SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

A Review of Genetic Algorithm: Operations and Applications

Faizatulhaida Md Isa[1,4,*], Wan Nor Munirah Ariffin[2,3], Muhammad Shahar Jusoh[1], Erni Puspanantasari Putri[5]

1 Faculty of Business and Comunication, Universiti Malaysia Perlis (UniMAP), Pusat Perniagaan Pengkalan Jaya Jalan Alor Setar-Kangar 01000 Kangar, Perlis, Malaysia
2 Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, Pagoh Higher Educational Hub, 84600 Pagoh, Johor, Malaysia
3 Centre of Excellent Social Innovation and Sustainability, University Malaysia Perlis (UniMAP), Kampus Pauh Putra, 02600 Arau, Perlis, Malaysia
4 Department of Mathematics, Science & Computer, Politeknik Tuanku Sultanah Bahiyah (PTSB), Kulim Hi-Tech Park, 09090 Kulim, Kedah, Malaysia
5 Department of Industrial Engineering, University of 17 Agustus 1945 Surabaya, Indonesia

ARTICLE INFO

Article history:
Received 22 June 2023
Received in revised form 7 September 2023
Accepted 7 November 2023
Available online 19 February 2024

Keywords:
Review; genetic algorithm; operation; application

ABSTRACT

This article presents a review of the Genetic Algorithm (GA), a prominent optimization technique inspired by natural selection and genetics. In the context of rapidly evolving computational methodologies, GA have gained considerable attention for their efficacy in solving complex optimization problems across various domains. The background highlights the growing significance of optimization techniques in addressing real-world challenges. However, the inherent complexity and diversity of problems necessitate versatile approaches like GA. The problem statement underscores the need to explore the underlying operations and applications of GA to provide a nuanced understanding of their capabilities and limitations. The objectives of this review encompass delving into the fundamental genetic operators, such as selection, crossover, and mutation, while examining their role in maintaining diversity and converging toward optimal solutions. Methodology-wise, a systematic analysis of existing literature is undertaken to distil key insights and trends in GA applications. The main findings show the adaptability of GA in tackling problems spanning engineering, economics, bioinformatics, and beyond. By facilitating the discovery of optimal or near-optimal solutions within large solution spaces, GA proves its mettle in scenarios where traditional methods fall short. The conclusion underscores the enduring relevance of GA in the optimization landscape, emphasizing their potential to remain a pivotal tool for addressing intricate real-world challenges, provided their parameters are fine-tuned judiciously to balance exploration and exploitation.

1. Introduction

Computers play a critical role in resolving difficulties in our everyday life. They provide numerous applications and benefits in biology, chemistry, physics, mathematics, geography, archaeology, engineering, and social sciences fields. Metaheuristic algorithms can be used to tackle complex

* *Corresponding author.*
E-mail address: faizatulhaida@studentmail.unimap.edu.my (Faizatulhaida Md Isa)

https://doi.org/10.37934/araset.40.1.134

problems in a variety of domains, including engineering, management, economics, and politics from the previous studies [1]. Intensification and diversity are crucial components of a metaheuristic algorithm. Metaheuristic algorithms are divided into two categories: single-solution-based and population-based metaheuristic algorithms. The classification of metaheuristic algorithms is shown in Figure 1.

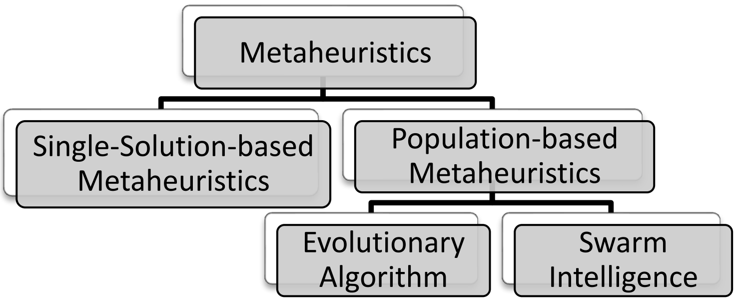


Fig. 1. Classification of metaheuristic algorithms

Simulated annealing (SA), taboo search (TS), microcanonical annealing (MA), and guided local search (GLS) are single-solution-based metaheuristics. Whereas population-based metaheuristics consist of genetic algorithm GA, particle swarm optimization (PSO), ant colony optimization (ACO), spotted hyena optimizer (SHO), emperor penguin optimizer (EPO), and seagull optimization (SOA) [2-7].

John Holland from University of Michigan introduced GA in 1975 [8]. Chromosome representation, fitness selection, and biologically inspired operators are all essential components in GA. Combinatorial optimization problems, i.e., obtaining optimal solution values to problems with numerous possible solutions, are one of the applications of GA. In terms of adaptation, each challenge can be expressed in genetic terms. GA based on operators, representation, and fitness are no longer being developed.

GA is one of the most widely used evolutionary algorithms, and it is based on Charles Darwin's natural selection principle [9]. This is reinforced by Gorunescu *et al.,* [10], who claimed that evolutionary algorithms can be used to find value searches and solutions to a variety of optimization problems. The capacity to cope with complicated and parallel issues is why GA are preferred over other optimization algorithms. It can handle various optimizations regardless of the fitness of the objective function (linear vs nonlinear, continuous vs noncontinuous, or existence of random noise). As a result, this paper focuses on these aspects of GA. In essence, this paper will discuss on below topics:

(i) general framework of GA;
(ii) various types of genetic operators together with their pros and cons;
(iii) variants of GA with their advantages and disadvantages; and
(iv) applicability of GA in various fields.

The article “Review of Genetic Algorithm: Operations and Applications" makes a significant contribution to the academic pursuit of advancing knowledge in the field of genetic algorithms by meticulously addressing a discernible research gap. Despite widespread acknowledgment of GA as effective tools for optimization and problem-solving, a full synthesis of their core operations and the large range of applications they cover has been significantly lacking in the scholarly literature. This article fills that hole well by undertaking a thorough and comprehensive examination of the fundamental processes of GA—selection, crossover, and mutation—shedding light on their complicated interaction and functional relevance.

2. Genetic Algorithm

GA is an algorithm that seeks to apply an understanding of natural evolution to solve everyday problems. It is commonly utilized to tackle optimization, research, and machine learning problems.

The approach taken by this algorithm is to combine various best solution options randomly in a set to get the best solution for the next generation to improve the solution's fitness (suitability). GA is a search algorithm-based natural selection mechanism, and it is also one of the most suitable algorithms used to solve complex optimization problems that cannot be tackled by conventional methods. The flowchart of the general GA can be seen in Figure 2 [11].

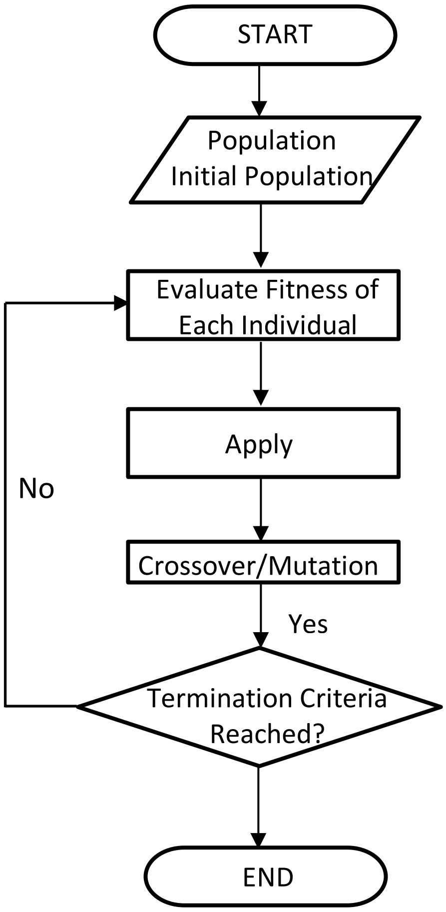


Fig. 2. Flowchart of the general GA

The solution produced by GA is called a chromosome and a set of chromosomes is called a population. Chromosomes are formed from an arrangement of components called genes and their values can be numerical, binaries, symbols, or characters depending on the problem to be solved.

Figure 3 shows the chromosome selection process uses Darwin's concept of evolutionary rules, i.e., chromosomes with high fitness values will have a greater chance of being reselected in future generations.

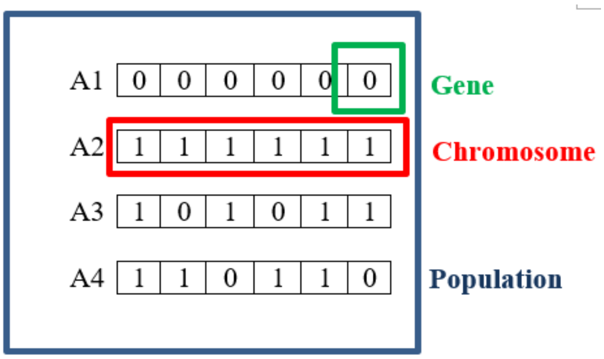


Fig. 3. Gene, chromosome, and population

The operators used in GA can be seen in Table 1.

Table 1
Operators used in GA

Type of operator	Example
(i) Encoding	Binary Encoding, Octal Encoding, Hexadecimal Encoding, Permutation Encoding, Value Encoding, Tree Encoding
(ii) Selection	Roulette Wheel Selection, Rank Selection Tournament Selection, Boltzmann Selection, Stochastic Universal Sampling
(iii) Crossover	Single Point Crossover K- Point Crossover Uniform Crossover Partially Mapped Crossover Order Crossover Precedence Preserving Crossover Shuffle Crossover Reduced Surrogate Crossover Cycle Crossover
(iv) Mutation	Displacement Mutation, Inversion Mutation Scramble Mutation, Bit Flipping Mutation, Reversing Mutation

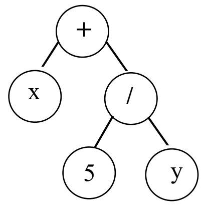
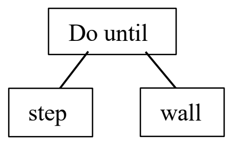
The operators of GA are discussed as shown in the next section.

2.1 Encoding Scheme

The encoding scheme an important role in converting the received information into a particular form. A common encoding scheme used in GA is Binary Encoding in which each gene or chromosome is represented as a string of 1 or 0 [12]. Octal Encoding, Hexadecimal Encoding, Permutation Encoding, Value Encoding, and Tree Encoding are also elements in the encoding scheme. The Binary Encoding scheme may not be suitable for some engineering design problems due to epistasis and natural representation. The comparison of different encoding schemes for GA is in Table 2. The fitness function for a certain encoding scheme, according to Goldberg [13], is determined by two factors: value and order.

(i) Encoding systems in which fitness is solely determined by order: f(o), for example, Permutation Encoding [14];
(ii) Value and order encoding schemes: f(v,o), e.g., Binary Encoding;
(iii) Value only encoding schemes: f(v), e.g., Value Encoding.

Table 2
Type and example of binary

Type of binary	Example	
(i) Binary Encoding: A binary string is used to represent each chromosome in this format (0-1) [15]. In Binary Encoding, there are four types of crossover operations: 1-point crossover, N-point crossover, Uniform crossover, and Arithmetic crossover.	Chromosome1 Chromosome2	110101110010 110101110010
(ii) Octal Encoding: The chromosome is represented using octal integers in this coding from the previous studies (0-7).	Chromosome1 Chromosome2	06254524 63726425
(iii) Hexadecimal Encoding: Hexadecimal numbers are used to represent the chromosome in this format (0-9, A-F).	Chromosome1 Chromosome2	97AE A2C6
(iv) Permutation Encoding: In Permutation Encoding, each chromosome represents a location in a series; for example, in the traveling salesman problem, the string of numbers represents the sequence of cities visited by the salesman.	Chromosome1 Chromosome2	1 5 2 3 5 2 6 4 6 9 8 8 6 3 6 3 9 6 3 1 5 8
(v) Value Encoding: A chromosome is represented as a string using Value Encoding, which can include integers, real numbers, characters, or objects. The crossover operator for integer values is the same as Binary Encoding. The values can be anything relevant to the problem, such as numbers or characters.	Chromosome1 Chromosome2	1.23, 2.12, 3.14, 0.34, 4.62 ABDJEIFJDHDDLDFLFEGT
(vi) Tree Encoding: Tree encoding is mostly used in genetic programming to evolve programs or expressions.	Chromosome1	
	Chromosome2	

Binary representation in genetic algorithms is efficient and simple for discrete optimization problems, but it has limitations in precision, expression, and exploration. It is crucial to consider the problem's characteristics before choosing this representation, and a hybrid approach may be beneficial to combine its strengths. Table 3 shows the comparison of different encoding of GA.

Table 3
Comparison of different encoding of GA

Encoding Schemes	Advantages	Disadvantages
Binary	Easy to implement	No support for an inversion operator
Octal	Easy to implement	No support for an inversion operator
Hexadecimal	Easy to implement	No support for an inversion operator
Permutation	Supports inversion operator	No support for binary operator
Value	Does not need value conversion	Requires specific crossover and mutation
Tree	Operators can be easily applied	Difficult to design a tree for some problems

2.2 Selection

Selection is an important step in GA to determine whether a particular string will participate in the reproductive process or not [16]. Reproduction operators are also known as selection steps. Selection stress is an important aspect in determining the rate of GA concentration. The ability of GA to make progressively better chromosomes is determined by the population's selection stress. Selective emphasis can be achieved in two ways: by producing more chromosomes in the population and then selecting only the best chromosomes for the next generation, or by selecting only the best chromosomes for the next generation. The latter approach yields higher-quality chromosomes [17].

The next step in selective emphasis is to pick the best parent for chromosomes. This strategy results in a population with a limited number of chromosomes for future generations. Roulette Wheel Selection, Rank Selection, Tournament Selection, Boltzmann Selection, Stochastic Universal Sampling, and Elitism are all common chromosomal selection strategies. Figure 4 shows the basic selection process in GA.

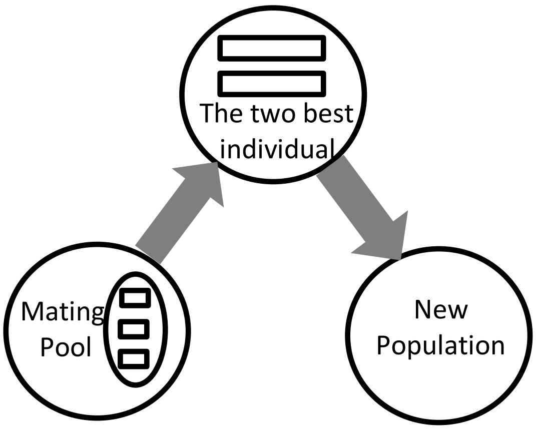
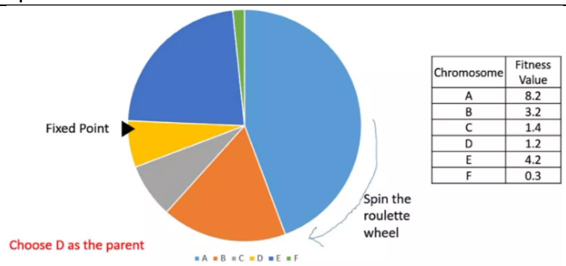
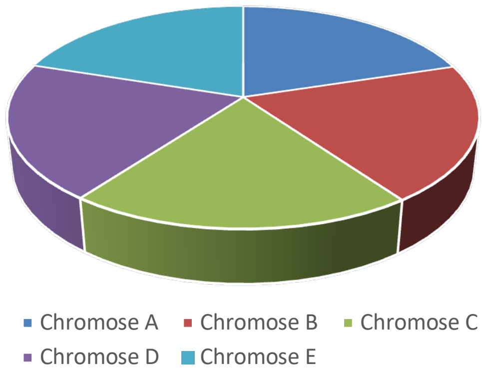
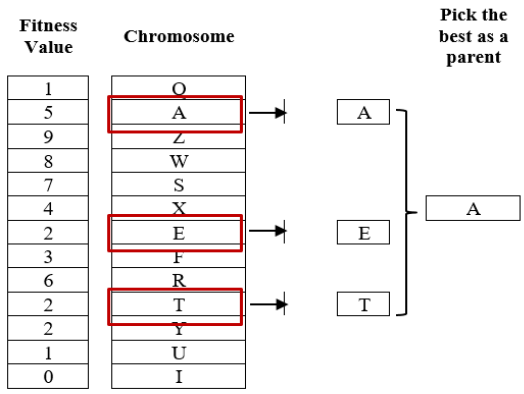
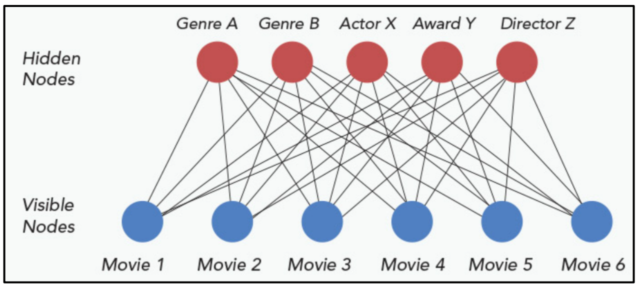


Fig. 4. Basic selection process

Table 4 shows the several established selection approaches, namely Roulette Wheel, Rank, Tournament, Boltzmann, Elitism and Stochastic Universal Sampling.

Table 4
Type and example of the selection

Type of selection	Example
(i) Roulette Wheel Selection Each individual is assigned a piece of the roulette wheel, the size of which is precisely proportional to the individual level of fitness as shown in Figure 5. As a result, a large segment size value will lead to a high level of fitness. This process will continue until the desired number of people has been chosen [18].	 **Fig. 5.** Roulette Wheel Selection
(ii) Rank Selection Roulette Wheel Selection will have issues if the fitness values are substantially diverse, and Rank Selection can be used to address this problem. It enables each chromosome to acquire fitness from its position and is utilized to allocate position to a population as shown in Figure 6. Positions are ranked from best to worst in the selection process. Each individual in the population was assigned a numerical rank based on their fitness, with selection dependent on their position [19].	 **Fig. 6.** Ranking between different chromosomes
(iii) Tournament Selection Figure 7 shows that several people are chosen at random from the population in the Tournament Selection process. The best fitness in the tournament determines the winner. The selection pressure can be easily modified as the size of the competition changes. According to Miller and Goldberg [20], enhanced selection pressure can be achieved by simply raising the size of the competition, as winners of larger tournaments will, on average, be fitter than winners of smaller tournaments. Tournament Selection is also extremely popular in literature as it can even work with negative fitness values.	 **Fig. 7.** Process in Tournament Selection
(iv) Boltzmann Selection Figure 8 shows that the selection rate is controlled by the constantly changing "temperature" according to the Boltzmann Selection schedule. The temperature rises quickly, indicating a low selection pressure. The selection pressure will rise as the temperature gradually drops, allowing the GA to narrow closer to the best section of the search area while retaining an appropriate level of diversity. In Monte Carlo simulation, this selection strategy is based on entropy and significance sampling methods [21]. It	 **Fig. 8.** Process in Boltzmann Selection

naturally leads to adaptive fitness, in which the fitness function changes with the environment rather than remaining constant.

(v) Stochastic Universal Sampling
Figure 9 shows that the Stochastic Universal Sampling is similar to Roulette Wheel Selection, except that instead of having only one fixed point, it has numerous fixed points. As a result, all the parents are chosen in a single-wheel spin. Furthermore, such a system encourages highly fit people to be chosen at least once.

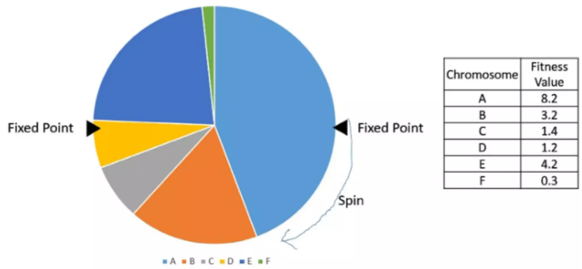


Fig. 9. Stochastic Universal Sampling

(vi) Elitism
Figure 10 shows that Elitism is a repeated selection process used to keep individuals with the greatest fitness values from deteriorating during evolution. This is because when crossovers are performed, it is quite likely that those with high fitness levels will lose fitness. No changes will be made by this technique because the best individuals can be passed on to the next generation [22].

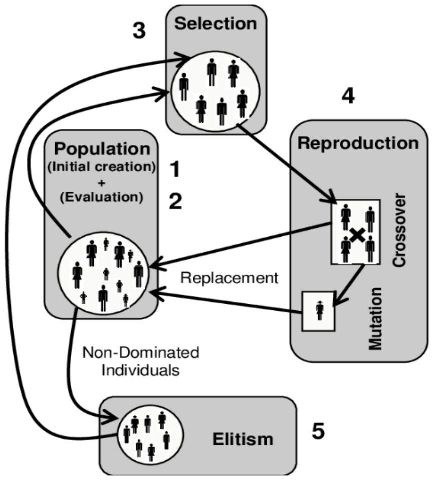


Fig. 10. Elitism

Table 5 presents a comparative analysis of several selection techniques employed in the context of (GA).

Table 5
Comparison of different selection techniques of GA

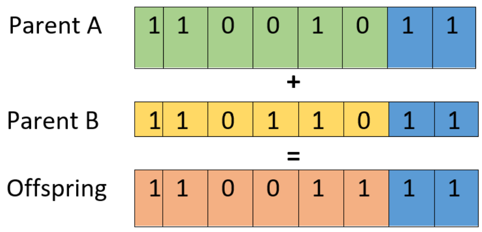
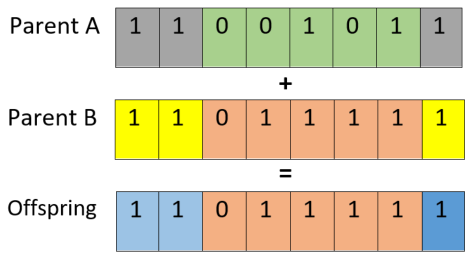
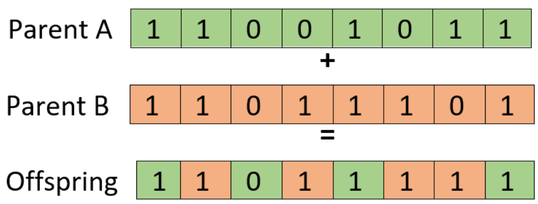
Selection Techniques	Advantage	Disadvantage
Roulette Wheel	Easy to implement	Premature convergence risk Depends upon variance
Rank	Preserve diversity Free from bias	Slow convergence Sorting required
Tournament	Preserve diversity	Loss of diversity when the tournament size is large
Boltzmann	Global optimum achieved	Computationally expensive
Stochastic Universal Sampling	Fast method	Premature convergence
Elitism	Preserve the best individual in the population	The best individual can be lost due to crossover and mutation operators

Hasan and Hussein [23] described and compared the performance of three types of GA selection strategies to solve the Travelling Salesman Problem. Roulette Wheel Selection was used to solve large-sized problems while Tournament Selection was used for small-sized problems. The best option is to use a ranking system. Poonam and Aggarwal [24] achieved the best result when applying Roulette Wheel Selection with two-point crossover and Tournament Selection with a one-point crossover.

2.3 Crossover

Crossover is the process of crossbreeding between genes and two parents. The crossover operator is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). Crossover occurs during evolution according to a user-definable crossover probability. The well-known crossover operators are single-point crossover, two-point crossover, and uniform crossover [25]. A number of chromosomes can be represented with a reduced number of alleles. 1-point crossover, N-point crossover, Uniform crossover, and Arithmetic crossover are all potential crossover procedures in binary encoding. This is generally used in the Knapsack problem.
The type and example of a crossover in GA can be seen in Table 6.

Table 6
Type and example of a crossover

Type of crossover	Example
(i) Single-point Crossover A random crossover point is chosen in a single-point crossover. The genetic information of two parents who have progressed beyond that point will be exchanged [26]. To create the new offspring, it replaced both parents' tail array bits as shown in Figure 11. When a single point crossover occurs, the head and tail of one chromosome separate, and if both have good genetic material, none of the offspring will inherit both good traits [27].	 **Fig. 11.** Single-point crossover
(ii) Two-point Crossover Figure 12 shows that two or more random crossover points are chosen in a two-point crossover, and the genetic information of the parents is swapped according to the segments formed [26]. To produce new offspring, the parents' middle segment is replaced.	 **Fig. 12.** Two-point crossover
(iii) Uniform (Intermediate) Crossover Figure 13 shows that a parent cannot be decomposed into segments in a uniform crossing. Each gene in the parent can be handled separately. A researcher can choose whether to switch the gene with the same position on another chromosome at random [26].	 **Fig. 13.** Uniform (intermediate) crossover

(iv) Arithmetic Crossover
Figure 14 shows that this crossover operator joins the two parent chromosomes in a linear way. Two chromosomes are selected at random for crossover, resulting in two children who are a linear mix of their parents [27].

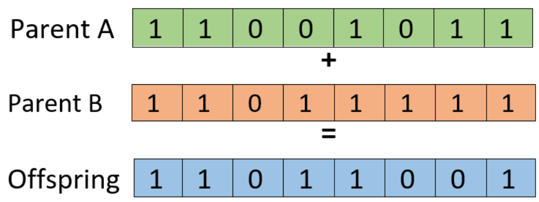


Fig. 14. Arithmetic crossover

Table 7 shows the comparison of different crossover techniques. From this table, it is observed that single-point and two-point crossover techniques are easy to implement while the uniform crossover is suitable for large subsets.

Table 7
Comparison of different crossover techniques

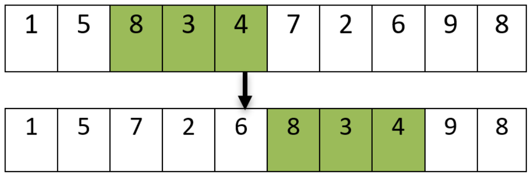
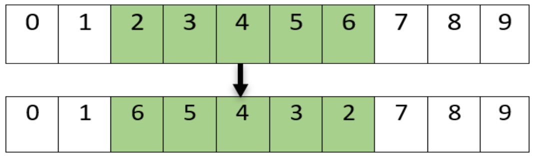
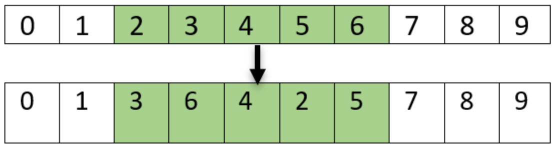
Crossover Techniques	Advantages	Disadvantages
Single-point	Easy to implement Simple	Less diverse solutions
Two-point	Easy to implement	Less diverse solutions Applicable on small subsets
Uniform	Unbiased exploration Applicable on large subsets Better recombination potential	Less diverse solutions

2.4 Mutation Operators

Mutation is the process of replacing one of the selected genes with a certain value. Mutations are operators that maintain genetic diversity from one population to the next. Shifts, simple inversions, and scrambling mutations are mutation controllers. The Evolutionary Algorithm (EA) community is becoming more aware of the significance of mutation and there is a rising interest in examining its characteristics more closely [28].

The Displacement Mutation operator replaces a substring of a particular individual solution within itself. The Simple Inversion Mutation operator inverts a randomly selected string and places it at a random location [16]. Whereas the Scramble Mutation operator serves as an individual solution in random order and checks whether the suitability value of the recently generated solution is improved or not by placing elements in a certain range. This article focuses on three types of mutations namely, Displacement Mutation, Simple Inversion Mutation, and Scramble Mutation. The type and example mutation of in GA may be shown in Table 8.

Table 8
Type and example of a mutation

Type of mutation	Example
(i) Displacement Mutation In individual solutions, Displacement Mutation is used to displace substrings as shown in Figure 15. The displacement places are chosen at random from the substrings calculated, resulting in both correct solutions and random displacement mutations [29]. Exchange mutations and insertion mutations are two types of Displacement Mutation. In the exchange operator and the insertion mutation, a part of the individual solution is either exchanged with another part or inserted at different sites [16].	 **Fig. 15.** Displacement Mutation
(ii) Simple Inversion Mutation Figure 16 shows that the Simple Inversion Mutation is like Scramble Mutation in that it selects a subset of genes, but instead of shuffling the selection, it simply inverts the entire string in the subset.	 **Fig. 16.** Simple Inversion Mutation
(iii) Scramble Mutation Figure 17 shows that the scramble mutation is a commonly used permutation operator where a random subset of genes in a chromosome are selected and their values are randomly shuffled.	 **Fig. 17.** Scramble Mutation

The mutation process aids in the prevention of GA descent towards local extremes [30]. This statement is supported by Sivanandam *et al.,* [12] where mutations work to improve beneficial genes that have been lost owing to genetic processes, but they also harm existing genetic information. The number of copies of the chosen gene will be tallied to determine which chromosomes and genes it resides on. The comparison of different mutation techniques in Table 9.

Table 9
Comparison of different mutation techniques

Mutation Techniques	Advantages	Disadvantages
Displacement Mutation	Easy to implement. Applicable for small problem instances	Risk of premature convergence
Simple-Inversion Mutation	Easy to implement	Premature convergence
Scramble Mutation	Affects a large number of genes Applicable on large problem instances	Disturbance in the population Deterioration of solution quality in some problems

3. Fitness Functions

The development of a viable fitness function is an important stage in GA because its performance is largely determined by the effort required to develop a workable fitness function [31]. The fitness function determines how near a given solution is to the ideal answer for the problem. It establishes the appropriateness of a solution. Each chromosome in the population will be counted for fitness in the GA to find the chromosome with the highest fitness value. GA is highly sensitive to fitness values, so it is critical to track fitness not only to avoid early concentration but also to keep the population diverse. Any fitness function should be able to meet the following criteria [32]:

(i) The fitness function must be defined precisely.
(ii) A reader should be able to grasp exactly how the fitness score is determined.
(iii) The fitness function should be well-implemented. The total efficiency of the GA will be diminished if the fitness function becomes the bottleneck of the algorithm.
(iv) The fitness function should be able to quantify how well a given solution fits the problem.
(v) The fitness function should produce understandable results.

Besides that, according to Helshani [33], the fitness function may be used to assess the overall performance of the operation rather than individual rules.

4. Applications

GA has been used successfully in a few application areas and various NP-hard tasks with high accuracy rates [29]. GA is also widely used for a wide variety of problems, including:

(i) Optimization

The term "optimization" refers to the process of determining input values in order to obtain the "optimal" output values. The term "best" has several meanings depending on the situation, but in mathematics, it refers to maximizing or reducing one or more objective functions by changing the input parameters [34]. In optimization, some sets of input are provided, and the output is generated based on the input as shown in Figure 18 [35].

Fig. 18. Processing of data

GA used in optimization includes numerical optimization and combined optimization such as Traveling Salesman Problem (TSP), Integrated Circuits (IC) Design, Work Scheduling, Video and Voice Optimization. The steps required in some GA optimization applications are as follows:

Step 1 - Model existing systems using WaterCAD, EPANET, etc.
Step 2 - Model calibration based on field measurements.
Step 3 - Determine future water needs to be met and the design and performance criteria to be met.
Step 4 - Identify possible options for new procurement or restoration of pipes, reservoirs, pumps, and valves as well as handling options.
Step 5 - Formulate a GA routine for the decision variables.
Step 6 - Connect the hydraulic model to the GA routine.
Step 7 - Run the GA and get input and instructions from the hydraulic expert.
Step 8 - Complete the generated alternatives and validate them.

(ii) Automatic Programming

GA for automated programming, among others, carries out the evolutionary process of computer programs in designing computational structures, such as cellular automata and isolation networks.

(iii) Machine Learning
GA has been successfully used to predict protein structure and design neural circuits to carry out the evolutionary process of rules in learning classroom systems or symbolic output systems. GA can also be used to control robots.
(iv) Economic Model
In economics, GA is used to model the process of innovation and development of breeding strategies.
(v) Immune System Model
Examples of the use of GA in this field are to model various aspects of the innate immune system, including somatic mutations during an individual's lifetime and finding multiple gene families throughout evolution.
(vi) Ecological Model
GA can also be used to model ecological phenomena such as the co-evolution of host parasites, symbiosis, and resource flow in ecology.

This article focuses on eight applications of GA: Scheduling Optimization, Traveling Salesman Problem (TSP), Support Vector Machine (SVM), Pipeline Network Optimization, Vehicle Routing Problem (VRP), Product Design, and Multimedia and Operation Management.

4.1 Scheduling Optimization

A schedule is a collection of several meetings at a particular time. Project duration should be emphasized during the planning stage to ensure optimal scheduling. Scheduling problems arise in various fields including construction, manufacturing, and software development. Due to its importance, project scheduling issues have been well researched by OR (Operations Research) and AI (Artificial Intelligence) researchers. The GA method has been widely used by previous researchers to tackle a variety of combination problems such as planning and production scheduling in the manufacturing industry. Apart from that, research proposed a multi-criteria optimization to schedule projects with cost and length limitations. To arrive at a suitable solution, they constructed a GA that uses co-evolution and Pareto principles [36]. Previous research has discussed how to create a good schedule by prioritizing minimum project duration. Some of the advantages of GA in problem-solving is it can be used to optimize discrete or continuous variables, is useful for analyzing large amounts of data, is beneficial in solving complex problems and provides a list of solutions to a problem. GA are also effective in tackling problems with the following characteristics:
(i) The problem space is large, quite complex, and quite difficult to understand.
(ii) No complicated calculation is required.
(iii) Researchers expect a solution that is good or acceptable. it does not have to be the optimal solution.
(iv) Problems related to time, such as problems in real-time systems.

Algorithms are necessary to optimize the duration of a project by scheduling activities. A chromosome, represented as a matrix, contains the sequence of activities and their start and completion times. The population is randomly formed, and the start and completion times are calculated. Crossbreeding and mutation processes are then carried out. Previous research has shown that these methods can produce a sequence of tasks that minimize objectives such as production time and meeting deadlines. Genetic algorithms have successfully solved scheduling problems like Job-shop Scheduling Problem, Flexible Job-shop Scheduling Problem, Distributed and Flexible Job-shop Scheduling Problem, No-wait Flow Shop Scheduling, and Integrated Process Planning and Scheduling.

4.1.1 Genetic algorithms operators used in scheduling optimization

These operators, including selection, crossover, mutation, and others, play a pivotal role in shaping the algorithm's performance and convergence characteristics. By dissecting and examining the intricacies of each operator, we intend to shed light on their specific roles and impacts within the context of scheduling problems.

(i) Initial Population

The first step in population development is to generate a random number based on the size of the chromosome. Machine numbers are used to calculate population size based on job numbers ($n \times m$). In the initial population, chromosomes are created by generating a random number of jobs with a limit on the appearance of numbers that is equal to the number of machines in order for new chromosomes to develop in a single population.

(ii) Selection

Reproduction is the stage of chromosomal string copying [37]. The process of selecting parental chromosomes for a population is known as selection. The fitness value is used during stain breeding and selection to determine which chromosome will be the parent. The higher the value of chromosomal fitness, the more likely it is to be chosen. Roulette Machine and Tournaments are the two most popular alternatives in JSP. A Roulette Machine is used to select parents based on the fitness value of each chromosome and random numbers are as follows [38]:

(a) Determine the time spent on each schedule (chromosome).
(b) Determine the schedule's overall duration (chromosome).
(c) Determine the fitness value of each chromosome.
(d) Determine how many chromosomes are there in total.
(e) Determine the fitness probability for each chromosome.
(f) Consider how much intelligence each chromosome has accumulated.

(iii) Crossover

The process of joining two people to form a new individual with a high fitness value is known as crossover [37]. By merging sections of the parent chromosome crosswise to generate a new chromosome, the crossover will modify different parts of the parent chromosome. Because the crossover process is governed by velocity and random values, not all chromosomal pairs undergo this crossover process.

(iv) Mutation

A mutation happens when one or more genes on a chromosome are replaced without the involvement of another chromosome. During evolution, mutations are carried out at a set rate. If the mutation rate is set too fast, the progeny will revert to primitive random search patterns [38]. However, if the rate is too slow, the process will be slowed, and the required result will not be achieved. The range of good mutations is between 0 to 0.3 [39]. If the mutation is too big or too little, it may be caught in the local optimum, making convergence difficult to achieve. At the bit stage, mutations are utilized to finish JSP [38].

4.2 Traveling Salesman Problem (TSP)

For future generations to be healthier, it is important for them to come from healthy families. This is because individuals who are in good physical shape are more likely to have beneficial mutations. To solve Traveling Salesman Problems (TSP), a similar concept is used where different solutions are combined to create a new and optimal solution. The TSP was first proposed in 1800 by William Rowan Hamilton and Thomas Kirkman, and it involves finding the shortest route between two places with the condition that each place is visited only once. There are several algorithms that can be used for TSP problems, including nearest neighbour heuristics, cheapest insertion heuristics, and GA. GA is advantageous for larger problems as it allows for faster computation times compared to other algorithms.

Ravichandran and Naganathan [40] addressed the problem of selecting route to a given destination on an actual map under a static environment by using GA. It is shown that the best route selection problem in network analysis can be solved with GA through efficient encoding, selection of fitness function, and various genetic operations. Helshani [33] solved TSP using GA to determine the optimum route on Google Maps. Lin *et al.,* [41] presented an improved Hybrid Genetic Algorithm (HGA) to solve two-dimensional Euclidean TSP, in which the crossover operator is enhanced with a local search.

4.2.1 Genetic algorithms operator used in TSP

In the realm of TSP, GA serves problem-solving tools. GA inspired by nature's evolution process, use a set of operators such as selection, crossover, and mutation to navigate through a vast solution space. These operators guide GAs in finding optimal or near-optimal solutions for complex scheduling challenges. The following are the GA operators used in TSP.

(i) Initial Population

Integer data representation is used in GA for TSP. Data are provided in the form of a series of integer sequences, in which one series represents individuals called chromosomes. Chromosomes consist of a set of genes that are integers. The genes in the chromosome represent the visiting position.

(ii) Evaluation

Evaluation is performed by calculating the distance between cities on a path that has been arranged in the chromosomes. Since each locus contains a number that represents each city, the calculation of suitability value is performed by summing the distance between the original city and the first locus, the city in the first locus and the second locus, and then returning to the original city. The evaluation process finds the distance between location genes and then adds them to the initial distance from each chromosome that has been generated.

(iii) Selection

Chromosome selection is performed so that only quality chromosomes can proceed to the next GA process. In the selection process, fitness parameters are used to determine how long a chromosome will survive. Selection is a crucial stage in genetic algorithms that determines whether a specific string will be involved in the reproduction process or not [42]. The ability of GA to progressively produce better chromosomes depends on the selective pressure applied to the population. Selection methods that are often used include Roulette Wheel Selection, Rank Selection, Elitism, and Tournament Selection. Figure 19 shows examples of TSP.

Journal of Advanced Research in Applied Sciences and Engineering Technology
Volume 40, Issue 1 (2024) 1-34

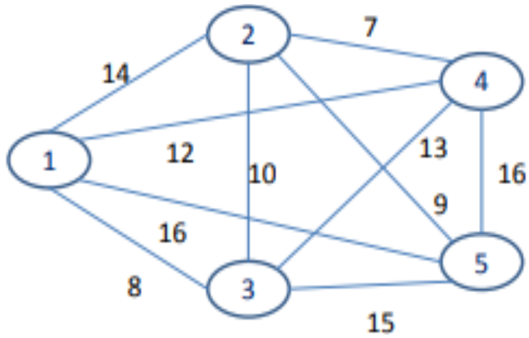


Fig. 19. Examples of TSP

Roulette Wheel Selection is the simplest and most widely used selection method. In this method, parents are selected based on fitness values where a greater fitness value is likely to be selected. Each chromosome is selected based on fitness value. Meanwhile, Elitism method selects individuals to be used in future generations based on the arrangement of intelligence values. While Stochastic universal sampling is effective for the Travelling Salesman issue, classic Roulette wheel selection becomes more efficient as the issue size grows [43]. This selection process is done by arranging all chromosomes in one generation and then taking the total population sizes desired. Each pathway in the initial population calculated distance, fitness value, fitness probability, and cumulative fitness probability.

(iv) Crossover

The hybridization process involves mating two chosen parents to create offspring. The offspring inherit genes from both parents. To ensure the arrangement of values on the chromosome remains consistent, a single-point crossover is used in conjunction with the Order Crossover (OX) operator. Chromosome crossing occurs in pairs, and if there is an odd number of chromosomes selected for crossing, some will be excluded.

(v) Mutations

Swap mutation is used in this case because permutation encoding is being used, which involves changing the values of two genes within a chromosome. The number of genes that will mutate needs to be determined beforehand in order to determine the length of a chromosome. The position of a mutated gene is determined by generating a random number based on the number of genes to be mutated. Reciprocal Exchange Mutation is a simple mutation method that randomly selects two positions and swaps their values.

4.3 Support Vector Machine (SVM)

SVM, a machine learning technique, is gaining popularity because of its attractive features and promising performance. It has been successfully used in various fields for both linear and non-linear classification problems. In order to classify data that cannot be separated linearly, kernel functions are employed with parameters that impact the accuracy of SVM. The optimal kernel parameters are determined using GA, through selection, crossover, and mutation, and are then used as model parameters for SVM. One of the challenges faced when using SVM is determining the best parameters. Setting the appropriate kernel parameters is essential for SVM to enhance its accuracy in classification and regression tasks. The parameters that should be optimized include penalty parameter (C) and kernel function parameters such as gamma (g) for the radial basis function (RBF) kernel. GA is recommended as the best parameter search method for SVM. The best C and g can be obtained through selection, crossover, and mutation, and the best parameters are used for the damage detection of SVM. The system of SVM optimized by GA is shown in Figure 8 [44]. Genetic method (GA) is a widely studied evolutionary method used for optimization challenges [45]. Figure 20 shows a linear separation of two classes with an SVM classifier. Samples on the margin are called the support vectors [46].

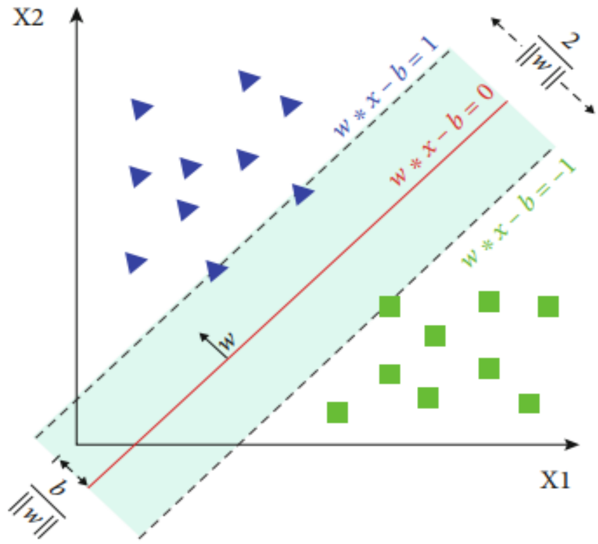


Fig. 20. Linear separation of two classes with an SVM classifier [46]

Algorithmic support vectors were developed to help decide whether these machines are capable of handling high-dimensional data sets and solving classification and regression problems with linear or nonlinear kernels. However, the SVM has problems in selecting the appropriate parameters to improve optimization. Thus, GA is used to select the appropriate parameters to be used in the SVM method.

SVM is a two-class classification method as shown in Figure 21, and this theory is based on ideas of structural risk. SVM uses kernel functions to map input data to high-dimensional space and find the optimal hyperplane to separate the data of the two classes [47]. The vector space should be separated by two different classes to obtain the optimal hyperplane. The optimal hyperplane is the furthest distance from the hyperplane of the two classes. SVM uses a kernel function to assist fit a hyperplane surface to the training data [48]. In linear separation problems, the optimal hyperplane can separate two different classes well and the vectors closest to the optimal hyperplane are called support vectors. SVM parameters used GA to successfully solve classification problems on 3 data namely UCI Machine Learning Repository (Image Letter Recognition, Pima Indians Diabetes, and Protein Localization Site). Syarif *et al.,* [49] found that the time required for GA to optimize SVM parameters is 15.9 times faster than the Grid Search method.

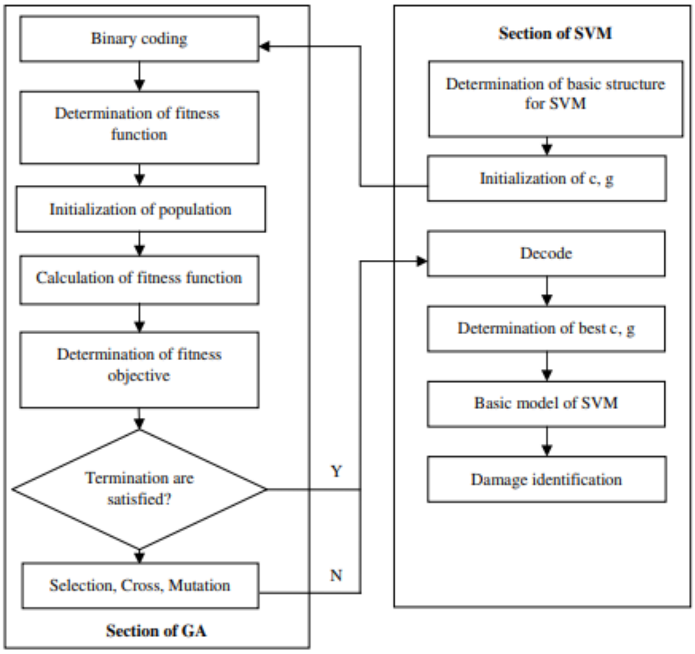


Fig. 21. SVM system optimized by GA

4.3.1 Genetic algorithms operator used in SVM

In the context of Support Vector Machines (SVM), Genetic Algorithm (GA) serve as a valuable tool for optimizing the selection of relevant features and tuning the hyperparameters, thereby enhancing the SVM's performance. One of the key GA operators employed in this context is the feature selection operator. This operator efficiently explores the high-dimensional feature space to identify the most informative features, consequently reducing the dimensionality and computational complexity of the SVM model. The selection, crossover, and mutation operators within the GA framework aid in systematically searching for the optimal combination of hyperparameters, thereby improving the SVM's ability to classify data accurately and efficiently. The following are the GA operators used in SVM.

(i) Fitness Function

Fitness function is an important factor in the evaluation and evolution of SVM. The fitness function directs the superordinated evolutionary learning process, which determines the likelihood that an individual can pass genetic information on to future generations. The most difficult part of the GA-based SVM parameter optimization process is designing a fitness function that produces SVM parameters that are dependable and effective for SVM models. K-fold cross-validation (CV) is a widely used approach for evaluating an SVM classifier's generalization capabilities. Instead of classification accuracy, cross-validation accuracy (CVA) is used as the fitness function in the evolutionary algorithm [50].

(ii) Selection

Selection is done in the form of a tournament with a two-player field. In addition, the Elitism technique is used to ensure that the best SVM is included in the following generation.

(iii) Crossover

The aim of this process is to combine the previously retained solutions to find interesting aspects (peaks and thresholds) of several solutions in new individuals [51]. The crossover operator is implemented as a uniform crossover, which means that all genes between two random sites on a chromosome are swapped between two genotypes representing parents for the two resulting new genotypes. For kernel aggregation and kernel exponent, the crossover can be used on chromosomes, whereas mutation can be used on all chromosomes. The actual utilization of a genetic procedure is determined by user-defined rates. Research suggested the use of a high rate of crossing over and a low rate of mutation [52].

(iv) Mutation

Mutation is carried out as a stepwise increment or decrement with a defined step size, resulting in a new value that falls between the minimum and maximum boundaries. Binary genes can be modified by flipping 0 to 1 and vice versa [52].

4.4 Pipeline Network Optimization

The pipeline is an important component in infrastructure and industry. For example, in clean water systems, GA can be used to determine the dimensions of pipeline hydraulic models and optimize new network planning. A relatively comprehensive approach to the use of GA for pipe network optimization has been developed over the last ten years [53]. Besides, GA is a powerful technique that can locate the lowest-cost network in a small number of hydraulic simulations compared to the search space's size. Furthermore, GA offers a variety of near-optimal options for a designer to evaluate. GA is simple to apply and could result in significant capital cost reductions for water delivery providers. The goal of the optimization model is to reduce the pipe network investment cost to ensure construction efficiency and reduce the discrepancy. The most economical set of pipe sizes can be determined through the GA method. In the problem of water pipeline optimization, the main purpose is to obtain the cheapest cost [54]. Nevertheless, the design of water pipeline networks is a difficult problem to solve. According to Jung and Karney [55], the use of the optimal design of a pipe network considering both steady and transient states using GA and particle swarm optimization (PSO). Integration of GA and PSO with a transient analysis technique can improve the search for hydraulic protection devices in a pipe network [56]. Both optimization programs GA and PSO were inspired by natural evolution and adaptation with excellent performance for solving moderately complex real-world problems which are highly nonlinear and demanding [55]. This statement is supported by Jung [57] that GA and PSO have a similar evolutionary history and produce optimal solutions. However, GA is very precisely used for solving complex and difficult optimization problems using conventional methods. In a pipe network optimization problem, the main notion of GA is to pick a population of initial solution points at random in the optimization space, then converge iteratively to better solutions until the required stopping requirements are met [58]. As with evolutionary processes in nature, a simple GA generally consists of three operators, namely: reproductive operator, crossover operator, and mutation operator. Four differences between GA and other methods are as follows:

(a) GA manipulates a set of encoded parameters.
(b) GA starts from a population of points and not from a single heuristic point.
(c) GA uses existing information directly and does not take advantage of derivatives or another additional knowledge.
(d) GA uses probability transition rules and not deterministic rules

To solve this problem, we will use eight different parts initializations of population, decoding of chromosomes, fitness values, parental selection, crossover, mutation, elitism, and population replacement. These parts work together to find the best solution. We can visualize how they are connected in Figure 22 [59]. The optimizer method will help us figure out the best sizes for the pipes.

This is important because smaller pipes can save money and create more pressure for a specific design [60].

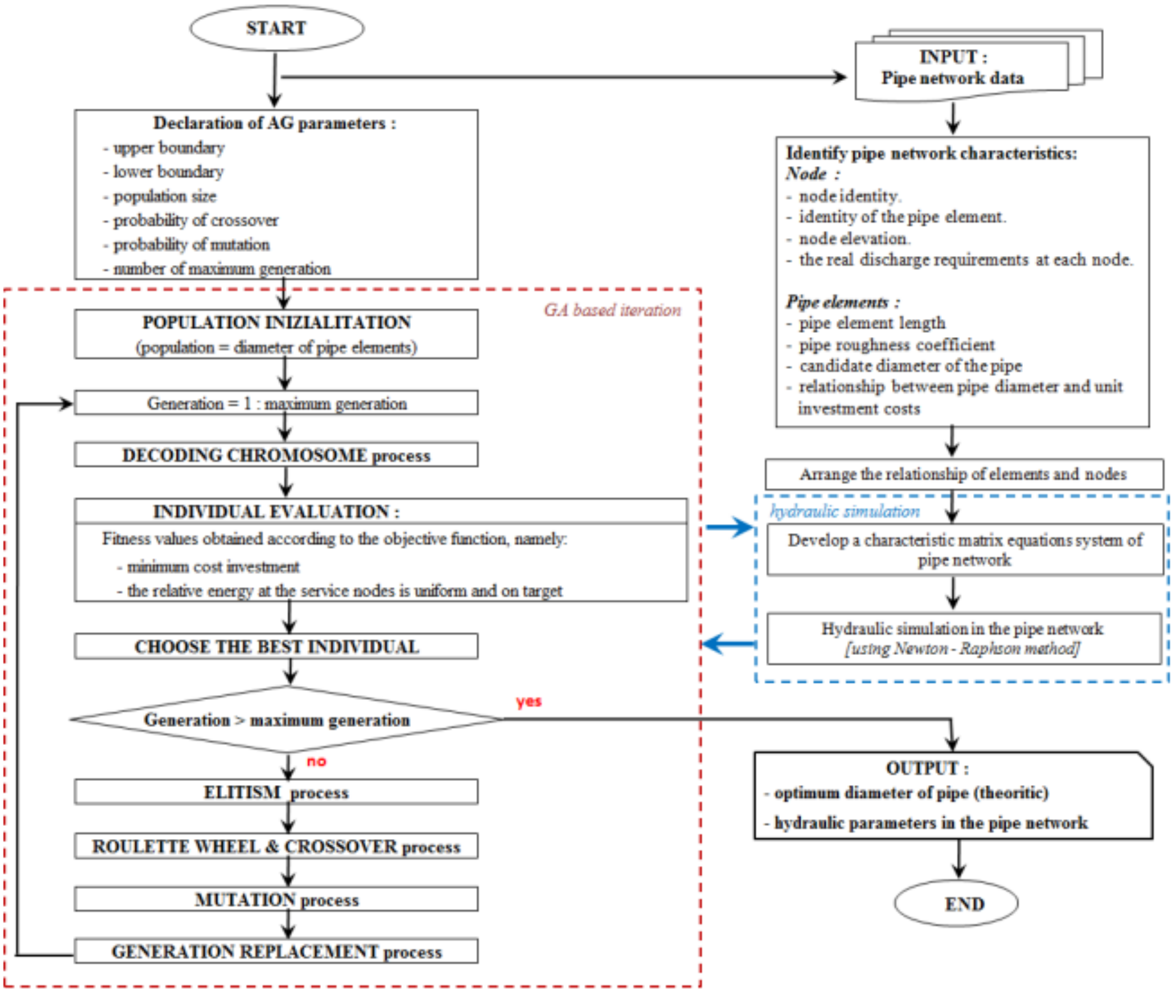


Fig. 22. The optimization model algorithm [59]

4.4.1 Genetic algorithms operator used in pipe network optimization

GA provides a variety of operators, including selection, crossover, and mutation, that are specifically designed to optimize pipe networks. These operators allow GA to experiment with and refine alternative network designs, pipe diameters, flow rates, and operating tactics over generations. By doing so, GA may efficiently balance competing objectives such as cost reduction, increased water distribution efficiency, and network resilience. The selection operator seeks superior solutions, the crossover operator combines promising elements from several designs, and the mutation operator adds diversity and adaptation to the expanding population. Thus, Genetic Algorithms play a critical role in pipe network optimization, providing a data-driven and automated way to address the numerous issues encountered by hydraulic engineers and urban planners in guaranteeing dependable and efficient water distribution systems. The following are the GA operators used in Pipe Network Optimization.

(i) Initial Population

A coding technique is used to generate an initial population of feasible solutions (pipe diameters). Coding is a crucial aspect of algorithm formulation in heuristic methods such as GA because it determines the size of the domain space and the level of discretization precision necessary [60]. GA use uses binary encoding to depict chromosome-like solutions. Therefore, an initial population of solutions is generated using a random number generator [61].

(ii) Selection

For the selection process, the Elitism method is used. Through crossover, the chosen solutions are merged to produce offspring at a predetermined rate (e.g., 0.60). A blended one-point crossover method is used to mix the variable values from two parents to improve the ability to generate new information [62]. At a randomly chosen crossing point, a single child variable value calculated from Eq. (1) ρ_{new}, is created by combining the values from the two parents [63].

$$\rho_{\text{new}} = round\ \{\gamma * \rho_{mn} + (1 - \gamma) * \rho_{dn}\} \tag{1}$$

(iii) Crossover

Each pair of strings in the new population is considered in the tum. Alternatively, a random crossover point is selected along the m-bit string (e.g., position 5). The digits from 6 to m of the 1st string are moved to the digit positions 6 to m of the 2nd string, while the corresponding digits of the 2nd string are moved to replace the end of the 1st string [61]. The crossover-blending method used in water distribution networks has the advantage of ensuring that the offspring variables (pipe diameters) fall within the range of commercially available diameters. Additionally, the coding scheme ensures that the variables always adhere to the velocity limitations based on the corresponding flow rates.

(iv) Mutation

The mutation operator is used to change a digit's value to another value by randomly selecting a digit location along the string. The mutation operator is also designed to change the diameter size for each parameter within the allowed specified values range with respect the commercial diameter boundaries and velocity constraints for each pipe [62].

4.5 Vehicle Routing Problem (VRP)

Vehicle Routing Problem (VRP) is a defining problem efficient and optimal route search i.e., using several vehicles to take off or pick up goods [64]. VRP is found in vehicle, service or goods distribution systems. Passenger capacity and total distance travelled must be considered in determining the route to be travelled by a vehicle. Among the components contained in VRP are customers, depot, guide, and vehicle traffic. VRP is one of the transportation problems in distributing products to consumers using vehicles with the aim of minimizing the predetermined distribution function [65]. VRP can be distributed to several types. The main problem in VRP is how to determine the route of the vehicle so that each customer is served exactly one vehicle, customer demand is met, and the total distance travelled is a minimum for all vehicles. Thus, the use of GA is important to find an optimal solution for a relatively short time [66]. The components needed to implement GA are the parameters, initial population, evaluation of fitness function, selection process and genetic operators such as crossover and mutation [67]. GA was implemented on the coordinated production and transportation problem in as a mixed-integer linear programming problem [67]. GA are procedures that resemble the evolutionary process (selection, crossover, and mutation processes). Data is presented in the form of a series of integer sequences, in which one series represents individuals or chromosomes which consist of a set of integer genes. The genes on the chromosome represents the customer being visited and the position of the gene represents the position of the visit; thus, the chromosome represents the path traversed by the vehicle. Figure 23 shows a flowchart of the VRP solution using GA [68].

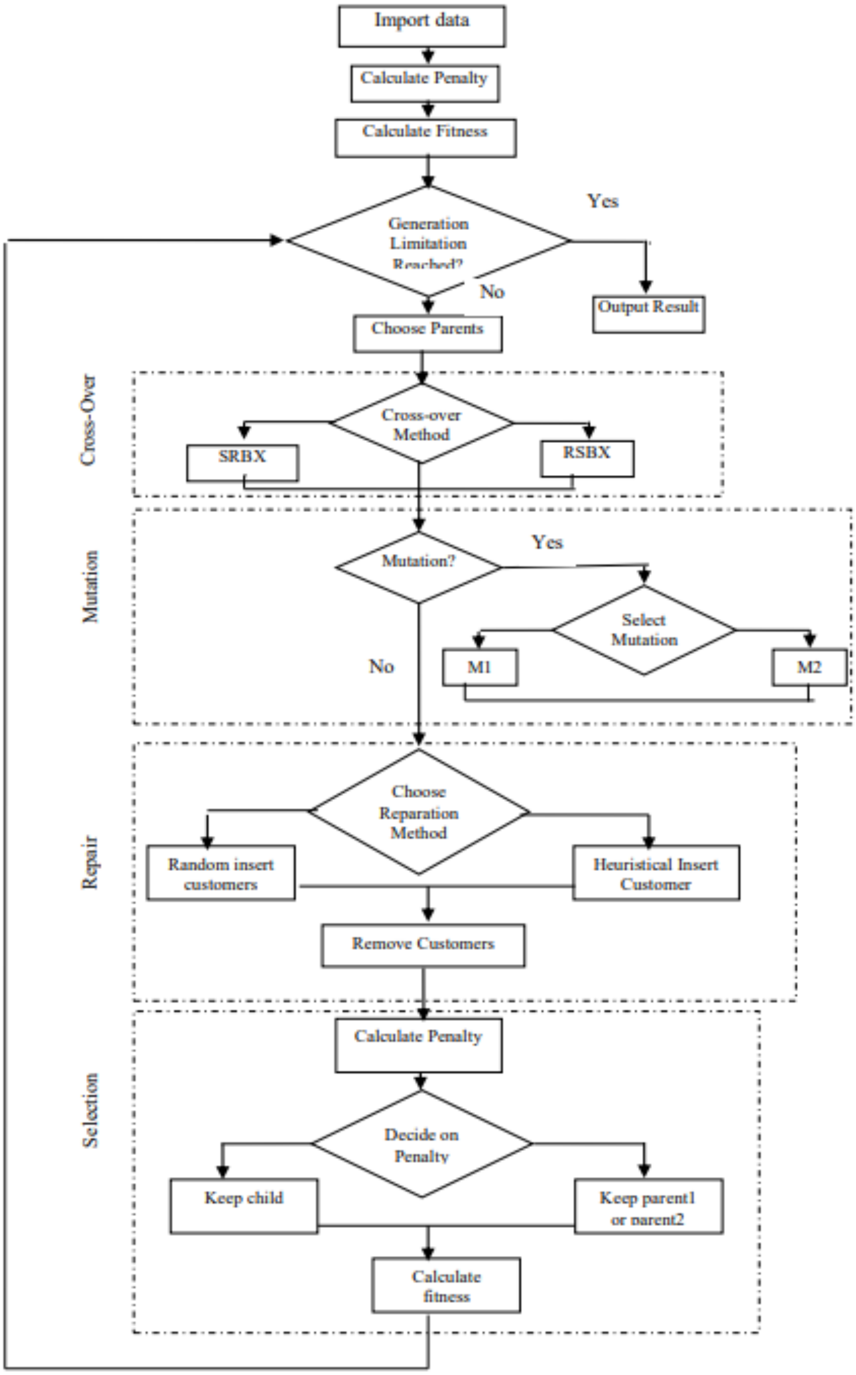


Fig. 23. Flowchart of VRP solution using GA

4.5.1 Genetic algorithms operator used in Vehicle Routing Problem (VRP)

Genetic Algorithms (GAs) are used to identify efficient solutions to the Vehicle Routing Problem (VRP). These operators include selection, crossover, and mutation. The selection operator discovers promising routes based on parameters such as distance, truck capacity, and time limits. The mutation operator adds diversity by exploring new paths. GA thrives not just in the conventional VRP but also in its complicated versions, making them excellent instruments in logistics and transportation optimization. Using these operators, GA methodically seek for ideal vehicle routes that minimize costs, maximize resource utilization, and conform to operational limitations, solving the practical issues of contemporary logistics. The following are the GA operators used in VRP.

(i) Initial Population

In each generation, the total size of the population that is transferred, mutated, and crossed over must equal to the initial population size. All additional parameters, such as number of vehicles, their capacity, locations with coordinates, and number of goods to be delivered, are provided as additional parameters, and are usually passed to the algorithm in the third module, along with the fitness function, because it is almost exclusively that function that uses them [69].

(ii) Selection

Selection guarantees that the most capable individuals, regarded as better solutions, have a higher chance of producing offspring. The only formal condition that must be met in most cases is that the fittest individuals have a higher chance of being chosen [70]. Examples of the selection process in VRP include:

(a) Roulette Wheel Selection (RWS)
(b) Elitism Selection (ES)
(c) Rank Selection (RS)
(d) Stochastic Universal Sampling Selection (SUSS)
(e) Tournament Selection (TS)

GA-based optimization system for drinking water distribution indicated that selection using the Elitism method produced a fitness value of 0.08294, which is significantly better than selection using Roulette Wheel (0.0583) [71].

(iii) Crossover

Crossover is a technique used in genetic algorithms to change the programming of a chromosome in the next generation. It is similar to biological crossover and reproduction, which are fundamental concepts in GA. The existing crossover methods do not allow for the exchange of genetic material between two parents. Instead, they combine information from two individuals to create a new offspring [68]. In this research, the crossover operator is used to improve specific gene sections in a chromosome by inheriting them from another chromosome. This is done to enhance the fitness of the offspring and preserve the best individuals [72]. The probability of using a specific crossover approach can be configured.

(iv) Mutation

Mutation is a genetic operator that is used to preserve genetic variation in a population of chromosomes from one generation to the next. It is comparable to biological mutation from Abdel-Kader [68] because as stated by Barba Rodríguez *et al.,* [73] mutation is the process of occasionally shifting bit values, which prevents the loss of a desirable potential genetic feature. In larger situations, the savings GA approach, which has a lower fitness variance, has generated relatively decent results [74]. Although the vertex sequencing technique converges to better solutions in small problems such as the 50 customers problem, the power of savings GA manifests itself in larger problems where the local optimizer used in the vertex sequencing method ceases to be effective [74].

4.6 Product Design

Product design is a critical activity with a positive impact on the performance of a firm. A firm will use optimal solution techniques for the design or redesign of a product. GA is applied in such cases to find the optimal solution as it is one of the heuristic algorithms that can be used to solve multi-objective problems. Heuristic techniques have been proposed for such optimal solutions since the optimal product design uses combined analysis data which is an NP-hard problem. It is not easy for product designers to determine the optimal settings of design attributes for producing effective product design because effective design is an important aspect in the development of new products to provide competitiveness in the market [75]. Failure to design a new product will negatively impact a firm, hence it is important to produce optimal product design before marketing the product. Research on the market of a product and its importance to consumers must be done by the industry before the manufacture of the product.

In order to solve a problem using GA, one needs to formulate the problem using a string structure. In the product design problem, we consider a chromosome (string) to represent a product. The attributes of the product would correspond to genes and its level. For example, soap product is represented by three attributes A1 (shape), A2 (colour), and A3 (scent) where attribute A1 has three levels (rectangular, square, spherical), attribute A2 has four levels (red, green, yellow, white) and attribute areas A3 three levels (fruity, floral, antiseptic). Evaluation is a settlement process, and it is an iterative procedure. Each product will be evaluated during the initial population stage and the evaluation is performed using the utility of the valuable parts obtained from the combined analysis. Searches are performed by combining chromosomes and then processing genetic controllers (selection, crossing, and mutation) by initiating genetic parameters (population size, crossover rate, mutation rate, and number of generations). GA is used in product design to solve problems such as:

(a) Parameter Optimization

Computational of fitness is a challenge for parameter optimizers. Previous studies used algorithms to identify solutions and find optimal solutions [76]. According to observations by Doyle *et al.,* [77], problems with more than 10 design variables are often expensive to assess. One approach to reducing costs is to use meta-models rather than simulation-based models. Mukhtar *et al.,* [78] conducted a kriging-assisted multi-objective GA in which a meta-model based on Gaussian process regression was used to evaluate several designs.

(b) Shape Optimization

Form optimization involves a large number of variables and requires expensive evaluations. Compressor blades, haptic devices, synchronous generator pole shapes, nozzle shapes, and free-form surfaces are part of the design. Costly calculation of optimization is addressed through the integration of game theory as presented by Jang *et al.,* [79] and Peng *et al.,* [80]. GA applications can reduce design evaluation where the main goal of affective design is to determine the optimal settings of design attributes to achieve maximum customer satisfaction. According to Mishra *et al.,* [81], among the optimization techniques that can be used for this purpose are linear programming and nonlinear programming. Alternative approaches such as GA have also been used to optimize products [82]. Therefore, based on previous studies, the selection of GA for design optimization is based on an undeniable heuristic algorithm.

4.6.1 Genetic algorithms operator used in product design

By utilizing specialized operators, GA function as essential instruments in the area of product development. These operators, which include selection, crossover, and mutation, are critical in the evolution and optimization of product designs. The selection operator discovers interesting design alternatives, whereas the crossover operator, similar to genetic recombination, mixes characteristics from chosen designs to develop new possibilities. The mutation operator adds diversity to the designs, allowing for the investigation of new solutions. The following are the GA operators used in product design.

(i) Initial Population

GA selection is quite precise in sustaining the selected chromosomal population. This is because population size is crucial in determining population variety at the start of a race as well as the time required to complete it. Furthermore, the size of the population can influence the number of optimal options available. According to Julirose [83], the increase in population means the run will take longer to complete. As a result, the size of the population is determined by the number of optimal solutions desired by the designer as well as the time required to calculate them.

(ii) Selection

Selection stress is an important aspect in determining the rate of GA in deciding. The Tournament Selection method was applied in GA where two responses from the public were chosen at random and compared [83]. The winner of this approach is selected by checking the solution with the highest rating. If both solutions have the same rating, the highest score (biggest congested distance) determines the winner. If both options provide the same outcome, the winner is chosen at random based on the crowd's position and distance.

(iii) Crossover

A pair of replicated product profiles is chosen, and genetic material between the two strings is transferred at a certain point on the string, resulting in offspring. Balakrishnan and Jacob [84] agreed with this assertion that the genetic material is changed in a randomly selected cross of string pairings from the resulting set of strings to produce offspring. According to Julirose [83], the type of cross-operators utilized, such as simple one-point, simple multi-point, and uniform random, might affect the efficacy of GA.

(iv) Mutation

A product profile is picked at random from the population at this point, and the value at a specific spot in the string is changed. The number of mutations in a generation must be balanced between the desire to maintain it high to cover more search space and the want to keep it low to avoid a basically random search [84].

4.7 Multimedia

GA is also applied in the field of multimedia for image processing, video processing, gaming, and medical imaging. GA is used to solve the problem of the relatively long computation time required to separate images. According to Sun *et al.,* [85], GA is used because it has the best search capability and is able to enhance natural contrast and magnify images [86]. In addition, to producing sound from the image provided, GA is also required. The use of GA in multimedia can remove haze, fog, and smoke from resulting images [87]. There were some problems during video sharing which can be solved by GA [88]. In addition, according to Ahmadi *et al.,* [89], GA can also be used for gesture recognition.

In games, GA is used for problem-solving such as route planning because it considers environmental constraints as well as obstacles in reaching a given destination. In addition, GA can record route planning problems through two-dimensional playground coordinate points. This statement is supported by Burchardt and Salomon [90] who researched GA route planning for football games. GA is used in medicine for medical imaging, specifically for edge detection in MRI and pulmonary nodules detection in CT scan images [91]. According to Shabankareh and Shabankareh [92], GA is also used to solve problems in biomechanics where it can be used to predict pathology during an examination. This is supported by Sari and Tuna [93] that used GA to solve problems during biomechanical treatment. GA is also utilized to treat breast cancer, where positron emission tomography (PET) pictures and magnetic resonance imaging (MRI) images are coupled with GA to provide coloured images of the tumour [94].

4.7.1 Genetic algorithms operator used in multimedia

GA provide a diverse approach in the realm of multimedia by utilizing specialized operators. These operators, which include selection, crossover, and mutation, are used in image and video processing, content optimization, and compression, among other things. The selection operator discovers prospective multimedia solutions, whereas the crossover operator, similar to genetic recombination, mixes elements from chosen media to produce new variants. The mutation operator adds variety, allowing for the study of new multimedia representations. The following are the GA operators used in multimedia.

(i) Initial Population

In cancer detection, an initial population that is reasonably diverse is suitable for simpler settlement space exploration. This, like any other step-by-step optimization process, can improve the algorithm's convergence speed. Factors involved in this phase include a few peaks, collection of peaks, and related peak thresholds [95]. According to Dash and Liu [96], peaks having large amplitudes between diverse spectra are more likely to be beneficial for discrimination. Peaks were then chosen based on their amplitude, and a random number of peaks was determined.

(ii) Selection

Individuals are evaluated based on their fitness worth, with the best receiving the highest ranking.

(iii) Crossover

Crossover is a random technique for exchanging genes between two chromosomes utilizing one or two sites of crossover. According to Mansoori *et al.,* [95], the goal of this stage is to combine the previously retained solutions to find interesting aspects (peaks and thresholds) of several solutions in new individuals. It is important to note that this stage is unrelated to optimization; in other words, a crossover might provide both excellent and bad results.

(iv) Mutation

Mutation is defined as a change in the value of a vector that corresponds to the solution that has been chosen. For each generation, the mutation rate (the rate at which a solution will undergo mutation) is specified as m. The chance of one mutation per individual is commonly set to 0.9 at the beginning of the algorithm [88]. Mutations will be separated into three types in cancer detection, namely Peak Removal, Peak Addition, and Threshold Relocation [95].

4.8 Operation Management (OM)

GA have shown to be a powerful technique for solving complicated optimization challenges in the realm of operations management. To systematically investigate and develop operational techniques, GA employ a set of specialized operators like as selection, crossover, and mutation. These algorithms may be used to solve a variety of operational problems, such as supply chain management, production scheduling, and facility layout optimization. Operational managers may improve overall operational performance by leveraging the power of GA to automate the search for efficient solutions, optimize procedures, and make data-driven choices. However, in this article, operational management is focused as follows:

(i) Facility Layout

According to Datta *et al.,* [97], crossover operators and modified GA mutations can be used to solve single-line facility layout (SRFLP) problems. GA is used for large-sized problems consisting of 60–80 instances. However, each solution to the problem depends on the parameters. This statement is supported by Datta [98] which states that GA is suitable for a wide range of facility layout products. GA is able to decrease the total cost by 7.2% compared to other algorithms due to the use of mutation and heuristics operators. In addition, a GA island model is also proposed to solve FLP [99]. The proposed technique maintains population diversity and generates better solutions than existing techniques.

(ii) Forecasting and Network Design

Among the important components in operation management are forecasts such as financial trade, logistics demand, and tourist arrivals. According to Chen *et al.,* [100] and Sermpinis *et al.,* [101], GA has been hybridized with the regression of support vectors, fuzzy, sets, and neural networks (NNs) to enhance its predictive ability. Besides, GA hybridized with other techniques for problem-solving such as multiple products and multiple periods [102]. According to Shi *et al.,* [103], GA is also used to optimize costs, profits, and carbon emissions.

(iii) Inventory Control

Inventory control plays an important role in operation management. According to Lee [17], backorders and lost sales are the two main issues in inventory control. Location inventory is used to find out the number and location of warehouses [104]. Various problems encountered and inventory control. The use of GA helps improve the design and in turn, solve inventory control problems.

4.8.1 Genetic algorithms operator used in Operation Management (OM)

In the realm of operation management, GA serve as effective problem-solving tools through the application of specialized operators. These operators, including selection, crossover, and mutation, play a pivotal role in optimizing various operational processes. The selection operator identifies promising solutions, while the crossover operator combines elements from selected strategies, much like genetic recombination, to create new approaches. The mutation operator introduces variability, allowing the exploration of innovative operational methods. GA enable operational managers to systematically explore and refine strategies, whether it's in supply chain logistics, production scheduling, or resource allocation, ultimately improving efficiency and decision-making in operations. This paragraph highlights the significance of GA and their operators in the realm of operation management by automating the search for optimal or innovative solutions to complex operational challenges. The following are the GA operators used in operation management.

(i) Initial Population

At the early population stage, two operations are required to determine the population size in the chromosome. The first process is to choose the starting population chromosome on a random basis, which means that each facility permutation can be chosen. By modifying the layout of the drawing and inserting it in the appropriate spot, different chromosomes can be formed for the original population.

(ii) Selection

The roulette Wheel Selection method is used to select the chromosomes due to its ease of implementation. Using this method, a random number between 0 and 1 will be created at each parent chromosome selection. According to Sun *et al.,* [105], the rank-based wheel selection mechanism in the proposed method is a modified version of the model.

(iii) Crossover

A crossover is a random process used to solve challenges on facility layout. At the beginning and end of the truncation section, each parent pair coded one line will yield two random integers. The heuristic technique, on the other hand, selects a group of facilities at random for each couple of parents and then selects a rectangular region as the cutting section that covers the greatest number of facilities in that group. The crossover is chosen from a legitimate cross-set for problems such as warehouse-retailer assignments. In the warehouse and retailer divisions, this set consists of border points between each time period. This is necessary to ensure that the offspring are viable.

(iv) Mutation

Mutations are used to search the solution space for new solutions. According to Hiassat *et al.,* [104], swap mutation and maximum warehouses open are the mutation operators used by GA to solve the warehouse-retailer assignment. For exchange mutations, a process occurs in which two genes are selected at random over a period and traded (according to value) throughout all time periods. Hiassat *et al.,* [104] selected chromosomes with fewer open warehouses as a mutation maximum warehouse. This is because chromosomes would find a method to organize retailers in such a way that better solutions could be found if more warehouses were required.

5. Conclusions

GA are intriguing and powerful relatives of optimization algorithms inspired by biological evolution concepts. This article takes a comprehensive tour of the world of GA, providing a complete overview of its functioning and numerous uses. GAs use a set of operators, such as selection, crossover, and mutation, to simulate the processes of natural selection, recombination, and adaptation, allowing them to navigate complicated solution spaces. The application of GA is growing rapidly in various fields such as Scheduling Optimization, Traveling Salesman Problem (TSP), Support Vector Machine (SVM), Pipeline Network Optimization, Vehicle Routing Problem (VRP), Product Design, and Operation Management. The use of GA as one of the popular meta-heuristics methods is well suited for the following problems:

(i) NP-hard problems with a simple statement and structure.
(ii) Challenging practical situations for which no perfect solutions are known; and
(iii) Well-known problems that have previously been studied require more accurate and computationally efficient answers.

Future research should apply GA to solve facility layout problems. A facility layout problem is an optimization problem that arises in a variety of situations, including shelf space allocation, equipment placement on a factory floor, and hospital or school layout designs.

Acknowledgment
This research was not funded by any grant.

References

[1] Kumar, Rakesh. "Novel encoding scheme in genetic algorithms for better fitness." *International Journal of Engineering and Advanced Technology* 1, no. 6 (2012): 214-219.
[2] Michalewicz, Zbigniew, Cezary Z. Janikow, and Jacek B. Krawczyk. "A modified genetic algorithm for optimal control problems." *Computers & Mathematics with Applications* 23, no. 12 (1992): 83-94. https://doi.org/10.1016/0898-1221(92)90094-X
[3] Edward, Guillen, Camargo Yeison, and Estupiñán Paola. "Hybrid approaches in network optical routing with QoS based on genetic algorithms and particle swarm optimization." *ACEEE International Journal of Network Security* 2, no. 4 (2011): 15-18.
[4] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." *IEEE Computational Intelligence Magazine* 1, no. 4 (2006): 28-39. https://doi.org/10.1109/CI-M.2006.248054
[5] Dhiman, Gaurav, and Vijay Kumar. "Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications." *Advances in Engineering Software* 114 (2017): 48-70. https://doi.org/10.1016/j.advengsoft.2017.05.014
[6] Dhiman, Gaurav, and Vijay Kumar. "Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems." *Knowledge-Based Systems* 150 (2018): 175-197. https://doi.org/10.1016/j.knosys.2018.03.011
[7] Dhiman, Gaurav, and Vijay Kumar. "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems." *Knowledge-Based Systems* 165 (2019): 169-196. https://doi.org/10.1016/j.knosys.2018.11.024
[8] Mitchell, Melanie, John Holland, and Stephanie Forrest. "When will a genetic algorithm outperform hill climbing." *Advances in Neural Information Processing Systems* 6 (1993).
[9] Yang, Xin-She. *Nature-Inspired Optimization Algorithms*. Elsevier, 2014. https://doi.org/10.1016/B978-0-12-416743-8.00010-5
[10] Gorunescu, Florin, Marina Gorunescu, and Ruxandra Gorunescu. "A metaheuristic GAs method as a decision support for the choice of cancer treatment." *Siberian Journal of Computational Mathematics* 7, no. 4 (2004): 301-307.
[11] Pereira, André G. C., José A. F. Roveda, Luiz Amorim Carlos, Viviane Simioli Medeiros Campos, and Sandra R. M. M. Roveda. "Convergence analysis of an elitist non-homogeneous genetic algorithm with mutation probability adjusted by a fuzzy controller." In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pp. 19-23. IEEE, 2013. https://doi.org/10.1109/IFSA-NAFIPS.2013.6608368
[12] Sivanandam, S. N., S. N. Deepa, S. N. Sivanandam, and S. N. Deepa. *Genetic algorithms*. Springer Berlin Heidelberg, 2008.
[13] Goldberg, David E. *Genetic algorithm in search optimization and machine learning*. Addison Wesley, 1989.
[14] Kar, Chinmoy, Vineet Kumar Rakesh, Tapas Samanta, and Sreeparna Banerjee. "A New Approach to Grid Scheduling using Random Weighted Genetic Algorithm with Fault Tolerance Strategy." *International Journal of Computer Applications* 975 (2012): 8887. https://doi.org/10.5120/7524-0632
[15] Sharma, Poonam, and Amit Wadhwa. "Analysis of selection schemes for solving an optimization problem in genetic algorithm." *International Journal of Computer Applications* 93, no. 11 (2014). https://doi.org/10.5120/16256-5714
[16] Jebari, Khalid, and Mohammed Madiafi. "Selection methods for genetic algorithms." *International Journal of Emerging Sciences* 3, no. 4 (2013): 333-344.
[17] Lee, Carmen Kar Hang. "A review of applications of genetic algorithms in operations management." *Engineering Applications of Artificial Intelligence* 76 (2018): 1-12. https://doi.org/10.1016/j.engappai.2018.08.011
[18] Abdulal, Wael, and S. Ramachandram. "Reliability-aware genetic scheduling algorithm in grid environment." In *2011 International Conference on Communication Systems and Network Technologies*, pp. 673-677. IEEE, 2011. https://doi.org/10.1109/CSNT.2011.145
[19] Saini, Nisha. "Review of selection methods in genetic algorithms." *International Journal of Engineering and Computer Science* 6, no. 12 (2017): 22261-22263.

[20] Miller, Brad L., and David E. Goldberg. "Genetic algorithms, tournament selection, and the effects of noise." *Complex Systems* 9, no. 3 (1995): 193-212.
[21] Chen, Mu-Chen, and Chia-Ping Lin. "A data mining approach to product assortment and shelf space allocation." *Expert Systems with Applications* 32, no. 4 (2007): 976-986. https://doi.org/10.1016/j.eswa.2006.02.001
[22] Saha, Sayandeep, Rajat Subhra Chakraborty, Srinivasa Shashank Nuthakki, Anshul, and Debdeep Mukhopadhyay. "Improved test pattern generation for hardware trojan detection using genetic algorithm and boolean satisfiability." In *Cryptographic Hardware and Embedded Systems--CHES 2015: 17th International Workshop*, Saint-Malo, France, September 13-16, 2015, Proceedings 17, pp. 577-596. Springer Berlin Heidelberg, 2015. https://doi.org/10.1007/978-3-662-48324-4_29
[23] Hasan, Zainab Falah, and Asraa Abdalluh Hussein. "Heart Disease Classification by Genetic Algorithm." *Journal of Babylon University, Pure and Applied Sciences* 9 (2016).
[24] Poonam, Maitreyee Dutta, and Naveen Aggarwal. "Meta-heuristics based approach for workflow scheduling in cloud computing: a survey." In *Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2015*, pp. 1331-1345. Springer India, 2016. https://doi.org/10.1007/978-81-322-2656-7_121
[25] Kaya, Yılmaz, and Murat Uyar. "A novel crossover operator for genetic algorithms: ring crossover." *arXiv preprint arXiv:1105.0355* (2011).
[26] Soon, Gan Kim, Tan Tse Guan, Chin Kim On, Rayner Alfred, and Patricia Anthony. "A comparison on the performance of crossover techniques in video game." In *2013 IEEE International Conference on Control System, Computing and Engineering*, pp. 493-498. IEEE, 2013. https://doi.org/10.1109/ICCSCE.2013.6720015
[27] Kora, Padmavathi, and Priyanka Yadlapalli. "Crossover operators in genetic algorithms: A review." *International Journal of Computer Applications* 162, no. 10 (2017). https://doi.org/10.5120/ijca2017913370
[28] De Falco, Ivan, Antonio Della Cioppa, and Ernesto Tarantino. "Mutation-based genetic algorithm: performance evaluation." *Applied Soft Computing* 1, no. 4 (2002): 285-299. https://doi.org/10.1016/S1568-4946(02)00021-2
[29] Bisen, Dhananjay, Sneha Mishra, and Praneet Saurabh. "K-means based cluster formation and head selection through artificial neural network in MANET." *Research Square* (2021). https://doi.org/10.21203/rs.3.rs-667651/v1
[30] Selvanathan, N., and Wee Jing Tee. "A genetic algorithm solution to solve the shortest path problem in OSPF and MPLS." *Malaysian Journal of Computer Science* 16, no. 1 (2003): 58-67.
[31] Razavi, Sayede Houri, E. Omid Mahdi Ebadati, Shahrokh Asadi, and Harleen Kaur. "An efficient grouping genetic algorithm for data clustering and big data analysis." *Computational Intelligence for Big Data Analysis: Frontier Advances and Applications* (2015): 119-142. https://doi.org/10.1007/978-3-319-16598-1_5
[32] Kanduč, Zoran. "Machines, humans and the question of control." In *Big Data, Crime and Social Control*, pp. 75-90. Routledge, 2017. https://doi.org/10.4324/9781315395784-4
[33] Helshani, Laurik. "An android application for Google map navigation system, solving the travelling salesman problem, optimization throught genetic algorithm." In *Proceedings of FIKUSZ Symposium for Young Researchers*, p. 89. Óbuda University Keleti Károly Faculty of Economics, 2015.
[34] Ting, Wong, and Chung. "Investigation of hybrid genetic algorithm/particle swarm optimization approach for the power flow problem." In *2005 International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 436-440. IEEE, 2005. https://doi.org/10.1109/ICMLC.2005.1526986
[35] Satrio, Pujo, Teuku Meurah Indra Mahlia, Niccolo Giannetti, and Kiyoshi Saito. "Optimization of HVAC system energy consumption in a building using artificial neural network and multi-objective genetic algorithm." *Sustainable Energy Technologies and Assessments* 35 (2019): 48-57. https://doi.org/10.1016/j.seta.2019.06.002
[36] O'Malley, Kathleen G., Felix Vaux, and Andrew N. Black. "Characterizing neutral and adaptive genomic differentiation in a changing climate: The most northerly freshwater fish as a model." *Ecology and Evolution* 9, no. 4 (2019): 2004-2017. https://doi.org/10.1002/ece3.4891
[37] Kelly Jr, James D., and Lawrence Davis. "A Hybrid Genetic Algorithm for Classification." In *IJCAI*, vol. 91, pp. 645-650. 1991.
[38] Soleimani, Hamed, and Govindan Kannan. "A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks." *Applied Mathematical Modelling* 39, no. 14 (2015): 3990-4012. https://doi.org/10.1016/j.apm.2014.12.016
[39] Tsujimura, Yasuhiro, Yuichiro Mafune, and Mitsuo Gen. "Effects of symbiotic evolution in genetic algorithms for job-shop scheduling." In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pp. 6-pp. IEEE, 2001.
[40] Ravichandran, S., and E. R. Naganathan. "Dynamic scheduling of data using genetic algorithm in cloud computing." *International Journal of Computing Algorithm* 2, no. 01 (2013): 127-133.

[41] Lin, Bao Lin, Xiaoyan Sun, and Sana Salous. "Solving travelling salesman problem with an improved hybrid genetic algorithm." *Journal of Computer and Communications*. 4, no. 15 (2016): 98-106. https://doi.org/10.4236/jcc.2016.415009

[42] Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." *Multimedia Tools and Applications* 80 (2021): 8091-8126. https://doi.org/10.1007/s11042-020-10139-6

[43] Sharma, Sonal, and Kusum Gupta. "Solving the traveling salesmen problem through genetic algorithm with new variation order crossover." In *2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pp. 274-276. IEEE, 2011. https://doi.org/10.1109/ETNCC.2011.6255903

[44] Liu, Han-Bing, and Yu-Bo Jiao. "Application of genetic algorithm-support vector machine (GA-SVM) for damage identification of bridge." *International Journal of Computational Intelligence and Applications* 10, no. 04 (2011): 383-397. https://doi.org/10.1142/S1469026811003215

[45] Zhang, Daqing, Jianfeng Xiao, Nannan Zhou, Mingyue Zheng, Xiaomin Luo, Hualiang Jiang, and Kaixian Chen. "A genetic algorithm based support vector machine model for blood-brain barrier penetration prediction." *BioMed Research International* 2015 (2015). https://doi.org/10.1155/2015/292683

[46] Tapak, Leili, Saeid Afshar, Mahlagha Afrasiabi, Mohammad Kazem Ghasemi, and Pedram Alirezaei. "Application of genetic algorithm-based support vector machine in identification of gene expression signatures for psoriasis classification: a hybrid model." *BioMed Research International* 2021 (2021). https://doi.org/10.1155/2021/5520710

[47] Aydin, Ilhan, Mehmet Karakose, and Erhan Akin. "A multi-objective artificial immune algorithm for parameter optimization in support vector machine." *Applied Soft Computing* 11, no. 1 (2011): 120-129. https://doi.org/10.1016/j.asoc.2009.11.003

[48] Xu, Li, Xiangjun Liu, and Lixi Liang. "A pore network model reconstruction method via genetic algorithm." *Journal of Natural Gas Science and Engineering* 21 (2014): 907-914. https://doi.org/10.1016/j.jngse.2014.09.038

[49] Syarif, Iwan, Adam Prugel-Bennett, and Gary Wills. "SVM parameter optimization using grid search and genetic algorithm to improve classification performance." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 14, no. 4 (2016): 1502-1509. https://doi.org/10.12928/telkomnika.v14i4.3956

[50] Wei, Li, Chen Junli, Shi Xinhua, and Liu Anna. "An optimization beam-position selection method based on genetic algorithm for Distributed Satellite-borne SAR system." In *Proceedings of 2011 IEEE CIE International Conference on Radar*, vol. 1, pp. 895-898. IEEE, 2011.

[51] Sakib, Shadman, Abyaz Kader Tanzeem, Ihtyaz Kader Tasawar, Fatema Shorna, Md Abu Bakr Siddique, and Saadia Binte Alam. "Blood Cancer Recognition Based on Discriminant Gene Expressions: A Comparative Analysis of Optimized Machine Learning Algorithms." In *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0385-0391. IEEE, 2021. https://doi.org/10.1109/IEMCON53756.2021.9623210

[52] Słota, Damian. "Influence of the mutation operator on the solution of an inverse Stefan problem by genetic algorithms." In *Computational Science-ICCS 2006: 6th International Conference*, Reading, UK, May 28-31, 2006, Proceedings, Part I 6, pp. 786-789. Springer Berlin Heidelberg, 2006. https://doi.org/10.1007/11758501_105

[53] Cho, Hang-Min, Suk-Joo Bae, Jungwuk Kim, and In-Jae Jeong. "Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm." *Computers & Industrial Engineering* 61, no. 3 (2011): 529-541. https://doi.org/10.1016/j.cie.2011.04.008

[54] Savic, Dragan A., and Godfrey A. Walters. "Genetic algorithms for least-cost design of water distribution networks." *Journal of Water Resources Planning and Management* 123, no. 2 (1997): 67-77. https://doi.org/10.1061/(ASCE)0733-9496(1997)123:2(67)

[55] Jung, Bong Seog, and Bryan Karney. "Fluid transients and pipeline optimization using GA and PSO: The diameter connection." *Urban Water Journal* 1, no. 2 (2004): 167-176. https://doi.org/10.1080/15730620412331289995

[56] Djebedjian, Berge. "Reliability-based water network optimization for steady state flow and water hammer." In *International Pipeline Conference*, vol. 42630, pp. 727-737. 2006. https://doi.org/10.1115/IPC2006-10234

[57] Jung, Sung Hoon. "Selective mutation for genetic algorithms." *International Journal of Computer and Information Engineering* 3, no. 8 (2009): 2060-2063.

[58] Sangroula, Uchit, Kuk-Heon Han, Kang-Min Koo, Kapil Gnawali, and Kyung-Taek Yum. "Leakage detection by using transient tests on an experimental looped water distribution network." *Preprints* (2022). https://doi.org/10.20944/preprints202201.0439.v1

[59] Sugianto, Welly, and Elva Susanti. "Optimasi Biaya Transportasi Pada UKM Di Kota Batam." *Journal of Industrial & Quality Engineering* 2303 (2021): 2715. https://doi.org/10.34010/iqe.v9i1.4278

[60] Chang, Hee-Chul, Won-Hee Park, Kyung-Beom Yoon, Tae-Kuk Kim, Duck-Hee Lee, and Woo-Sung Jung. "Inverse estimation of properties for charring material using a hybrid genetic algorithm." *Journal of Mechanical Science and Technology* 25 (2011): 1429-1437. https://doi.org/10.1007/s12206-011-0402-3

[61] Simpson, A., L. Murphy, and G. Dandy. "Pipe network optimisation using genetic algorithms." *American Society of Civil Engineers*, 1993.

[62] Khoshnevisan, Benyamin, Mohammad Ali Rajaeifar, Sean Clark, Shahaboddin Shamahirband, Nor Badrul Anuar, Nor Liyana Mohd Shuib, and Abdullah Gani. "Evaluation of traditional and consolidated rice farms in Guilan Province, Iran, using life cycle assessment and fuzzy modeling." *Science of the Total Environment* 481 (2014): 242-251. https://doi.org/10.1016/j.scitotenv.2014.02.052

[63] Haupt, Randy L., and Sue Ellen Haupt. *Practical genetic algorithms*. John Wiley & Sons, 2004. https://doi.org/10.1002/0471671746

[64] Kurnia, Hari, Elyza Gustri Wahyuni, Elang Cergas Pembrani, Syifa Tri Gardini, and Silfa Kurnia Aditya. "Vehicle routing problem using genetic algorithm with multi compartment on vegetable distribution." In *IOP Conference Series: Materials Science and Engineering*, vol. 325, no. 1, p. 012012. IOP Publishing, 2018. https://doi.org/10.1088/1757-899X/325/1/012012

[65] Shariff, SS Radiah, Noor Hasnah Moin, and Mohd Omar. "Location allocation modeling for healthcare facility planning in Malaysia." *Computers & Industrial Engineering* 62, no. 4 (2012): 1000-1010. https://doi.org/10.1016/j.cie.2011.12.026

[66] Wibawa, Aji Prasetya, Wayan F. Mahmudy, Agung Mustika Rizki, Gusti Eka Yuliastuti, and Ishardita Pambudi Tama. "Multi-Site Aggregate Production Planning Using Particle Swarm Optimization." *Journal of Engineering, Project, and Production Management* 12, no. 1 (2022): 62.

[67] Omar, Mohamed K., and Ajitha Angusamy. "Hybrid genetic algorithm for coordinated production and transportation planning problem." In *2012 IEEE Business, Engineering & Industrial Applications Colloquium (BEIAC)*, pp. 34-37. IEEE, 2012. https://doi.org/10.1109/BEIAC.2012.6226081

[68] Abdel-Kader, Rehab F. "Hybrid discrete PSO with GA operators for efficient QoS-multicast routing." *Ain Shams Engineering Journal* 2, no. 1 (2011): 21-31. https://doi.org/10.1016/j.asej.2011.05.002

[69] Ochelska-Mierzejewska, Joanna, Aneta Poniszewska-Marańda, and Witold Marańda. "Selected genetic algorithms for vehicle routing problem solving." *Electronics* 10, no. 24 (2021): 3147. https://doi.org/10.3390/electronics10243147

[70] Alabsi, Firas, and Reyadh Naoum. "Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system." *Journal of Emerging Trends in Computing and Information Sciences* 3, no. 7 (2012): 1053-1058.

[71] Gimnastian, Danu Fawwaz, Rahmat Yasirandi, and Dita Oktaria. "Analysis and Design of a Route Recommendation System and Bicycle Rental Fees at Tourist Destinations with Genetic Algorithms." *Jurnal Media Informatika Budidarma* 6, no. 2 (2022): 837-847. https://doi.org/10.30865/mib.v6i2.3749

[72] Shon, Sudeok, Sangwook Jin, and Seungjae Lee. "Minimum weight design of sinusoidal corrugated web beam using real-coded genetic algorithms." *Mathematical Problems in Engineering* 2017 (2017). https://doi.org/10.1155/2017/9184292

[73] Barba Rodríguez, Irene, Carmelo del Valle Sevillano, and Diana Borrego Núñez. "A Constraint-based Job-Shop Scheduling Model for Software Development Planning." *ADIS 2009: Decision-making in Software Engineering in Conjunction with JISBD 2009* (2009), pp. 1-12. (2009).

[74] Moin, Noor Hasnah. "Hybrid genetic algorithms for vehicle routing problems with time windows." *Computers & Operations Research* (2002).

[75] Chu, M. C., K. Y. Fung, T. Kwok, J. K. C. Leung, Y. C. Lin, H. Liu, Kam Biu Luk, H. Y. Ngai, C. S. J. Pun, and H. L. H. Wong. "Development of a Bonner sphere neutron spectrometer from a commercial neutron dosimeter." *Journal of Instrumentation* 11, no. 11 (2016): P11005. https://doi.org/10.1088/1748-0221/11/11/P11005

[76] Yepes, Victor, José Vicente Martí, Tatiana García-Segura, and Fernando González-Vidosa. "Heuristics in optimal detailed design of precast road bridges." *Archives of Civil and Mechanical Engineering* 17, no. 4 (2017): 738-749. https://doi.org/10.1016/j.acme.2017.02.006

[77] Doyle, Josh, Roy Hartfield, and Christopher Roy. "Aerodynamic optimization for freight trucks using a genetic algorithm and CFD." In *46th AIAA Aerospace Sciences Meeting and Exhibit*, p. 323. 2008. https://doi.org/10.2514/6.2008-323

[78] Mukhtar, Azfarizal, N. Khai Ching, and M. Zamri Yusoff. "Optimal design of opening ventilation shaft by Kriging metamodel assisted multi-objective genetic algorithm." *International Journal of Modeling and Optimization* 7, no. 2 (2017): 92-97. https://doi.org/10.7763/IJMO.2017.V7.565

[79] Jang, Sung Ho, Tae Young Kim, Jae Kwon Kim, and Jong Sik Lee. "The study of genetic algorithm-based task scheduling for cloud computing." *International Journal of Control and Automation* 5, no. 4 (2012): 157-162.

[80] Peng, Yidong, Xiuli Qu, and Jing Shi. "A hybrid simulation and genetic algorithm approach to determine the optimal scheduling templates for open access clinics admitting walk-in patients." *Computers & Industrial Engineering* 72 (2014): 282-296. https://doi.org/10.1016/j.cie.2014.03.026

[81] Mishra, Bhabani Shankar Prasad, Satchidananda Dehuri, Rajib Mall, and Ashish Ghosh. "Parallel single and multiple objectives genetic algorithms: A survey." In *Modeling Applications and Theoretical Innovations in Interdisciplinary Evolutionary Computation*, pp. 73-110. IGI Global, 2013. https://doi.org/10.4018/978-1-4666-3628-6.ch006

[82] Pedrola, Oscar, Davide Careglio, Miroslaw Klinkowski, Luis Velasco, Keren Bergman, and Josep Solé-Pareta. "Metaheuristic hybridizations for the regenerator placement and dimensioning problem in sub-wavelength switching optical networks." *European Journal of Operational Research* 224, no. 3 (2013): 614-624. https://doi.org/10.1016/j.ejor.2012.08.011

[83] Julirose, Gonzales. "Genetic algorithm optimization of product design for environmental impact reduction." *PhD diss., University of Malaya*, 2018.

[84] Balakrishnan, P. V., and Varghese S. Jacob. "Genetic algorithms for product design." *Management Science* 42, no. 8 (1996): 1105-1117. https://doi.org/10.1287/mnsc.42.8.1105

[85] Sun, Xue, Lien-Fu Lai, Ping Chou, Liang-Rui Chen, and Chao-Chin Wu. "On GPU implementation of the island model genetic algorithm for solving the unequal area facility layout problem." *Applied Sciences* 8, no. 9 (2018): 1604. https://doi.org/10.3390/app8091604

[86] Karthikeyan, K., and P. K. Dhal. "Investigation of optimal location and tuning of STATCOM by genetic algorithm based transient stability improvement." *Journal of Electrical Systems* 14, no. 2 (2018): 103-117.

[87] de Paiva, Jonatas Lopes, Claudio FM Toledo, and Helio Pedrini. "An approach based on hybrid genetic algorithm applied to image denoising problem." *Applied Soft Computing* 46 (2016): 778-791. https://doi.org/10.1016/j.asoc.2015.09.013

[88] Reynes, Christelle, Robert Sabatier, Nicolas Molinari, and Sylvain Lehmann. "A new genetic algorithm in proteomics: Feature selection for SELDI-TOF data." *Computational Statistics & Data Analysis* 52, no. 9 (2008): 4380-4394. https://doi.org/10.1016/j.csda.2008.02.025

[89] Ahmadi, Mehdi, Mazdak Arabi, Dana L. Hoag, and Bernard A. Engel. "A mixed discrete-continuous variable multiobjective genetic algorithm for targeted implementation of nonpoint source pollution control practices." *Water Resources Research* 49, no. 12 (2013): 8344-8356. https://doi.org/10.1002/2013WR013656

[90] Burchardt, Hagen, and Ralf Salomon. "Implementation of path planning using genetic algorithms on mobile robots." In *2006 IEEE International Conference on Evolutionary Computation*, pp. 1831-1836. IEEE, 2006.

[91] Kavitha, A. R., and C. Chellamuthu. "Brain tumour segmentation from MRI image using genetic algorithm with fuzzy initialisation and seeded modified region growing (GFSMRG) method." *The Imaging Science Journal* 64, no. 5 (2016): 285-297. https://doi.org/10.1080/13682199.2016.1178412

[92] Shabankareh, Sajjad Ghazanfari, and Saeid Ghazanfari Shabankareh. "Improvement of edge-tracking methods using genetic algorithm and neural network." In *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1-7. IEEE, 2019. https://doi.org/10.1109/ICSPIS48872.2019.9066026

[93] Sari, Murat, and Can Tuna. "Prediction of pathological subjects using genetic algorithms." *Computational and Mathematical Methods in Medicine* 2018 (2018). https://doi.org/10.1155/2018/6154025

[94] Baum, Karl G., Evan Schmidt, Kimberly Rafferty, Andrzej Król, and María Helguera. "Evaluation of novel genetic algorithm generated schemes for positron emission tomography (PET)/magnetic resonance imaging (MRI) image fusion." *Journal of Digital Imaging* 24 (2011): 1031-1043. https://doi.org/10.1007/s10278-011-9382-1

[95] Mansoori, Tanzeem Khan, Amrit Suman, and Sadhana K. Mishra. "Feature selection by genetic algorithm and SVM classification for cancer detection." *International Journal of Advanced Research in Computer Science and Software Engineering* 4, no. 9 (2014).

[96] Dash, Manoranjan, and Huan Liu. "Consistency-based search in feature selection." *Artificial Intelligence* 151, no. 1-2 (2003): 155-176. https://doi.org/10.1016/S0004-3702(03)00079-1

[97] Datta, Dilip, André RS Amaral, and José Rui Figueira. "Single row facility layout problem using a permutation-based genetic algorithm." *European Journal of Operational Research* 213, no. 2 (2011): 388-394. https://doi.org/10.1016/j.ejor.2011.03.034

[98] Sadrzadeh, Amir. "A genetic algorithm with the heuristic procedure to solve the multi-line layout problem." *Computers & Industrial Engineering* 62, no. 4 (2012): 1055-1064. https://doi.org/10.1016/j.cie.2011.12.033

[99] Mateo, Carlos, Pedro L. Carro, Paloma García-Dúcar, Jesús De Mingo, and Íñigo Salinas. "Radio-over-fiber linearization with optimized genetic algorithm CPWL model." *Optics Express* 25, no. 4 (2017): 3694-3708. https://doi.org/10.1364/OE.25.003694

[100] Chen, Rong, Chang-Yong Liang, Wei-Chiang Hong, and Dong-Xiao Gu. "Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm." *Applied Soft Computing* 26 (2015): 435-443. https://doi.org/10.1016/j.asoc.2014.10.022

[101] Sermpinis, Georgios, Charalampos Stasinakis, Konstantinos Theofilatos, and Andreas Karathanasopoulos. "Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms-Support vector

regression forecast combinations." *European Journal of Operational Research* 247, no. 3 (2015): 831-846. https://doi.org/10.1016/j.ejor.2015.06.052
[102] Soleimani, Hamed, and Govindan Kannan. "A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks." *Applied Mathematical Modelling* 39, no. 14 (2015): 3990-4012. https://doi.org/10.1016/j.apm.2014.12.016
[103] Shi, Jianmai, Zhong Liu, Luohao Tang, and Jian Xiong. "Multi-objective optimization for a closed-loop network design problem using an improved genetic algorithm." *Applied Mathematical Modelling* 45 (2017): 14-30. https://doi.org/10.1016/j.apm.2016.11.004
[104] Hiassat, Abdelhalim, Ali Diabat, and Iyad Rahwan. "A genetic algorithm approach for location-inventory-routing problem with perishable products." *Journal of Manufacturing Systems* 42 (2017): 93-103. https://doi.org/10.1016/j.jmsy.2016.10.004
[105] Sun, Xiaoqin, Dongli She, Yuanhang Fei, Xiao Han, and Lei Gao. "An improved pore-solid fractal model for predicting coastal saline soil hydraulic properties based on changepoints determined by genetic algorithm-support vector regression." *Soil and Tillage Research* 224 (2022): 105502. https://doi.org/10.1016/j.still.2022.105502