http://dx.doi.org/10.21622/ACE.2026.06.1.2069

DETERMINATION OF PARAMETERS FOR TEST CASE OPTIMIZATION IN ADAPTIVE CUCKOO SEARCH TECHNIQUE: A FUZZY DELPHI ANALYSIS

James Maina Mburu[1], John Gichuki Ndia[2] and Samson Wanjala Munialo[3]

[1,2] Murang'a University of Technology, Department of Information Technology, Kenya.

[3] Meru University of Science and Technology, Department of Information Technology, Kenya.

Emails: { jmburu48@gmail.com, jndia@mut.ac.ke, smunialo@must.ac.ke }

Received 09 April 2026 - Accepted 01 June 2026 - Published 07 June 2026

ABSTRACT

Cuckoo Search Algorithm (CSA) is widely used in test case optimization due to its simplicity and ease of implementation. However, the performance of CSA and its adaptive variants mainly relies on parameter configuration. Current studies mainly rely on heuristic tuning, empirical experimentation, and problem-specific adjustments, with limited use of systematic parameter selection approaches, which may potentially affect the efficiency, reproducibility, and consistency of the technique. To address this gap, this study proposes a structured Parameter Selection and Validation Framework for identifying and confirming critical parameters for the development of the Enhanced Adaptive Cuckoo Search Technique (EACST). The framework combines literature-based parameter identification, expert evaluation, and Fuzzy Delphi analysis to determine and validate critical parameters. The study conducted an expert opinion survey involving 56 software testing professionals, where parameters were initially identified through an extensive literature review and subsequently refined based on expert feedback. These responses were analyzed using the Fuzzy Delphi Method, which incorporates a consensus threshold($d \leq 0.2$) and defuzzification to assess agreement and rank parameters. The findings indicate that all the identified parameters satisfied the threshold condition and achieved a consensus level above 75%, with an overall agreement of 91%, signifying strong expert consensus. All parameters were accepted (α-cut ≥ 0.5), demonstrating their suitability for inclusion in the proposed technique. These results establish a reproducible framework for parameter selection that supports exploration–exploitation balance in test case optimization. However, the study is limited to expert-based evaluation and requires further validation through implementation. The proposed framework integrates literature-based parameter selection with Fuzzy Delphi analysis, bridging the gap between theoretical identification and practical technique development.

Keywords: *Enhanced Adaptive Cuckoo Search Technique (EACST), Expert Consensus, Exploration–Exploitation Balance, Fuzzy Delphi Method, Model-Based Testing, Parameter Selection, Software Testing, Test Case Optimization.*

1. INTRODUCTION

Software testing is a process in software engineering that ensures the production of quality, reliable, and correct software. It typically entails three main phases, namely: test case production, test case selection, and test execution. The most critical stage of software development, which consumes resources in terms of time, cost, and effort, is the test case generation [1]. Subsequently, the design of efficient techniques for generating optimal test cases has attracted significant research attention.

Currently, metaheuristic optimization techniques are employed to Unified Modeling Language

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

(UML), a standard modeling approach applied to represent system structure and behavior, which has been widely used for automated test case production. These techniques aim to improve testing efficiency by intelligently exploring for optimal test cases within large solution spaces. Techniques based on the Cuckoo Search Algorithm (CSA) and its variants [2],[3], Genetic Algorithms (GA) [4], Genetic–Crow Search hybrids [5], and Firefly–Bee Colony hybrids [6] have recorded promising performance in improving test coverage and optimization outcomes. However, they still experience premature convergence or slow search progression. These drawbacks culminate from an imbalance between global search and local refinement mechanisms within the optimization process [7].

CSA is among the well-established metaheuristic techniques due to its simplicity, low parameter dependency, and ease of implementation. The technique applies Lévy flight-based randomization and a probabilistic replacement approach to explore the search space. Despite these gains, the CSA displays notable shortcomings, namely, susceptibility to local optima, inadequate exploitation capability, and relatively slow convergence rates in challenging problem domains [8]. Various CSA variants, such as Adaptive Cuckoo Search (ACSA) [2], Adaptive Cuckoo Search based on a Dynamic Adjustment Mechanism (ACS-DAM)[9], Multi-Strategy Adaptive Cuckoo Search (MSACS) [10], Enhanced Cuckoo Search Algorithm (ECSA) [11], and Enhanced Cuckoo Search Optimization with Opposition-Based Learning (ECSO-OBL) [12], have been proposed and rely primarily on parameter adjustment mechanisms such as step size and abandonment probability.

The adjustment of parameters is often insufficient to effectively maintain population diversity, achieve stable convergence, or prevent premature convergence to local optima. Consequently, maintaining a consistent balance between exploration and exploitation remains a significant challenge. Furthermore, integrating multiple adaptive strategies may increase algorithmic complexity and introduce greater parameter dependency, hence the need to develop an Extended Adaptive Cuckoo Search Technique (EACST).To develop a new technique, it is necessary to determine the parameters required for its design. According to Shadkam [13], the efficiency of metaheuristic techniques largely depends on the appropriate Selection and adjustment of parameters. However, existing studies lack a systematic framework for parameter selection, often relying on arbitrary choices that may lead to the omission of critical parameters needed for the development of efficient techniques. They largely rely on heuristic tuning, empirical experimentation, and problem-specific adjustments for parameter configuration, with limited use of a structured, validated parameter selection framework. These limitations may result in inconsistent optimization performance, reduced reproducibility, and lower reliability of the proposed techniques. Therefore, there is a need for a structured parameter selection and validation mechanism to identify critical parameters that support an effective balance between exploration and exploitation, thereby improving optimization efficiency and ensuring consistent performance of the technique.

This study implements a structured framework for the identification and validation of critical parameters for the development of EACST. The approach integrates literature-based parameter identification, expert evaluation, and the Fuzzy Delphi Method (FDM) to establish a reproducible and validated basis for parameter selection. By incorporating expert knowledge and fuzzy analysis, the approach aims to determine the parameters that can enhance optimization in test case generation.

The remainder of this paper is as follows: Section 2 presents related literature, Section 3 describes the research methodology, Section 4 presents the findings, Section 5 discusses the results, Section 6 presents the threats to validity, and Section 7 concludes the study with recommendations for future research.

2. LITERATURE REVIEW

This section reviews existing studies on CSA variants with emphasis on parameter configurations, optimization approaches, and reported challenges. The review provides a basis for understanding current advancements and identifying critical parameters relevant to the proposed EACST Parameter Selection and Validation Framework.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Salgotra et al. [14] suggest a Self-Adaptive Cuckoo Search (SACS) that integrates a Gaussian bare-bones sampling mechanism with adaptive control of critical parameters, such as population size, search space, switch probability, and maximum iteration. The SACS was evaluated using benchmark optimization functions, where it demonstrated good performance, positioned first with a mean f-rank of 3.10. However, the Gaussian bare-bones sampling approach only enhances the exploration of the SACS and does not sufficiently support exploitation capability. Furthermore, the technique remains sensitive to parameter configurations, and the parameter settings were not systematically validated. In addition, the evaluation is mainly limited to benchmark problems, raising concerns about generalizability and practical applicability.

Also, hybrid and application-specific improvements have been discovered. Jeyaboopathiraja et al. [15] introduced an enhanced Adaptive Cuckoo Search Algorithm (ACSA) integrated with Artificial Neural Networks (ANNs) for cybersecurity applications, specifically for intrusion detection in cloud and IoT domains. The technique employs ACSA to enhance network routing and feature selection, while ANNs are used for classification of regular and malicious activities. The algorithm includes adaptive mechanisms such as an adaptive step-size adjustment, adaptive population control, and inertia weight to enhance search efficiency and convergence. The model was evaluated with a real-world cybersecurity dataset and demonstrated better performance in terms of accuracy, precision, recall, and F1-score compared to existing methods. Nonetheless, the combination of multiple adaptive strategies and learning models increases its complexity and establishes additional parameters that need careful configuration. Furthermore, the parameter configurations rely on predefined settings without a systematic parameter selection strategy. Moreover, the technique is mainly domain-specific to cybersecurity applications, which may constrain its generalizability to other optimization problems.

Likewise, Samal et al. [16] proposed an Adaptive Cuckoo Search (ACS) approach for optimal relay node placement in Wireless Body Area Networks (WBANs). In this approach, the step size is dynamically adjusted based on the fitness function to enhance convergence and avoid local optima. The technique optimizes multiple objectives, such as cost, coverage, energy consumption, delay, and load balancing, attaining optimal relay placement with less energy consumption by the 100th iteration. However, the technique is characterized by a large number of iterations to achieve optimal solutions. In addition, the use of a multi-objective fitness function increases computational complexity and sensitivity to parameter tuning. Moreover, parameter settings are mainly based on problem-specific configurations, which may reduce adaptability across different optimization scenarios. Furthermore, the evaluation is restricted to simulated WBAN environments with predefined conditions, which may limit generalizability to real-world applications.

Similarly, Yang et al. [9] proposed an Adaptive Cuckoo Search based on a Dynamic Adjustment Mechanism (ACS-DAM). In this approach, both discovery probability and step size are adaptively adjusted using exponential and logarithmic functions to maintain global exploration and exploitation capabilities. The technique was evaluated on 23 benchmark functions and demonstrated superior performance in terms of convergence speed and solution accuracy compared to standard CSA and several existing variants. It was further used to enhance Support Vector Machine (SVM) parameters, resulting in improved classification accuracy and faster convergence. Nevertheless, the adjustment approach relies on predefined mathematical functions and fixed parameter adjustment rules, which may restrict adaptability across diverse problem domains. Furthermore, the technique was primarily evaluated using benchmark functions and controlled datasets, raising concerns regarding its applicability and reproducibility in real-world optimization problems.

Likewise, Sahoo et al.[2] Suggest an Adaptive Cuckoo Search Algorithm (ACSA) for model-based test case production using a UML sequence model. The approach transforms a sequence diagram into a graph and employs ACSA to enhance test paths for improved coverage. Results indicate faster convergence, reaching optimal solutions at 100 iterations compared to 220 for standard CSA. Nevertheless, the technique requires a relatively large number of iterations to achieve optimal solutions, which may limit efficiency in

Volume 6, Issue 1, June 2026- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2026.06.1.2069

large applications. Furthermore, the technique relies on predefined parameters without a systematic parameter selection strategy. Although adaptive step size improves the balance between exploration and exploitation, its rule-based nature, rather than feedback-driven adaptation, may limit its effectiveness in large and complex search spaces. Moreover, the evaluation was conducted on a single case study, which may restrict its applicability to complex software testing scenarios.

Moreover, Gao et al. [10] introduced a Multi-Strategy Adaptive Cuckoo Search (MSACS) technique that combines several search strategies with dynamic selection mechanisms to enhance the balance between exploration and exploitation. The technique integrates crucial parameters such as population size, search space, fitness function discovery probability, step size, and strategy selection probability, which impact convergence behavior. The technique was evaluated on 24 benchmark functions and demonstrated enhanced accuracy and stability compared to existing techniques. However, the technique still requires a relatively large number of iterations (1000–2000) to achieve optimal solutions, which may limit efficiency. In addition, the incorporation of multiple strategies increases algorithmic complexity and parameter dependency, while parameter settings are mostly determined through trial-and-error tuning rather than a systematic selection approach. Furthermore, the evaluation is constrained to benchmark functions, restraining applicability to real-world optimization problems.

Likewise, Xiong et al. [3] suggested a Cuckoo Search algorithm based on a cloud model (CCS), where the step size factor is dynamically modified using fitness-based membership functions to enhance convergence and search accuracy. The technique integrates vital parameters such as discovery probability, step size factor, and cloud model parameters, which influence search behavior. The technique was validated on 25 benchmark functions and chaotic time-series prediction problems, demonstrating improved convergence speed, accuracy, and stability compared to several CSA and non-CSS algorithms. However, convergence remains dependent on parameter tuning, and the parameter configurations were not systematically validated. Moreover, the technique was mainly validated in simulation environments, which may limit efficiency and applicability in real-world optimization problems.

In addition, Abdulwahab et al. [17] suggested a Multi-Objective Binary Cuckoo Search Algorithm (MOBCSA) for feature selection in bioinformatics, enhancing both classification accuracy and the number of nominated genes. The technique combines parameters such as population size, abandonment probability, and maximum iterations, along with mechanisms such as an external archive and crowding distance to preserve diversity. The technique was evaluated on biomedical datasets, attaining classification accuracy between 92.79% and 98.42% and an average of 15.67 to 27.88 genes. Nevertheless, the technique establishes extra parameters and algorithmic complexity without systematic parameter validation. Also, much emphasis is placed on exploration, with relatively limited exploitation mechanisms, leading to an imbalanced search process. In addition, the evaluation is restricted to biomedical datasets, limiting applicability to real-world problems such as model-based test case optimization.

Similarly, Anwariningsih et al. [18] introduced an enhanced Cuckoo Search algorithm with adaptive parameters and hybrid distribution (Dynamic Hybrid Cuckoo Search with Contrast Limited Adaptive Histogram Equalization (DH-CSA-CLAHE)) for improving Contrast Limited Adaptive Histogram Equalization (CLAHE) parameters in medical image enhancement. The technique combines parameters such as adaptive discovery rate and step size with a hybrid distribution and combination of normal and uniform strategies to enhance the balance between exploration and exploitation. The technique demonstrates better performance in terms of image quality metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Mean Squared Error (MSE). However, the integration of dynamic and hybrid approaches increases algorithmic complexity. In addition, the balance between exploration and exploitation is guided by predefined parameter schedules, which may limit adaptability across diverse problem domains. Furthermore, the technique relies on predefined parameter configurations without a structured parameter selection and validation framework. Moreover, evaluation is limited to medical image processing applications, restricting generalizability to other domains.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Likewise, Villanueva et al.[11] developed an Enhanced Cuckoo Search Algorithm (ECSA) incorporating Sobol sequence initialization and adaptive parameter control using cosine annealing to enhance population distribution and convergence efficiency. The technique adaptively adjusts core parameters such as discovery rate and step size, achieving up to a 30% improvement in mean fitness and outperforming the standard CSA in 11 out of 13 benchmark functions, with statistically significant differences at the 5% significance level. Conversely, the technique establishes additional approaches that increase its complexity and parameter dependency. In addition, the performance in discrete optimization problems is not statistically significant, limiting its applicability to different problem domains.

In addition, Meena et al. [19] proposed a Cuckoo Search Optimization (CSO) technique improved with fuzzy logic for influence maximization in dynamic social networks. The technique integrated parameters such as discovery probability, maximum iterations, population size, and dynamic seed size, local search, while using closeness and degree centrality with influence strength in a fuzzy-based update mechanism to improve solutions. The technique was evaluated on several real-world datasets and indicated good performance in terms of activated nodes compared to benchmark techniques. However, the technique incurs computational cost due to fuzzy logic and centrality calculations, specifically in dense networks. Although the balance between exploration and exploitation is implicitly achieved through the interaction of Cuckoo Search operations and fuzzy-based enhancement, the absence of explicit adaptive control limits the technique's ability to respond effectively to dynamic search conditions. Furthermore, the technique relies on predefined parameter settings and lacks a structured parameter selection and validation framework. Moreover, the technique is mainly evaluated in social network contexts, restricting its applicability to other domains.

Lastly, Reddy et al. [12] suggested an Enhanced Cuckoo Search Optimization with Opposition-Based Learning (ECSO-OBL) for optimal sensor node placement in wireless sensor networks to improve coverage and connectivity. The technique combines parameters such as step size, population size, discovery probability, mutation probability, maximum iterations, and objective function while incorporating opposition-based learning, adaptive step-size control, and particle swarm optimization to improve exploration and exploitation. The technique achieved a maximum coverage rate of 98.45% in 143 iterations and reduced the execution time to 2.85 seconds, outperforming current techniques. However, the approach establishes several adaptive mechanisms, increasing parameter dependency and band complexity without structured validation; although convergence is enhanced, it still requires a large number of iterations. Overall, the reviewed studies highlight common trends in parameter usage, design strategies, performance outcomes, and associated limitations, which are summarized in Table 1.

Table 1: Summary of Reviewed Studies

Study	Parameters	Design / Strategy	Results Achieved	Identified Gaps
Salgotra et al. [14]	Population size, switch probability, Maximum iterations, Lévy flight (λ), search space, objective function	Self-Adaptive CS (Gaussian bare-bones)	Best performance (f-rank = 3.10)	Parameter sensitivity, weak exploitation, limited to benchmarks, and a lack of structured parameter validation
Jeyaboopathiraja et al. [15]	Population size, step size, inertia weight, Lévy flight (λ), objective function, Maximum iterations	Hybrid ACSA + ANN	Improved accuracy, precision, and recall	High complexity; parameter dependency; domain-specific, limited systematic parameter selection strategy
Samal et al. [16]	Population size, step size, iterations, multi-objective fitness function, Lévy flight (λ), Maximum iterations	Adaptive CS (WBAN optimization)	Optimal placement; minimal energy at 100 iterations	High iterations; complexity; limited to simulation, parameter settings based on problem-specific adjustment

Yang et al. [9]	Population size, step size, discovery probability, Lévy flight (λ), objective function	ACS with dynamic adjustment	Improved convergence, accuracy, stability	Predefined control; limited adaptability; benchmark-based, fixed parameter adjustment strategy
Sahoo et al. [2]	Population size, step size, iterations, lévy flight, fitness function, search space, Maximum iterations	ACSA for UML test case optimization	Improved convergence at 100 iterations	Iteration-dependent; single case study; parameter reliance, limited parameter optimization strategy
Gao et al. [10]	Population size, step size, Maximum iterations strategy probability, Lévy flight (λ), objective function	Multi-strategy adaptive CS	Improved accuracy and stability	High iterations complexity; trial-and-error tuning, lack of systematic parameter selection
Xiong et al. [3]	Population size, discovery probability, step size, cloud parameters, objective function	Cloud-based CS (CCS)	Improved convergence and stability	Parameter-dependent; simulation-based evaluation, limited parameter validation
Abdulwahab et al. [17]	Population size, abandonment probability (pa), Lévy flight, maximum iterations, fitness function, search space	Multi-objective binary CS	High accuracy (92.79%–98.42%)	Too many parameters; weak exploitation; limited domain, parameter settings based on problem-specific adjustment
Anwariningsih et al. [18]	Population size, abandonment probability (pa), iterations, fitness function	Dynamic hybrid CS	Improved PSNR, SSIM, MSE	Complexity; predefined balance; domain-specific, lack of structured parameter selection
Villanueva et al. [11]	Population size, step size, pa, Lévy flight (λ), objective function, Maximum iterations	ECSA with Sobol + cosine annealing	30% fitness improvement; statistically significant	Complexity; parameter dependency; sensitivity to parameter settings
Meena et al. [19]	Population size, pa, iterations, fitness function, Maximum iterations, local search (centrality), search space	CS + fuzzy logic	Competitive influence maximization results	High computational cost; implicit balance; domain-specific, lacks structured parameter selection
Reddy et al. [12]	Population size, probability of abandoning, step size, local search, mechanism, maximum iterations, mutation probability, Lévy flight (λ), objective function	ECSO-OBL (CS + PSO + OBL)	98.45% coverage; convergence at 143 iterations;	High parameter dependency; complexity; High iterations; complexity; limited to simulation, absence of a structured parameter selection mechanism

2.1. SYNTHESIS AND RESEARCH GAP

The reviewed studies indicate that, despite the variety of application domains and algorithmic improvements, a common set of parameters is consistently used in CSA and its adaptive variants. Parameters such as step size, population size, maximum iterations, abandonment probability, Lévy flight, search space, local search, and objective function play a fundamental role in governing exploration–exploitation balance, convergence behavior, and solution quality. However, the Selection of these parameters in existing studies largely relies on heuristic tuning, empirical experimentation, and problem-specific adjustments. Furthermore, although key parameters are commonly reported, limited evidence exists regarding a systematic procedure for selecting and validating parameter values, potentially affecting reproducibility, consistency, and applicability across studies.

Furthermore, Cuckoo Search variants still exhibit premature or slow convergence due to limitations in effectively balancing exploration and exploitation. Although adaptive, hybrid, and multi-strategy techniques have enhanced search performance, the balance remains inadequately controlled or not fully dynamic, resulting in inefficient search processes and a comparatively large number of iterations required to attain best solutions. In addition, the combination of numerous adaptive and hybrid mechanisms has meaningfully increased algorithmic complexity and parameter dependency, making these techniques hard to configure, computationally expensive, and less practical for real-world domains.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

In addition, most advanced Cuckoo Search variants are mostly evaluated in simulation-based environments, with limited application in model-based test case optimization. Existing implementations are frequently limited to single UML models and lack combined multi-UML frameworks, thereby restricting test coverage and reducing practical applicability in real-world software systems. Likewise, performance evaluation is often grounded on isolated metrics, focusing either on effectiveness or efficiency rather than a comprehensive assessment. The dependence on benchmark functions and small-scale case studies further limits generalizability and reduces confidence in real-world deployment.

To address these challenges, this study proposes an EACST Parameter Selection and Validation Framework that incorporates literature-based parameter identification guided by criteria such as frequency of use, role in exploration–exploitation balance, and empirical performance impact. The framework is further reinforced through expert evaluation and analysis using the Fuzzy Delphi Method, providing a systematic and reproducible basis for the design of EACST.

3. METHODOLOGY

This study used a structured expert opinion survey to validate the proposed EACST parameters using the Fuzzy Delphi Method (FDM). FDM is an enhancement of the traditional Delphi technique that combines fuzzy logic to manage uncertainty and vagueness in expert opinions [20]. The overall validation process was conducted in four phases to guarantee that the parameters selected are theoretically founded, practically relevant, and reinforced by expert consensus. The whole validation workflow is demonstrated in Fig.1 (EACST Parameter Selection and Validation Framework).

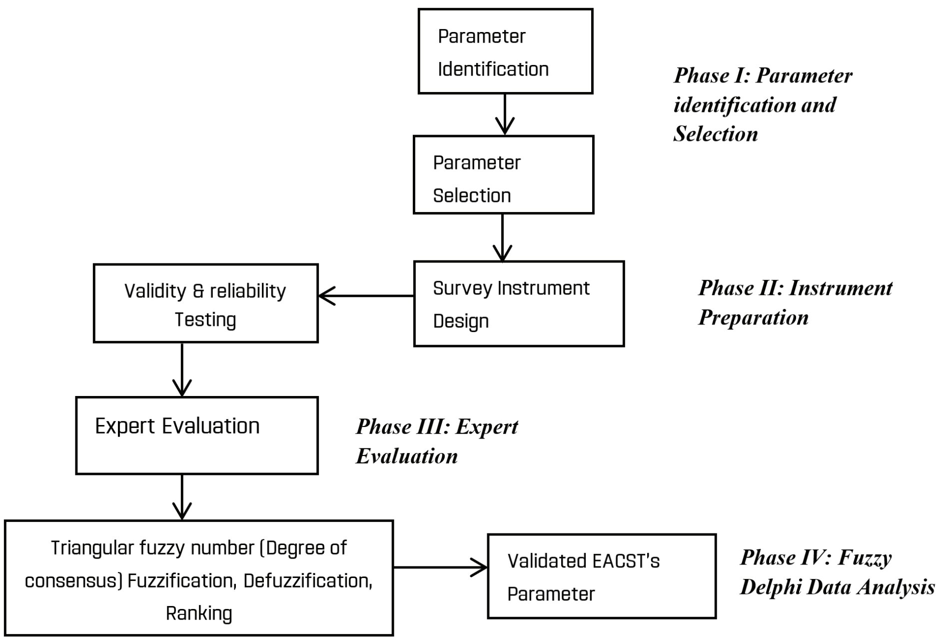

Figure 1: EACST Parameter Selection and Validation Framework

3.1. PHASE I: PARAMETER IDENTIFICATION AND SELECTION

This section describes the process used to identify and justify the critical parameters of the proposed Enhanced Adaptive Cuckoo Search Technique (EACST). The parameter identification process was guided by insights derived from reviewed studies on Cuckoo Search-based techniques to ensure that the selected parameters are supported by existing literature and practical applications. The process involves establishing selection criteria, mapping the identified parameters against these criteria, and providing justification for their inclusion in the proposed framework.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

3.1.1. Parameters Selection Criteria

The parameters of EACST were selected using a literature-driven approach based on three criteria, as shown in Table 2.

Table 2: Parameter Selection Criteria

Criterion	Description	Justification from Literature
C1: Frequency of Use	Parameter appears consistently across multiple ACS studies	Indicates maturity and wide research acceptance
C2: Role in Exploration–Exploitation Balance	The parameter directly influences global exploration or local exploitation behavior.	Essential for avoiding premature convergence
C3: Empirical Performance Impact	Parameter shown to improve convergence speed, solution quality, or coverage.	Supported by experimental evidence

3.1.2. Mapping of Parameters Selected

The mapping of selected parameters against these criteria (Table 3) shows that the majority satisfied all three conditions, indicating strong theoretical and empirical support in the literature. Parameters such as Population Size, Lévy Flight, Probability of abandonment, Search Space, Step Size, and Fitness/Objective Function satisfied all criteria. Maximum Iterations and Local Search Mechanism satisfied two of the three criteria but were reserved due to their demonstrated empirical impact on algorithmic performance.

Table 3: Mapping of Selected Parameters to Selection Criteria

Parameter	C1 (Frequency of Use)	C2 (Exploration-Exploitation Role)	C3(Empirical Performance Impact)	Ref
Population size (N)	✓	✓	✓	[2, 11, 12]
Maximum iterations	✓	–	✓	[10, 18, 19]
Abandonment probability (pa)	✓	✓	✓	[2, 10, 17]
Step size (α)	✓	✓	✓	[11, 16, 20]
Lévy flight (λ)	✓	✓	✓	[3, 16], 17]
Local search mechanism	–	✓	✓	[14, 17, 20]
Search space	✓	✓	✓	[17, 19, 20]
Fitness / objective function	✓	✓	✓	[2, 12, 16]

✓=Satisfy

–=Not Satisfy

3.1.3. Rationale for Parameter Selection

This section provides a parameter-specific justification for their inclusion in EACST. Each selected parameter is justified on grounds of its functional role in the optimization process and its empirical effectiveness as reported in prior studies. The rationale outlines how specific parameters contribute to search diversity, convergence behavior, and overall optimization efficiency, thereby supporting their integration into the proposed EACST framework. Table 4 maps the identified parameters to the relevant literature sources that support their inclusion.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Table 4: Rationale for Parameter Selection

Parameter	Rationale from literature	Ref
Population Size (N)	Determines diversity of solutions; larger N improves exploration but increases computational cost.	[2, 3, 11, 12]
Max Iterations (I)	Defines a stopping criterion to prevent unnecessary computation.	[10, 18, 19]
Step Size (α)	Controls Lévy flight step length; adaptive adjustment improves convergence.	[14, 16, 20]
Abandonment Probability (pa)	Probability of replacing poor solutions; balances exploration and exploitation.	[2, 10, 17]
Lévy Flight (λ)	Enhances global exploration by allowing long jumps.	[2, 13, 16]
Local Search Mechanism	Refines promising regions for exploitation.	[14, 17, 20]
Search space	Defines the feasible region within which the algorithm operates	[17, [21], [12]
Objective Function	Guides the optimization process	[2, 12, 16]

The parameter determination procedure provides a systematic literature-driven basis for selecting the critical parameters of EACST. However, these parameters require validation to guarantee their relevance and practical applicability.

For that reason, the next section presents the methodology, where the identified parameters are operationalized into a survey instrument and confirmed using the Fuzzy Delphi Method to achieve expert consensus.

3.2. PHASE II: INSTRUMENT PREPARATION

To begin with, the identified parameters were converted into a semi-structured questionnaire for expert assessment. Every parameter was clearly defined and evaluated using a five-point Likert scale (Table 5) to preserve consistency in responses.

Table 5: Likert Scale

Scale	Response
1	Strongly Disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly Agree

To ensure content validity, the instrument was checked by supervisors and three academic experts, who confirmed its appropriateness. In addition, a pilot study comprising nine software testing practitioners was performed to verify clarity and logical flow, leading to slight adjustments.

The reliability of the instrument was assessed using Cronbach's alpha, where a value of 0.70 or higher is generally considered acceptable [23],[24]. The instrument attained a coefficient of 0.939 across eight items, demonstrating a high level of internal consistency (as shown in Table 6.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Table 6: Internal consistency

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
0.939	0.943	8

3.3. PHASE III: EXPERT EVALUATION

In this phase, software testers (experts) were chosen using a combination of purposive and snowball sampling. Purposive sampling ensured that only individuals with relevant expertise were included, while snowball sampling was applied to identify extra qualified experts through referrals. The validated questionnaire was subsequently distributed to the experts, who rated each parameter according to their level of agreement using a Likert scale. The responses received formed the base for determining the level of agreement among experts.

3.4. PHASE IV: FUZZY DELPHI ANALYSIS

The gathered responses were examined using the Fuzzy Delphi Method (FDM) to derive consensus on the suggested parameters. Each Likert-scale response was transformed into a Triangular Fuzzy Number where denote the least, most likely, and highest values, respectively. This conversion was conducted using the fuzzy scale presented in Table 7, which connects linguistic responses into equivalent fuzzy numerical values. This translation facilitates qualitative expert judgments to be conveyed quantitatively.

Table 7: Fuzzy Scale Agreement Levels

Response Category	Fuzzy Value Range
Strongly Disagree	(0.0,0.1,0.2)
Disagree	(0.1,0.2,0.4)
Neutral	(0.2,0.4,0.6)
Agree	(0.4,0.6,0.8)
Strongly Agree	(0.6,0.8,1.0)

To assess the level of consensus among experts, the threshold value was calculated using the distance between fuzzy numbers as shown in Equation (1):

$$d(m,n) = \sqrt{\frac{1}{3}\left[(m_1 - n_2)^2 + (m_2 - n_2)^2 + (m_3 - n_3)^2\right]} \qquad (1)$$

A value of indicates satisfactory agreement among experts. Furthermore, a minimum consensus level of 75% was required for each parameter to be considered valid. These criteria ensure that only parameters with sufficient expert agreement are retained for further analysis [25],[26].

After evaluating expert agreement, defuzzification was performed to obtain an individual representative value for each parameter. This process transforms the Triangular Fuzzy Numbers into a crisp value to facilitate ranking and decision-making. In this study, the centroid method was applied as shown in Equation (2)[26]:

$$\mathrm{A} = \frac{1}{3} * (m_1 + m_2 + m_3) \qquad (2)$$

The subsequent defuzzified value was then employed to determine the acceptance of each parameter. α-cut value of 0.5 was adopted as the decision threshold. Parameters with were acknowledged, indicating consensus among experts, while those with values below 0.5 were disallowed [20],[26]. Lastly, the parameters were ranked based on their defuzzified values to determine their relative importance within the proposed EACST framework.

4. RESULTS

Throughout the application of the FDM, expert comments and proposals were integrated to enrich the parameters. After prioritization, the statements were reviewed, with no additional parameters proposed by the experts. Even though 56 responses were received, only 45 were considered valid, with 11 excluded due to incompleteness or duplication.

4.1. EXPERT PERSONAL INFORMATION

4.1.1. Academic Qualification

The academic qualifications of the experts who took part in the survey are shown in Fig.2. The majority of experts (71.4%) hold a Bachelor's degree, followed by 19.6% with a Diploma, while a smaller percentage (8.9%) hold a Master's degree.

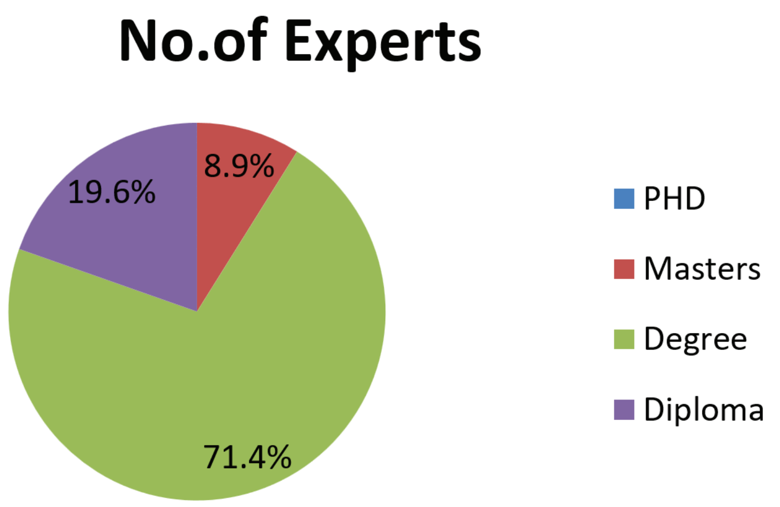

Figure 2: Level of education

4.1.2. Number of years in the software industry

Fig.3 shows the distribution of industry experience among the experts. The results demonstrate strong representation of experienced professionals. A substantial percentage (26.8%) has over eight years of experience, while 25% have 2–3 years of experience. Furthermore, 23.2% fall within the 4–5 year range, and 17.9% have 6–7 years of experience. Outstandingly, only 7.1% of the experts have less than two years of experience (0–1 year).

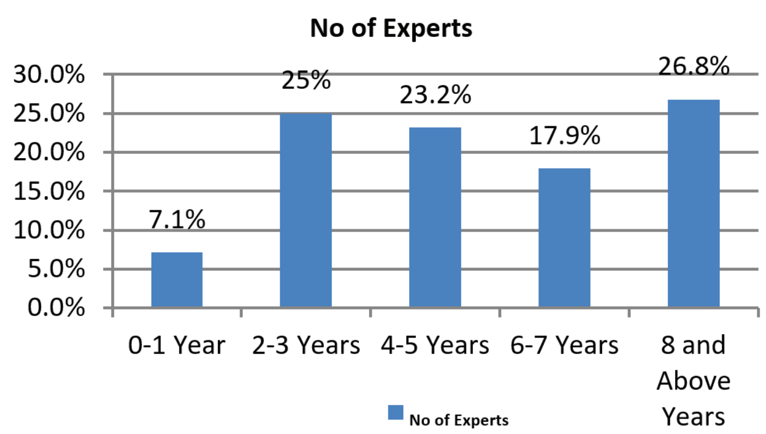

Figure 3: Number of years in the software industry

4.1.3. Knowledge in Software Testing

The results from Table 8 indicate that the largest number of experts ranked themselves between 3 (moderate knowledge) and 4 (high knowledge), with 4 being the most common rank.

Table 8: Knowledge in software testing

Rank	Frequency	Percentage (%)
1(Very Low)	0	0%
2 (Low)	5	11.1%
3 (Moderate)	15	33.3%
4 (High)	19	42.2%
5 (Very High)	6	13.3%
Total	45	100%

4.2. PARAMETER ANALYSIS

4.2.1. Expert Consensus Analysis

The findings of the Fuzzy Delphi analysis are illustrated in Table 9, which includes threshold value, percentage of expert consensus, fuzzy score, and parameter rankings. All parameters satisfied the threshold condition, demonstrating agreement among experts. Furthermore, the consensus level exceeded 75%, with an overall agreement of 91%, confirming strong expert consensus.

TABLE 9: Expert consensus about parameters of EACST

E	PS	I	LF	SS	PA	LS	OF	SP
1	0.2	0.4	0.2	0.4	0.2	0.2	0.2	0
2	0.2	0.2	0.2	0.2	0.2	0	0	0
3	0	0	0.2	0.2	0.2	0	0	0
4	0.2	0.2	0	0	0.2	0.2	0	0
5	0.2	0.2	0	0	0.2	0.2	0	0
6	0.2	0	0	0.2	0	0.2	0.2	0.2
7	0	0.2	0.2	0	0	0.2	0.2	0.2
8	0	0.2	0.2	0	0	0	0	0
9	0	0	0	0.2	0	0	0	0.2
10	0	0	0	0	0.2	0.2	0.2	0.2
11	0	0.2	0	0.2	0.4	0.2	0	0.4
12	0	0.4	0.2	0.2	0.2	0.2	0	0.2
13	0	0.2	0.2	0.2	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2
16	0.2	0	0.4	0.2	0.2	0.2	0	0
17	0	0.2	0	0	0	0.2	0.2	0.4
18	0.2	0.2	0	0.2	0.2	0.4	0.2	0.4
19	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

20	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2
21	0	0.2	0	0.2	0.2	0	0	0.2
22	0	0	0	0.2	0.2	0	0	0.2
23	0	0.2	0.2	0	0	0	0.2	0
24	0.2	0.2	0	0	0	0	0	0
25	0.2	0	0	0.4	0	0.2	0	0.2
26	0.2	0.2	0	0.4	0	0.2	0	0.2
27	0.2	0.2	0	0	0	0.2	0.2	0
28	0	0.2	0	0.2	0	0.2	0	0
29	0.2	0	0.2	0	0.2	0	0	0
30	0	0	0.2	0	0	0	0	0
31	0	0	0	0	0.2	0.2	0.2	0.2
32	0	0.2	0	0	0.2	0.2	0	0
33	0	0.2	0	0	0	0	0.2	0
34	0	0.2	0	0	0.2	0.2	0.2	0
35	0.2	0.2	0.2	0	0.2	0.2	0.2	0.2
36	0.2	0.4	0.2	0.2	0.2	0.4	0.2	0.2
37	0.2	0.2	0	0.4	0	0.2	0.2	0
38	0.2	0.2	0.2	0	0	0.2	0	0.2
39	0.2	0.2	0.7	0.5	0.4	0	0	0.2
40	0	0	0.7	0.6	0.4	0	0	0.2
41	0	0.2	0	0	0.2	0	0	0
42	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
43	0.2	0	0.2	0.2	0	0	0	0
44	0	0	0	0.2	0	0.2	0.2	0
45	0.1	0	0.2	0.0	0	0.1	0.1	0.1
D	0.1	0.1	0.1	0.1	0.1	0.1	0	0.1
% CEEP	91%	93%	82%	89%	87%	93%	100%	96%
% CEAP				91%				
F	0.653	0.667	0.598	0.650	0.601	0.702	0.716	0.702
R	5	4	8	6	7	2	1	2

Key; E-Expert, **PS**-Population Size, **I**- No of Iterations, **LF**- Lévy Flight, **S**-Step Size, **P**-Probability of Abandoning, **LS**-Local Search, **OF**-Objective Function, **SP**- Search Space.
D-Threshold value, **CEEP**- Percentage Consensus of Experts for Every Parameter, **CEAP**- Percentage Consensus of Experts for all Parameters, **F**-Fuzzy Score, **R**-Parameter Ranking

The threshold values obtained for all parameters were within the acceptable limit, confirming consistency in expert opinions. In addition, the percentage consensus for individual parameters surpassed the required 75% threshold, while the overall consensus among experts reached 91%, showing strong agreement concerning the relevance of the proposed EACST parameters. These results indicate that the identified parameters satisfied the predefined criteria for subsequent ranking and acceptance analysis.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

4.2.2. Parameter Ranking and Acceptance

Following the evaluation of expert consensus, the validated parameters were ranked based on their defuzzified values to determine their relative importance within the proposed EACST framework. Parameters with values greater than 0.5 were accepted for inclusion in the technique. The ranking and acceptance results are presented in Table 10.

Table 10: EACST's Parameters Ranking Based on Fuzzy Score

Parameters	Fuzzy score (f)	Ranking (r)	Expert Consensus
Population Size(N)	0.653	5	Accept
Max. Iterations(I)	0.667	4	Accept
Lévy Flight	0.598	8	Accept
Step Size	0.650	6	Accept
Probability of Abandoning(pa)	0.601	7	Accept
Local Search Method	0.702	2	Accept
Objective Function	0.716	1	Accept
Search Space	0.702	2	Accept

The findings show that the objective function achieved the highest rankfollowed by search space and local search mechanisms . Lévy flight obtained the lowest score ; however, it still satisfied the predefined acceptance criteria. These findings confirm that all identified parameters are valid for inclusion in EACST.

4.2.3. Integration of Validated Parameters into EACST

The validated parameters were subsequently mapped into the proposed EACST framework to demonstrate their functional roles within the optimization process. As illustrated in Fig. 4, population size and search space will be used to generate and define the initial candidate solutions. The objective function will guide the optimization process by evaluating solution quality, while Lévy flight supports exploration, and step size regulates movement during the search process. Furthermore, the local search mechanism will facilitate exploitation of promising regions, abandonment probability will be applied to preserve population diversity, and maximum iterations will be used to define the stopping criterion for attaining optimal solutions.

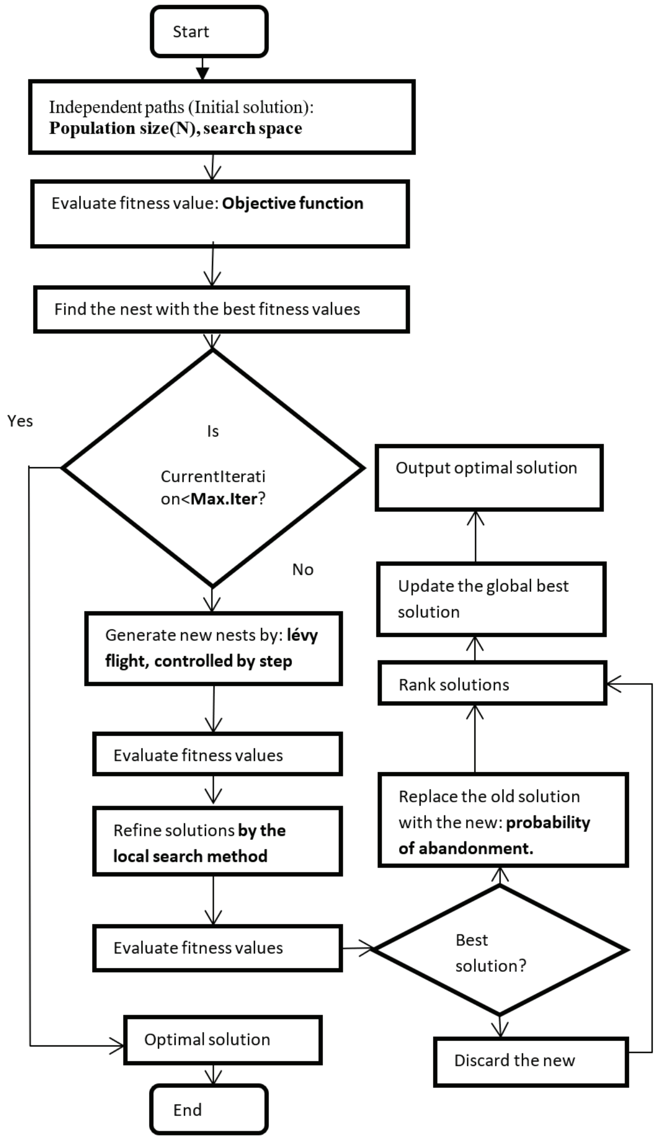

Figure 4: Parameter integration within EACST

5. DISCUSSION

5.1. EXPERT DEMOGRAPHIC DATA

The demographic findings demonstrate that the survey was conducted with a competent and experienced group of experts, guaranteeing the reliability of the results. The majority of respondents hold Bachelor's and Master's degrees, indicating a strong academic base in ICT-related fields, while their diverse industry experience offers a balanced blend of practical and theoretical perspectives. Furthermore, the reported moderate to very high knowledge in software testing approves the participants' capability to give informed evaluations. These characteristics reinforce the validity of the expert consensus and enhance confidence in the reliability of the parameter validation process, addressing common concerns in the literature regarding the subjectivity of expert-based methods.

5.2. PARAMETER ANALYSIS

The findings indicate that all proposed EACST parameters achieved strong expert consensus and satisfied the predefined acceptance criteria, confirming their relevance for adaptive metaheuristic design. The high ranking of the objective function **(f = 0.716)** emphasizes its importance in guiding the optimization process through effective fitness evaluation. This observation is consistent with studies by Sahoo et al.[2] , Samal et al. [16] and Abdulwahab et al. [17], which identified the objective function as a major factor influencing convergence behavior and solution quality. The strong agreement among experts further demonstrates that an appropriate fitness evaluation mechanism is essential for determining the quality of candidate solutions and directing the search process toward optimal outcomes.

Similarly, the ranking of search space and local search mechanisms **(f = 0.702)** reflects the importance of exploitation strategies in refining promising solutions and improving convergence performance. Comparable observations were reported by Gao et al. [10] and Villanueva et al. [11], where adaptive local search mechanisms enhanced solution refinement and optimization efficiency. Defining an appropriate search space is equally important because it determines the region within which feasible solutions are explored and reduces unnecessary computational effort associated with irrelevant search areas. These findings suggest that effective exploitation strategies contribute significantly to improving optimization performance while minimizing premature convergence.

Parameters including maximum iterations, step size, and abandonment probability were also identified as important components influencing convergence behavior, search dynamics, and population diversity. Similar findings were reported by Yang et al. [9], who observed that adaptive adjustment of step size improved convergence speed and solution accuracy, and by Salgotra et al. [14], where adaptive parameter control enhanced search efficiency. Collectively, these parameters support the balance between exploration and exploitation, a concept widely recognized in metaheuristic optimization as a requirement for obtaining high-quality solutions while maintaining search efficiency.

Although population size and Lévy flight obtained comparatively lower rankings, their acceptance indicates that they remain important for preserving search diversity and strengthening global exploration capability. Similar observations were reported by Xiong et al. [3] and Anwariningsih et al. [18], where these parameters contributed to search-space exploration and prevention of premature convergence. Their relatively lower ranking may be attributed to expert perceptions that these parameters have a more indirect influence on optimization performance compared to parameters such as the objective function and local search mechanisms, which directly affect solution evaluation and convergence behavior.

The strong consensus achieved across all identified parameters suggests that a structured approach to parameter determination may improve consistency in the design of metaheuristic techniques. Existing studies often rely on heuristic tuning and problem-specific parameter adjustments, which can lead to variations in optimization behavior and reproducibility. The present findings show that expert-based validation provides a more systematic basis for identifying significant parameters, thereby supporting more reliable parameter configuration in adaptive Cuckoo Search techniques.

From a practical perspective, the validated framework may assist researchers and practitioners in configuring adaptive Cuckoo Search techniques with reduced dependence on trial-and-error parameter tuning. This can improve consistency in optimization performance and support more efficient test case generation within software testing environments. Nevertheless, the current findings are based on expert consensus and therefore require further experimental validation through implementation and performance evaluation of the proposed EACST in real-world software testing scenarios.

6. THREATS TO VALIDITY

6.1. INTERNAL THREATS

Sampling bias was a potential threat because access to industry experts was limited; therefore, snowball sampling was initially used to identify software testers through referrals from other qualified participants. However, snowball sampling may introduce bias because participants are often recruited from existing professional networks, potentially excluding less-connected experts and limiting diversity within the sample. To minimize this threat, purposive sampling was employed to ensure that only participants with relevant expertise and experience in software testing were selected for inclusion in the study.

Instrumentation threat was considered because a poorly designed questionnaire could have misled experts and affected the accuracy of their responses. To minimize this threat, the questionnaire was reviewed by three academic experts who provided feedback and confirmed that the instrument met the required criteria. In addition, the instrument was tested for internal consistency using Cronbach's alpha, where the results indicated a high level of reliability.

6.2. EXTERNAL THREAT

Generalizability was considered a potential external threat because the findings of this study were derived from the opinions of software testing professionals. Therefore, the results may be applicable within software engineering contexts but may not fully represent perspectives from other domains, such as broader engineering fields or computer networks. Consequently, the applicability of the findings across diverse contexts should be interpreted with caution.

7. CONCLUSION AND FUTURE WORK

This study aimed to establish a structured and validated framework for identifying and validating critical parameters for the Enhanced Adaptive Cuckoo Search Technique (EACST) in test case optimization. The findings demonstrated strong agreement among experts, with all identified parameters satisfying the predefined acceptance criteria. Parameters, including objective function, search space, local search mechanism, maximum iterations, population size, step size, abandonment probability, and Lévy flight, were confirmed as important components for supporting effective and efficient test case optimization. The findings further indicate that optimization performance is influenced not only by the inclusion of these parameters but also by their systematic Selection and configuration.

The primary contribution of this study is the proposed EACST Parameter Selection and Validation Framework, which integrates literature-based parameter identification, expert evaluation, and Fuzzy Delphi analysis into a structured and validated process. Unlike many existing studies that rely on heuristic tuning and problem-specific parameter settings, the proposed framework provides a reproducible and systematic basis for parameter determination. This contributes toward improving consistency, reliability, and optimization efficiency in adaptive Cuckoo Search techniques while strengthening exploration–exploitation balance during optimization. From a practical perspective, the validated framework provides guidance for researchers and practitioners in configuring adaptive metaheuristic techniques for software testing applications. Nevertheless, the results are based on expert consensus and therefore require further validation through implementation and performance evaluation within real-world environments.

Future work will focus on the design, implementation, and evaluation of EACST software testing scenarios. Further refinement of parameters and integration with other optimization approaches may improve scalability, robustness, and applicability across broader optimization domains.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to the panel of experts who participated in the survey.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

REFERENCES

[1] A.Tamizharasi , P. Ezhumalai, S. Remya Rose, P. Suresh, and S. Logesswarie, "Bio Inspired Approach for Generating Test data from User Stories," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 2, pp. 412–419, Apr. 2021, doi: https://doi.org/10.17762/turcomat.v12i2.826.

[2] R. K. Sahoo, S. Satpathy, S. Sahoo, and A. Sarkar, "Model driven test case generation and optimization using adaptive cuckoo search algorithm," *Innovations in Systems and Software Engineering*, vol. 18, no. 2, pp. 321–331, Jan. 2021, doi: https://doi.org/10.1007/s11334-020-00378-z.

[3] Y. Xiong, Z. Zou, and J. Cheng, "Cuckoo search algorithm based on cloud model and its application," *Scientific Reports*, vol. 13, no. 1, pp. 1–13, Jun. 2023, doi: https://doi.org/10.1038/s41598-023-37326-3.

[4] R. K. Sahoo, M. Derbali, H. Jerbi, D. Van Thang, P. Pavan Kumar, and S. Sahoo, "Test Case Generation from UML-Diagrams Using Genetic Algorithm," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2321–2336, 2021, doi: https://doi.org/10.32604/cmc.2021.013014.

[5] A. Tamizharasi and P. Ezhumalai, "Genetic-based Crow Search Algorithm for Test Case Generation," *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, vol. 13, no. 4, pp. 1–11, 2022, doi: https://doi.org/10.14456/ITJEMAST.2022.74.

[6] S. S. Panigrahi, P. K. Sahoo, B. P. Sahu, A. Panigrahi, and A. K. Jena, "Model-driven Automatic Paths Generation and Test Case Optimization Using Hybrid FA-BC," *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 263–268, Mar. 2021, doi: https://doi.org/10.1109/esci50559.2021.9396999.

[7] L. Liu, X. Liu, N. Wang, and P. Zou, "Modified Cuckoo Search Algorithm with Variational Parameters and Logistic Map," *Algorithms*, vol. 11, no. 3, p. 30, Mar. 2018, doi: https://doi.org/10.3390/a11030030.

[8] A. Sharma, A. Sharma, V. Chowdary, A. Srivastava, and P. Joshi, "Cuckoo Search Algorithm: A Review of Recent Variants and Engineering Applications," *Studies in Computational Intelligence*, pp. 177–194, Oct. 2020, doi: https://doi.org/10.1007/978-981-15-7571-6_8.

[9] Y. Yang, M. Fu, S. Yu, C. Jia, and X. Zuo, "Adaptive Cuckoo Search Algorithm Based on Dynamic Adjustment Mechanism," 電腦學刊, vol. 32, no. 5, pp. 171–183, Oct. 2021, doi: https://doi.org/10.53106/199115992021103205014.

[10] J. Cheng and Y. Xiong, "Multi-strategy adaptive cuckoo search algorithm for numerical optimization," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 2031–2055, Jun. 2022, doi: https://doi.org/10.1007/s10462-022-10222-4.

[11] M. A. Villanueva, C. M. Ching, and K. Mata, "An Enhancement of Cuckoo Search Algorithm for Optimal Earthquake Evacuation Space Allocation in Intramuros, Manila City," *arXiv (Cornell University)*, Feb. 2025, doi: https://doi.org/10.48550/arxiv.2502.13477.

[12] M. R. Reddy, M. L. R. Chandra, and R. Dilli, "Enhanced Cuckoo Search Optimization with Opposition-Based Learning for the Optimal Placement of Sensor Nodes and Enhanced Network Coverage in Wireless Sensor Networks," *Applied Sciences*, vol. 15, no. 15, p. 8575, Aug. 2025, doi: https://doi.org/10.3390/app15158575.

[13] E. Shadkam, "Parameter setting of meta-heuristic algorithms: a new hybrid method based on DEA and RSM," *Environmental Science and Pollution Research*, vol. 29, no. 15, pp. 22404–22426, Nov. 2021, doi: https://doi.org/10.1007/s11356-021-17364-y.

[14] R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, "Self adaptive cuckoo search: Analysis and experimentation," *Swarm and Evolutionary Computation*, vol. 60, p. 100751, Aug. 2020, doi: https://doi.org/10.1016/j.swevo.2020.100751.

[15] J. Jeyaboopathiraja, P. Mariajohn, S. S. Maidin, and J. Sun, "An Adaptive Cuckoo Search Algorithm with Deep Learning for Addressing Cyber Security Problem," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1977–1988, Dec. 2024, doi: https://doi.org/10.47738/jads.v5i4.366.

[16] T. K. Samal, S. C. Patra, and M. R. Kabat, "An adaptive cuckoo search based algorithm for placement of relay nodes in wireless body area networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1845–1856, May 2022, doi: https://doi.org/10.1016/j.jksuci.2019.11.002.

[17] H. M. Abdulwahab, S. Ajitha, A. Naji, B. Abdullah, and F. A. Ghanem, "MOBCSA: Multi-Objective Binary Cuckoo Search Algorithm for Features Selection in Bioinformatics," *IEEE access*, vol. 12, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3362228.

[18] S. H. Anwariningsih, Wahyono, and R. Sumiharto, "An Improved Cuckoo Search Algorithm with Dynamic Parameters and Hybrid Distribution for Enhanced CLAHE," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 24148–24158, Aug. 2025, doi: https://doi.org/10.48084/etasr.10652.

[19] M. Sharma and B. Pathik, "Crow Search Algorithm with Improved Objective Function for Test Case Generation and Optimization," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1125–1140, 2022, doi: https://doi.org/10.32604/iasc.2022.022335.

[20] N. K. Ismail, S. Mohamed, and M. I. Hamzah, "The Application of the Fuzzy Delphi Technique to the Required Aspect of Parental Involvement in the Effort to Inculcate Positive Attitude among Preschool Children," *Creative Education*, vol. 10, no. 12, pp. 2907–2921, 2019, doi: https://doi.org/10.4236/ce.2019.1012216.

[21] S. K. Meena, S. S. Singh, and K. Singh, "Cuckoo Search Optimization-Based Influence Maximization in Dynamic Social Networks," *ACM Transactions on the Web*, vol. 18, no. 4, pp. 1–25, Oct. 2024, doi: https://doi.org/10.1145/3690644.

[22] S.-Y. Yang, Y.-H. Xiang, D.-W. Kang, and K.-Q. Zhou, "An Improved Cuckoo Search Algorithm for Maximizing the Coverage Range of Wireless Sensor Networks," *Mağallaẗ baġdād li-l-ʿulūm*, vol. 21, no. 2(SI), pp. 0568–0568, Feb. 2024, doi: https://doi.org/10.21123/bsj.2024.9707.

[23] H. C. Dandekar, "Qualitative methods in planning research and practice," *Journal of Architectural and Planning Research*, vol. 22, no. 2, pp. 129–137, 2005.

[24] S. Hari Sugiharto, "Comparative Test of Cronbach's Alpha Reliability Coefficient, Kr-20, Kr-21, And Split-Half Method," *Journal of Education Research and Evaluation*, vol. 8, no. 1, pp. 47–57, Feb. 2024, doi: https://doi.org/10.23887/jere.v8i1.68164.

[25] T. T. Yin and H. Hanif, "Fuzzy Delphi Method: A Step-by-Step Guide to Obtain Expert Consensus on MUETBot Functionalities," *International journal of academic research in business & social sciences*, vol. 14, no. 4, pp. 1097–1104, Apr. 2024, doi: https://doi.org/10.6007/ijarbss/v14-i4/21307.

[26] S. N. A. Mohamad, M. A. Embi, and N. Nordin, "Determining e-Portfolio Elements in Learning Process Using Fuzzy Delphi Analysis," *International Education Studies*, vol. 8, no. 9, Aug. 2015, doi: https://doi.org/10.5539/ies.v8n9p171.