ISSN 2347 - 3983
Volume 8. No. 2, February 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter29822020.pdf
https://doi.org/10.30534/ijeter/2020/29822020

An Exploratory Retrospective Assessment on the Usage of Bio-Inspired Computing Algorithms for Optimization

Omdev Dahiya[1], Kamna Solanki[2], Sandeep Dalal[3], Amita Dhankhar[4]
[1,2,4]University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak-124001, India
[3]Department of computer science and applications, Maharshi Dayanand University, Rohtak-124001, India
[1]Omdahiya21792@gmail.com, [2]Kamna.mdurohtak@gmail.com
[3]Sandeepdalal.80@gmail.com, [4]Amita.infotech@gmail.com

ABSTRACT

In the era of digitalization, every task is performed with the help of software-dependent applications. Therefore, the developed software is required to be robust, reliable, and fault free. Testing is performed to check the functioning of the developed software to evaluate whether the software product is error-free or not. Test cases play a vital role in the testing process. However, with the advancement of time, a particular test suite becomes so lengthy that the execution of all the test cases is not possible due to limited time and resources. Researchers have proposed diverse techniques to make the testing process an effective one. This study has worked towards finding the usage of bio-inspired computing algorithms used for optimization. The reason behind this is because these algorithms have performed exceptionally well in addressing complex problems to provide workable solutions in a reasonable time. It is observed that only a handful of these algorithms were applied in testing, such as ant colony optimization, bee colony optimization, neural networks, and genetic algorithms. Even progress is made in the limited area of these algorithms. This study was conducted with a motive to sort out the most popular bio-inspired algorithms and to explore their working principles, developments made till now, along with the scope of their application. This paper has discussed how the development of these algorithms has progressed from already explored algorithms to the development of many new ones such as cuckoo search, artificial bee colony, bat algorithm, firefly algorithm, flower pollination algorithm, and many more. This study will help the researchers to gain insight into choosing the algorithm and explore them in developing new techniques for optimization.

Key words: Bio-inspired computing algorithms, Exploratory review, Software testing, Test optimization.

1. INTRODUCTION

Globalization has integrated the economies and has led to interaction among companies, peoples, and governments worldwide. There is an exchange of information, technology, and exchange of goods and services. The advancements in digital technology boost this. Now we can get anything at our doorstep just by making a few clicks. The researchers have developed various software-based applications. With so much dependence on software applications, there comes a great responsibility also to develop fault-free software [1]. Testing of software is performed to find potential faults in the software so that those can be removed, and the delivery of reliable software be ensured [2-3]. Of many testing techniques available, exhaustive testing intends to test the software for all possible combinations of test cases to make sure that the developed software is completely bug-free. Although its limitation is that its usage is only limited to its applicability to small programs, performing exhaustive testing of large applications is not a feasible idea considering the limited resources, time, and cost [4-5]. Regression testing plays a significant role as its techniques intent to rationalize the test suite size. Regression testing is performed to check whether no new faults have crept in after the modifications were made to the software [6-8]. Finding the optimum solution to a problem is an uphill task as applications have to deal with NP-hard problems. For the solution of these problems, various optimization tools were used. Still, there is no surety that the optimum solution will be obtained. With so much of complexities involved in this process, there is a requirement to develop intelligent approaches so that workable and suitable solutions to the problems can be discovered. Based on these contexts, intelligent metaheuristics algorithms can fulfill the need to find answers to complex issues. Within the domain of metaheuristics, bio-inspired computing algorithms are gradually gaining popularity as they can learn and adapt like biological organisms, and they have supremacy in addressing complex problems to provide workable solutions in a reasonable time. It is also getting difficult to track the developments in this domain because many algorithms are getting introduced at a fast pace. Still, this study was conducted to find the most popular bio-inspired algorithms and to explore their working principles, developments made till now, along with the scope of their application. Earlier attempts have also been made by researchers to address this issue [9-10]. Due to limited awareness amongst the new

researchers, they are just fitting the algorithms in their study instead of exploring them according to the problem statement. This study has worked towards addressing this issue. We have reviewed the most popular as well as newly discovered bio-inspired algorithms that hold great potential for the developed applications. In this study, we have tried to explore the applications and scope of these algorithms in a specific context. It will help researchers to gain insight into choosing the algorithm and explore them in developing new techniques for test optimization. We are not able to provide the in-depth detail of an algorithm to show how it will be implemented and how potential enhancement can be made in them to make them more efficient as this is beyond the scope of this study. We can provide a generalized overview based on the other reviews performed by numerous researchers [11-34]. Fig. 1 shows the generalized taxonomy of bio-inspired computing algorithms.

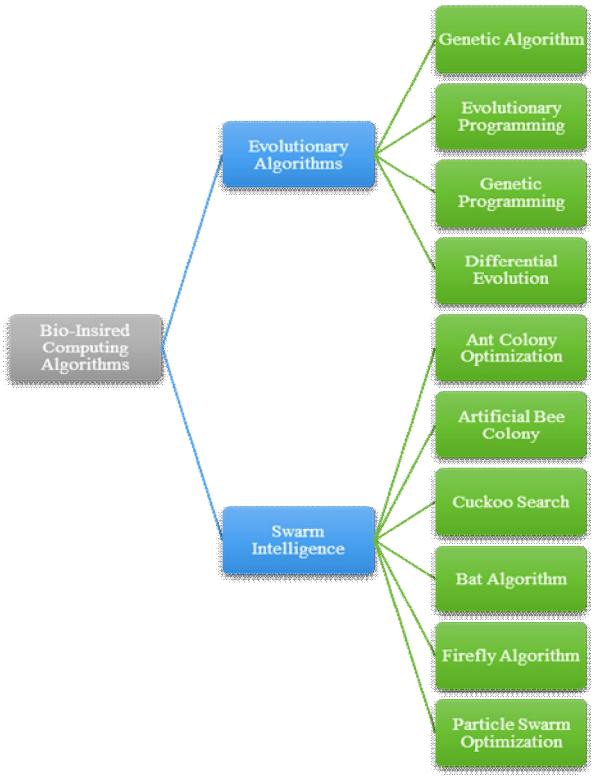

Figure 1: Generalized classification of bio-inspired computing algorithms

In the following sections, this research work discussed how the development of these algorithms has progressed from already explored algorithms to the development of many new ones such as cuckoo search, artificial bee colony, bat algorithm, firefly algorithm, flower pollination algorithm and many more. This study will help the researchers to gain insight into choosing the algorithm and explore them in developing new and effective techniques for test optimization as well as in solving complex real-life problems.

2. RESEARCH METHODOLOGY

This study was conducted in three different phases. In the first phase, we searched articles on google scholar and Scopus database by the combination of various search words and techniques of software testing such as "bio-inspired computing algorithms, metaheuristic algorithms, heuristics, nature-inspired algorithms, application of nature-inspired algorithms and software testing." The searched results were refined, utilizing various parameters, inclusion, and exclusion criteria. First, here is title-based exclusion, then abstract based and then conclusion based. If a study has passed through these filters, then it is studied in detail to find whether it is relevant to the importance and usage of bio-inspired computing algorithms for optimization and its usage in software testing. To explore the usage of Bio-inspired algorithms in solving the software testing problems, again searching of studies was performed using keywords like, "Ant colony optimization <and> Software testing," "Genetic algorithm <and> Software testing," "Artificial bee colony <and> Software testing," "Particle swarm optimization <and> Software testing," "Neural networks <and> Software testing," "Bat algorithm <and> Software testing," Cuckoo search <and> Software testing," "Firefly algorithm <and> Software testing," Flower pollination algorithm <and> Software testing." In the second phase, the objective was to arrange the selected studies according to the different algorithms. After arranging the studies according to the algorithms implemented, there comes the third stage, which is to identify different domains and problems for which they were used to solve them. This was done by studying the literature regarding that particular algorithm so that its applicability can be studied in detail. We have worked towards ensuring that most of the useful studies were included in this study. For it, references from selected studies were also used to find other relevant studies. After these phases, we have presented an overall conclusion of this study, followed by the references. Now various algorithms are presented in detail along with their contributors who have either worked towards proposing that particular technique or have presented/applied them for test optimization.

2.1 Ant Colony Optimization

The basic concept behind the ant colony approach is to resolve optimization problems by taking actions similar to real ants. This approach was proposed by Dorigo [35]. Ants have an excellent capability to search for the nearest source of food and then reaching to a destination using their chemical called pheromone. More and more ants traverse the same path for reaching the food source, and the intensity of pheromone

dropped at that path increases, which attracts other ants to traverse that path. Numerous researchers have exploited the potential benefits of solving complex problems using the ant colony system. Table 1 below enlists the prominent researchers who have proposed different testing techniques using this approach and have used to solve various problems as well.

Table 1: Contributors in Ant Colony Optimization Approach

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Dorigo	1995	[35]
2.	Srivastava *et al.*	2008	[36]
3.	Donghua and Wenjie	2011	[37]
4.	Singh *et al.*	2010	[38]
5.	Srivastava	2010	[39]
6.	Suri and Singhal	2011	[40]
7.	Li *et al.*	2009	[41]
8.	Noguchi *et al.*	2015	[42]
9.	Srivastava and Baby	2010	[43]
10.	Chengying *et al.*	2015	[44]
11.	Suri and Singhal	2011	[45]
12.	Suri and Singhal	2012	[46]
13.	Gao *et al.*	2015	[47]
14.	Zhou *et al.*	2017	[48]
15.	Singh *et al.*	2010	[49]
16.	Gao *et al.*	2015	[50]
17.	Dorigo and Birattari	2010	[51]
18.	Dorigo *et al.*	2006	[52]
19.	Bououden *et al.*	2015	[53]
20.	Ghasab *et al.*	2015	[54]
21.	Hong *et al.*	2012	[55]
22.	Liao *et al.*	2014	[56]
23.	Liao *et al.*	2014	[57]
24.	Mandloi and Bhatia	2014	[58]
25.	Romdhane *et al.*	2013	[59]
26.	Ramli *et al.*	2016	[60]
27.	Popentiu-Vladicescua nd Alabeanu	2016	[61]
28.	Panthi and Mohapatra	2016	[62]
29.	Ansari *et al.*	2016	[63]
30.	Guo	2017	[64]
31.	Rauf	2017	[65]
32.	Zhang *et al.*	2017	[66]
33.	Khanna	2017	[67]
34.	Ahmad *et al.*	2018	[68]
35.	Zhang *et al.*	2019	[69]
36.	Zheng	2019	[70]

The above table presents the details of studies that have worked towards using an ant colony algorithm for solving combinatorial optimization problems. This provides complete detail of the articles and will help the newcomers in providing them an in-depth analysis of the algorithm. In the present scenario trend of ant colony optimization algorithm is going on with authors proposing its modified versions, as well as their usage in the collaboration of other factors.

2.2 Genetic Algorithm

A genetic algorithm was designed by getting inspiration from the biological concept of evolution and is based on the "survival of the fittest" theory [71-73]. The algorithm works by selecting a set of solutions that are depicted by chromosomes and are called population. This is used to form a new population by applying basic operators such as "selection," "crossover," "mutation," and this process goes on until an optimal solution is attained. The underlying idea is that the new population will be efficient than the previous one. This algorithm has been widely used to solve a variety of optimization problems. The algorithm has shown its supremacy in solving complex and real-life problems. Due to this, many researchers have worked towards using a genetic algorithm for solving problems and proposing different techniques. Table 2 below enlists the prominent researchers who have proposed different testing techniques using this approach and as well as to solve various optimization problems.

Table 2: Contributors in Genetic Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Varshney and Mehrotra	2013	[74]
2.	Khor and Grogono	2004	[75]
3.	Gulia and Chillar	2012	[76]
4.	Bhasin	2014	[77]
5.	Srivastava *et al.*	2009	[78]
6.	Rathore *et al.*	2011	[79]
7.	Prakash *et al.*	2015	[80]
8.	Rao *et al.*	2013	[81]
9.	Mahajan *et al.*	2012	[82]
10.	Lodha *et al.*	2014	[83]
11.	Srivastava and Kim	2004	[84]
12.	Froser and Arcuri	2013	[85]
13.	Ghiduk	2014	[86]
14.	Umbarkar and Sheth	2015	[87]
15.	Boopathi *et al.*	2014	[88]
16.	Mishra *et al.*	2017	[89]
17.	Ahmed *et al.*	2012	[90]
18.	Aytug *et al.*	2003	[91]
19.	Colin and Jonathan	2002	[92]

20.	Grefenstette	2013	[93]
21.	Mitchell *et al.*	1992	[94]
22.	Srinivas and Patnaik	1994	[95]
23.	Siarry *et al.*	2002	[96]
24.	Yao *et al.*	2005	[97]
25.	Zhou *et al.*	2014	[98]
26.	Sharma *et al.*	2016	[99]
27.	Khan *et al.*	2016	[100]
28.	Qi *et al.*	2016	[101]
29.	Khanna	2016	[102]
30.	Soltani	2017	[103]
31.	Kudjo	2017	[104]
32.	Bahaweres	2017	[105]
33.	Yadav and Dutta	2017	[106]
34.	Goyal *et al.*	2018	[107]
35.	Mansour *et al.*	2018	[108]
36.	Bala and Chillar	2018	[109]
37.	Boopathi *et al.*	2019	[110]
38.	Mishra *et al.*	2019	[111]
39.	Bhattacharjee and Saluja	2019	[112]
40.	Dubey	2019	[113]
41.	Habtemariam and Mohapatra	2019	[114]
42.	Bhatia	2020	[115]
43.	Fan *et al.*	2020	[116]

The above table presents a detail of the articles with the authors who have worked towards using a genetic algorithm to solve various problems like job scheduling, traveling salesman problem, and constrained optimization problems. This will enable the coming fellows who want to propose different techniques for solving complex problems of computer science as well as real-life problems using a genetic algorithm. After analysis of the studies, they may deduce how the algorithm can be used and what are its application areas.

2.3 Artificial Bee Colony

The artificial Bee colony algorithm was introduced by Karaboga [117]. The basic idea behind this technique is deduced from the intelligent behavior of honey bees. In this bee colony, there are three types of bees (a) "Employed bees" (b) "Onlooker bees" (c) Scouts bees. The number of "employed bees" corresponds to the number of food sources near a beehive. They search for a potential food source with a random stimulus, and then the fitness of the food source is assessed. This information is communicated amongst them. Various activities done by them shows their behavior, such as how the task is allocated, how they interact, their navigational behavior, and many more. Many studies were conducted to explain the functioning of the queen, their communication and dance strategy, how they mate and reproduce, and their navigational behavior. Researchers have used this technique to solve various optimization problems amongst different domains of engineering and sciences. Table 3 enlists prominent contributors in the field of artificial bee colony optimization.

Table 3: Contributors in Artificial Bee Colony Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Drias *et al.*	2005	[118]
2.	Karaboga and Basturk	2007	[119]
3.	Lucic and Teodorovic	2001	[120]
4.	Teodorovic and Dell'Orco	2005	[121]
5.	Jia *et al.*	2016	[122]
6.	Yu	2016	[123]
7.	Nseef *et al.*	2016	[124]
8.	Biswas *et al.*	2014	[125]
9.	Li *et al.*	2015	[126]
10.	Karaboga and Basturk	2008	[127]
11.	Karaboga *et al.*	2014	[128]
12.	Karaboga and Ozturk	2011	[129]
13.	Gao and Liu	2012	[130]
14.	Karaboga and Akay	2009	[131]
15.	Bansal *et al.*	2016	[132]
16.	Liu *et al.*	2016	[133]
17.	Ma *et al.*	2016	[134]
18.	Aghdam and Arasteh	2017	[135]
19.	Alazzawi *et al.*	2017	[136]
20.	Roeva	2018	[137]
21.	Chhabra	2018	[138]
22.	Yilmaz and Bascifti	2018	[139]
23.	Sheoran *et al.*	2019	[140]
24.	Peng *et al.*	2019	[141]
25.	Luo	2019	[142]
26.	Sidek	2019	[143]
27.	Akay and Akay	2020	[144]
28.	Mishra *et al.*	2020	[145]
29.	Alazzawi *et al.*	2020	[146]

The above table presents a detail of the articles with the authors who have worked towards using an artificial bee colony algorithm to solve various maximization or minimization problems, routing problems, searching problem, task allocation problem. This provides full detail of the articles and will help the newcomers in providing them an in-depth analysis of the algorithm.

2.4 Particle Swarm Optimization

In particle swarm optimization, Swarm refers to a large number of homogeneous agents who interact amongst themselves in their environment. This algorithm is based on the behavior of organisms in a group, such as depicted by bird flock, fish, or insects to achieve an optimal solution [147-148]. In this, members of the group try to make a shared objective according to feedback from the other group members. Each member of the group tries to find a possible solution at any instant of time. After that suitability of that candidate solution is communicated to other members of the swarm via signals. Other members, therefore, senses the strength of the transmitted signal, and according to the fitness function, the suitability of the candidate solution is assessed. This algorithm has helped the researchers in solving complex problems by reaching an optimal solution based on varying criteria. Table 4 enlists prominent contributors who have explored the domains where particle swarm optimization approaches have been used to solve various multi population-based optimization problems.

Table 4: Contributors in Particle Swarm Optimization Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Kennedy	1997	[149]
2.	Jiang *et al.*	2015	[150]
3.	De Souza *et al.*	2011	[151]
4.	Zhan *et al.*	2011	[152]
5.	Couceiro and Ghamisi	2016	[153]
6.	Gandomi *et al.*	2013	[154]
7.	Hong	2009	[155]
8.	Kennedy	2011	[156]
9.	Liu *et al.*	2005	[157]
10.	Shi and Eberhart	1998	[158]
11.	Niu *et al.*	2005	[159]
12.	Liang and Suganthan	2005	[160]
13.	Zhao *et al.*	2010	[161]
14.	Liang and Suganthan	2005	[162]
15.	Yang and Li	2010	[163]
16.	Xu *et al.*	2015	[164]
17.	Chen *et al.*	2010	[165]
18.	Zheng and Liu	2009	[166]
19.	Bolufe and Chen	2011	[167]
20.	El Dor *et al.*	2012	[168]
21.	Zhang and Ding	2011	[169]
22.	Zhang *et al.*	2011	[170]
23.	Wang *et al.*	2012	[171]
24.	Fan and chang	2010	[172]
25.	Marinakis	2014	[173]
26.	Liang and Suganthan	2006	[174]
27.	Wang and Lai	2009	[175]
28.	Xiao and Cheng	2013	[176]
29.	Niu *et al.*	2006	[177]
30.	Jatana *et al.*	2016	[178]
31.	Sheng *et al.*	2017	[179]
32.	Hajihassani *et al.*	2018	[180]
33.	Sun *et al.*	2018	[181]
34.	Wang and Liu	2018	[182]
35.	Jianqi *et al.*	2018	[183]
36.	Khatibsyarbini *et al.*	2018	[184]
37.	Allawi *et al.*	2018	[185]
38.	Nayak and Ray	2019	[186]
39.	Malhotra and Khanna	2019	[187]
40.	Islam *et al.*	2019	[188]

The above table presents a detail of the articles with the authors who have worked towards the particle swarm optimization algorithm for solving various scheduling problems, multi-criteria-based decision problems, and constraint-based optimization problems. This will enable the coming fellows who want to propose different techniques for solving complex problems of computer science as well as real-life problems using this optimization algorithm.

2.5 Neural Networks

Neural Networks work as a human neuron system, and data is processed, which mimics in the same way a human brain operates. It is a non-linear data processing algorithm; numerous processing unit is combined in a different layered network [189]. These networks are adaptive, organizing in nature, and possess the ability to learn based on feedback and input from their operating environment. Depending on the accuracy of results, feedback could either be negative or positive. The whole network is just like a black box where the provided input and the output is visible to the user. If the output is not as expected, then the feedback of the result is fetched back so that the processing model could be improved. The most straightforward implementation of the neural networks is that of a perceptron network. In it, there is a single layer for internal functioning, and feedback is provided to improve output. Neural networks have been implemented on various domains ranging from different types of systems to different kinds of interfaces according to the requirement. Numerous studies have been conducted to examine different criteria on how the network can be trained. According to the necessity of the problem, neural networks have also been used in conjunction with other algorithms so that the prediction capability of a system could be improved further. Table 5 enlists prominent contributors in the field of neural network implementation to solve various optimization problems.

Table 5: Contributors in Neural Networks

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Kar	2013	[190]
2.	Sadegh	1993	[191]
3.	Kar	2015	[192]
4.	Hornik	1991	[193]
5.	Specht	1990	[194]
6.	Schmidhuber	2015	[195]
7.	Oja	1992	[196]
8.	Chen *et al.*	2008	[197]
9.	Fausett	1994	[198]
10.	Craven and Shavlik	1997	[199]
11.	Lampinen and Vehtari	2001	[200]
12.	Mensah *et al.*	2016	[201]
13.	Makondo *et al.*	2016	[202]
14.	Pang *et al.*	2017	[203]
15.	Sathyavathy	2017	[204]
16.	Sun *et al.*	2018	[205]
17.	Ma *et al.*	2018	[206]
18.	Byun *et al.*	2019	[207]
19.	Karpov *et al.*	2018	[208]
20.	Joffe and Clark	2019	[209]
21.	Mannarswamy *et al.*	2020	[210]
22.	Ghosh and Singh	2020	[211]

The above table presents a detail of the articles with the authors who have worked towards using neural network-based approaches to solve various problems such as classification problems, pattern recognition, association rules, missing data prediction, data normalization, and various optimization problems. This will enable the coming fellows who want to propose different techniques for solving complex problems of computer science as well as real-life problems. After analysis of the studies, they may deduce how the algorithm can be used and what are its application areas.

2.6 Bat Algorithm

Bat algorithm was developed based on the behavior of bats, which uses echo-based parameters for locating their prey. It is one of the recently developed bio-inspired computing algorithms. In this, bats navigate in the surroundings to identify and catch their prey even in the dark, by using the echoes of sound emitted by them [212-213]. This process is known as echolocation. Using this approach, bats can find their food source by identifying the distance. While searching for their prey, bats can easily adjust their flight velocity, frequency, and loudness of their cry. The loudness and frequency are updated in such a fashion that when the prey is identified, the frequency increases, and loudness decreases. This algorithm is used to solve various multi-objective optimization problems. Various researchers have used this in combination with nature-inspired algorithms as well. Table 6 enlists prominent contributors who have worked for the approaches of the Bat algorithm.

Table 6: Contributors in Bat Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Wang *et al.*	2015	[214]
2.	Heraguemi *et al.*	2015	[215]
3.	Mirjalili *et al.*	2014	[216]
4.	Jaddi *et al.*	2015	[217]
5.	Gandomi *et al.*	2013	[218]
6.	Heraguemi *et al.*	2016	[219]
7.	Gandomi and Yang	2014	[220]
8.	Meng *et al.*	2015	[221]
9.	Rodrigues *et al.*	2014	[222]
10.	Yang and He	2013	[223]
11.	Yang and Gandomi	2012	[224]
12.	Alsariera and Zamli	2017	[225]
13.	Ashish *et al.*	2018	[226]
14.	Sharma and Sehgal	2018	[227]
15.	Ozturk	2018	[228]
16.	Alsariera *et al.*	2018	[229]
17.	Huang *et al.*	2019	[230]

The above table presents a detail of the articles with the authors who have worked towards using bat algorithm to solve various classification problems, multi-objective optimization problems, clustering, and combinatorial optimization problems. Many studies have reported its supremacy over other nature-inspired approaches. This table provides complete detail of the articles and will help the newcomers in providing them an in-depth analysis of the algorithm.

2.7 Cuckoo Search

The cuckoo search algorithm is developed by replicating the breeding behavior of cuckoos [231]. Other birds' nests were often used by cuckoos to lay their eggs, and they remove the eggs of those birds to ensure that the hatching probability of their eggs is ensured. This algorithm imitates three types of brood parasitism, namely a). "co-operative breeding" b). "intra-specific brood parasitism" and c). "Nest takeover." One or more eggs were laid by the cuckoo bird, and for doing that, "levy flight" is done for the identification of nests. Then eggs were put in the randomly chosen nest. The best nests which have eggs of high quality will be carried over to the next generations. The probability of host parents identifying the eggs laid by the cuckoo is also high because a limited number of host nests are available. If that happens, the host

parent either abandons the nests or eggs were thrown away, and a new nest is built. A "levy flight" is performed whenever a new set of a candidate solution is generated. To identify globally optimal solution utilizing a diversity of eggs in the nest, mixing and redistribution of eggs of two or more nests is done. This algorithm is proposed to address single and multi-objective problems under complex non-linear constraints where achieving global optimization is not an easy task. It provides an efficient, workable solution for challenging problems. Table 7 enlists prominent contributors in the field of the cuckoo search algorithm to solve various optimization problems.

Table 7: Contributors in Cuckoo Search Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Walia and Kapoor	2014	[232]
2.	Yang and Deb	2013	[233]
3.	Yang and Deb	2014	[234]
4.	Araghi *et al.*	2015	[235]
5.	Yang and Deb	2014	[236]
6.	Bhandari *et al.*	2014	[237]
7.	Gandomi *et al.*	2013	[238]
8.	Gotmare *et al.*	2015	[239]
9.	Kumar and Rawat	2015	[240]
10.	Yang and Deb	2014	[241]
11.	Srivastava	2012	[242]
12.	Khari and Kumar	2016	[243]
13.	Dhabal *et al.*	2016	[244]
14.	Khari and Kumar	2017	[245]
15.	Haixian and Jing	2018	[246]
16.	Shehab *et al.*	2018	[247]
17.	Sharma *et al.*	2019	[248]
18.	Dhareula and Ganpati	2019	[249]
19.	Benkhaira and Layeb	2020	[250]

The above table presents a detail of the articles with the authors who have worked towards using a cuckoo search algorithm to solve various single or multi-objective problems, scheduling problems, knapsack problems, and many more. This will provide an insight into the coming fellows who want to propose different techniques for solving complex problems of computer science as well as real-life problems. After analysis of the studies, they may deduce how the algorithm can be used and what are its application areas.

2.8 Firefly Algorithm

The inspiration for the firefly algorithm is achieved from the flashing behavior of the fireflies. The fireflies produce a flashlight through a process known as bioluminescence [251]. The fireflies use this flashlight for food foraging and to attract potential mating partners as well as to remind them about predators. They exhibit characteristics of swarm intelligence through decentralized decision making and self-organization. The brightness in their flash indicates the fitness of male fireflies. There are certain assumptions for the standard firefly algorithm to operate, such as A). "A firefly will be attracted to each other regardless of their sex because they are unisexual." B). "Attractiveness is proportional to their brightness, whereas the less bright firefly will be attracted to the brighter firefly. However, the attractiveness decreased when the distance of the two fireflies increased." C). "If the brightness of both fireflies is the same, the fireflies will move randomly." In this, the initial population of fireflies was created, which was followed by modifying a fitness parameter. Then the evaluation of fitness for each firefly in the population was performed. Based on this, ranking to the fireflies is provided. For the next round of evaluation, only the best solutions are taken forward. Many computations are planned to control the iteration. The firefly algorithm was also used in conjunction with other algorithms as well to obtain the enhanced results. This algorithm can more efficiently use Multimodal functions as compared to different swarm-based algorithms. The algorithm has found its applicability in dealing with multi-objective search problems, combinatorial optimizations problems, and many more. Table 8 enlists prominent contributors in the field of the firefly algorithm.

Table 8: Contributors in Firefly Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Yang	2010	[252]
2.	Srivastava *et al.*	2013	[253]
3.	Yang and Deb	2010	[254]
4.	Xu and Liu	2013	[255]
5.	Gandomi *et al.*	2013	[256]
6.	Ozsoydan and Baykasoglu	2015	[257]
7.	KavousiFard *et al.*	2014	[258]
8.	Gandomi *et al.*	2011	[259]
9.	Long *et al.*	2015	[260]
10.	Lukasik and Zak	2009	[261]
11.	Rahmani and MirHassani	2014	[262]
12.	Yang and He	2013	[263]
13.	Mishra *et al.*	2014	[264]
14.	Verma *et al.*	2016	[265]
15.	Yang *et al.*	2012	[266]
16.	Khatibsyarbini *et al.*	2019	[267]
17.	Arora and Saha	2018	[268]
18.	Hashim and Dawood	2018	[269]
19.	Rathee *et al.*	2019	[270]
20.	Pandey and Banerjee	2019	[271]

The above table presents a detail of the articles with the authors who have worked towards the firefly algorithm for solving various classification problems, discrete search-based problems, and NP-hard problems. This will enable the coming fellows who want to propose different techniques for solving complex problems of computer science as well as real-life problems using this optimization algorithm.

2.9 Flower Pollination Algorithm

Flower pollination algorithm was developed based on the pollination mechanism of the flowers [272]. Pollination is a process of spreading the pollen of a flower, also known as the reproduction unit of a flower, to another flower of a plant for germination/reproduction. This process is achieved through agents known as pollinators. Usually, biotic pollination is done by most of the flowers by agents such as birds or insects. In this pollination, flowers of the different plants are involved. It is known as cross-pollination and is considered a global/optimization pollination process. Some plants also perform abiotic pollination through agents like water or wind. Here flowers of the same plant are involved. It is known as self-pollination, and it is considered as a local/optimization pollination process. The process of pollination would try to enhance the reproduction of the fittest. Table 9 enlists prominent contributors to the flower pollination algorithm.

Table 9: Contributors in Flower Pollination Algorithm

Sr. No.	Authors Name	Publication Year	Reference No.
1.	Yang *et al.*	2013	[273]
2.	Bekdas *et al.*	2015	[274]
3.	Yang *et al.*	2014	[275]
4.	Nigdeli *et al.*	2016	[276]
5.	Kabir *et al.*	2017	[277]
6.	Naseer *et al.*	2019	[278]
7.	Naseer *et al.*	2018	[279]
8.	Abdel-Basset and Shawky	2019	[280]
9.	Naseer and Zamli	2018	[281]
10.	Dhareula and Ganpati	2019	[282]
11.	Dhareula and Ganpati	2019	[283]

The above table presents a detail of the articles with the authors who have worked towards using the flower pollination algorithm for solving complex problems. This algorithm has found its applicability for global optimization problems with multiple objectives and various criteria. It is also used to solve significant integer programming problems, global and local search problems. It has been used in numerous disciplines of engineering and sciences. This will enable the coming fellows who want to propose different techniques for solving complex problems of computer science as well as real-life problems. After analysis of the studies, they may deduce how the algorithm can be used and what are its application areas.

2.10 Other Nature-Inspired Algorithms

This study has tried to cover most of the famous and upcoming bio-inspired computing algorithms for test optimization. Though there were few algorithms as well, which have been developed by the researchers but have not become so popular either due to limited area of application or other researchers could not explore their usability in their particular domain. Considering the aim of this study, we have still documented those algorithms in the form of a table. This will aid the upcoming researchers in exploring them further for their scope of application across domains. Researchers may explore the quality of outcome by using trying and mixing these algorithms with other existing theories. Table 10 enlists such algorithms along with the details of their references.

Table 10: Contributors in other Nature-Inspired Algorithms

Sr. No.	Algorithm Details	Authors Name	Publication Year	Reference No.
1.	Monkey Search	Mucherino and Seref	2007	[284]
2.	Fruit Fly Algorithm	Pan	2012	[285]
3.	Virus Colony Search	Li *et al.*	2016	[286]
4.	Dolphin Swarm Algorithm	Yong *et al.*	2016	[287]
5.	Squirrel Search	Jain *et al.*	2019	[288]
6.	Butterfly Algorithm	Qi *et al.*	2017	[289]
7.	Krill Herd	Gandomi and Alavi	2012	[290]
8.	Bacterial Foraging	Passino	2002	[291]
9.	Wolf Search	Tang *et al.*	2012	[292]
10.	Bean Algorithm	Zhang *et al.*	2010	[293]
11.	Amoeba Algorithm	Zhang *et al.*	2013	[294]
12.	Shark Algorithm	Hersovici	1998	[295]
13.	Lion Algorithm	Yazdani and Jolai	2015	[296]
14.	Dove Algorithm	Su *et al.*	2009	[297]

3. RESULT AND DISCUSSION

This study has worked towards presenting a comprehensive source of information so that further research can be conducted where this research will serve as a base or starting point. We have worked towards documenting the work of numerous researchers on different bio-inspired algorithms to solve large-scale real-world problems for finding optimal solution and their usage in software testing as well. As testing overall improves the quality of software [298]. Researchers were inspired in diverse ways to develop such diverse algorithms getting inspired from nature. We observed that some algorithms were more popular than others, as they were developed a long time back. People were using these algorithms for an extended period; that is why the number of studies published for them has been increased. However, with time, researchers have also proposed new algorithms, and they are in trend. Figure 1 shows the publication statistics for the studies that were selected for this article. A constant increase in published studies is observed starting from 1990 to 2020. A period from 2010 onwards shows a much more trend for the relevant studies. So, it can be said that the coming years will experience a much more emphasis on the usage of Bio-inspired algorithms for solving optimization problems. To further categorize the selected studies based on the algorithms used, we have prepared a pie chart distribution. In this, it can be observed that out of 251 studies selected for different algorithms, 36 were based on ant colony optimization algorithm which is about 14%, 43 were based on genetic algorithm which is about 17%, 29 were based on artificial bee colony which is about 11%, 40 were based on particle swarm optimization which is about 16%, 22 were based on neural networks which are about 9%, 17 were based on bat algorithm which is about 7%, 19 were based on cuckoo search which is about 8%, 20 were based on firefly algorithm which is about 8%, 11 were based on flower pollination algorithm which is about 4% and 14 were based on other remaining nature-inspired algorithms which is about 6%. From the analysis of Figures 1 and 2, this study believes that much work has been done on ant colony optimization algorithms, neural networks, particle swarm optimization, genetic algorithms, and artificial bee colony. Now the current era is experiencing the development and usage of algorithms like bat algorithm, cuckoo search, firefly algorithm, and flower pollination algorithm.

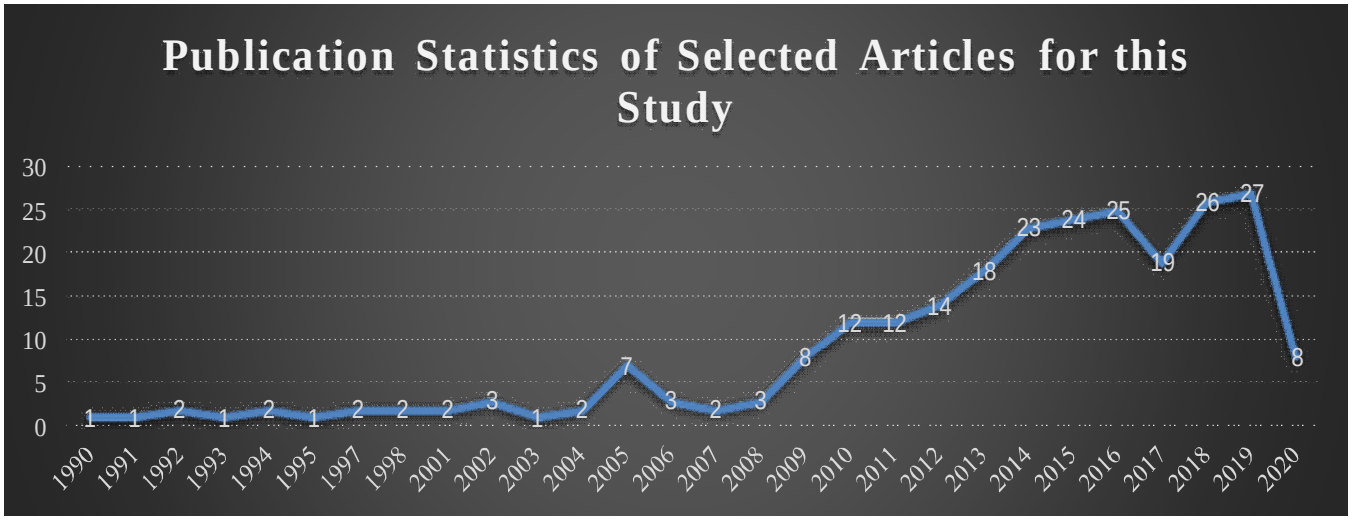

Figure 1: Trend in publication for studies based on the usage of Bio-inspired computing algorithms for optimization

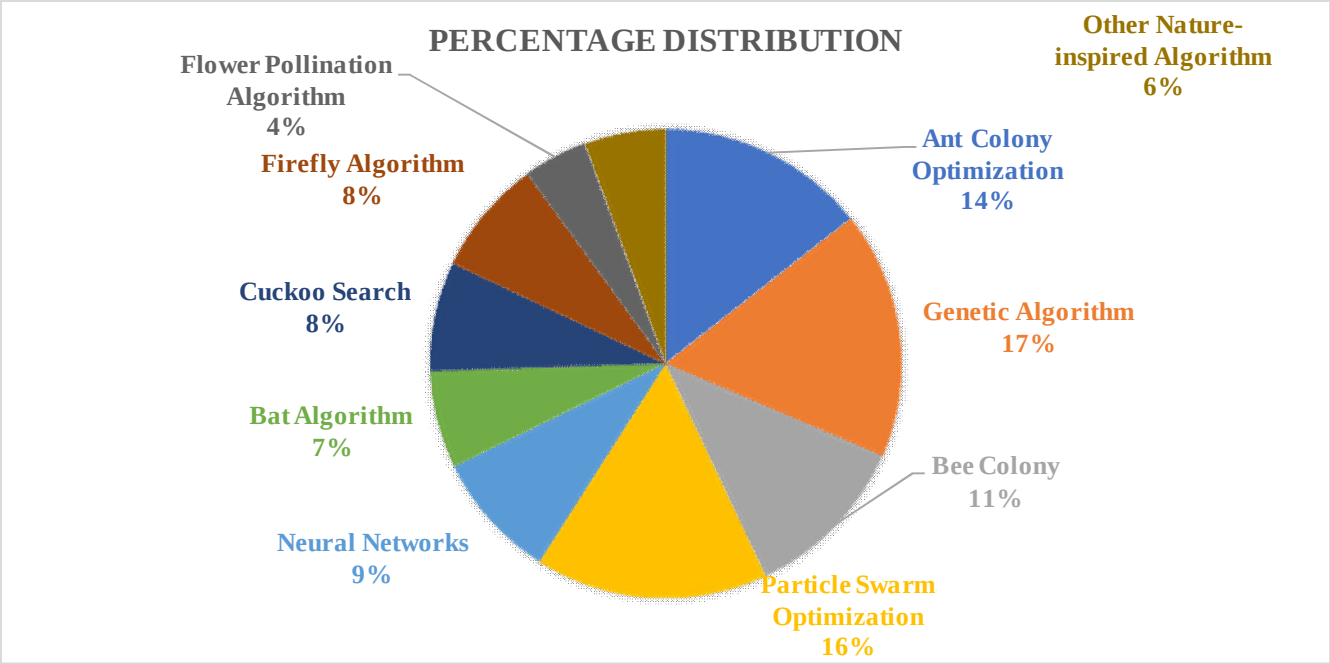

Figure 2: Percentage usage of different Bio-inspired computing algorithms in the selected studies

4. CONCLUSION

Optimization is a process of finding the best possible solutions. There is no certainty whether the optimal solution will be achieved or not. This study has grouped the various bio-inspired algorithms after reviewing the existing literature available for them. We have then presented the details of numerous researchers who have worked in that particular approach. The area of the applicability and scope of the algorithms is also documented. We have shown how, with the advent of time, different algorithms were proposed and what is current trends in terms of the development of those algorithms. The aim of this study is not to highlight how real-life problems can be solved using them, but to provide and act as a base to understand the underlying objective and scope of these algorithms. This paper will help the readers to find review papers of the documented algorithms so that further understanding could be developed for the development of theories and in the scope of their application. After understanding these algorithms, practitioners may use this study to explore more such bio-inspired algorithms for their application across different domains. Future work lies in exploring these algorithms further to improve their performance by introducing different dimensions and new improvements.

REFERENCES

1. Sommerville, Ian ***"What is software engineering?".*** Software Engineering (8th ed.). Harlow, England: Pearson Education p.p. 7, 2007.
2. R. Mukherjee and K. S. Patnaik, **"A survey on different approaches for software test case prioritization,"** Journal of King Saud University-Computer and Information Sciences, 2018.
3. M. Khatibsyarbini, M. A. Isa, D. N. Jawawi, and R. Tumeng**, "Test case prioritization approaches in regression testing: A systematic literature review."** Information and Software Technology, 93, pp.74-93, 2018.
4. O. Dahiya, K. Solanki and S. dalal, **"Comparative Analysis of Regression Test Case Prioritization Techniques,"** International Journal of advanced trends in computer science and engineering, Vol. 8 No. 4, pp. 1521-1531, 2019. https://doi.org/10.30534/ijatcse/2019/74842019
5. K. Solanki, and Y. Singh, **"Importance of Selecting Test Cases for Regression Testing,"** IOSR Journal of Computer Engineering (IOSRJCE) e-ISSN, pp. 2278-0661, 2014.
6. O. Dahiya and K. Solanki, **"Comprehensive cognizance of Regression Test Case Prioritization Techniques,"** International journal of emerging trends in engineering research, Vol. 7 No. 11, pp. 638-646, 2019. https://doi.org/10.30534/ijeter/2019/377112019
7. K. Solanki, and Y. Singh, **"Novel Classification of Test Case Prioritization Techniques,"** International Journal of Computer Applications, Vol. 975, pp. 8887, 2014.
8. O. Dahiya and K. Solanki, **"A systematic literature study of regression test case prioritization approaches,"** International Journal of Engineering & Technology, Vol. 7, No. 4, pp.2184-2191, 2018.

9. Jr. I. Fister, X.S. Yang, I. Fister, J. Brest, and D. Fister**, "A brief review of nature-inspired algorithms for optimization."** arXiv preprint arXiv:1307.4186.
10. A. K. Kar, **"Bio inspired computing–a review of algorithms and scope of applications,"** Expert Systems with Applications, 59, pp.20-32, 2016.
11. H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, and H. Zhou, H., **"Multi-population techniques in nature inspired optimization algorithms: a comprehensive survey."** Swarm and evolutionary computation, 44, pp.365-387, 2019.
12. A. Gogna and A. Tayal, **"Metaheuristics: review and application."** Journal of Experimental & Theoretical Artificial Intelligence, 25(4), pp.503-526, 2013.
13. S. X. Yang**, "Review of meta-heuristics and generalised evolutionary walk algorithm,"** International Journal of Bio-Inspired Computation, 3 (2), pp. 77–84, 2013.
14. G. Theraulaz, **"Task differentiation in Polistes wasp colonies: a model for self-organizing groups of robots,"** In *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animates, 1991* (pp. 346-355). The MIT Press, 1991.
15. M. Hersovici, M. Jacovi, Y.S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, **"The shark-search algorithm. An application: tailored Web site mapping".** Computer Networks and ISDN Systems, 30(1-7), pp.317-326, 1998.
16. C. Reeves and J. Rowe, **"Genetic Algorithms: Principles and Perspectives,"** Kluwer, 2006.
17. H.G. Beyer**, "Toward a theory of evolution strategies: the (μ, λ)-theory,"** Evol. Comput. 2 (4) pp. 381–407, 1994.
18. X. Yao, Y. Liu and G. Lin, **"Evolutionary programming made faster,"** *IEEE Trans. Evol. Comput*. 3 (2), 82–102, 1999.
19. M. Clerc, **"Particle Swarm Optimization,"** ISTE Publishing, 2006.
20. G. Zhu, and S. Kwong, **"Gbest-guided artificial bee colony algorithm for numerical function optimization,"** *Applied mathematics and computation*, *217*(7), pp.3166-3173, 2010. https://doi.org/10.1016/j.amc.2010.08.049
21. I. Fister, I. Fister Jr., X.S. Yang, and J. Brest**, "A comprehensive review of firefly algorithms,"** Swarm Evolutionary Comput. 13, pp.34–46, 2013.
22. A. Bolaji, M. Al-Betar, M. Awadallah, A. Khader, and L. Abualigah, **"A comprehensive review: krill herd algorithm (KH) and its applications,"** Appl. Soft Comput. 49, pp. 437–446, 2016.
23. S. Mirjalili, S.M. Mirjalili, and A. Lewis, **"Grey wolf optimizer,"** Adv. Eng. Software 69, pp. 46–61, 2014.
24. M. Baghmisheh, K. Madani and A. Navarbaf, **"A discrete shuffled frog optimization algorithm,"** Artif. Intell. Rev. 36 (4), pp. 267–284, 2011.
25. J. Bansal, H. Sharma, S. Jadon, and M. Clerc, **"Spider monkey optimization algorithm for numerical optimization,"** Memet. Comput. 6 (1), pp. 31–47, 2014.
26. S. Mirjalili and A. Lewis, **"The whale optimization algorithm,"** Adv. Eng. Softw. 95, pp. 51–67, 2016.
27. W.B. Langdon and S.M. Gustafson, **"Genetic programming and evolvable machines: ten years of reviews,"** Genet. Program. Evolvable Mach. 11 (3/4) pp. 321–338, 2010.
28. M. Hauschild and M. Pelikan, **"An introduction and survey of estimation of distribution algorithms,"** Swarm Evolutionary Comput. 1 (3), pp. 111–128, 2011.
29. S. Das and P.N. Suganthan, **"Differential evolution – a survey of the state-of-the-art,"** *IEEE Trans. Evol. Comput.* 15 (1), pp. 4–31, 2011.
30. S. Das, S. Mullick, and P.N. Suganthan, **"Recent advances in differential evolution – an updated survey,"** Swarm Evolutionary Comput. 27, pp. 1–30, 2016.
31. D. Simon, **"Biogeography-based optimization,"** *IEEE Trans. Evol. Comput*. 12 (6), pp. 702–713, 2009.
32. H. Ma and D. Simon, **"Evolutionary Computation with Biogeography-based Optimization,"** Wiley-ISTE, 2016.
33. S. Ye, H. Ma, S. Xu, and M. Fei, **"An effective fireworks algorithm for warehouse scheduling problem,"** *Trans. Inst. Meas. Contr*. 39 (1), pp. 75–85, 2017.
34. M. Dorigo and L. Gambardella, **"Ant colony system: a cooperative learning approach to the traveling salesman problem,"** *IEEE Trans. Evol. Comput.* 1 (3), pp. 53–66, 1997. https://doi.org/10.1109/4235.585892
35. M. Dorigo. **"Optimization, learning, and natural algorithms."** Ph. D. Thesis, *Politecnico di Milano*, Italy, 1992.
36. P. R. Srivastava, V. Ramachandran, M. Kumar, G. Talukder, V. Tiwari, P. Sharma**, "Generation of test data using meta-heuristic approach."** *In: TENCON 2008 IEEE Region 10 Conference*. Hyderabad, 2008.
37. C. Donghua, and Y. Wenjie, **"The research of test-suite reduction technique."** *In: Consumer Electronics, Communications, and Networks (CECNet)*. XianNing, 2011.
38. Y. Singh, A. Kaur, and B. Suri: **"Test case prioritization using ant colony optimization."** ACM SIGSOFT Softw. Eng. Notes 35, 2010.
39. P. R. Srivastava: **"Structured testing using Ant colony optimization."** *In: First International Conference on Intelligent Interactive Technologies and Multimedia*, 2010.
40. B. Suri and S. Singhal: **"Analyzing test case selection & prioritization using ACO."** ACM SIGSOFT Softw. Eng. Notes 36, 2011.

41. K. Li, Z. Zhang, and W. Liu: **"Automatic test data generation based on ant colony optimization,"** vol. 6. Tianjin, 2009.
42. T. Noguchi, H. Washizaki, Y. Fukazawa, A. Sato, and K. Ota: **"History-based test case prioritization for black box testing using ant colony optimization."** Graz,2015.
43. P. R. Srivastava and K. Baby: **"Automated software testing using meta-heuristic technique based on an Ant colony optimization."** *In: Electronic System Design (ISED). Bhubaneswar*, 2010. https://doi.org/10.1109/ISED.2010.52
44. M. Chengying X. Lichuan and C. Jinfu, **"Adapting Ant Colony Optimization to generate test data for software structural testing,"** Swarm and Evolutionary Computation, Elsevier, 2015.
45. B. Suri, & S. Singhal, **"Implementing ant colony optimization for test case selection and prioritization."** International journal on computer science and engineering, 3(5), 1924-1932, 2011.
46. B. Suri, & S. Singhal, **"Literature survey of ant colony optimization in software testing."** *In Software Engineering (CONSEG), 2012 CSI Sixth International Conference on* (pp. 1-7). IEEE, 2012.
47. D. Gao, X. Guo, and L. Zhao**, "Test case prioritization for regression testing based on ant colony optimization."** *In Software Engineering and Service Science (ICSESS), 6th IEEE International Conference on* (pp. 275-279). *IEEE*, 2015.
48. J. Zhou, C. Wang, and Y. Li, **"A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system,"** Appl. Math. Model. 45, pp. 684–704, 2017.
49. Y. Singh, A. Kaur, and B. Suri, **"Test case prioritization using ant colony optimization,"** ACM SIGSOFT Softw. Eng. Notes, vol. 35, no. 4, pp. 1-7, 2010
50. D. Gao, X. Guo, and L. Zhao, **"Test case prioritization for regression testing based on ant colony optimization,"** *In 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), IEEE*, pp. 275-279. 2015.
51. M. Dorigo & M. Birattari, **"Ant colony optimization."** In Encyclopaedia of machine learning (pp. 36–39). Springer US, 2010.
52. M. Dorigo, M. Birattari, & T. Stützle**, "Ant colony optimization".** *IEEE Computational Intelligence Magazine*, 1 (4), pp. 28–39, 2006.
53. S. Bououden, M. Chadli, & H. R. Karimi, **"An ant colony optimization-based fuzzy predictive control approach for nonlinear processes."** Information Sciences, 299, pp. 143–158, 2015.
54. M. A., J. Ghasab, S. Khamis, F. Mohammad, & H. J. Fariman, **"Feature decision-making ant colony optimization system for an automated recognition of plant species."** Expert Systems with Applications, 42 (5), 2361–2370, 2015.
55. T. P. Hong, Y. F. Tung, S. L. Wang, Y. L. Wu, & M. T. Wu, **"A multi-level ant–colony mining algorithm for membership functions."** Information Sciences, 182 (1), pp. 3–14, 2012. https://doi.org/10.1016/j.ins.2010.12.019
56. T. Liao, K. Socha, M. Montes de Oca, T. Stutzle, & M. Dorigo**, "Ant colony optimization for mixed-variable optimization problems."** *IEEE Transactions on Evolutionary Computation*, 18 (4), pp. 503–518, 2014.
57. T. Liao, T. Stützle, M. A. M. deOca, & M. Dorigo, **"A unified ant colony optimization algorithm for continuous optimization."** European Journal of Operational Research, 234 (3), pp. 597–609, 2014.
58. M. Mandloi & V. Bhatia, **"Congestion control-based ant colony optimization algorithm for large MIMO detection."** Expert Systems with Applications, 42 (7), pp. 3662–3669, 2014.
59. L. B. Romdhane, Y. Chaabani, & H. Zardi, **"A robust ant colony optimization-based algorithm for community mining in large scale oriented social graphs."** Expert Systems with Applications, 40 (14), pp. 5709–5718, 2013.
60. N. Ramli, R. R. Othman, and M.S.A.R. Ali, **"Optimizing combinatorial input-output based relations testing using Ant Colony algorithm."** *In 2016 3rd International Conference on Electronic Design (ICED)* (pp. 586-590). IEEE, 2016.
61. F. Popentiu-Vladicescu, and G. Albeanu, **"Nature-inspired approaches in software faults identification and debugging."** Procedia Computer Science, 92, pp.6-12, 2016.
62. V. Panthi, and D.P. Mohapatra, **"A.C.O. based embedded system testing using UML Activity Diagram".** *In 2016 IEEE Region 10 Conference (T.E.N.C.O.N.)* (pp. 237-242). IEEE, 2016.
63. A. Ansari, A. Khan, A. Khan, and K. Mukadam, **"Optimized regression test using test case prioritization."** Procedia Computer Science, 79, pp.152-160, 2016.
64. Q. Guo, **"Task scheduling based on ant colony optimization in cloud environment."** *In A.I.P. Conference Proceedings* (Vol. 1834, No. 1, p. 040039). A.I.P. Publishing L.L.C, 2017.
65. A. Rauf, **"Data flow testing of UML state machine using ant colony algorithm (A.C.O.)."** International Journal of Computer Science and Network Security, 17(10), pp.40-44, 2017.
66. Y.N. Zhang, H. Yang, Z.K. Lin, Q. Dai, and Y.F. Li, **"A test suite reduction method based on novel quantum ant colony algorithm."** *In 2017 4th International Conference on Information Science and Control Engineering (I.C.I.S.C.E.)* (pp. 825-829). IEEE, 2017.
67. M. Khanna, **"Application of Ant Colony Algorithm in Regression Testing of Web Applications–A novel approach.",** 2017.
68. S.F. Ahmad, D.K. Singh, and P. Suman, **"Prioritization for regression testing using ant colony optimization**

based on test factors." In Intelligent Communication, Control, and Devices (pp. 1353-1360). Springer, Singapore, 2018.

69. W. Zhang, Y. Qi, X. Zhang, B. Wei, M. Zhang, and Z. Dou**, "On Test Case Prioritization Using Ant Colony Optimization Algorithm,"** *In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 2767-2773, IEEE, 2019.
70. T. Zheng, **"Automatic test case generation method of parallel multi-population self-adaptive ant colony algorithm."** *In Recent Developments in Intelligent Computing, Communication, and Devices* (pp. 469-476). Springer, Singapore, 2019.
71. C. Darwin, **"On the origins of species by means of natural selection."** London: Murray, 1859.
72. J. H. Holland, **"Adaptation in natural and artificial systems":** An introductory analysis with applications to biology, control, and artificial intelligence University of Michigan Press, 1975.
73. D. E. Goldberg, **"Genetic Algorithms: In Search, Optimization and Machine Learning."** Addison Wesley, MA,1989.
74. S. Varshney and M. Mehrotra, **"Search based software test data generation for structural testing: a perspective."** ACM SIGSOFT Softw. Eng. Notes 38, 2013. https://doi.org/10.1145/2492248.2492277
75. S. Khor and P. Grogono, **"Using a genetic algorithm and formal concept analysis to generate branch coverage test data automatically."** *In: 19th IEEE International Conference on Automated Software Engineering*, 2004.
76. P. Gulia and R. S. Chillar, **"A new approach to generate and optimize test cases for UML state diagram using genetic algorithm."** ACM SIGSOFT Softw. Eng. Notes 37, 2012.
77. H. Bhasin, **"Artificial life and cellular automata based automated test case generator."** ACM SIGSOFT Softw. Eng. Notes 39, 2014.
78. P. R. Srivastava, P. Gupta, Y. Arrawatia and S. Yadav, **"Use of genetic algorithm in generation of feasible test data.**" ACM SIGSOFT Softw. Eng. Notes 34, 2009.
79. A. Rathore, A. Bohara, P. R. Gupta, P. T. S. Lakshmi, P. R. Srivastava, **"Application of genetic algorithm and tabu search in software testing."** *In: Fourth Annual ACM Bangalore Conference*, 2011.
80. S. S. K. Prakash, S. U. M. Dhanyamraju Prasad, D. Gopi Krishna, **"Recommendation and regression test suite optimization using heuristic algorithms."** *In: 8th India Software Engineering Conference*, 2015.
81. K. K. Rao, G. S. V. P. Raju, and S. Nagaraj, **"Optimizing the software testing efficiency by using a genetic algorithm—a design methodology."** ACM SIGSOFT Softw. Eng. Notes 38, 10, 2013.
82. M. Mahajan, S. Kumar, R. Porwal, **"Applying genetic algorithm to increase the efficiency of a data flow-based test data generation approach."** ACM SIGSOFT Softw. Eng. Notes, 2012. https://doi.org/10.1145/2347696.2347707
83. G. M. Lodha, R. S. Gaikward, **"Search Based Software Testing with Genetic using fitness function,"** *IEEE Conference on Computational Intelligence on Power, Energy, and Controls with their impact on Humanity CIPECH)* 2014.
84. P. R. Srivastava and T. H. Kim, **"Application of Genetic Algorithm in Software Testing**," International Journal of Software Engineering and its application 3, 4,2009.
85. G. Fraser and A. Arcuri, **"Whole Test Suite Generation,"** *IEEE Transaction On Software Engineering*, 39, 2,2013.
86. A. S. Ghiduk, **"Automatic Generation of basis test paths using variable length genetic algorithm,"** *Information Processing Letters* 114, 304, 2014.
87. A. J. Umbarkar and P. D. Sheth, **"Crossover operators in genetic algorithms: a review."** ICTACT J. Soft Comput. 6(1), 2015.
88. M. Boopathi, R. Sujatha, C. S. Kumar, S. Narasimman, **"The mathematics of software testing using genetic algorithm."** *In: 2014 3rd International Conference on Reliability, Infocom Technologies, and Optimization (ICRITO) (Trends and Future Directions)*, pp. 1–6. IEEE, 2014.
89. D. B. Mishra, R. Mishra, K. N. Das, and A. A. Acharya, **"A systematic review of software testing using evolutionary techniques."** *In: Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, pp. 174–184. Springer, Singapore (2017)
90. A. A. Ahmed, M. Shaheen, and E. Kosba, **"Software testing suite prioritization using multi-criteria fitness function."** *In: 2012 22nd International Conference on Computer Theory andApplications (ICCTA)*, pp. 160–166. IEEE,2012.
91. H. Aytug, M. Khouja, & F. E. Vergara, **"Use of genetic algorithms to solve production and operations management problems: A review."** International Journal of Production Research, 41 (17), pp. 3955–4009, 2003.
92. R. R. Colin and E. R. Jonathan, **"Genetic Algorithms-Principles and perspectives, A guide to GA Theory."** Kluwer Academic Publishers, Springer, 2002.
93. J. J. Grefenstette, **"Genetic algorithms and their applications":** *Proceedings of the second international conference on genetic algorithms. Hillsdale*, New Jersey: Psychology Press, 2013.
94. M. Mitchell, S. Forrest, and J. H. Holland, **"The royal road for genetic algorithms: Fitness landscapes and GA performance."** *In Proceedings of the first*

European conference on artificial life (pp. 245–254). Cambridge: The MIT Press, 1992.

95. M. Srinivas and L. M. Patnaik, **"Adaptive probabilities of crossover and mutation in genetic algorithms,"** *IEEE Transactions on Systems, Man and Cybernetics*, 24 (4), 656–667, 1994. https://doi.org/10.1109/21.286385
96. P. Siarry, A. Petrowski, and M. Bessaou, **"A multipopulation genetic algorithm aimed at multimodal optimization,"** Adv. Eng. Soft. 33 (4), pp. 207–213, 2002.
97. J. Yao, N. Kharma, and P. Grogono, **"BMPGA: a bi-objective multi-population genetic algorithm for multi-modal function optimization,"** *in Proceeding of IEEE Congress on Evolutionary Computation, Edinburgh*, UK, Sep. 2005, pp. 816–823.
98. X. Zhou, R. Zhao, and F. You, **"EFSM-based test data generation with multi-population genetic algorithm,"** in *Proceeding of the 5th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, Jun. 2014, pp. 925–928.
99. A. Sharma, P. Rishon, and A. Aggarwal, **"Software testing using genetic algorithms."** Int. J. Comput. Sci. Eng. Surv.(IJCSES), 7(2), pp.21-33, 2016.
100. R. Khan, M. Amjad, and A.K. Srivastava, **"Optimization of automatic generated test cases for path testing using genetic algorithm."** *In 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)* (pp. 32-36). IEEE, 2016.
101. R.Z. Qi, Z.J. Wang, and S.Y. Li, **"A parallel genetic algorithm based on spark for pairwise test suite generation."** Journal of Computer Science and Technology, 31(2), pp.417-427, 2016.
102. E. Khanna, **"Regression testing based on genetic algorithms."** International Journal of Computer Applications, 154(8), 2016.
103. M. Soltani, A. Panichella, and A. Van Deursen, **"A guided genetic algorithm for automated crash reproduction."** *In 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)* (pp. 209-220). IEEE, 2017.
104. P.K. Kudjo, E. Ocquaye, and W. Ametepe, **"Review of genetic algorithm and application in software testing."** International Journal of Computer Applications, 160(2), pp.1-6, 2017.
105. R.B. Bahaweres, K. Zawawi, D. Khairani, and N. Hakiem**, "Analysis of statement branch and loop coverage in software testing with genetic algorithm."** *In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (pp. 1-6). IEEE, 2017.
106. D.K. Yadav, and S. Dutta, **"Regression test case prioritization technique using genetic algorithm."** In Advances in computational intelligence (pp. 133-140). Springer, Singapore, 2017.
107. S. Goyal, P. Mishra, A. Lamichhane, and P. Gandhi**, "Software test case optimization using genetic algorithm."** Int. J. Sci. Eng. Sci, 1(12), pp.69-73, 2018.
108. N. Mansour, R. Haraty, and H. Zeitunlian, **"Genetic Algorithm for Testing Web Applications,"** 2018.
109. A. Bala, and R.S. Chillar, **"A NOVEL TEST CASE PRIORITIZATION USING MENDEL OPERATOR BASED GENETIC ALGORITHM,"** Journal of Theoretical & Applied Information Technology, 96(9), 2018.
110. M. Boopathi, R. Sujatha, C. S. Kumar, S. Narasimman, and A. Rajan, **"Markov approach for quantifying the software code coverage using genetic algorithm in software testing."** International Journal of Bio-Inspired Computation, 14(1), pp.27-45, 2019.
111. D.B.Mishra, N. Panda, R. Mishra, and A. A. Acharya, **"Total fault exposing potential based test case prioritization using genetic algorithm."** International Journal of Information Technology, 11(4), pp.633-637, 2019.
112. G. Bhattacharjee, and A. S. Saluja, **"A Path-Oriented Test Data Generation Approach Hybridizing Genetic Algorithm and Artificial Immune System."** *In Computational Intelligence in Data Mining* (pp. 649-658). Springer, Singapore, 2019.
113. L. Dubey, **"Modified Genetic Algorithm for Performing the Regression Testing."** Journal of Artificial Intelligence Research & Advances, 6(2), pp.69-75, 2019.
114. G. M. Habtemariam, and S. K. Mohapatra, **"A Genetic Algorithm-Based Approach for Test Case Prioritization."** *In International Conference on Information and Communication Technology for Development for Africa* (pp. 24-37). Springer, Cham, 2019.
115. P. K. Bhatia, **"Test Case Minimization in COTS Methodology Using Genetic Algorithm: A Modified Approach."** *In Proceedings of ICETIT 2019* (pp. 219-228). Springer, Cham, 2020.
116. S. Fan, B. Ma, N. Yao, Y. Zhang, C. Xia, and D. Zhang, **"Generating Test Data for Path Coverage Based on Genetic Algorithm."** *In IOP Conference Series: Materials Science and Engineering* (Vol. 719, No. 1, p. 012070). IOP Publishing, 2020.
117. D. Karaboga, **"An idea based on honey bee swarm for numerical optimization"** Technical report-tr06, Erciyes University, Computer Engineering Department, Vol. 200, 2005.
118. H. Drias, S. Sadeg, and S. Yahi. **"Cooperative bees swarm for solving the maximum weighted satisfiability problem."** In *Computational Intelligence and Bioinspired Systems*, pages 318–325. Springer, 2005. https://doi.org/10.1007/11494669_39
119. D. Karaboga and B. Basturk, **"A powerful and efficient algorithm for numerical function optimization:**

artificial bee colony (ABC) algorithm," Journal of global optimization, 39(3):459–471, 2007.

120. P. Lucic and D. Teodorovic, **"Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence,"** *in Preprints of the TRISTAN IV triennial symposium on transportation analysis*, pages 441–445, 2001.
121. D. Teodorovic and M. Dell'Orco. **"Bee colony optimization–a cooperative learning approach to complex transportation problems,"** *In Advanced OR and AI Methods in Transportation*: pp. 51–60, 2005.
122. D. Jia, S. Qu, L. Li**, "A multiswarm artificial bee colony algorithm for dynamic optimization problems,"** *in Proceeding of International Conference on Information System and Artificial Intelligence*, Hong Kong, China, pp. 441–445, 2016.
123. G. Yu, **"A new multi-population-based artificial bee colony for numerical optimization,"** Int. J. Comput. Sci. Math. 7 (6), pp. 509–515, 2016.
124. S. Nseef, S. Abdullah, A. Turky, G. Kendall, **"An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems,"** Knowl. Base Syst. 104, 14–23, 2016.
125. S. Biswas, S. Das, S. Debchoudhury, S. Kundu, **"Co-evolving bee colonies by forager migration: a multi-swarm based artificial bee colony algorithm for global search space,"** Appl. Math. Comput. 232 (3), 216–234, 2014.
126. J. Li, Y. Zhao, J. Li, and X. J. Liu, **"Artificial bee colony optimizer with bee-to-bee communication and multipopulation coevolution for multilevel threshold image segmentation,"** Math. Probl Eng., 272947, 2015.
127. D. Karaboga and B. Basturk, **"On the performance of artificial bee colony (ABC) algorithm,"** Applied soft computing, 8 (1), 687–697, 2008.
128. D. Karaboga, B. Gorkemli, C. Ozturk, & N. Karaboga, **"A comprehensive survey: Artificial bee colony (ABC) algorithm and applications."** Artificial Intelligence Review, 42 (1), pp. 21–57, 2014.
129. D. Karaboga, and C. Ozturk, **"A novel clustering approach: Artificial bee colony (ABC) algorithm.** Applied soft computing, 11 (1), pp. 652–657, 2011.
130. W. F. Gao, and S. Y. Liu, **"A modified artificial bee colony algorithm. Computers & Operations Research",** 39 (3), pp. 687–697, 2012.
131. D. Karaboga and B. Akay, **"A survey: Algorithms simulating bee swarm intelligence,"** Artificial Intelligence Review, 31 (1-4), pp. 61–85, 2009.
132. P. Bansal, S. Sabharwal, N. Mittal, and S. Arora, **"ABC-CAG: covering array generator for pair-wise testing using artificial bee colony algorithm."** e-Informatica Software Engineering Journal, 10(1), 2016.
133. B. Liu, W.M. Li, and S. Pan, **"A novel adaptive cooperative artificial bee colony algorithm for solving numerical function optimization."** *In Theory, Methodology, Tools, and Applications for Modeling and Simulation of Complex Systems* (pp. 25-36). Springer, Singapore, 2016.
134. W. Ma, Z. Sun, J. Li, M. Song, and X. Lang, **"An improved artificial bee colony algorithm based on the strategy of global reconnaissance,"** Soft Computing, 20(12), pp.4825-4857, 2016.
135. Z.K. Aghdam, and B. Arasteh, **"An efficient method to generate test data for software structural testing using artificial bee colony optimization algorithm,"** International Journal of Software Engineering and Knowledge Engineering, 27(06), pp.951-966, 2017.
136. A.K. Alazzawi, A.A.B. Homaid, A.A. Alomoush, and A.A. Alsewari, **"Artificial bee colony algorithm for pairwise test generation."** Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(1-2), pp.103-108, 2017.
137. O. Roeva, **"Application of artificial bee colony algorithm for model parameter identification."** Innovative Computing, Optimization, and Its Applications (pp. 285-303). Springer, Cham, 2018.
138. J. K. Chhabra, **"Many-objective artificial bee colony algorithm for large-scale software module clustering problem."** Soft Computing, 22(19), pp.6341-6361, 2018. https://doi.org/10.1007/s00500-017-2687-3
139. Z. Yilmaz, and F. Basciftci, **"Binary Artificial Bee Colony Algorithm to Solve Single Objective Resource Allocation Problem."** International Journal of Future Computer and Communication, 7(1), 2018.
140. S. Sheoran, N. Mittal, and A. Gelbukh, **"Artificial bee colony algorithm in data flow testing for optimal test suite generation."** International Journal of System Assurance Engineering and Management, pp.1-10, 2019.
141. H. Peng, C. Deng, and Z. Wu, **"Best neighbor-guided artificial bee colony algorithm for continuous optimization problems."** Soft Computing, 23(18), pp.8723-8740, 2019.
142. K. Luo, **"A hybrid binary artificial bee colony algorithm for the satellite photograph scheduling problem."** Engineering Optimization, pp.1-20, 2019.
143. N. A. Sidek, S. A. Bareduan, and A. Nawawi, **"Performance Investigation of Artificial Bee Colony (ABC) Algorithm for Permutation Flowshop Scheduling Problem (PFSP)."** In Journal of Physics: Conference Series (Vol. 1150, No. 1, p. 012060). IOP Publishing, 2019.
144. R. Akay, and B. Akay, **"Artificial Bee Colony Algorithm and an Application to Software Defect Prediction."** In Nature-Inspired Methods for Metaheuristics Optimization (pp. 73-92). Springer, Cham, 2020.
145. S. Mishra, B. K. Mishra, and H. K. Tripathy, **"Significance of Biologically Inspired Optimization Techniques in Real-Time Applications."** In Robotic Systems: Concepts, Methodologies, Tools, and Applications (pp. 224-248). IGI Global, 2020.

146. A. K. Alazzawi, H. M. Rais, S. Basri, and Y. A. Alsariera, **"Pairwise Test Suite Generation Based on Hybrid Artificial Bee Colony Algorithm."** *In Advances in Electronics Engineering* (pp. 137-145). Springer, Singapore, 2020.
147. Y. Shi and R. C. Eberhart, **"Parameter selection in particle swarm optimization."** In *Evolutionary programming VII* (pp. 591–600). Berlin: Springer, 1998.
148. Y. Shi and R. C. Eberhart, **"Empirical study of particle swarm optimization,"** in *Proceedings of the 1999 IEEE congress on evolutionary computation (pp. 1945–1950)*, 1999.
149. J. Kennedy, **"The particle swarm: Social adaptation of knowledge."** *In Proceedings of the IEEE international conference on evolutionary computation*, pp. 303–308, 1997.
150. S. Jiang, J. Shi, Y. Zhang, and H. Han, **"Automatic test data generation based on reduced adaptive particle swarm optimization algorithm."** *Neurocomputing, 158*, pp.109-116, 2015.
151. L. S. de Souza, P. B de Miranda, R. B. Prudencio, and F. D. A. Barros, **"A multi-objective particle swarm optimization for test case selection based on functional requirements coverage and execution effort."** In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence* (pp. 245-252). IEEE, 2011.
152. Z. H. Zhan, J. Zhang, Y. Li & Y. H. Shi, **"Orthogonal learning particle swarm optimization."** *IEEE Transactions on Evolutionary Computation, 15* (6), pp. 832–847, 2011.
153. M. Couceiro, and P. Ghamisi, **"Particle swarm optimization,"** In *Fractional Order Darwinian Particle Swarm Optimization* (pp. 1–10). Springer International Publishing, 2016.
154. A. H. Gandomi, G. J. Yun, X. S. Yang, & S. Talatahari, **"Chaos-enhanced accelerated particle swarm optimization,"** *Communications in Nonlinear Science and Numerical Simulation, 18* (2), 327–340, 2013.
155. W. C. Hong, **"Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model."** Energy Conversion and Management, 50 (1), pp. 105–117,2009.
156. J. Kennedy, **"Particle swarm optimization."** In Encyclopaedia of machine learning (pp. 760–766). New York: Springer US, 2011.
157. B. Liu, L. Wang, Y. H. Jin, F. Tang, & D. X. Huang, **"Improved particle swarm optimization combined with chaos."** *Chaos, Solitons & Fractals*, 25 (5), 1261–1271, 2005.
158. Y. Shi and R. C. Eberhart, **"Parameter selection in particle swarm optimization."** *In International conference on evolutionary programming (pp. 591-600). Springer, Berlin, Heidelberg, 1998.*
159. B. Niu, Y. Zhu, X. He, **"Multi-population cooperative particle swarm optimization"** in *Proceeding of European Conference on Artificial Life, Canterbury*, UK, pp. 874–883, 2005.
160. J. J. Liang and P. N. Suganthan, **"Dynamic multi-swarm particle swarm optimizer with local search,"** *in: Proceeding of IEEE Congress on Evolutionary Computation*, Edinburgh, UK, pp. 522–528, 2005.
161. S. Zhao, P. N. Suganthan, and S. Das, **"Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search,"** *in Proceeding of 2010 IEEE World Congress on Computational Intelligence*, Barcelona, Spain, pp. 18–23, 2010.
162. J. J. Liang and P. N. Suganthan, **"Dynamic multi-swarm particle swarm optimizer,"** in *Proceeding of IEEE Swarm Intelligence Symposium, Pasadena*, CA, pp. 124–129, 2005.
163. S. Yang and C. Li, **"A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments,"** *IEEE Trans. Evol. Comput.* 14 (6), pp. 959–974, 2010.
164. X. Xu, Y. Tang, J. Li, C. Hua and X. Guan, **"Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy,"** Appl. Soft Comput. 29, pp. 169–183, 2015.
165. H. Chen, Y. Zhu and K. Hu, **"Discrete and continuous optimization based on multiswarm coevolution,"** Nat. Comput. 9 (3), pp. 659–682, 2010.
166. X. Zheng and H. Liu, **"A different topology multi-swarm PSO in dynamic environment,"** in *Proceeding of IEEE International Symposium on IT in Medicine & Education*, Jinan, China, pp. 790–795, 2009.
167. R. Bolufe and S. Chen, **"An analysis of sub-swarms in multi-swarm systems,"** *in Proceeding of the 24th Australasian Joint Conference on Artificial Intelligence*, Perth, Australia, pp. 271–278, 2011.
168. A. El Dor, M. Clerc, P. Siarry, **"A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization,"** Comput. Optim. Appl. 53 (1), pp. 271–295, 2012.
169. J. Zhang and X. Ding, **"A multi-swarm self-adaptive and cooperative particle swarm optimization,"** Eng. Appl. Artif. Intell. 24 (6), pp. 958–967, 2011.
170. Y. Zhang, D. Gong, and Z. Ding, **"Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer,"** Expert Syst. Appl. 38 (11), pp. 13933–13941, 2011.
171. J. J. Liang, B. Qu, P. N. Suganthan, and B. Niu, **"Dynamic multi-swarm particle swarm optimization for multi-objective optimization problems,"** *in: Proceeding of IEEE Congress on Evolutionary Computation (CEC)*, Brisbane, Australia, pp. 1–8, 2012.
172. S. Fan and J. Chang, **"Dynamic multi-swarm particle swarm optimizer using parallel PC cluster systems for global optimization of large-scale multimodal functions,"** Eng. Optim. 42 (5), pp. 431–451, 2010.

173. Y. Marinakis, M. Marinaki, and A. Migdalas, **"Adaptive tunning of all parameters in a multi-swarm particle swarm optimization algorithm: An application to the probabilistic traveling Salesman problem,"** *in Proceeding of Conference on Optimization Control and Applications in the Information Age*, Macedonia, Greece, pp. 187–207, 2014.
174. J. J. Liang and P. N. Suganthan, **"Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism."** *In: Proceeding of IEEE Congress on Evolutionary Computation*, Vancouver, Canada, pp. 9–16, 2006.
175. Y. Wang and Z. Cai, **"A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems,"** Front. Comput. Sci. China 3 (1), pp.38–52, 2009.
176. J. Xiao and Z. Cheng, **"A multi-swarm particle swarm optimization to solve DNA encoding in DNA computation,"** J. Comput. Theor. Nanosci. 210 (5), pp. 1129–1136, 2013.
177. B. Niu, Y. Zhu and X. He, **"A multi-population cooperative particle swarm optimizer for neural network training,"** *in Proceeding of International Symposium on Neural Networks*, Chengdu, China, pp. 570–576, 2006. https://doi.org/10.1007/11759966_85
178. N. Jatana, B. Suri, S. Misra, P. Kumar, and A.R. Choudhury, **"Particle swarm based evolution and generation of test data using mutation testing."** *In International Conference on Computational Science and its Applications* (pp. 585-594). Springer, Cham, 2016.
179. Y. Sheng, C. Wei, and S. Jiang, **"Constraint test cases generation based on particle swarm optimization,"** International Journal of Reliability, Quality and Safety Engineering, 24(05), p.1750021, 2017.
180. M. Hajihassani, D.J. Armaghani, and R. Kalatehjari, **"Applications of particle swarm optimization in geotechnical engineering: a comprehensive review,"** Geotechnical and Geological Engineering, 36(2), pp.705-722, 2018.
181. J. Sun, J. Chen, and G. Wang, **"Multi-Objective Test Case Prioritization based on Epistatic Particle Swarm Optimization."** International Journal of Performability Engineering, 14(10), 2018.
182. Z. Wang, and Q. Liu, **"A software test case automatic generation technology based on the modified particle swarm optimization algorithm."** *In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)* (pp. 156-159). IEEE, 2018.
183. S. Jianqi, H. Yanhong, L. Ang, and C. Fangda, **"An optimal solution for software testing case generation based on particle swarm optimization."** Open Physics, 16(1), pp.355-363, 2018.
184. M. Khatibsyarbini, M. A. Isa, and D.N.A. Jawawi, **"Particle Swarm Optimization for Test Case Prioritization Using String Distance."** Advanced Science Letters, 24(10), pp.7221-7226, 2018.
185. H. M. Allawi, W. Al Manaseer, and M. Al Shraideh, **"A greedy particle swarm optimization (GPSO) algorithm for testing real-world smart card applications."** International Journal on Software Tools for Technology Transfer, pp.1-12, 2018.
186. G. Nayak, and M. Ray, **"Modified condition decision coverage criteria for test suite prioritization using particle swarm optimization."** International Journal of Intelligent Computing and Cybernetics, 2019.
187. R. Malhotra, and M. Khanna, **"Dynamic selection of fitness function for software change prediction using particle swarm optimization."** Information and Software Technology, 112, pp.51-67, 2019.
188. R. Islam, M. N. Akhtar, B. R. Ahmad, U. K. Das, M. Rahman, and Z.I.A. Khalib, **"An approach to building energy clusters using particle swarm optimization algorithm for allocating the tasks in computational grid."** Indonesian Journal of Electrical Engineering and Computer Science, 14(2), pp.826-833, 2019.
189. S. Grossberg, **"Nonlinear neural networks: Principles, mechanisms, and architectures."** *Neural networks, 1* (1), pp. 17–61, 1988.
190. A. K. Kar, **"Using artificial neural networks and analytic hierarchy process for the supplier selection problem."** In *2013 IEEE international conference on signal processing, computing, and control* (pp. 1–6), 2013.
191. N. Sadegh, **"A perceptron network for functional identification and control of nonlinear systems,"** *IEEE Transactions on Neural Networks, 4* (6), pp. 982–988, 1993.
192. A. K. Kar, **"A hybrid group decision support system for supplier selection using analytic hierarchy process, fuzzy set theory and neural network,"** *Journal of Computational Science, 6,* pp. 23–33,2015.
193. K. Hornik, **"Approximation capabilities of multilayer feed-forward networks."** *Neural networks, 4* (2), pp. 251–257, 1991.
194. D. F. Specht, **"Probabilistic neural networks."** *Neural networks, 3* (1), pp. 109–118, 1990.
195. J. Schmidhuber, **"Deep learning in neural networks: An overview".** *Neural Net- works, 61,* 85–117, 2015.
196. E. Oja, **"Principal components, minor components, and linear neural networks,"** *Neural Networks, 5* (6), pp. 927–935,1992.
197. Z. Chen, Y. Zhao, and Y. Zheng, **"Neural network electrical machine faults diagnosis based on multi-population GA,"** *in Proceeding of International Joint Conference on Neural Networks*, Hong Kong, China, Jun. 2008, pp. 3795–3801.
198. L. Fausett, **"Fundamentals of neural networks: Architectures, algorithms, and applications,"** Upper Saddle River, New Jersey: Prentice-Hall, Inc, 1994.

199. M.W. Craven, and J.W. Shavlik, **"Using neural networks for data mining."** *Future generation computer systems*, *13*(2-3), pp.211-229, 1997.
200. J. Lampinen and A. Vehtari, **"Bayesian approach for neural networks—review and case studies,"** *Neural networks*, *14*(3), pp.257-274, 2001.
201. S. Mensah, J. Keung, K.E. Bennin, and M. F. Bosu, **"Multi-objective optimization for software testing effort estimation."** SEKE, 2016.
202. W. Makondo, R. Nallanthighal, I. Mapanga, and P. Kadebu, **"Exploratory test oracle using multi-layer perceptron neural network."** *In 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)* (pp. 1166-1171). IEEE, 2016.
203. Y. Pang, X. Xue, and H. Wang, **"Predicting vulnerable software components through deep neural network."** *In Proceedings of the 2017 International Conference on Deep Learning Technologies* (pp. 6-10), 2017.
204. V. Sathyavathy, **"Evaluation of software testing techniques using artificial neural network."** Int. J. Electr. Comput. Sci, 6(3), pp.20617-20620, 2017.
205. Y. Sun, X. Huang, and D. Kroening, **"Testing deep neural networks,"** arXiv preprint arXiv:1803.04792, 2018.
206. L. Ma, F. Zhang, M. Xue, B. Li, Y. Liu, J. Zhao, and Y. Wang, **"Combinatorial testing for deep learning systems,"** arXiv preprint arXiv:1806.07723, 2018.
207. T. Byun, V. Sharma, A. Vijayakumar, S. Rayadurgam, and D. Cofer, **"Input prioritization for testing neural networks,"** *In 2019 IEEE International Conference On Artificial Intelligence Testing (AITest)* (pp. 63-70). IEEE, 2019. https://doi.org/10.1109/AITest.2019.000-6
208. Y.L. Karpov, L.E. Karpov, and Y.G.Smetanin, **"Adaptation of general concepts of software testing to neural networks."** Programming and Computer Software, 44(5), pp.324-334, 2018.
209. L. Joffe, and D. Clark, **"Constructing Search Spaces for Search-Based Software Testing Using Neural Networks."** *In International Symposium on Search Based Software Engineering* (pp. 27-41). Springer, Cham, 2019.
210. S. Mannarswamy, S. Roy, and S. Chidambaram, **"Tutorial on Software Testing & Quality Assurance for Machine Learning Applications from research bench to real world."** *In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (pp. 373-374), 2020.
211. D. Ghosh, and J. Singh, **"A Novel Approach of Software Fault Prediction Using Deep Learning Technique."** In Automated Software Engineering: A Deep Learning-Based Approach (pp. 73-91). Springer, Cham, 2020.
212. X.S. Yang. **"A new metaheuristic bat-inspired algorithm**." *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74, 2010.
213. X. S. Yang, **"Bat algorithm for multi-objective optimization,"** International Journal of Bio-Inspired Computation, 3 (5), pp. 267–274, 2011.
214. G. Wang, B. Chang, and Z. Zhang, **"A multi-swarm bat algorithm for global optimization,"** *in Proceeding of IEEE Congress on Evolutionary Computation, Sendai, Japan*, May 2015, pp. 480–485.
215. K. Heraguemi, N. Kamel, H. Drias, **"Multi-population cooperative bat algorithm for association rule mining,"** in *Proceeding of Computational Collective Intelligence*, Madrid, Spain, pp. 265–274,2015.
216. S. Mirjalili, S. M, Mirjalili, and X. S. Yang, **"Binary bat algorithm."** Neural Computing and Applications, 25 (3-4), pp. 663–681, 2014.
217. N. S. Jaddi, S. Abdullah, A. R. Hamdan, **"Multi-population cooperative bat algorithm-based optimization of artificial neural network model."** Information Sciences, 294, 628–644, 2015.
218. A. H. Gandomi, X. S. Yang, A. H. Alavi, and S. Talatahari, **"Bat algorithm for constrained optimization tasks."** Neural Computing and Applications, 22 (6), pp. 1239–1255, 2013.
219. K. Heraguemi, N. Kamel, H. Drias, **"Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies,"** Appl. Intell. 45 (4), pp.1–13, 2016.
220. A. H. Gandomi and X. S. Yang, **"Chaotic bat algorithm."** Journal of Computational Science, 5 (2), 224–232, 2014.
221. X. B. Meng, X. Z. Gao, Y. Liu and H. Zhang, **"A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization."** Expert Systems with Applications, 42 (17), 6350–6364, 2015.
222. D. Rodrigues, L. A., Pereira, R. Y. Nakamura, K. A. Costa, X. S. Yang, A. N. Souza and J. P. Papa**, "A wrapper approach for feature selection based on bat algorithm and optimum-path forest."** Expert Systems with Applications", 41 (5), pp. 2250–2258, 2014.
223. X. S. Yang and X. He, **"Bat algorithm: Literature review and applications,"** International Journal of Bio-Inspired Computation, 5 (3), pp. 141–149,2013.
224. X. S. Yang and A. H. Gandomi, **"Bat algorithm: A novel approach for global engineering optimization,"** Engineering Computations, 29 (5), pp. 464–483, 2012.
225. Y. A. Alsariera, and K.Z. Zamli, **"A real-world test suite generation using the bat-inspired t-way strategy."** *In The 10th Asia Software Testing Conference (SOFTEC2017)* (Vol. 10, pp. 71-79), 2017.
226. T. Ashish, S. Kapil, and B. Manju, **"Parallel bat algorithm-based clustering using MapReduce."** *In Networking Communication and Data Knowledge Engineering* (pp. 73-82). Springer, Singapore, 2018.
227. A. Sharma, and N. Sehgal, **"Enhanced test case prioritization technique using bat algorithm,"** 2018.
228. M. M. Öztürk, **"A bat-inspired algorithm for prioritizing test cases."** Vietnam Journal of Computer Science, 5(1), pp.45-57, 2018.

229. Y.A. Alsariera, H.A.S Ahmed, H. S. Alamri, M.A. Majid, and K.Z. Zamli, **"A bat-inspired testing strategy for generating constraints pairwise test suite."** Advanced Science Letters, 24(10), pp.7245-7250, 2018.
230. X. Huang, P. Li, and Y. Pu, **"Amplitude angle modulated bat algorithm with application to zero-one knapsack problem."** IEEE Access, 7, pp.27957-27969, 2019.
231. X. S. Yang and S. Deb, **"Cuckoo search via Lévy flights,"** In *Nature & Biologically Inspired Computing, 20 09. NaBIC 20 09. World Congress on* (pp. 210–214). IEEE, 2009.
232. G. S. Walia and R. Kapoor, **"Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search."** *Expert Systems with Applications, 41* (14), pp. 6315–6326, 2014.
233. X. S. Yang and S. Deb, **"Engineering optimisation by cuckoo search."** *International Journal of Mathematical Modelling and Numerical Optimisation, 1* (4), pp. 330–343, 2010.
234. X. S. Yang, and S. Deb, **"Multiobjective cuckoo search for design optimization."** *Computers & Operations Research, 40* (6), pp. 1616–1624, 2013. https://doi.org/10.1016/j.cor.2011.09.026
235. S. Araghi, A. Khosravi, and D. Creighton, **"Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network."** *Expert Systems with Applications, 42* (9), pp. 4422–4 431, 2015.
236. X. S. Yang, and S. Deb, **"Cuckoo search: Recent advances and applications."** *Neural Computing and Applications, 24* (1), pp. 169–174, 2014.
237. A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Singh, **"Cuckoo search algorithm and wind driven optimization-based study of satellite image segmentation for multilevel thresholding using Kapur's entropy."** *Expert Systems with Applications, 41* (7), pp. 3538–3560,2014.
238. A. H. Gandomi, X. S. Yang, and A. H. Alavi, **"Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems,"** *Engineering with computers, 29* (1), 17–35, 2013.
239. A. Gotmare, R. Patidar, and N. V. George, **"Nonlinear system identification using a cuckoo search optimized adaptive Hammerstein model,"** *Expert Systems with Applications, 42* (5), pp. 2538–2546, 2015.
240. M. Kumar and T. K. Rawat, **"Optimal design of FIR fractional order differentiator using cuckoo search algorithm."** *Expert Systems with Applications, 42* (7), pp. 3433–3449, 2015.
241. X. S. Yang and S. Deb, **"Cuckoo search: Recent advances and applications,"** *Neural Computing and Applications, 24* (1), pp. 169–174, 2014.
242. P. R. Srivastava, D. V. P. kumarreddy, M. Srikanthreddy, Ch. V.B. RamarajuIch. Manikanthnath, **"Test case prioritization using cuckoo search,"** *Advanced automated software testing*, 2012.
243. M. Khari, and P. Kumar, **"A novel approach for software test data generation using cuckoo algorithm."** *In Proceedings of the second international conference on information and communication technology for competitive strategies* (pp. 1-6), 2016.
244. S. Dhabal, S.Tagore, and D. Mukherjee, **"An improved Cuckoo Search Algorithm for numerical optimization."** *In 2016 International Conference on Computer, Electrical & Communication Engineering (ICCECE)* (pp. 1-7). IEEE, 2016.
245. M. Khari, and P. Kumar, **"An effective meta-heuristic cuckoo search algorithm for test suite optimization."** Informatica, 41(3), 2017.
246. H. Haixian, and F. Jing, **"Research on Multi-objective Test Case Generation Based on Cuckoo Search."** *In 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 1619-1623). IEEE, 2018.
247. M. Shehab, A.T. Khader, and M. Laouchedi, **"A hybrid method based on cuckoo search algorithm for global optimization problems."** Journal of ICT, 17(3), pp.469-491, 2018.
248. S. Sharma, S.A.M. Rizvi, and V. Sharma, **"A Framework for Optimization of Software Test Cases Generation using Cuckoo Search Algorithm."** *In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 282-286). IEEE, 2019.
249. P. Dhareula, and A. Ganpati, **"Cuckoo Search Algorithm for Test Case Prioritization in Regression Testing,"** International Journal of Recent Technology and Engineering (IJRTE), Volume-8 Issue-3, September 2019.
250. S. Benkhaira, and A. Layeb**, "Face Recognition Using RLDA Method Based on Mutated Cuckoo Search Algorithm to Extract Optimal Features."** International Journal of Applied Metaheuristic Computing (IJAMC), 11(2), pp.118-133, 2020.
251. X. S. Yang, **"Firefly algorithms for multimodal optimization,"** *In Stochastic algorithms: Foundations and applications* (pp. 169–178). Berlin Heidelberg: Springer, 2009.
252. X. S. Yang. **"Firefly algorithm, stochastic test functions, and design optimization,"** *International Journal of Bio-Inspired Computation*, 2(2): pp.78–84, 2010.
253. P. R. Srivastava, B. Mallikarjun, and X. S. Yang, **"Optimal Test Sequence Generation using firefly algorithm,"** Swarm and Evolutionary Computation," Elsevier, 8, 44, 2013.
254. X. S. Yang and S. Deb. **"Eagle strategy using levy walk and firefly algorithms for stochastic optimization."** *In Nature Inspired Cooperative Strategies for Optimization (NICSO2010),* pp. 101–111, Springer, 2010.
255. M. Xu and G. Liu, **"A multipopulation firefly algorithm for correlated data routing in underwater**

wireless sensor networks," Int. J. Distrib. Sens. Netw., pp. 245–253, 2013.

256. A. H. Gandomi, X. S. Yang, S. Talatahari, A. H. Alavi, "**Firefly algorithm with chaos."** Communications in Nonlinear Science and Numerical Simulation", 18 (1), 89–98, 2013.
257. F. Ozsoydan and A. Baykasoglu, **"A multi-population firefly algorithm for dynamic optimization problems,"** *in Proceeding of International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, Douai, France, pp.1–7, 2015.
258. A. Kavousi-Fard, H. Samet, and F. Marzbani, **"A new hybrid modified firefly algorithm and support vector regression model for accurate short-term load forecasting."** Expert systems with applications, 41 (13), pp. 6047–6056, 2014.
259. A. H. Gandomi, X. S. Yang, and A. H. Alavi, **"Mixed variable structural optimization using firefly algorithm,"** Computers & Structures, 89 (23), pp. 2325–2336, 2011.
260. N. C. Long, P. Meesad and H. Unger, **"A highly accurate firefly based algorithm for heart disease prediction,"** Expert Systems with Applications, 42 (21), pp. 8221–8231, 2015.
261. S. Lukasik and S. Zak, **"Firefly algorithm for continuous constrained optimization tasks,"** *In Computational collective intelligence. Semantic web, social networks, and multiagent systems* (pp. 97–106). Berlin: Springer, 2009.
262. A. Rahmani and S. A. MirHassani, **"A hybrid Firefly-Genetic Algorithm for the capacitated facility location problem,"** Information Sciences, 283, pp. 70–78, 2014.
263. X. S. Yang and X. He, **"Firefly algorithm: Recent advances and applications,"** International Journal of Swarm Intelligence, 1 (1), pp. 36–50, 2013.
264. A. Mishra, C. Agarwal, A. Sharma, and P. Bedi, **"Optimized gray-scale image watermarking using DWT–SVD and Firefly Algorithm,"** Expert Systems with Applications, 41 (17), pp. 7858–7867, 2014.
265. O. P. Verma, D. Aggarwal, and T. Patodi**, "Opposition and dimensional based modified firefly algorithm,"** Expert Systems with Applications, 44, pp. 168–176, 2016.
266. X. S. Yang, S. S. S. Hosseini, and A. H. Gandomi, **"Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect,"** Applied Soft Computing, 12 (3), pp. 1180–1186, 2012.
267. X. S. Yang, **"Flower pollination algorithm for global optimization."** In *Unconventional computation and natural computation* (pp. 240–249). Berlin Heidelberg: Springer, 2012.
268. M. Khatibsyarbini, M.A. Isa, D. N. Jawawi, H.N.A. Hamed, and M.D.M. Suffian, **"Test Case Prioritization Using Firefly Algorithm for Software Testing."** *IEEE Access*, 7, pp.132360-132373, 2019.
269. I. Arora, and A. Saha, **"Software fault prediction using firefly algorithm."** International Journal of Intelligent Engineering Informatics, 6(3-4), pp.356-377, 2018.
270. N.L. Hashim, and S.Y. Dawood, **"Test case minimization applying firefly algorithm."** Int J Adv Sci Eng Inf Technol, 8(4), pp.1777-1783, 2018.
271. N. Rathee, R.S. Chillar, S. Vij, and S. Kukreja, **"Automatic Optimization of Test Path Using Firefly Algorithm."** In Harmony Search and Nature Inspired Optimization Algorithms (pp. 717-729). Springer, Singapore, 2019.
272. A. Pandey, and S. Banerjee, **"Test Suite Optimization Using Firefly and Genetic Algorithm."** International Journal of Software Science and Computational Intelligence (IJSSCI), 11(1), pp.31-46, 2019.
273. X. S. Yang, M. Karamanoglu, and X. He, **"Multiobjective flower algorithm for optimization."** *Procedia Computer Science*, 18, pp. 861–868, 2013.
274. G. Bekdas, S. M. Nigdeli, and X. S. Yang, **"Sizing optimization of truss structures using flower pollination algorithm."** *Applied Soft Computing, 37*, pp. 322–331, 2015.
275. X. S. Yang, M. Karamanoglu, and X. He, **"Flower pollination algorithm: A novel approach for multi-objective optimization."** *Engineering Optimization, 46* (9), pp.1222–1237, 2014.
276. S. M. Nigdeli, G. Bekda, and X. S. Yang, **"Application of the flower pollination algorithm in structural engineering."** In *Metaheuristics and optimization in civil engineering* (pp. 25–42). Berlin: Springer, 2016.
277. M.N. Kabir, J. Ali, A.A. Alsewari, and K.Z. Zamli, **"An adaptive flower pollination algorithm for software test suite minimization."** *In 2017 3rd International Conference on Electrical Information and Communication Technology (EICT)* (pp. 1-5). *IEEE*, 2017. https://doi.org/10.1109/EICT.2017.8275215
278. A.B. Nasser, K.Z. Zamli, and B.S. Ahmed, **"Dynamic Solution Probability Acceptance Within the Flower Pollination Algorithm for Combinatorial."** *Intelligent and Interactive Computing: Proceedings of IIC 2018*, 67, p.3, 2019.
279. A.B. Nasser, K.Z. Zamli, A.A. Alsewari, and B.S. Ahmed, **"Hybrid flower pollination algorithm strategies for t-way test suite generation."** PloS one, 13(5), 2018.
280. M. Abdel-Basset, and L.A. Shawky, **"Flower pollination algorithm: a comprehensive review."** Artificial Intelligence Review, 52(4), pp.2533-2557, 2019.
281. A.B. Nasser, and K.Z. Zamli, **"Self-adaptive population size strategy based on flower pollination algorithm for T-way test suite generation."** *In International Conference of Reliable Information and Communication Technology* (pp. 240-248). Springer, Cham, 2018.

282. P. Dhareula, and A. Ganpati, **"Software Test Case Prioritization Using Genetically Modified Flower Pollination Algorithm (Gm-Fpa)."** International journal of scientific and technology research, vol. 8, issue 12, pp 298-306, 2019.
283. P. Dhareula, and A. Ganpati, **"Flower Pollination Algorithm for Test Case Prioritization in Regression Testing."** *In the 4th International Conference on Information and Communication Technology for Sustainable Development (ICT4SD 2019)* on July 5th -6th, Springer, Goa, India, 2019.
284. A. Mucherino, and O. Seref, **"Monkey search: a novel metaheuristic search for global optimization."** In *AIP conference proceedings* (Vol. 953, No. 1, pp. 162-173). American Institute of Physics, 2007.
285. W. T. Pan, **"A new fruit fly optimization algorithm: taking the financial distress model as an example."** *Knowledge-Based Systems*, *26*, pp.69-74, 2012.
286. M. D. Li, H. Zhao, X. W. Weng, and T. Han, **"A novel nature-inspired algorithm for optimization: virus colony search,"** Adv. Eng. Software 92 pp. 65 – 88, 2016.
287. W. Yong, W. Tao, Z. Cheng-Zhi, H. Hua-Juan, **"A new stochastic optimization approach dolphin swarm optimization algorithm,"** Int. J. Comput. Intell. Appl. 15 (02), 2016.
288. M. Jain, V. Singh, and A. Rani, A., **"A novel nature-inspired algorithm for optimization: Squirrel search algorithm,"** *Swarm and evolutionary computation*, *44*, pp.148-175 2019.
289. X. Qi, Y. Zhu, and H. Zhang, **"A new meta-heuristic butterfly-inspired algorithm,"** Journal of computational science, 23, pp.226-239, 2017.
290. A. H. Gandomi, and A. H. Alavi, **"Krill herd: a new bio-inspired optimization algorithm,"** Commun. Nonlinear Sci. Numer. Simulat. 17 (12) pp. 4831 – 4845, 2012.
291. K. M. Passino, **"Biomimicry of bacterial foraging for distributed optimization and control."** *IEEE control systems magazine*, *22*(3), pp.52-67, 2002.
292. R. Tang, S. Fong, X. S. Yang, and S. Deb, **"Wolf search algorithm with ephemeral memory,"** In *Seventh International Conference on Digital Information Management (ICDIM 2012)* (pp. 165-172). *IEEE*, 2012.
293. X. Zhang, B. Sun, T. Mei, and R. Wang, **"Post-disaster restoration based on fuzzy preference relation and bean optimization algorithm,"** In *IEEE youth conference on information computing and telecommunications* (pp. 271–274), 2010.
294. X. Zhang, S. Huang, Y. Hu, Y. Zhang, S. Mahadevan, and Y. Deng, **"Solving 0–1 knapsack problems based on amoeboid organism algorithm",** *Applied Mathematics and Computation*, *219*, 9959–9970, 2013.
295. M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, **"The sharksearch algorithm. An application: Tailored web site mapping,"** *Computer Networks and ISDN Systems, 30*, pp. 317–326, 1998.
296. M. Yazdani and F. Jolai, **"Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm."** *Journal of Computational Design and Engineering*, *3* (1), 24–36, 2015.
297. M.C. Su, S.Y. Su, and Y. X. Zhao, **"A swarm-inspired projection algorithm."** *Pattern Recognition*, *42*(11), pp. 2764-2786, 2009. https://doi.org/10.1016/j.patcog.2009.03.020
298. O. Dahiya and K. Solanki, S. Dalal, A. Dhankhar, **"Regression Testing: Analysis of its Techniques for Test Effectiveness,"** International Journal of advanced trends in computer science and engineering, Vol. 9, No. 1, 2020. *[Accepted] [In Press]*.