### *A. Classical Genetic Algorithm (Fig.4)*

Genetic algorithm (GA)(Fig.4) is an optimization algorithm that is inspired from natural selection. It is a population based search algorithm, which utilizes the concept of survival of fittest [2]. 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 chromo-somes say C1 and C2 are selected from the population Y according to the fitness value. The single-point crossover operator with crossover probability (Cp) 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 (Mp) to generate O′. The new offspring O′ is placed in the new population. The selection, crossover, and mutation operations will be repeated on the current population until the new population is complete. Each iteration of the algorithm is called a *generation*.

GA dynamically changes the search process through the probabilities of crossover and mutation and reaches an 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.

### *B. Genetic Operators (Fig.3)*

*Owing to the vast number of genetic operators in use, only a couple of them are studied in this paper. The others are provided with suitable references for further study.*

***a) Encoding Scheme****:* For most of the computational problems, the encoding scheme [6] (i.e., to convert in particular form) plays an important role. The given information has to be encoded in a particular bit string [7]. 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 [8]. In binary encoding, each bit represents the character-istics 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 to the problem. In octal encoding scheme, the gene or

**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,....,n)
Set iteration counter t = 0
**while** (t < MAX)
Fitness:
Compute the fitness value of each chromosomes
Selection:
Select a pair of chromosomes from initial population based on fitness
Crossover:
Apply crossover operation on selected pair with crossover probability
Mutation:
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**

Fig.4. Genetic Algorithm

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) [7, 8]. The permutation encoding 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 a string of some values. These values can be real, integer number, or character [9]. 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 [10]. This type of

encoding is generally used in evolving programs or expressions.

***b) Selection****:* Selection or Parent Selection is an important step in genetic algorithms that determines whether the particular string (chromosome) will participate in the reproduction process or not. The selection step is sometimes also known as the reproduction operator [9, 10]. The convergence rate of GA depends upon the selection pressure. However, care should be taken to prevent one extremely fit solution from taking over the entire population in a few generations, as this leads to the solutions being close to one another in the solution space thereby leading to a loss of diversity. Maintaining good diversity in the population is extremely crucial for the success of a GA. This taking up of the entire population by one extremely fit solution is known as premature convergence and is an undesirable condition in a GA. The well-known selection techniques are roulette wheel, rank, tournament, and stochastic universal sampling, boltzmann and elitism[10]. Parent selection is very crucial to the convergence rate of the GA as good parents drive individuals to better and fitter solutions.

*i) Roulette wheel:* Fitness proportionate selection, also known as roulette wheel selection, is a genetic operator used in genetic algorithms for selecting potentially useful solutions for recombination. In fitness proportionate selection, as in all selection methods, the fitness function assigns a fitness to possible solutions or chromosomes. This fitness level is used to associate a probability of selection with each individual chromosome. This could be imagined similar to a Roulette wheel in a casino. Usually a proportion of the wheel is assigned to each of the possible selections based on their fitness value. This could be achieved by dividing the fitness of a selection by the total fitness of all the selections, thereby normalizing them to 1. A fixed point is chosen on the wheel circumference and the wheel is rotated. Then a random selection is made similar to how the roulette wheel is rotated, by choosing the individual pointed to by the fixed point.

*ii) Rank selection*: 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 [10].

*iii) Tournament selection*: 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 [10]. In this method of selection, each individual is compared with all n-1 other individuals if it reaches the final population of solutions [10].

*iv) Stochastic universal sampling*: 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. It gives equal chance to all the individuals in getting selected for participating in crossover for the next generation.

***c) Crossover****:* 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-point, two-point, k-point, uniform, partially matched, order, precedence preserving crossover, shuffle, reduced surrogate and cycle. In a single point crossover (Fig.5), a random crossover point is selected. The genetic information of two parents which is beyond that point will be swapped with each other [11]. It replaced the tail array bits of both the parents to get the new offspring. In a two point (Fig.6) 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 [11].In a uniform crossover (Fig.7), the 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 [11].

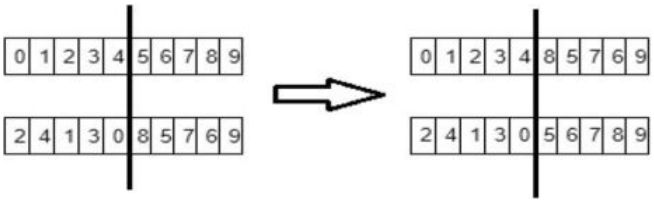


Fig.5. Single point crossover

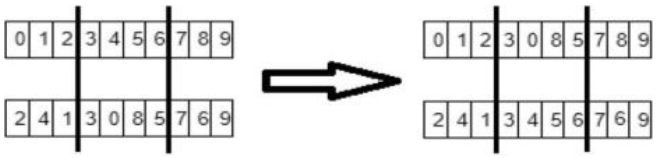


Fig.6. Two point crossover

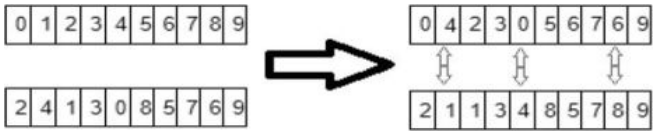


Fig.7. Uniform crossover

***d) Mutation****:* Mutation operators generate diversity in the population. Two main challenges have to be tackled during the application of mutation. First, the proba-bility of mutation operator that was applied on the population. Second, the outlier produced in chromosomes after the mutation process.Crossover involves reproduction between strings in the existing population. It may happen that none of the parents in the existing population have the necessary resources to converge to a solution. Mutation is the only operator that can introduce diversity in the current population, by introducing new allele values in the genes.

*Mutation rate (probability)*: This rate determines how many chromosomes should be mutated in one generation; mutation rate is in the range of [0, 1]. Some commonly used mutation techniques include bit-flip, random resetting, swap, scramble, inversion and displacement.[10].

*i) Bit Flip Mutation*: In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs. (Fig.8)

*ii) Random Resetting*: Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

*iii) Swap Mutation*: In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings. (Fig.9)

*iv) Scramble Mutation*: Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly. (Fig.10)

*v) Inversion Mutation*: In inversion mutation,a subset of genes is selected like in scramble mutation, but instead of shuffling the subset, the entire subset is merely inverted.(Fig.11)

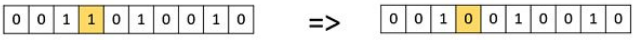


Fig.8. Bit Flip Mutation

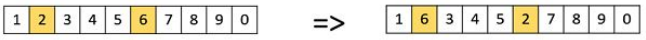


Fig.9.Swap Mutation

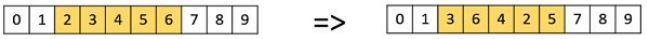


Fig.10.Scramble Mutation

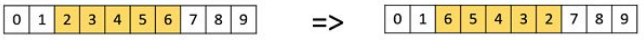


Fig.11.Inversion Mutation

## V. 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 [16].
The details of these advanced GAs are beyond the scope of this paper, but have been mentioned for the sake of completeness.

## VI. CONCLUSION

Genetic Algorithm has proven effective due to its parallelism, easy modifiability, adaptability to different problems and problem types and multi-objective optimization [17] support.

***Positives:***

It excels when applied to solutions where there are multiple optima. GA searches parallelly from a population of points. Therefore, it has the ability to avoid being trapped in local optimal solution like traditional methods, which search from a single point. They excel in cases where the objective function is not smooth so derivative methods like gradient descent cannot be applied and where the number of parameters is very large or even when the objective function is noisy or stochastic.

***Negatives:***

Contrary to the advantages that it offers, GA implementation is still an art. GA requires less information about the problem, but designing an objective function and getting the representation and operators right can be difficult. Furthermore, GA is computationally expensive, since the fitness function has to be calculated for each generation for each member of the population.

## REFERENCES


[1] 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

[2] Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs. Springer-Verlag, New York

[3]https://link.springer.com/article/10.1007/s11042-020-10139-6

[4]https://www.cs.ubc.ca/~hutter/EARG.shtml/stack/2013_Sorensen_MetaheuristicsTheMetaphorExposed.pdf

[5] Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. Evol Comput 4(1):1–32

[6] Kumar A (2013) Encoding schemes in genetic algorithm. Int J Adv Res IT Eng 2(3):1–7

[7] 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.

[8] Sivanandam SN, Deepa SN (2008) Introduction to genetic algorithm, 1st edn. Springer-Verlag, Berlin Heidelberg

[9] 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.

[10] Jebari K (2013) Selection methods for genetic algorithms. Abdelmalek Essaâdi University. International Journal of Emerging Sciences 3(4):333–344

[11] 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, Mindeb, pp 493–498

[12] Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, Inc

[13] Dhiman G, Kumar V (2017) Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70

[14] Dhiman G, Kumar V (2018) Emperor penguin optimizer: A bio-inspired algorithm for engineering
problems. Knowl-Based Syst 159:20–50

[15] Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl-Based Syst 165:169–196

[16] El-Mihoub TA, Hopgood AA, Lars N, Battersby A (2006) Hybrid genetic algorithms: A review. Eng Lett
13:2

[17] Emmerich MTM, Deutz AH (2018) A tutorial on multiobjective optimization: fundamentals and evolu-tionary methods. Nat Comput 17(3):585–609