International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

Role of Evolutionary Algorithms in Construction Projects Scheduling

Noha Essam[a]*, Laila Khodier[b], Fatma Fathy[c]

[a]*MSc Student, Architecture Department, Ain-Shams University, Cairo, Egypt*
[b]*Professor of Project Management and Sustainable Development, Architecture Department , Ain-Shams University, Cairo, Egypt*
[c]*Assistant Professor, Architecture Department , Ain-Shams University, Cairo, Egypt*
[a]*Email: G18053070@eng.asu.edu.eg,* [b]*Email: leila.mohammed@eng.asu.edu.eg*
[c]*Email: fatma.fathy@eng.asu.edu.eg*

Abstract

Due to the increase in the stakeholders and their objectives the construction projects have significantly been affected by the ongoing demands leading to increase in complexity of scheduling problems, research in the field of Multi-Objective Optimization (MOO) have increased significantly. Through their population-based search methodologies, Evolutionary Algorithms drove attention to their efficiency in addressing scheduling problems involving two or three objectives. Genetic Algorithms (GA) particularly have been used in most of the construction optimization problems due to their ability to provide near-optimal Pareto solutions in a reasonable amount of time for almost all objectives. However, when optimizing more than three objectives, the efficiency of such algorithms degrades and trade-offs among conflicting objectives must be made to obtain an optimal Pareto Frontier. To address that, this paper aims to provide a comparative analysis on four evolutionary algorithms (Genetic algorithms – Memetic algorithms – Particle Swarm – Ant colony) in the field of construction scheduling optimization, gaps are addressed, and recommendations are proposed for future research development.

Keywords: Evolutionary Algorithms (EAs); Construction Scheduling; Multi-Objective Optimization (MOO).

----

* Corresponding author.

1. Introduction

Construction Industry has recently risen to the top of the list of sectors directly influencing economies and national development [1]. It usually engages various stakeholders with competing objectives, which implies that nearly all decision-making problems must address conflicts between several objectives bound to various limitations at the same time [2]. An ideal balance between these goals must be achieved for a project to be completed successfully. Each construction project consists of several activities that is unique yet related to each other. Each activity can be performed using different alternative methods, which means the impact on the overall objectives of the project is directly dependent on the decision maker choice of alternatives [2] Figure 1. Therefore, great care must be taken while implementing the relevant alternatives selected for each specific project activity in order to create an optimal balance among these elements. Due to the various ways these options can be combined, and the numerous project activities involved, which further complicate the process of attaining the optimum, this process is, however, extremely difficult to complete using standard approaches. The techniques and approaches used in optimization range from conventional algorithms (like gradient-based search) to sophisticated metaheuristic algorithms (like Genetic Algorithms). However, it has been demonstrated that metaheuristic algorithms, such as GA, outperform classical algorithms due to their flexibility in solving mega-scale, complicated optimization problems as well as their decision variables, functions, and constraints [3].

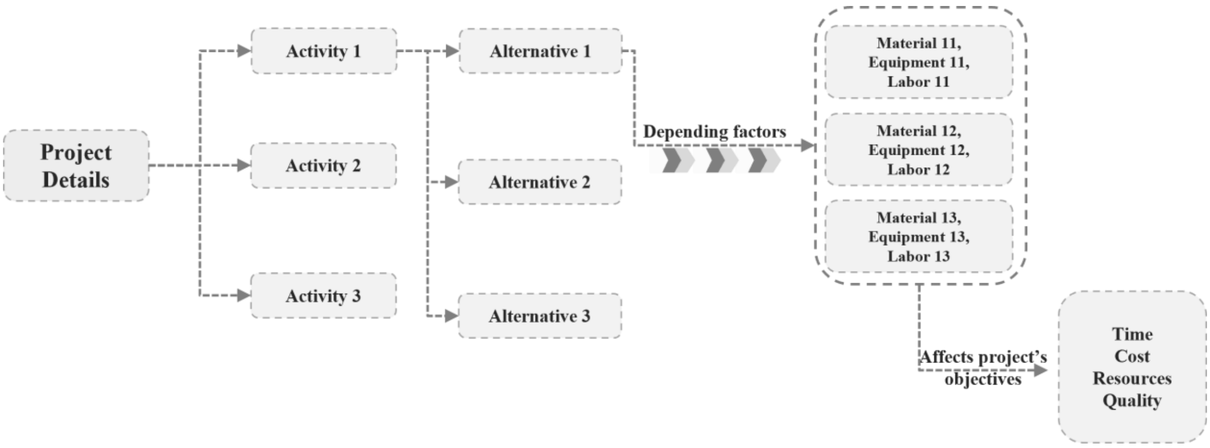

Figure 1: Input Data for the optimization model. Source: author adopted after Panwar and Jha [4].

In this paper, we aim to offer review on previous applications of EAs in optimizing construction scheduling to solve Multi-Objective Scheduling Problem (MOSP) with two or more objectives. Finally, gaps in the literature are addressed, and suggestions are made based on the findings for future research development in this field.

2. Construction Scheduling Optimization Problem (CSOP)

The schedule for carrying out the project's tasks is the conventional definition of a construction schedule [5]. Basically, construction scheduling focuses on efficiently allocating resources and ordering tasks over time. With the project deadline typically being pre-specified, the time factor is a crucial consideration for a stakeholder when assessing and choosing proposals. In order to get an edge during the assessment of a proposal, contractors constantly work to reduce the project duration. Nevertheless, time and cost are interdependent which means crashing a project's time schedule requires extra resources which consequently increases the cost. On the other hand, stake holders such as developers seek to minimize the project's overall cost and duration in order to lower their capital expenses and increase their return on investment. Such conflicting objectives require a precise consideration trade-offs among objectives during project scheduling phase [6]. Also, the construction site is a dynamic atmosphere; thus, it is anticipated that the original timeline may experience numerous changes as work progresses. In furthermore, schedules for building projects frequently include a lot of activities. As the number of construction activities grows, resource utilization options at the activity level leads to an exponentially growing number of feasible schedule combinations. This results in numerous execution sequences. It is necessary to create and choose an acceptable alternative while taking into account the project objectives, such as the reduction of the overall execution time or expenses [7].

As a result, the construction activities such as materials, personnel, tools, and site layout are represented by variables when the construction schedule issues are modelled as constraint satisfaction problems. Different scheduling constraints can then be defined between these variables. Before starting the related construction task, these limitations must be met. Regular scheduling constraints can be capacity, spatial features, dependencies on technology, availability of resources.etc. In order to represent such variables and constraints a type of representation known as constraint graphs are used where each type of constraint has its corresponding constraint graph calculated [8].

Optimization has existed for quite a long time. However, following the 1950's to 1970's period of massive theoretical expansion, optimization resurfaced once again as a topic of intense international research. A rising public awareness of restricted resource availability is one of the reasons why this subject of study remains so enthralling. Also, enhancement in the computing programs and enlarging the hardware capacities contributed to increasing the level of interest directed towards discovering more potentials in the field. In today's management advancements, quantitative assistance for decision-making processes in projects is recognized as being quite essential. However, what distinct managing and optimizing construction projects is their stochastic and dynamic environment. Thus, making the flow of adequate information vital for achieving optimality in construction processes, leading to the necessity of choosing the most appropriate technique for the given problem [9].

Numerous techniques and algorithms have been created in an ongoing effort to address specific obstacles, notably significant practical problems, to solve CSO problems [6]. Optimization can be defined the process of choosing a set of variables subjected to a set of constraints and obtaining the minimum or maximum value of a function. The optimization value that is typically calculated using optimization techniques is known as the fitness, or objective function [9]. A multi-objective optimization problem can be solved in two steps: optimizing the various associated objective functions and figuring what kinds of "trade-offs" are appropriate from the decision-maker perspective. Coello, and colleagues [10] described a categorization of the Multi-Objective Optimization Problem (MOOP) in detail, the categories were divided depending on the sequence of the decision-making and search processes (A priori Preference Articulation - A posteriori Preference Articulation - Progressive Preference Articulation).

With that being said, an efficient approach for handling a specific optimization problem must be properly selected in order to get significant results. Before choosing a feasible solution approach, the optimization problem should be evaluated in terms of its functions, constraints, and decision variables [11]. Optimization algorithms are generally made to assist in identifying practical combinations of trade-offs between objectives. In essence, an optimization algorithm seeks to efficiently traverse the space of possible practical or near-practical solutions [12]. More and more alternative methods have emerged to address the challenges associated with the MOOPs. Mathematical approaches for instance were found to be quite challenging when optimizing large-scale engineering problems. Linear Programming and Dynamic Programming approaches also frequently fail while searching for solutions and often get trapped in local optima [13]. Therefore, studies have suggested evolutionary-based algorithms as an approach to get around these challenges through discovering close to optimal solutions to problems [14].

3. **Evolutionary Algorithms**

There has been an increase in interest in algorithms based on the notion of natural evolution over the last two decades [15]. Evolutionary-based methods have been introduced by researchers as a type that is guided by meta-heuristic techniques for finding near-optimal solutions to such problems [13, 14]. EAs are random search techniques inspired by the biological evolution of nature and the social behavior and act of survival of the species living in it. In the field of artificial intelligence research, EAs are quickly expanding because of their population-based nature which allows them to deal simultaneously with a set of possible solutions in multi-objective optimization problems and several components of the Pareto optimal set to be generated in a single run [10]. Numerous researchers have investigated the ability to emulate the efficient behavior of these species through devising computational systems that have the ability to provide rapid and robust solutions to solve difficult optimization problems [14]. The EAs population consists of a set of individuals where each one forms a searching point in the pool of feasible solutions and the creation of the first population is started at random and then exposed to selection, crossover, and mutation across numerous generations, causing the newly produced generations to progress towards more desirable search space regions. The search is advanced by assessing the fitness of the entire population, identifying the individuals with the highest fitness values, and joining them to develop new individuals with a higher possibility of improving fitness [15].

A set of chromosomes forms a population. Consequently, the individuals are decoded into several parameters defining the targeted function. Generally, Evolutionary Operators (EVOPs) act on the population of an EA in the same way as they do in nature, seeking to create solutions with improving fitness. EAs cover both search and multi-objective decision making, which makes the EAs particularly appealing MOP solution approaches [10].

In principle, when it comes to applying EAs to a problem, they all use the same strategy and structure. To begin, the problem requires a representation that is appropriate for each method. The evolutionary search algorithm is then used to arrive at a near-optimal solution repeatedly [14].

Maier, and colleagues [16] provided a study on the mechanism of functioning of single and multi-objective EAs. It also offered a detailed explanation on what makes EAs the convenient alternative for selecting optimal solutions. Those reasons can be summarized as follows:

- EAs can find global optimum solutions due to being population-based which broaden the exploration span throughout the entire search space recognizing the more efficient spaces in a quicker and efficient way.
- Unlike most traditional optimization methods, EAs' searching strategy can be tailored to a particular problem at hand by adjusting them to find the values that indicate an ideal balance between exploration and exploitation.
- EAs can be smoothly linked to simulation and integrated assessment models through two-way connection between the optimization algorithm and a simulation model. During this connection the optimization algorithm identify the decision variables and provided them to the simulation model which in turn analyzes the related objective function and constraint values before passing them back to the optimization algorithm.
- The adoption of EAs is straightforward since the optimization technique employed is similar to the informal optimization approach utilized in simulation models. However, the way solutions are adjusted is the primary difference between the informal optimization methods and formal EA-based optimization methods. as the evolutionary operators automatically obtain adjustments in decision variable values throughout the optimization iterations.
- Additional feature of EAs is that, unlike most traditional optimization methods, they can deal with both discrete and continuous decision variables.
- EAs can deal with constraints in a simple manner through using constrained handling strategies (e.g. penalty functions, constrained tournament selection for multi-objective optimization)
- EAs can handle many-objectives simultaneously without the need to turn the multi-optimization problem into single optimization problem. To obtain this feature EAs use multi-objective analogies that travel throughout the search space and identifying an approximate Pareto front in a single algorithm run.
- EAs can save significant computational time when compared with the traditional methods because they are particularly suitable to being deployed in parallel computing systems.
- EAs operate with population of near-optimal solutions which share the same objective function space but differ in decision variables space, allowing decision-makers to have more flexibility to use their judgement and insight to choose the final solution from a pool of optimal solution.
- EAs boost trust in the optimal solutions identified from the optimization process, as the EAs use the outputs of environmental simulation models to evaluate objectives and constraints.

(Elbeltagi and colleagues (2005) offered a compared and briefly discussed the methods of using five Evolutionary algorithms: genetic algorithms, memetic algorithms, particle swarm, ant-colony systems, and shuffled frog leaping.

The performance of EAs was also addressed based on this comparative research, where all five algorithms were used to solve discrete optimization test and a comparative result were presented. The results discussed that the PSO approach outperformed other algorithm based on success rate and solution quality and the second in computational processing time.

3.1. Genetic Algorithms

Based on based on Charles Darwin's theory of natural evolution and survival of the fittest, Genetic algorithms (GA) were first introduced in the 1960s, as an optimization technique [17]. The GA is a fast global concurrent searching technique that can adaptively find an optimum or sub-optimal solution by gathering data from the search space [6]. In evolution theory, each potential solution, or individual, is expressed as a string of bits called chromosomes. These chromosomes store all the decision variables, and each chromosome is made up of genes that are expressed in real number, binary digits or integers. GAs grant each individual a fitness score, individuals that are the fittest are much more capable of surviving and pass on their genetic material as they battle for resources in the environment. As a result, the population's quality improves during the evolution process [18]. The most essential processes in GA are selection, crossover, and mutation are employed [6] where a selection method is utilized to select the parents according to the fitness values, and a prospective population is formed for the next process. Then a crossover is performed, enabling swapping of information between parents to generate new children known as offspring. To avoid the premature problem created by the crossover, genes are modified on a randomly chosen positions during the mutation process. For each following round iteration, a new population is created and the whole processes is performed once again. See Figure 1

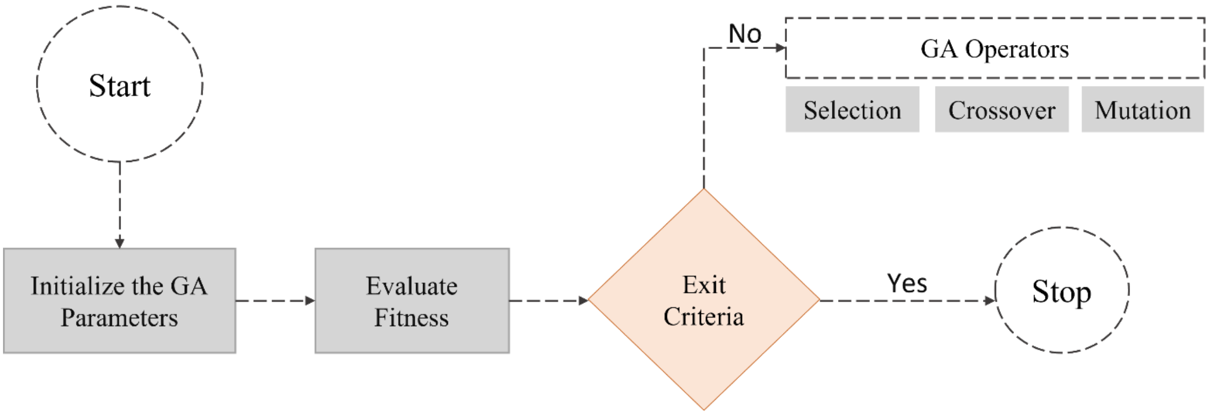

Figure 2: Implementation structure of genetic algorithms - Source: Shabir and Singla [19] modified by Author.

3.1.1. Selection Operator

In genetic algorithms, selection is a crucial phase that decides whether or not a given string will take part in the reproduction process [20].The GA search is carried out by generating a series of chromosomal populations, the procedure of selecting parents for pairing and recombining to produce offspring for the following generation is known as parent selection, with each population's members being the children of the previous population and the parents of the subsequent one. Usually, the initial population is created by selecting the elements of all strings at random. while at each new iteration the next population is created by altering the strings of the previous population to produce a new population with a higher fitness value. This pattern repeats until a stop requirement or the required number of iterations is met [21]. The choice of parents can directly and critically affect the GA's convergence rate, since competent parents motivate the search for superior and more fit solutions in the offspring. Nevertheless, a single exceptionally fit solution can take over the whole population

within a few generations if not enough care is taken during the selection process. This is phenomenon is known as premature convergence that would result in reduction in variety among solutions as the solutions is being too near to one another in the pool of solutions. Such scenario is un desirable as maintaining a high level of population variety is critical to a GA's success [22].The pace of convergence of GA is determined by the selection pressure [20].
Following are the most well-known selection techniques found in literature :

3.1.1.1. Roulette Wheel Selection (Proportionate selection)

This approach uses the fitness function to assign a score to each genome then the overall population's cumulative fitness is computed. The probability of selecting each genome is then computed according to the fitness score (the higher the score the more likely it will be picked) [23].

3.1.1.2. Ranking Selection

In linear ranking selection which was first introduced by [24] individual genomes are given ranks according to their fitness values, where the higher the fitness values of genomes are the higher they are ranked and vise verse. followed by the process of choosing genomes which are based on a probability that is linearly proportional to the population's genome rank [23].

3.1.1.3. Tournament Selection

To choose an individual, Tournament selection entails running multiple competitions between a few randomly selected individual genomes from the population. During this process feasible points are always considered better than infeasible points also the fittest according to fitness function (the tournament champion) of two viable solutions is chosen for the next step which is cross-over; however when choosing between two infeasible solutions the one with the smallest total of constraint breaches is chosen [25]. Also, the pattern of selecting and pairing parents will be repeated till the new population has the necessary number of genomes set by the user and is equal to the size of the ancestral population [23].

In order to speed up the search process and provide effective GA solutions, a balance between the two explorative and exploitative elements of GA, which are described by the two operators cross over and mutation, is required [26].

3.1.2. Crossover Operator

Considered as the prime search operator in Genetic Algorithms (GA), Cross-over comes as important as the selection, mutation and coding in GA. It's main function is to enhance the GA's behavior through examining information that is available through the search space [26]. The operator starts functioning directly after the process of selecting the parents, by crossing the two parents' chromosomes and instantly replacing the two resultant individuals in the new population then moving on to the selection of another pair. However, before identifying the next pair of parents a fitness index must be instantaneously modified so as to allow diversity by selecting from a constantly shifting population [21].

There are two prime approaches a crossover operator can adopt [26] :

a) Mean centric operator that produces children in the region of the mean of the participant parents around their centroid.
b) Parent centric operator that produces children near their parents.

As for the operators, there are many crossover operators discussed in literature, following are brief description of some of the frequently used operators [26]:

3.1.2.1. Single point crossover

The operator generates offspring by randomly selecting a split point on the parent's chromosome then performs the crossover at this point. The average crossover probability is between 0.2 and 1.0. The single crossover can be generalized into N-point crossover where the number of crossover points is equal to N.

3.1.2.2. Uniform crossover

This operator divides the parent's chromosomes into heads and tails, then adopts the coin flipping technique, so that each gene of the first child gets head from one parent and tail from the other. for the second child, an inverted copy of the gene is made.

3.1.2.3. Simulated binary crossover (SBX)

Unlike the previous operators who depend only on exploitative search where the offspring is produced in the area defined by parents, thus resulting in early convergence. This operator produces offspring in the exploration search region near the parents, rather than only inside the region delimited by them. However, the operation of one-point crossover on a string of binary alphabets in a continuous region is simulated using simulated binary crossover (SBX)

3.1.3. Mutation Operator

In biological systems during chromosomal swaps, some defects may occur while duplicating DNA content. The result of that defected duplication is known as mutation. In GA optimization, mutation is considered as a secondary operator and is created using a particular probability to modify the solution variable DNA [27]. A mutation operator will modify the genes of the offspring and promote population variety to the next generation, allowing GAs to target potential portions of the search space while avoiding local solutions [26]. The form of the mutation usually depends on whether the GA uses binary or integer numbers as a representation [27]. Also , Some mutation operators are specifically intended to solve particular sorts of problems while ignoring others.
References [28] reported several types of mutation operators such as Mirror mutation and binary bit-flipping

mutation, Random (uniform) mutation, Mutation based on directed variation techniques, Directed mutation ,polynomial mutation (PLM) etc.). A brief summary of some of the frequently used mutation operators is provided following [26].

3.1.3.1. Binary bit-flipping mutation

This operator functions mainly by substituting a gene with its counterpart value at the center point of the gene's border span.

3.1.3.2. Uniform mutation

This operator performs mutation according to a pre-specified range of random uniform values to replace the value of the specified gene.

3.1.3.3. Polynomial mutation

This operator uses an index to determine its strength where small index means small diversity and vice versa .PM is basically based on polynomial distribution.

3.1.3.4. Non-uniform mutation

This operator aims to achieve a balance between exploration and exploitation, while function is based on population size the search starts consistently and turns quite locally at the end of the search.

3.1.3.5. Adaptive mutation

This operator uses the recurrence of the finest chromosomal genes and adapts accordingly using its adaptive characteristics.

3.1.4. Selection of Optimal Solution

In order to solve a real-world Multi-Objective Optimization Problem (MOOP), a single optimal solution needs to be selected from the non-dominated solutions set. To do so , the Pareto-optimal set is first identified using evolutionary algorithms and then applying a strategy that considers decision Maker's preferences (DM) and is capable of selecting a single solution from a set of non-dominated solutions. ***[29]*** explained in details three distinct methods for determining solutions while taking decision maker's preferences into consideration. The study aimed to compare the actual performance of those methods using certain benchmark problems. Following is a brief of these methods:

3.1.4.1. Reference Point based EMO Approach (RP-EMO)

This Approach is considered apriori approach and was first introduced by [30] where a reference point or points are pre-defined with a goal to attain the Pareto-front or set of solutions around those specific points. However, there are some challenges associated with using this approach in real world problems such as an adequate knowledge with the criterion space is required in advance, such process is not easy to get obtain especially for problems where complicated functions are used to generate the Pareto front or when the problem has more than one criterion. The key concepts of this approach are [29]: In the criterion domain, solutions that are nearer to the reference points get to be more emphasized and noticed. Then in order to preserve a varied range of solutions around each reference point, solutions that are adjacent to a near reference point are also emphasized.

3.1.4.2. Weighted Metrics Method (WMM)

This Approach was first recognized by [31] and is considered posteriori approach as the manifestation of preference information is employed a posteriori. This means that after the outcomes of the optimization process is provided in the form of a set of solutions, the final solution is picked from this set. References [29] stated that reducing range between the optimal criteria vector and the solutions on the criteria space is the basic goal of weighted metrics. Thus this criterion vector is considered the optimal solution if it is located within the space of permitted solutions due to it's minimization of objective functions. Nevertheless, when the conditions are not met, this optimal criteria vector can only belong to the permissible region.

3.1.4.3. Weighted Stress Function Method (WSFM)

This method is considered a posterior approach where the decision maker preferences are incorporated sequentially after the search process is carried out. In order for the WSFM method to provide an optimum solution satisfying the decision maker's preferences the solution must be part of the Pareto Frontier. Also, the optimal objective vector that is responsible of maximizing/minimizing each of the objective functions should be properly considered during the selection process. In multi-objective optimization, however, the relative priority assigned to each criterion causes "stress" while looking for solutions that maximize each of the multiple criteria.

Panwar and Jha [4] managed to optimize four objectives (time-cost-resources-environmental impact) by developing a Non-Sorting Genetic Algorithm (NSGA III)-based many-objective scheduling optimization model (MOSM) that produces a pareto front of possible solutions and present it as a weighted sum for decision-makers to choose a solution according to the preference and added Several trade-off configurations were examined to identify the advantages of adding a fourth objective. The study validated its model through considering two case studies. Moreover, the MOSM was proved to be performing well when optimizing all four targeted project objectives as well as being more advanced in its capabilities when compared other models found in literature.

Sharma and Trivedi [32] developed an opposition-based non-dominated sorting genetic algorithm III (OBSGA III) model that provides two improvements over the standard NSGA III.

The first is employing opposition-based learning (OBL) to create a well-diversified basic population, and the second is the generation leaping using OBL. The created model's validity and superiority are shown by similarities and comparisons between its results and those from earlier studies. The suggested model is used to resolve a time-cost-resources-quality trade-off optimization issue in order to show the model's applicability.

3.2. Memetic algorithms

While MAs and GAs have many similarities, memes, as opposed to genes, make up chromosomes. All chromosomes and off-springs are permitted to gather significant information, through a local search, before becoming part of the evolutionary process, which is a distinctive feature of the MAs algorithm [33]. This feature led to using the term MAs to describe GAs that make extensive utilization of local search [34]. MAs operate by generating an initial population at random, much like the GAs. Then to enhance each population member's experience and offer a population of local optimal solutions, a local search is then conducted on each member of the population. Off-spring is then created through performing crossover and mutation operations. The local search is then applied on their progeny in order to always preserve local optimality [14]. See Figure 3

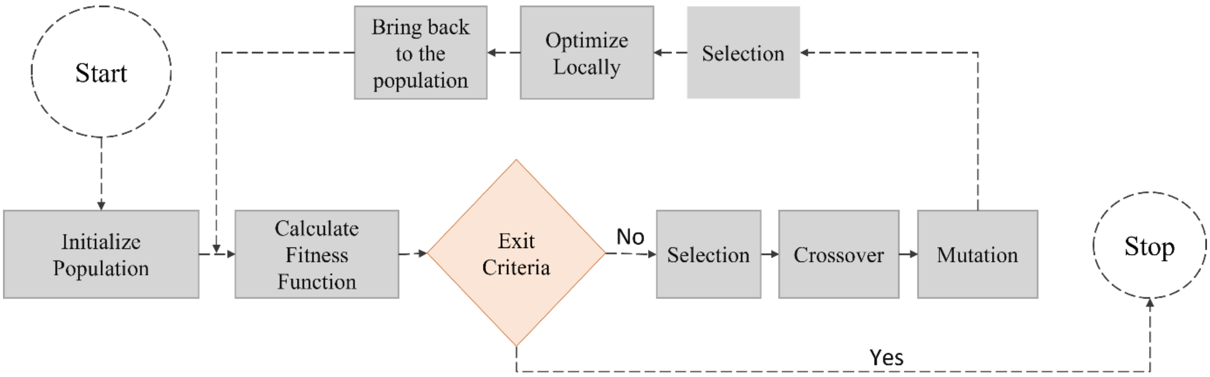

Figure 3: Implementation structure of memetic algorithms. Source: Singh, and colleagues [35] modified by author.

Rahman, and colleagues [25] proposed an efficient memetic approach for solving Resource constrained project scheduling problem (RCPSP).

The suggested solution combines a properly thought-out genetic algorithm with local search methods and adaptive mutation. In order to direct the GA search and save processing time while still finding the best solution, the study proposed a modified approach to create workable solutions inside the original population. Additionally, the study used a very basic chromosomal representation to express the predecessor-successor associations in a binary matrix that can be examined quickly to save processing time and employed two distinct localized queries to take use of the benefits of both methods and speed up the exploitation process. The fittest individuals were ranked as per their fitness value and breach value in the suggested method, which created an initial population at random with 20% of population size being mended to be practical. The selection process used the tournament approach, and the off-spring is mutated following the cross-over process.

The suggested local search was then utilized to perform more exploration in order to quickly get the best solution. One local search was performed to one chromosome at a time with various rates in the suggested hybrid local search, which employed two separate local searches to alter the created chromosomes. The study recommended further development for the algorithm to solve more challenging problems with much more activities in a considerable time and more accurate results.

Particle Swarm Algorithms

Instead of focusing solely on individual cognitive capacities, Kennedy and Eberhart's original theories on particle swarms intended to create artificial intelligence technologies by utilizing straightforward analogies of social interaction [36]. PSO basically operates in a way that the search space of a particular issue or function is filled with several small, simple elements called particles, each of which assesses the objective function where it is currently located. Then, by integrating multiple factors, each particle selects how to navigate through the search space by fusing a historical component of its present and ideal locations with those of one or more swarm members, along with some random disturbances. Once every particle has been shifted, the process moves on to the next iteration. The swarm will eventually move near to the fitness function's maximum, much like a flock of birds looking for food together. It is practically impossible for a particle to overcome a challenge on its own; advancement only happens when particles connect. The ability to solve problems arises from the individual behaviors of the particles through their interactions, and it affects the entire population. [37]. See Figure 4

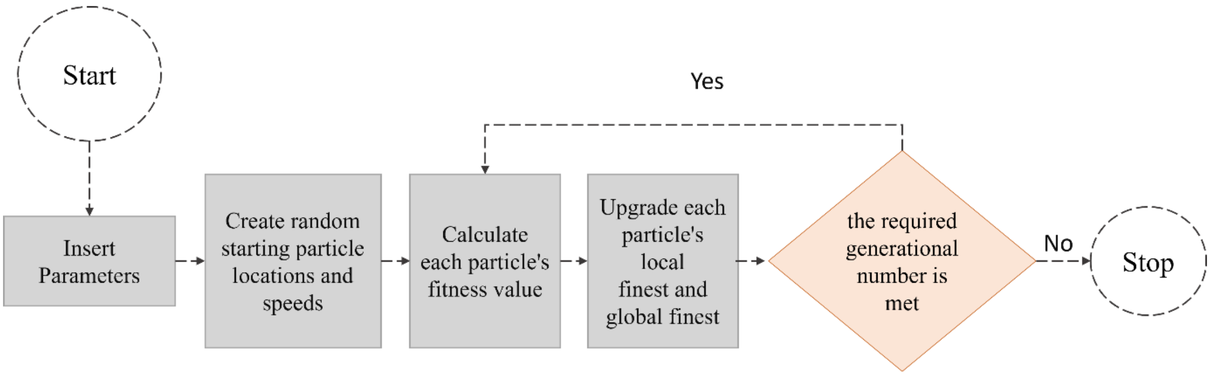

Figure 4: Implementation structure of particle swarm algorithms. Source: Shabir and Singla [19] modified by author.

Zhang and Xing [38] addressed the problem of minimizing cost and time while maximizing quality of construction by developing a study that uses fuzzy-multi-objective particle swarm optimization (FMOPSO) to solve the fuzzy time-cost-quality trade-off (TCQT) problem by employing fuzzy arithmetic operations to express the inequality limitations among the relevant fuzzy numbers. The suggested fuzzy multi- attribute utility approach has been used to assess the time, cost, and quality of each set of building methods. The resulted solutions in a form of particles are represented using the PSO approach. Also, to assist in putting the suggested methods into practice, the FMOPSO framework has been developed. To test the effectiveness of the FMOPSO a TCQT problem of a bridge building project is solved. The study validated the usefulness of the FMOPSO through solving the TCQT problem of a cas study of a bridge located in Iran. Additionally, the FMOPSO's effectiveness or convergence performance has been contrasted with a GA approach which also uses the constrained fuzzy arithmetic and fuzzy multi-attribute utility suggested in the study. However, the model could only generate a single optimal solution unlike the pareto set that can be provided when using genetic algorithms. Also, this study didn’t consider the fuzziness during estimating the weights of the objectives for calculating the fuzzy composite attribute utility.

Han, and colleagues [2] proposed a Multi-Objective Particle Swarm (MOPSO) Solver which is basically a multi-optimization system employing the MOPSO algorithm that is compatible with Autodesk Revit and its visual programming environment Dynamo. The proposed model offered insights on the demonstration of multi-objective optimization for architectural design using a parametric BIM environment and was proven for its great potential to be a helpful tool for optimizing designs more effectively. it was proven that the model can produce accurate precise findings when using the typical test functions as well as providing reasonable results when optimizing simple building design problem. However, the model was limited to run using the versions Dynamo and Revit used in the study and needs to be developed to run in more environment. With that being said, the progress and implementation of multi-objective optimization for architectural design using a parameterized BIM environment is the research's main contribution.

3.3. Ant Colony Algorithms

Ant-colony optimization (ACO) algorithms evolve similarly to PSO, but through social behavior rather than genetic change. Pheromone trails, that ants leave behind as they walk, are used for communicating. The first pheromone discharges will be identical in all directions when ants randomly revolve around a barrier as they exit their colony to look for a food supply. When the ants on the shorter way locate a source of food, they transport the food and begin turning around, leaving behind more pheromone as they go. An ant will probably take the shortest route while returning to the nest with food because this path will contain the most pheromones placed along it. For this reason, newly emerged ants eventually leave the nest in search of food [14]. See Figure 5

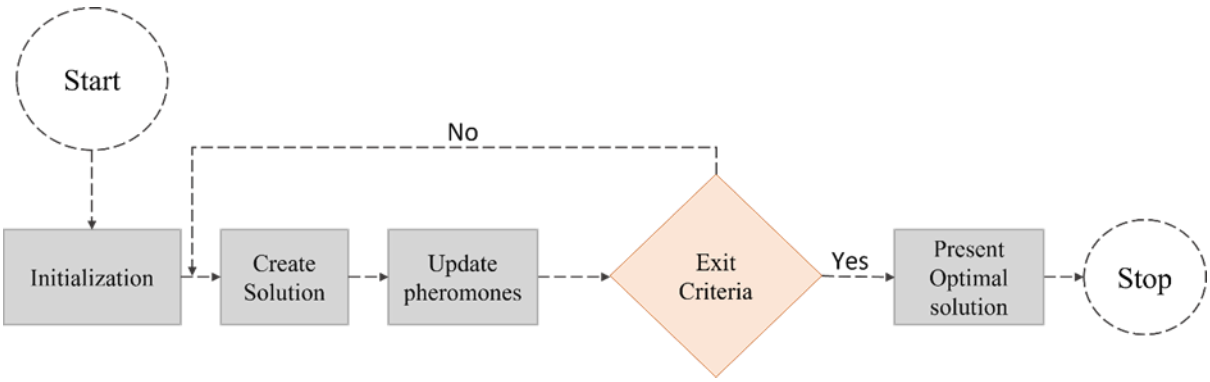

Figure 5: Implementation structure of ant colony algorithms. Source: Zhao, and colleagues [39] modified by author.

Zuo, and colleagues [40] proposed (PBACO) which is an improved ant-colony algorithm with an approach to multi-objective optimization of task scheduling problems in cloud computing.

The approach offered a resource cost model that specifies the requirement of tasks on resources with a focus on the variation of resources and activities in cloud computing. Moreover, this approach achieves multi-objective optimization among both performance and cost by considering the make-span and the user's budget expenditures as optimization problem limitations.

These two constraint functions forced the algorithm to promptly modify the quality of the solution in response to feedback in order to get the best outcome. Four metrics—the make-span, cost, rate of deadline violations, and resource utilization—were used in certain simulation trials to assess this approach's effectiveness. The study results discussed that multi-objective optimization approach outperforms other comparable methods based on these four metrics, especially given that it raised 56.6% in the optimal scenario.

Li and He [41] proposed a novel optimization approach for optimizing quality, time limit, cost, and safety that is based on the analysis of standard multi-objective optimization model where the fitness function that considers the four algorithms decreased the local and avoided the subjective weight analysis.

The model also optimized the global search of the algorithm through integrating the upgraded ant colony factors. The study also discussed that the heuristic data will be obscured when too much pheromone remains on the route, which will eventually slow convergence down greatly, this route is called pheromone concentration path. As a result, the pheromone path computation included a fitness function and the improvement path identified when the optimal is modified by adding the pheromone concentration to the dynamic update path of the fitness function. The convergence speed is accelerated by this pheromone concentration path.

Table 1: Comparative analysis of advantages and disadvantages of four evolutionary algorithms. Source: Author.

Algorithm	Abbreviation	Prime Member	Advantages	Disadvantages
Genetic Algorithm	GA	Genome	• Population based with a good ability to perform global search using its exploring and exploiting features. • Reaches the global optima in a relatively short time. • Adaptable to different problems including large scale projects. • Undergoes probabilistic transition rules not deterministic rules. • Have several customized toolboxes implemented in the programming interfaces such as MATLAB and python.	• fails to converge sometimes, especially when initial search point is not selected carefully.
Memetic Algorithm	MA	Meme	• Combines exploration with exploitation and uses a local search technique to reduce the likelihood of premature convergence to sub-optimal solutions. • Similar to GAs in implementation and operators.	• Small memory size leads to insufficient information which causes convergence to non-optimal solutions. • Too large memory size increases the complexity of the algorithm which decreases the performance efficiency of the algorithm.
Particle Swarm	PSO	Particle	• resistant to design variable sizing. • Simple to perform multiple simultaneous processing. • free of derivatives and has a very few variables in the algorithm. • a powerful algorithm and more robust for global search.	• Poor in local optimum searching
Ant Colony	ACO	Ant	• Convergence is guaranteed. • free of derivatives. • Efficient to TSP and related problems.	• Coding is complicated and not straightforward. • By iteration, the probability distribution evolves. • Uncertainty in convergence time.

To sum it up, Table 1 summarizes the advantages and disadvantages of the four evolutionary algorithms discussed in this paper as follows:

4. Discussion and conclusion

Being a meta-heuristic approach, Evolutionary algorithms have been developed to overcome the practical challenges highlighted by the mathematical approach which are both complex and time-consuming. Such challenges are like the dependence of CSO problem solutions on constraints factor and the requirement for a strong mathematical background. Also, some mathematical approaches are rendered to be vulnerable to local optimality trapping as a result of being oriented towards single objectives only. Therefore, EAs have proven to be more promising in the field of solving the majority of the common scheduling problems. These algorithms are also regarded as problem-independent optimization algorithms because they function by adopting exploration and exploitation searching mechanisms to find the global optimum solutions within a pool of possible solutions. However, the solutions found using EAs are approximate or near-optimal solutions where the solution cannot be determined to be the optimum solution like the solutions obtained from the deterministic methods such as the mathematical approaches. As a result, as compared to deterministic techniques, they take a lot less time to operate. A lot of work has been put into creating multi-objective optimization methods that can find a number of optimum solutions in a single simulation run.

GAs in particular have attracted most of the recent attention found in literature since they can handle a variety of scheduling problems in a reasonable time frame by utilizing their relatively quick concurrent searching strategy that can find an ideal solution by gathering data from the search space. Nevertheless, genetic algorithms are intrinsically challenging in solving scheduling problems since there aren't any clear limitations which consequently makes it challenging to formulate the problem in terms of mathematics. Consequently, the issue becomes entangled in less-than-ideal search space regions. To hasten the convergence of the optimum scheduling solution, it is often preferable to begin the optimization approach at a feasible point.

Relative to GA, PSO has a number of compelling advantages, such as its memory that maintains information about the best solutions inside its particles. Also, its coding is straightforward and free of derivatives with a very few variables in the algorithm as it only follows the present optimal solution. However, several studies have discussed that GA has proven to be more effective with problems involving more than 10 parameters in the optimization as it requires less time and smaller number of iterations to reach an optimum solution compared to PSO search. Also, several developments on genetic algorithms have been proposed such as the Memetic algorithm in order to enhance its efficiency in searching for the optimum solutions.

Basically, MA is a hybrid genetic algorithm that uses local search in crossover to reduce premature convergence to sub-optimal solutions. It also utilizes the local to improve the offspring through improving the members located between the two parents. Nevertheless, when performing with small memory the information collected might not be enough to reach optimal solution while too large memory size will decrease the performance of the algorithm as a result of increasing its complexity. Ant-Colony is one of the evolutionary algorithms used in several dynamic applications as the research in its field is experimental rather than theoretical. Its real advantage is that it gives a good conclusive feedback that helps in improving the discovery of feasible solutions in a robust manner. However, its coding is difficult compared to other EAs and its convergence time span is uncertain despite the convergence itself is guaranteed.

In general, one of the most crucial problems related to all EAs is the premature convergence where population converge to a sub-optimal solution rather than an optimal one. To overcome this problem the entire population should split up into various sub-populations in order to decrease the rate of genetic material transfer between individuals. Another problem is determining the ideal balance between the EA's exploitation and exploring attributes. Controlling the degree of selection pressure by providing specific genetic operators that ensure large population variation at the beginning of the algorithm procedure and a low population variation at the end of the procedure is one approach to solve this problem. Moreover, to solve real world problems, EAs should be developed to run using any available computer technologies to perform a high-dimensional domain optimization.

All in all, EA parameter settings that are adequate in one scenario may not be acceptable in another. Which means that the objective function and search space used to describe each optimization problem differs and the outcomes of applying the chosen EA technique will be significantly influenced by the quality of the selected objective function. It is found to be exceedingly difficult to define a suitable mathematical model with an acceptable objective function when dealing with complex construction problems. Also, the EA approaches lack consistency in results which means each time an optimization process is run a different result is provided. Therefore, maintaining the consistency of the outcomes produced by EA approaches should be the key priority for future development studies [42]. Additionally, more study should be done to investigate the potential of EAs in optimizing more than three objectives as there is a clear shortage of research on this topic. Also, further studies can be done in the field of enhancing EAs performance using hybrid integrations among there operators. The integration of optimization models to produce optimal schedules during the course of a building project is determined to be a promising area of research and further development.

References

[1] V. Toğan and M. A. Eirgash, "Time-Cost Trade-off Optimization of Construction Projects using Teaching Learning Based Optimization," *KSCE Journal of Civil Engineering,* vol. 23, pp. 10-20, 2019/01/01 2019.

[2] Z. Han, N. Cao, G. Liu, and W. Yan, "MOPSO for BIM: a multi-objective optimization tool using particle swarm optimization algorithm on a BIMbased visual programming platform," 2019.

[3] R. Evins, "A review of computational optimisation methods applied to sustainable building design," *Renewable and sustainable energy reviews,* vol. 22, pp. 230-245, 2013.

[4] A. Panwar and K. N. Jha, "A many-objective optimization model for construction scheduling," *Construction Management and Economics,* vol. 37, pp. 727-739, 2019/12/02 2019.

[5] H. Adeli and A. Karim, *Construction scheduling, cost optimization and management*: CRC Press, 2001.

[6] J. Zhou, P. E. D. Love, X. Wang, K. L. Teo, and Z. Irani, "A review of methods and algorithms for

optimizing construction scheduling," *Journal of the Operational Research Society,* vol. 64, pp. 1091-1105, 2013/08/01 2013.

[7] B. Dasović, M. Galić, and U. Klanšek, "A Survey on Integration of Optimization and Project Management Tools for Sustainable Construction Scheduling," *Sustainability,* vol. 12, p. 3405, 2020.

[8] M. König and U. Beißert, "Construction scheduling optimization by simulated annealing," in *Proc. of the 26th Annual International Symposium on Automation and Robotics in Construction, Texas, USA*, 2009.

[9] V. Machairas, A. Tsangrassoulis, and K. Axarli, "Algorithms for optimization of building design: A review," *Renewable and sustainable energy reviews,* vol. 31, pp. 101-112, 2014.

[10] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems* vol. 5: Springer, 2007.

[11] V. Venkrbec, M. Galić, and U. Klanšek, "Construction process optimisation–review of methods, tools and applications," *Građevinar,* vol. 70, pp. 593-606, 2018.

[12] P. Ballesteros-Pérez, K. M. Elamrousy, and M. C. González-Cruz, "Non-linear time-cost trade-off models of activity crashing: Application to construction scheduling and project compression with fast-tracking," *Automation in Construction,* vol. 97, pp. 229-240, 2019.

[13] M. Løvbjerg, "Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality," 2002.

[14] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics,* vol. 19, pp. 43-53, 2005/01/01/ 2005.

[15] M. Janga Reddy and D. Nagesh Kumar, "Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review," *H2Open Journal,* vol. 3, pp. 135-188, 2020.

[16] H. R. Maier, S. Razavi, Z. Kapelan, L. S. Matott, J. Kasprzyk, and B. A. Tolson, "Introductory overview: Optimization using evolutionary algorithms and other metaheuristics," *Environmental Modelling & Software,* vol. 114, pp. 195-213, 2019/04/01/ 2019.

[17] V. Bhosale, S. Shastri, and M. Khandare, "A review of genetic algorithm used for optimizing scheduling of resource constraint construction projects," *International Research Journal of Engineering and Technology,* vol. 4, pp. 2869-2872, 2017.

[18] H. Al-Tabtabai and A. P. Alex, "Using genetic algorithms to solve optimization problems in construction," *Engineering, Construction and Architectural Management,* vol. 6, pp. 121-132, 1999.

[19] S. Shabir and R. Singla, "A comparative study of genetic algorithm and the particle swarm optimization," *International Journal of electrical engineering,* vol. 9, pp. 215-223, 2016.

[20] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications,* vol. 80, pp. 8091-8126, 2021/02/01 2021.

[21] M. Marseguerra, E. Zio, and S. Martorell, "Basics of genetic algorithms optimization for RAMS applications," *Reliability Engineering & System Safety,* vol. 91, pp. 977-991, 2006/09/01/ 2006.

[22] T. Point. (MAY). *Genetic Algorithms - Parent Selection*. Available: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm#

[23] V. Faghihi, K. F. Reinschmidt, and J. H. Kang, "Construction scheduling using Genetic Algorithm based on Building Information Model," *Expert Systems with Applications,* vol. 41, pp. 7565-7578, 2014/11/15/ 2014.

[24] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proceedings of the second international conference on genetic algorithms*, 1987, pp. 14-21.

[25] H. F. Rahman, R. K. Chakrabortty, and M. J. Ryan, "Memetic algorithm for solving resource constrained project scheduling problems," *Automation in Construction,* vol. 111, p. 103052, 2020.

[26] S. M. Lim, A. B. M. Sultan, M. N. Sulaiman, A. Mustapha, and K. Y. Leong, "Crossover and mutation operators of genetic algorithms," *International journal of machine learning and computing,* vol. 7, pp. 9-12, 2017.

[27] M. Hamdan, "On the disruption-level of polynomial mutation for evolutionary multi-objective optimisation algorithms," *Computing and Informatics,* vol. 29, pp. 783-800, 2010.

[28] L. S. MOOI, "Crossover and mutation operators of real coded genetic algorithms for global optimization problems," 2016.

[29] J. C. Ferreira, C. M. Fonseca, and A. Gaspar-Cunha, "Methodology to select solutions from the pareto-optimal set: a comparative study," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 789-796.

[30] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 635-642.

[31] K. Miettinen, "Nonlinear multiobjective optimization kluwer academic publishers," *Boston, Massachusetts,* 1999.

[32] K. Sharma and M. K. Trivedi, "Development of Multi-Objective Scheduling Model for Construction Projects Using Opposition-Based NSGA III," *Journal of The Institution of Engineers (India): Series A,* vol. 102, pp. 435-449, 2021/06/01 2021.

[33] P. Merz and B. Freisleben, "A Genetic Local Search Approach to the Quadratic Assignment Problem," *Proceedings of the 7th international conference on genetic algorithms*, 06/23 1997.

[34] P. Moscato, L. Plata, and M. Norman, "A "Memetic" Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems," vol. 1, 07/22 1999.

[35] S. Singh, A. Chand, and S. Lal, *Improving Spam Detection Using Neural Networks Trained by Memetic Algorithm*, 2013.

[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, pp. 1942-1948.

[37] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence,* vol. 1, pp. 33-57, 2007.

[38] H. Zhang and F. Xing, "Fuzzy-multi-objective particle swarm optimization for time–cost–quality tradeoff in construction," *Automation in Construction*, vol. 19, pp. 1067-1075, 2010.

[39] H. Zhao, H. Zhou, and G. Yang, "Research on Global Path Planning of Artificial Intelligence Robot Based on Improved Ant Colony Algorithm," *Journal of Physics: Conference Series,* vol. 1744, p. 022032, 02/01 2021.

[40] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing," *IEEE Access*, vol. 3, pp. 2687-2699, 2015.

[41] Y. Li and Y. He, "Multi-objective optimization of construction project based on improved ant colony algorithm," *Tehnički vjesnik,* vol. 27, pp. 184-190, 2020.

[42] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Computing and Applications*, vol. 32, pp. 12363-12379, 2020.