A Systematic Literature Review on Application of Genetic and ACO Algorithm in Software Testing

Tanya Singh, Kriti Dubey, Tamanna, Shrishti Chauhan

Department of Information Technology
Indira Gandhi Delhi Technical University For Women, Delhi, India

Abstract— Software testing is a crucial phase in the software development life cycle. It ensures security, reliability, cost-effectiveness, product quality, performance, and customer satisfaction. Automation and optimization of the software testing tasks are needed to reduce the cost and time of testing. Many popular meta-heuristic optimizations (search) based techniques like the ACO algorithm and Genetic algorithm becoming popular for the optimization and automation of testing. In this paper, we conducted a systematic literature review(SLR) on the applications of genetic and aco algorithms in software testing like test case generation, optimization, etc. The purpose of this paper is to examine the trend in the use of genetic and aco algorithms in software testing. We selected 60 related research papers using search strings and relevant titles. After reading in full papers, we selected 37 papers for review. According to our study, most of the research work is done on test optimization, test case generation and automation using Genetic and ACO algorithms.

Keywords— Software Testing (ST), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Systematic Literature Review (SLR), Testing Optimization, Test case Generation

I. INTRODUCTION

Software testing is a process of verifying and validating the bugs or errors or faults in the software. It is an integral part of software development. It is done with the objective and intent to find maximum faults in the software by designing test cases based on the functionality of the software. It is done to ensure that the software is bug-free in order to produce a quality product. Software testing ensures security, reliability, cost-effectiveness, product quality, performance, and customer satisfaction. Manual Testing and Automation Testing are different types of software testing. Black-Box testing and White-Box testing are different types of software testing techniques. The four levels of software testing are Unit Testing, Integration Testing, System Testing, Acceptance Testing.

Due to the increasing demand and usage of the software, it is very important to reduce the time consumption of its testing. Meta-heuristic optimization (search) based techniques have been used to automate and optimize software testing tasks. Some of the widely used search-based algorithms are Ant Colony Optimization (ACO), Genetic Algorithm(GA), Particle Swarm Optimization (PSO), Cuckoo Search (CS), etc. In this paper, we are mainly focusing on the Genetic algorithm (GA) and Ant Colony Optimization (ACO) algorithm.

a. Genetic Algorithm

The GA may be a meta-heuristic inspired by evolution that belongs to the large class of evolutionary algorithms in computational mathematics and informatics. These algorithms are often used to generate a higher-quality solution to optimize and search problems by specializing in bio-inspired operators like convergence, selection, and mutations. In 1988, author, John Holland, created GA supported Darwin's evolutionary theory. In 1992 he expanded GA. This algorithm is an evolutionary algorithm. Evolutionary algorithms are used to solve problems for which there is no well-defined efficient solution method used to solve the optimization problem (such as scheduling and shortest path) Moreover, as modelling and simulation with random functions. A genetic algorithm is an optimized solution designed to enhance options for the candidate population (also called people, animals, or genotypes). [1] Each candidate solution contains a set of traits (genes or phenotype) that can be evolved and changed. Solutions are usually represented in binary as a string of 0's and 1's. but other codecs are, permitted. Evolution is often initiated by a random group of people and It is an iterative process in which the population provides services as a production technique for each reprint. Everyone within the population's fitness is measured for every generation. In contrast, fitness is often defined because the target feature values are resolved.

If well-matched individuals are randomly selected from an existing population and are further genetically altered to initiate every new generation cycle (which may be recombined and randomly mutated). The next generation in the process uses the newer generation candidate strategy. The algorithm is often terminated when the utmost number of generations or satisfaction is reached. As a result, each successive generation is best suited to the population's environment. The populations are kept within the scope of the search methods. Everyone within the computational complexity may be a solution to a selected problem. A finite-length vector of

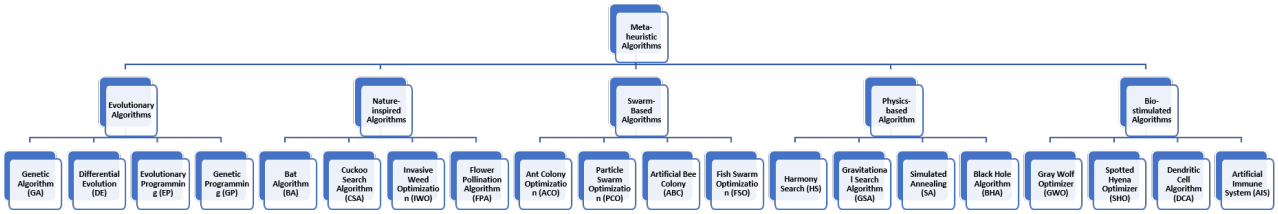

Fig. 1: Meta-heuristic Optimization Algorithms

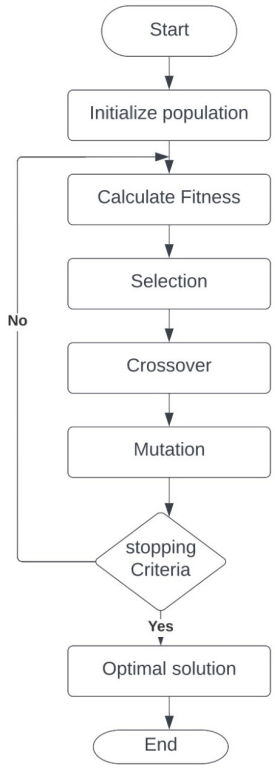

Fig. 2: Genetic Algorithm Flowchart

components represents everyone within the population. The component is comparable to genes, and plenty of genes contribute to chromosome formation. a personality's ability to perform is represented by their fitness score. An individual's optimal fitness score might be calculated. GA can keep track of an n-person population's fitness levels. Everyone includes a high fitness score, which increases the likelihood of reproduction. By grouping the chromosome of that generation, someone with a high fitness score is chosen for mating and producing healthy offspring. Because the population size is constant, space is created when a brand-new baby is born. As a result, several people die and are replaced by newcomers, who eventually produce a brand new generation of the breeding potential of the whole older human population is low. Whether or not the less suitable life. On average, younger generations provide more "good genes" than older generations. As a result, each succeeding generation features a better solution than the one before it. When the offspring of two populations haven't any significant differences from the offspring of the opposite population, the populations are converging. This algorithm is said to translate into a set of solutions for individual problems the subsequent are a number of the benefits of GAs.

1. GA is strong and durable.
2. Provides calming solutions to a huge population
3. GA is immune to small changes in inputs and noise during the method.

b. ACO Algorithm

The social activities of insects and other animals are the basis for the relatively recent problem-solving strategy known as swarm intelligence. Ant colony optimization is a general-purpose optimization methodology that was specifically inspired by ants. This technique has been explored the most and is the most effective. The foraging behaviour of some ant species is the source of inspiration for ant colony optimization (ACO). To identify a preferred path that other colony members should take, these ants leave pheromone deposits on the ground. The optimization problem is solved using an analogous approach by ant colony optimization. As a result, other optimization methods were developed based on such evolutionary algorithms, expanding the field of meta-heuristics Swarm intelligence and meta-heuristic algorithms including Ant Colony Optimization (ACO and Particle Swarm Optimization (PSO).

Designing intelligent multi-agent systems using the collective behaviour of social insects such as ants, termites, bees, wasps, and other fauna such as flocks of birds and schools of fish is the goal of swarm intelligence. Ants are

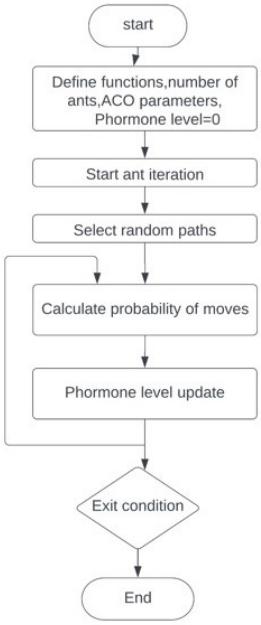

Fig. 3: ACO Algorithm Flowchart

eusocial insects that choose communal survival and sustainability over that of individual species. They use pheromones, sound, and touch to interact with one another. Pheromones are organic chemical substances released by ants that cause other individuals of the same species to act socially. These substances can influence how other people behave as they act like hormones outside the body of the person who secretes them. Since most ants live on the ground, they use the surface of the soil to leave pheromone trails that other ants can follow (smell). Ants reside in communal nests, and the fundamental idea behind ACO is to watch the ants leave their homes in order to find food in the quickest amount of time. Ants initially begin to wander aimlessly in and out of their nests in search of food. Numerous paths connect the nest to the food source as a result of this random search. Now, depending on the food's quality and amount, Ants bring back some of it with the right pheromone concentration on their way home. The likelihood that the following ants would choose a particular path would depend on these pheromone trials, and this probability would operate as a guiding factor to the food source. Clearly, the concentration and rate of pheromone evaporation are the foundations for this possibility. The length of each path can also be seen to be easily accounted for because the rate at which the pheromone evaporates is also a determining factor.

II. Selection And Search Criteria

The goal of this systematic review is to address the search-based approaches applied to the application of Genetic Algorithm and ACO Algorithm in software testing. The systematic literature review (SLR) is the major important step in the research process conducted to precisely gather, analyse and report the ongoing literature in the field. This SLR focuses specially on Meta-heuristic search-based optimization techniques, Genetic Algorithm and ACO Algorithm. Various studies are recently published and the authors have also explored the use of meta-heuristic optimization techniques in the context of the Genetic algorithm and ACO algorithm. The study is very rigorous, and detailed and reviews the publications of the last 10 years. This SLR primarily aims to review studies in the field of Genetic and ACO algorithm from the year 2012. These latest research papers were gathered initially based on the title and then further shortlisted based on the abstract of the respective papers published in the last 10 years which cited the studies included in the current SLR. The study and analysis of numerous papers collected from various sources involve time as well as research effort as it also encloses the review and modification of the paper. Therefore, in this paper, the relevant studies from the last 10 years were completely analysed. This SLR consists of various figures (Fig 4) and tables (Table 1-2) to give easy access to comprehensive knowledge on the topic.

III. Literature Review

There are numerous nature-inspired approaches. [7] presented a hybrid deep neural network model for enhanced prediction of software bugs. Different Nature-Inspired algorithms have been applied. Experimental investigations have been conducted using NASA dataset for the prediction of software defects and evaluation measures such as accuracy, computational time and F1 score. The approach based on combination of Genetic and Coral Reef meta-heuristics, achieving accuracy of around 96% and average F1 score of 0.92. [3] The proposed methodology is a memetic algorithm in which reinforcement learning is used as a local search method within a genetic algorithm. [4] Aim to shorten the test lengths and improve test process by adding a genetic algorithm to GraphWalker. It shows that genetic algorithm speeds up the test process by 45%. [2] Customizes and improves Ant Colony Optimization (ACO). It employs the idea of adaptive random testing in the local search. Test suites generated by the proposed approach has 9% better mutation score. [8] Proposed a methodology for automated software design using search-based software engineering ap-

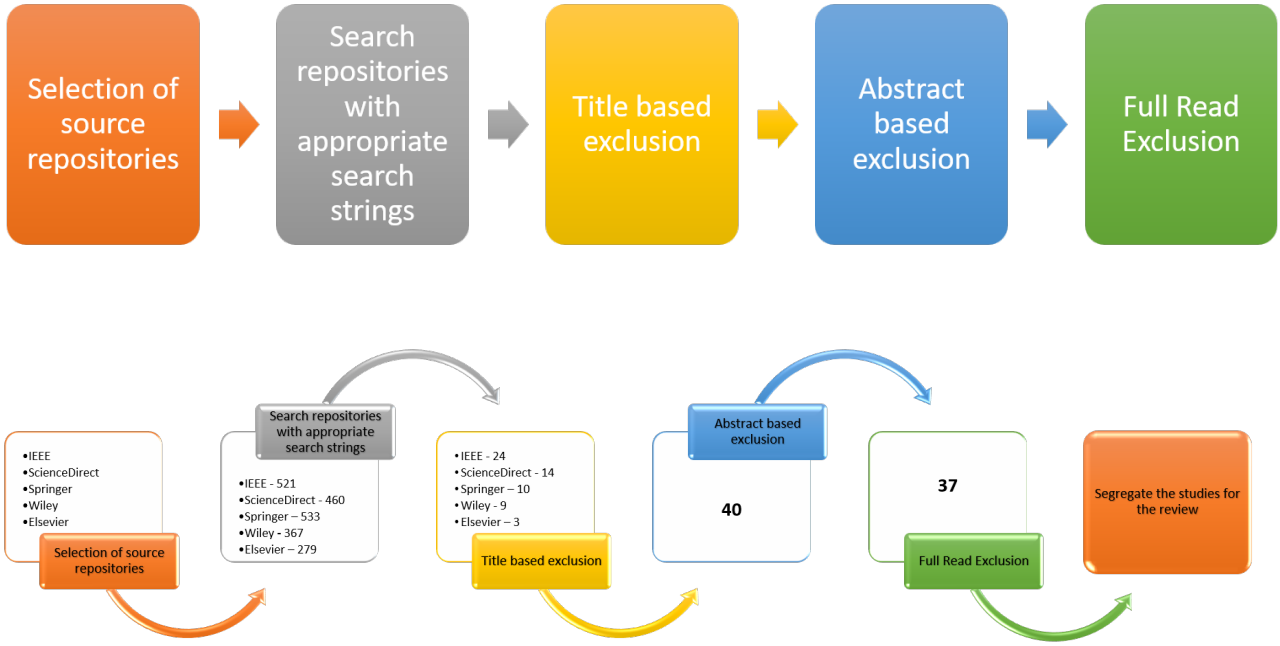

Fig. 4: Search and selection criteria

TABLE 1: REPOSITORIES AND THEIR CORRESPONDING SEARCH STRINGS

Repositories	Search Strings
IEEE	((“Software” AND (“testing” OR “test case”) AND (“application” OR “optimization” OR “prioritization”) AND (“genetic and ACO algorithm” OR “genetic algorithm” OR “ACO algorithm” OR “meta-heuristics”)))
ScienceDirect	(((“genetic OR ACO” OR “genetic algorithm” OR “genetic programming” OR “search-based optimization techniques” OR “metaheuristic”) AND (“testing" OR "test case") AND "Software")))
Springer	((“Software” AND (“testing” OR “test case”) AND (“ACO Algorithm” OR “Genetic Algorithm” OR “Genetic and ACO Algorithm” OR “Optimization Algorithm” OR ”Meta heuristic Algorithms” OR ”Regression testing” OR ”Prioritization”) AND (“optimization” OR “Application” Or “classification”)))
Wiley	((“Software” AND (“test case”) AND (“Ant colony Algorithm” OR “Genetic Algorithm” OR ”Regression testing” OR “Whale Algorithm”) AND (“optimization” OR “optimal”)
Elsevier	((“Software” AND (“testing” OR “test case”) AND (“ACO Algorithm” OR “Genetic Algorithm” OR “Genetic and ACO Algorithm” OR “Optimization Algorithm” OR ”Meta heuristic Algorithms”) AND (“optimisation” OR “Application” Or “prediction”)))

TABLE 2: INITIAL NUMBER OF PAPERS OBTAINED BY SEARCHING THE DATA REPOSITORIES

Repositories	Initial Count
IEEE	643
ScienceDirect	460
Springer	533
Wiley	367
Elsevier	279

TABLE 3: RESEARCH QUESTIONS (RQS)

No.	Research Questions
RQ1	What is the trend in the research of the application of Genetic and ACO algorithms in software testing?
RQ2	How is the test process improvised with the use of genetic and ACO algorithms?
RQ3	How can the total flaws present in the system can be detected along with the maximum possible paths in minimum time?
RQ4	What are the points of comparison between Genetic and ACO algorithms?

proach. Ant Colony Optimization is used as a meta-heuristic search algorithm in both single-objective and multi-objective modes. A semantic network is used. [5] In proposed approach, Hybrid Genetic Algorithm applied for civilizing the excellence of test case generation. Path coverage and mutation score are scrutinized from a result of an infinite number of test cases along with global search. Finally accomplished that the superior test cases with better quality is generated only by proposed approach. [6] Study is to develop optimum test cases by a modified Ant Colony Optimization (ACO) technique in an automated method. The prediction model used ensures better accuracy of the design of test inputs.

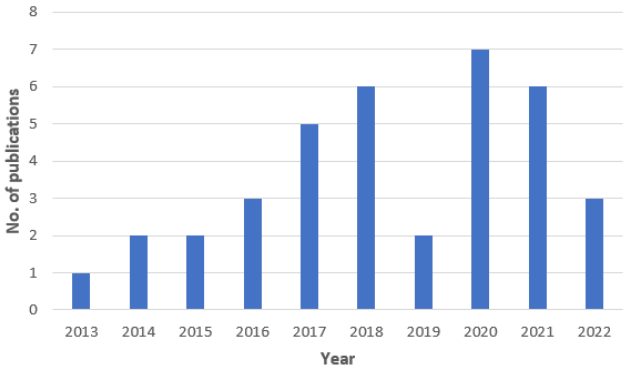

Fig. 5: Publication Trend in Genetic and ACO Algorithm

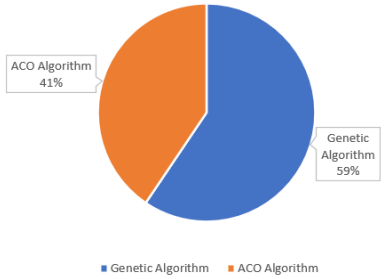

Fig. 6: Distribution of Research Papers Based on Algorithm

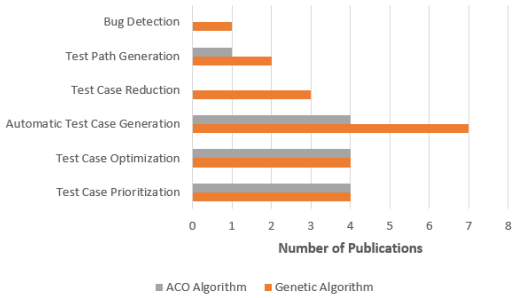

Fig. 7: Distribution of Research Papers Based on Applications

IV. Analysis of Studies

This section provides a final analysis of our SLR (systematic literature review) and gives the result of the research questions.

a. RQ1

We have analysed the trend of the research papers on the applications of the Genetic and ACO algorithms in software testing. Fig 5 shows the number of publications over the last 10 years. From Fig 5 we can see the increase in the number of research work in this field. 2019 has the highest number of publications and research. The study of Genetic and ACO algorithms is becoming popular among researchers due to their performance and high efficiency. As per our findings, both biologically inspired techniques, Genetic and ACO algorithms, are considered better than all the traditional techniques for the optimization of software testing. 59% research is on Genetic Algorithm Application and 41% on ACO Algorithm. We found that both algorithms are used mainly for testing optimization, test case generation and automation(Fig 7).

b. RQ2

Nowadays, applications of meta-heuristic algorithms for achieving optimized solutions have so much growth. Qualitative parameters are used by these algorithms whose values are easily adjustable. The speed of convergence of the meta-heuristic algorithm is high in the possibility of finding a universal optimized solution.

c. RQ3

During the selection of test cases, choosing a good test case sequence is very important and should be able to pass all errors with minimal execution time. As a result of our analysis, regression testing can be considered one of the most expensive, time-consuming and critical operations performed within an environment, with certain limitations to ensure the effectiveness of modified software where important test cases are executed before the lower important test cases. Research has shown that prioritization approaches based on test fac-

tors such as importance, volatility, complexity, failure rate, duration, and relevance work well and are comparable to sophisticated techniques such as ant colonies and the genetic algorithms-based approach has been shown to be superior.

d. RQ4

Genetic algorithms are search-based optimization techniques based on the fundamentals of Genetics and Natural Selection whereas Ant Colony Optimization is meta-heuristic influenced by the foraging behaviour of ant colonies. ACO adapts to changes in real-time. Genetic algorithms are used to find optimal solutions for optimization problems whereas Ant Colony Optimization can be used to find approximate solutions to complex optimization problems. Genetic Algorithms result in high-quality solutions. Genetic algorithms require more iterations as compared to Ant Colony Optimization. ACO gives better solutions but it takes more time compared to the Genetic Algorithm.

V. Conclusion and Future Scope

In this review paper, we did a detailed analysis of the trend in the research work of the application of Genetic and ACO algorithms in software testing. Due to high performance and efficiency, Genetic and ACO algorithms have become a popular topic among researchers. Genetic and ACO algorithms have been used in a variety of software testing fields such as test case optimization, test case generation, path generation, and so on.

According to our research, most of the research papers are on test optimization, test case generation and automation using Genetic and ACO algorithms. Future research can involve other applications of Genetic and ACO algorithms. More research work is required on ACO algorithms.

This review paper will guide the researchers in their future research work on the advances in the application of genetic and ACO algorithms in various software testing areas.

References

[1] Tanweer Alam, Shamimul Qamar, Amit Dixit, and Mohamed Benaida. Genetic algorithm: Reviews, implementations, and applications. *arXiv preprint arXiv:2007.12673*, 2020.

[2] Atieh Monemi Bidgoli and Hassan Haghighi. Augmenting ant colony optimization with adaptive random testing to cover prime paths. *Journal of Systems and Software*, 161:110495, 2020.

[3] Mehdi Esnaashari and Amir Hossein Damia. Automation of software test data generation using genetic algorithm and reinforcement learning. *Expert Systems with Applications*, 183:115446, 2021.

[4] Ramazan Koç, Alper Şen, and İsmail Öztürk. Accelerating user interface testing process with genetic algorithm. In *2021 15th Turkish National Software Engineering Symposium (UYMS)*, pages 1–3. IEEE, 2021.

[5] Priyanka. S and Subhashni. R. Automatic test case generation using hybrid genetic algorithm. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 1549–1556, 2021.

[6] Vinod Chandra SS et al. An ant colony optimization algorithm based automated generation of software test cases. In *International Conference on Swarm Intelligence*, pages 231–239. Springer, 2020.

[7] Kajal Tameswar, Geerish Suddul, and Kumar Dookhitram. A hybrid deep learning approach with genetic and coral reefs metaheuristics for enhanced defect detection in software. *International Journal of Information Management Data Insights*, 2(2):100105, 2022.

[8] Vali Tawosi, Saeed Jalili, and Seyed Mohammad Hossein Hasheminejad. Automated software design using ant colony optimization with semantic network support. *Journal of Systems and Software*, 109:1–17, 2015.