

Metaheuristic Approaches for Test Redundancy Reduction: A Systematic Review of Algorithms, Optimization Strategies, and Research Trends

*Norasyikin Safieny¹, Kamal Z. Zamli², Hasneeza L. Zakaria³ and Najwa Ayuni Jamaludin¹

¹Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Melaka, Malaysia

²Fakulti Komputeran, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Pekan, Pahang, Malaysia

³Jabatan Komputer, Fakulti Komputeran Cerdas, Universiti Malaysia Perlis (UniMAP), Malaysia

*Corresponding Author

DOI: <https://doi.org/10.47772/IJRISS.2026.100801034>

Received: 12 September 2026; Accepted: 17 September 2026; Published: 23 September 2026

ABSTRACT

Test redundancy reduction (TRR) addresses the continued growth of regression test suites by selecting a smaller test set while preserving required testing adequacy. This updated systematic review maps metaheuristic approaches to TRR and closely related test-suite reduction/minimization across algorithm family, search architecture, task structure, objective structure, evaluation practice, and reproducibility. A seeded-update review design was used: the test-suite-reduction evidence base reported by Habib et al. was re-screened, followed by targeted bibliographic searching and backward and forward snowballing to update the evidence through September 2026. The quantitative corpus contains 48 peer-reviewed primary studies published from 2010 to 2026. Thirty-three studies (68.8%) use single or modified-single architectures, while 15 (31.3%) use hybrid architectures. Swarm-intelligence methods account for 24 studies (50.0%). Multi-objective optimization appears in 18 studies (37.5%), whereas 30 (62.5%) use single-objective formulations. Six studies (12.5%) explicitly employ multi-task or multifactorial optimization, with this cluster emerging in 2024–2025, while single-task formulations remain dominant (42/48, 87.5%). The review therefore reframes multi-task TRR from an absent direction to a recent but still comparatively underrepresented research direction. Across the corpus, direct algorithm ranking remains inappropriate because benchmark suites, adequacy criteria, search budgets, parameter reporting, and runtime environments differ substantially. The review further distinguishes algorithm family from search architecture, task structure, and objective structure so that hybrid, multi-objective, and multi-task formulations are not conflated. The resulting evidence map provides an auditable basis for future hybrid, multi-objective, and multi-task TRR research.

Keywords: test redundancy reduction, test suite minimization, metaheuristics, hybrid optimization, multi-objective optimization

INTRODUCTION

Regression testing is essential for determining whether software changes have affected previously working behaviour. As software evolves, regression suites tend to grow and may contain tests that exercise overlapping requirements or code elements. Test-suite minimization, selection and prioritization are related but distinct regression-testing activities; minimization specifically seeks a smaller suite while retaining a defined adequacy criterion (Khan et al., 2018; Yoo & Harman, 2012).

TRR is naturally formulated as a combinatorial optimization problem. Search-based software engineering has therefore motivated genetic, swarm, physics-inspired and hybrid search strategies for reducing testing effort while maintaining adequacy. Earlier surveys show substantial diversity in reduction frameworks and search-based methods (Habib et al., 2023; Khan et al., 2016; Rahman et al., 2024).

The literature has already been reviewed from several perspectives. Khan et al. synthesized test-suite-reduction approaches and experimental quality, while Habib et al. organized 57 selected search-based TSR studies from 2007–2022 into evolutionary, swarm-intelligence, human-inspired, physics-based and hybrid families (Habib et al., 2023; Khan et al., 2018). The present review uses that historical evidence as a discovery anchor and extends the evidence through September 2026.

This study therefore reviews metaheuristic approaches for TRR with four connected emphases: algorithm families, optimization strategy, evaluation practice, and research trends. In particular, the review distinguishes standalone from hybrid algorithms and single-task from multi-task or multi-objective formulations. This distinction is important because combining algorithms is not equivalent to solving multiple optimization tasks, and optimizing multiple objectives is not equivalent to multi-task optimization.

The contribution of this review is an updated, study-level synthesis that connects algorithmic development with optimization formulation and evidence quality. It separates algorithm architecture (single/modified versus hybrid), task structure (single-task versus multi-task/multifactorial), and objective structure (single- versus multi-objective), because these dimensions answer different research questions. All included primary studies are individually coded in Supplementary Table S1 and mapped to the reference list.

Historical TSR evidence contains both deterministic and search-based methods, and prior reviews emphasize that reduction, selection, and prioritization should not be treated as interchangeable concepts (Khan et al., 2016; Khan et al., 2018; Yoo & Harman, 2012). The present synthesis focuses on metaheuristic/search-based optimization studies that directly reduce, minimize, select, or jointly optimize test suites under an explicit testing adequacy or effectiveness goal.

Single or modified-single metaheuristics span evolutionary, swarm, human-inspired and physics-inspired search. Representative examples include genetic test-suite optimization (Jeyaprakash & Alagarsamy, 2015; Nachiyappan et al., 2010; Sheikh et al., 2023), simulated-annealing TRR (Zamli et al., 2015), particle-swarm minimization (Ahuja & Bhatia, 2024; Deneke et al., 2022; Sun & Wang, 2010), ant-colony variants (Bharathi & Sangeetha, 2018; Mohanty et al., 2020; Zhang et al., 2017), flower-pollination methods (Chintapalli & Singh, 2019; Kabir et al., 2017), and bat/ant-lion variants (Bajaj et al., 2022; Reda et al., 2022; Wei et al., 2022).

Multi-objective formulations explicitly optimize more than one criterion within the same reduction task. Examples combine suite size or execution cost with coverage, fault detection, localization, or related effectiveness measures (Ahmed, 2016; Anwar et al., 2019; Choudhary et al., 2018; Freeda & Rajendran, 2024; Gupta et al., 2022; Habib et al., 2024; Reda et al., 2022; Sharma et al., 2010; Turner et al., 2016). This dimension is distinct from multi-task optimization, where knowledge is transferred or coordinated across related optimization tasks.

Hybridization combines complementary search mechanisms or algorithmic components. The corpus includes early GA/bee-colony and concept-analysis/PSO combinations (Selvakumar et al., 2013; Suri et al., 2011), hybrid cuckoo/bee and FA-BC approaches (Lakshminarayana & SureshKumar, 2021; Panigrahi et al., 2021), GNA-SA (Zamli et al., 2018), improved ABC with TLBO mechanisms (Ahuja & Bhatia, 2023), hybrid fault-detection/localization optimization (Freeda & Rajendran, 2024), relay-based PSO-Jaya (Mojumder et al., 2025), and adaptive SCA-PSO (Kader et al., 2025).

Existing reviews cover much of the historical landscape (Habib et al., 2023; Khan et al., 2018; Rahman et al., 2024). The main reason for updating the evidence is the recent appearance of explicitly multi-task/multifactorial TRR and new hybrid strategies, which materially changes the earlier interpretation of this research gap.

RELATED LITERATURE AND EVOLUTION OF METAHEURISTIC TRR

From deterministic reduction to search-based optimization

Early and intermediate studies demonstrate the diversity of search-based reduction: genetic search (Jeyaparakash & Alagarsamy, 2015; Nachiyappan et al., 2010), Tabu Search (Sharma et al., 2010), simulated annealing (Zamli et al., 2015), PSO (Sun & Wang, 2010), and greedy/genetic combinations (Yamuç et al., 2017). These approaches establish the optimization basis on which later swarm, hybrid, multi-objective and multi-task formulations build.

Single Metaheuristics

Single-algorithm approaches remain useful baselines because they expose the behaviour of one search family without cross-family hybrid effects. The corpus includes ACO-Min and Lion Search (Asthana et al., 2020; Mohanty et al., 2020), quantum-ant and dragonfly variants (Sugave et al., 2017; Zhang et al., 2017), Firefly (Sahoo et al., 2016), Flower Pollination (Chintapalli & Singh, 2019), PSO (Ahuja & Bhatia, 2024; Deneke et al., 2022), Bat (Bajaj et al., 2022; Reda et al., 2022), Ant Lion (Wei et al., 2022), Sparrow Search (Rahman et al., 2024), and Social Network Search (Mekeng et al., 2026).

Multi-objective optimization

Multi-objective TRR reflects the practical reality that testing quality cannot be reduced to cardinality alone. The included studies optimize or evaluate combinations of reduction, execution cost, coverage, fault detection, fault localization, or similarity-related objectives (Ahmed, 2016; Anwar et al., 2019; Bharathi & Sangeetha, 2018; Bharathi, 2022; Choudhary et al., 2018; Freeda & Rajendran, 2024; Gan et al., 2021; Gupta et al., 2022; Habib et al., 2024; Reda et al., 2022; Sharma et al., 2010; Turner et al., 2016). Because formulations and metrics differ, these studies should be interpreted through their stated trade-offs rather than a single global rank.

Hybrid metaheuristics

Hybrid methods seek complementary exploration and exploitation or combine an optimizer with another selection/refinement mechanism. The included hybrid evidence spans BCO-GA (Suri et al., 2011), cuckoo/combinatorial filtering (Ahmed, 2016), PSO/firefly (Bharathi, 2022), FA-BC (Panigrahi et al., 2021), GNA-SA (Zamli et al., 2018), ABC-TLBO mechanisms (Ahuja & Bhatia, 2023), and recent PSO-Jaya and SCA-PSO strategies (Kader et al., 2025; Mojumder et al., 2025).

Multi-task and emerging optimization strategies

Multi-task optimization is conceptually different from multi-objective optimization. Six included studies explicitly use multi-task or multifactorial formulations: population-division multi-task SCA (Zamli et al., 2024), multifactorial Rat Swarm Optimization (Kang et al., 2025), multi-task Brown Bear (Sidek et al., 2025b), multi-task SCA (Zamli et al., 2025), adaptive multi-task SCA-PSO (Kader et al., 2025), and MT-ZOA (Rizwan et al., 2025).

Experimental quality and reproducibility

The credibility of metaheuristic comparisons depends on transparent experimental design. Prior software-engineering SLR guidance and TSR reviews emphasize systematic study identification, reproducible extraction and careful interpretation of heterogeneous empirical evidence (Brereton et al., 2007; Habib et al., 2023; Khan et al., 2018; Kitchenham & Charters, 2007). Accordingly, this review does not infer statistical superiority when papers use different benchmarks, budgets, adequacy measures, parameter settings, or execution environments.

METHODOLOGY

Research design

This study adopts an updated evidence-mapping SLR design guided by established software-engineering review practice (Brereton et al., 2007; Kitchenham & Charters, 2007). It is not presented as a de novo PRISMA review because database hit, duplicate-removal and screening counts were not reconstructed retrospectively. Instead, the method transparently reports a seeded historical re-screen plus a verified update search.

Research questions

Table 1: Research questions and motivations

ID	Research question	Motivation
RQ1	What types of metaheuristic-based TRR/TSR studies have been published from 2010 to 2026, and how has publication activity evolved?	Map publication year, venue, study type, and empirical context across the review window.
RQ2	Which metaheuristic algorithm families are used and how have they evolved?	Identify dominant and emerging search families.
RQ3	To what extent are standalone, modified, hybrid, single-task, multi-task and multi-objective strategies investigated?	Characterize optimization strategy and the direction of methodological innovation.
RQ4	Which objectives, adequacy criteria, datasets and performance metrics are used?	Determine how effectiveness, adequacy and efficiency are evaluated.
RQ5	Under what comparable evaluation conditions do recent approaches report competitive performance?	Avoid unsupported universal rankings across incompatible benchmarks.
RQ6	What methodological, practical and reproducibility limitations remain?	Assess baselines, repeated runs, statistics, parameters, artifacts and validation.
RQ7	What research directions emerge from the evidence, particularly for hybrid, multi-objective, and multi-task optimization?	Identify opportunities supported by observed distributions and recurring methodological gaps.

Participants, materials or data sources

Study discovery used a seeded-update strategy. Habib et al.'s 57-study search-based TSR review for 2007–2022 was used as the historical discovery anchor (Habib et al., 2023). Candidate historical studies were re-screened against this review's scope, and targeted bibliographic searching plus backward/forward snowballing were used for 2023 through 8 September 2026. Publisher pages, DOI records, proceedings, or institutional repositories were used to verify bibliographic identity and relevance where available.

The core search concept is: ("test redundancy reduction" OR "test suite reduction" OR "test suite minimization") AND (metaheuristic OR "meta-heuristic" OR "search-based" OR optimization OR evolutionary OR swarm OR hybrid). Database-specific variants will also include terms for genetic algorithm, particle swarm, simulated annealing, Grey Wolf, Sparrow Search, Sine Cosine, multi-objective, and multi-task optimization.

Table 2: Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
English peer-reviewed journal or conference paper	Non-English or non-peer-reviewed work

Published from January 2010 to September 2026	Published before 2010
Addresses TRR, test suite reduction or minimization	Only test prioritization or test generation without reduction
Uses or empirically compares search/metaheuristic optimization	No relevant optimization evidence
Provides sufficient information to answer at least one RQ	Insufficient method/evaluation information
Full text or sufficient publisher/proceedings evidence available for eligibility and coding	Bibliographic identity or methodological relevance cannot be verified

Procedure and data analysis

Study selection for the historical window followed re-screening of the earlier systematic corpus, followed by targeted update searching and snowballing for 2023–2026. Records were retained when they addressed test-suite reduction/minimization or closely related test optimization using a metaheuristic/search-based optimization strategy and contained enough methodological information for coding. The quantitative corpus contains 48 studies.

Reporting quality was evaluated descriptively using eight criteria: clarity of aim; adequacy of dataset or subject-system description; reproducibility of the optimization procedure; appropriateness of baselines and metrics; reporting of repeated runs or statistical analysis; reporting of parameter settings and stopping criteria; availability or identifiability of artifacts; and alignment between conclusions and evidence. Because several historical studies were available only through bibliographic/secondary-review evidence, these criteria are used to characterize reporting completeness rather than as an exclusion threshold. No numerical quality score is fabricated where full-text evidence is unavailable.

Table 3: Data extraction framework

Dimension	Extracted information
Bibliographic	Year, venue, document type
Problem context	TRR/TSR/TSM; white-box/black-box; regression context
Algorithm	Family, variant, standalone/modified/hybrid
Optimization formulation	Single-task, multi-task, single-objective, multi-objective
Adequacy	Requirement, statement, branch, checked coverage, mutation/fault detection
Evaluation	Benchmarks, project scale, baselines, run count, statistical tests
Efficiency	Reduction %, suite size, execution/search time, convergence
Reproducibility	Parameters, seeds, implementation, hardware, code/data availability
Evidence	Main findings, limitations, future directions

RESULTS

The final quantitative evidence map contains 48 individually coded primary studies published from 2010–2026. The final corpus includes the verified 2025 Brown Bear/Roulette Wheel Brown Bear TRR study (Sidek et al., 2025a). Supplementary Table S1 supplies the complete study-level coding and bibliographic traceability.

Publication and algorithm trends (RQ1-RQ2)

Among the 48 included studies, swarm-intelligence methods form the largest coded family (24/48, 50.0%). Single or modified-single architectures account for 33 studies (68.8%) and hybrid architectures for 15 (31.3%). Evolutionary approaches account for 6 studies (12.5%), human-inspired search for 2 (4.2%), and physics-based search for 1 (2.1%). The update period includes improved swarm methods (Ahuja & Bhatia, 2024; Mekeng et al., 2026; Rahman et al., 2024; Sidek et al., 2025a) and hybrid strategies (Ahuja & Bhatia, 2023; Freeda & Rajendran, 2024; Kader et al., 2025; Mojumder et al., 2025), while explicit multi-task/multifactorial work appears mainly in 2024–2025 (Kader et al., 2025; Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025).

Table 4: Evidence-corpus classification of 48 primary studies

Dimension	Category	n	%
Architecture	Single / modified single	33	68.8
Architecture	Hybrid	15	31.3
Task structure	Single-task	42	87.5
Task structure	Multi-task / multifactorial	6	12.5
Objective structure	Single-objective	30	62.5
Objective structure	Multi-objective	18	37.5
Algorithm family	Swarm intelligence	24	50.0
Algorithm family	Hybrid	15	31.3
Algorithm family	Evolutionary	6	12.5
Algorithm family	Human-inspired	2	4.2
Algorithm family	Physics-based	1	2.1

Table 5: Publication-year distribution of the 48 included primary studies

Year	Studies	Year	Studies
2010	3	2011	1
2013	1	2015	2
2016	4	2017	4
2018	3	2019	2

2020	3	2021	4
2022	6	2023	2
2024	5	2025	7
2026	1		

Optimization strategies (RQ3)

Table 6: Proposed taxonomy of optimization strategies for TRR

Strategy	Description	Evaluation implication
Standalone	One metaheuristic performs the search	Simpler attribution; requires strong baseline comparison
Modified improved	Operators, parameters or schedules of one method are changed	Ablation is needed to show the modification contributes
Hybrid	Two or more search mechanisms are integrated	Component contribution and fair baselines should be reported
Single-task	One TRR optimization task is solved independently	Does not exploit knowledge transfer among related tasks
Multi-task	Related optimization tasks are solved jointly or with knowledge transfer	Requires task definition and transfer analysis
Multi-objective	Several objectives are optimized simultaneously	Requires Pareto-aware or trade-off evaluation

The coding confirms that architecture, task structure, and objective structure are separate dimensions. Thirty-three studies (68.8%) use single or modified-single architectures and 15 (31.3%) are hybrid. Forty-two studies (87.5%) are single-task and six (12.5%) are multi-task/multifactorial (Kader et al., 2025; Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025). Thirty studies (62.5%) are coded as single-objective and 18 (37.5%) as multi-objectives. Representative multi-objective studies include neuro-fuzzy regression-suite optimization (Anwar et al., 2019), Pareto Harmony Search (Choudhary et al., 2018), fault detection/localization optimization (Freeda & Rajendran, 2024; Gupta et al., 2022), adapted Binary Bat (Reda et al., 2022), and similarity-based multi-objective optimization (Habib et al., 2024).

Evaluation objectives and metrics (RQ4)

Table 7: Evaluation dimensions for metaheuristic TRR

Dimension	Representative measures	Interpretation
Reduction	Reduced suite size; reduction percentage	Efficiency of redundancy removal
Coverage	Requirement, statement, branch, checked coverage	Preservation of adequacy
Fault effectiveness	Mutation score; fault detection	Preservation of defect-detection value

Cost	Execution time; search time; computational budget	Practical efficiency
Search behaviour	Best/mean fitness; convergence; stability	Optimization quality and stochastic behaviour
Statistical evidence	Variance; significance tests; effect size	Strength of comparative claims

Comparative performance (RQ5)

The corpus does not support a universal best-algorithm ranking because studies differ in subject systems, adequacy criteria, optimization budgets, and performance measures. For example, particle-swarm, genetic, swarm-hybrid, and multi-objective studies are evaluated under different experimental contexts (Deneke et al., 2022; Habib et al., 2024; Mojumder et al., 2025; Sheikh et al., 2023). The defensible synthesis is therefore comparative at the level of design patterns, objectives, and reporting practice rather than a pooled performance league table.

Methodological and reproducibility limitations (RQ6)

Recurring comparability problems include heterogeneous benchmarks, incomplete parameter reporting, different stopping rules, inconsistent repeated-run reporting, and missing runtime context. These concerns are consistent with earlier TSR syntheses (Habib et al., 2023; Khan et al., 2018; Rahman et al., 2024). Consequently, claims of algorithmic superiority should be restricted to matched experimental conditions within the original studies.

Emerging research directions (RQ7)

Three directions are supported by the updated corpus. First, hybridization is established but remains less common than single/modified-single search (31.3% versus 68.8%); recent examples such as relay PSO-Jaya and adaptive SCA-PSO motivate stronger ablation and matched-budget comparisons (Kader et al., 2025; Mojumder et al., 2025). Second, multi-objective TRR is established (18/48, 37.5%) and should report trade-offs explicitly (Freeda & Rajendran, 2024; Gupta et al., 2022; Habib et al., 2024; Reda et al., 2022). Third, multi-task/multifactorial optimization forms a recent six-study cluster (12.5%) (Kader et al., 2025; Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025). Future work should examine knowledge-transfer mechanisms, negative transfer, heterogeneous tasks, reproducible benchmarks, and industrial-scale validation.

Corpus provenance and coding reliability

Supplementary Table S1 provides the study-level audit trail for all 48 included primary studies, including bibliographic identity, year, venue, algorithm/strategy, architecture, task structure, objective structure, algorithm family, source/DOI, and verification basis. The historical evidence was cross-checked against established TSR syntheses (Habib et al., 2023; Khan et al., 2018; Rahman et al., 2024), while the 2023–2026 update studies were verified against publisher, DOI, proceedings, or repository records.

DISCUSSION

The principal value of this review is to connect algorithm choice with optimization formulation and evidence quality. Earlier syntheses establish the breadth of test-suite reduction and search-based optimization (Habib et al., 2023; Khan et al., 2016; Khan et al., 2018; Rahman et al., 2024), while the updated corpus shows that recent work is increasingly differentiated by hybrid architecture, multiple objectives, and cross-task knowledge transfer.

In the 48-study corpus, single/modified-single architectures dominate (68.8%) over hybrids (31.3%), while multi-objective formulations account for 37.5%. Task structure is more uneven: 87.5% are single-task and 12.5%

are explicitly multi-task/multifactorial. The six multi-task studies are recent (Kader et al., 2025; Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025), so the research opportunity concerns generalization, transfer design and validation rather than the first introduction of multi-task TRR.

This review has limitations. Terminology varies across TRR, test-suite reduction, minimization, selection, prioritization, and broader test optimization, which can affect search sensitivity. The historical component is a seeded re-screen of an earlier systematic corpus rather than a reconstructed database search, and therefore the review does not report unobserved PRISMA identification or deduplication counts. These limitations are stated to preserve auditability rather than imply exhaustiveness beyond the documented evidence set (Habib et al., 2023).

The evidence indicates three overlapping but non-equivalent research axes: algorithm family, optimization formulation, and task structure. A single swarm algorithm can be multi-objective without being hybrid (Reda et al., 2022); a hybrid can remain single-task (Mojumder et al., 2025; Zamli et al., 2018); and an explicitly multi-task method can use a single search family (Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025). Keeping these axes separate prevents taxonomy errors and yields a clearer research agenda.

This separation is also useful for interpreting research maturity. A field containing many named metaheuristics does not necessarily contain strong comparative evidence. If algorithms are tested on different datasets and objectives, it is unsafe to conclude that one family is universally superior. The review should instead identify patterns under comparable settings and highlight where replication or shared benchmarks are missing.

For future TRR research, multi-task optimization deserves attention because it is emerging but not yet broad. Current evidence covers SCA, Rat Swarm, Brown Bear, Zebra Optimization, and SCA-PSO formulations (Kader et al., 2025; Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025). Stronger future studies should evaluate when transfer helps, when negative transfer occurs, and whether multi-task advantages persist under matched search budgets and heterogeneous industrial tasks.

CONCLUSION

This updated systematic review maps 48 primary studies on metaheuristic test redundancy and test-suite reduction from 2010 to 2026 using separate coding dimensions for algorithm family, search architecture, task structure, and objective structure. Single or modified-single architectures remain dominant (68.8%), while hybrids account for 31.3%. Swarm-intelligence methods represent 50.0% of the corpus, and multi-objective optimization appears in 37.5%. Six studies (12.5%) explicitly use multi-task or multifactorial optimization (Kader et al., 2025; Kang et al., 2025; Rizwan et al., 2025; Sidek et al., 2025b; Zamli et al., 2024; Zamli et al., 2025), making this an emerging but still comparatively underrepresented direction. The central methodological priority is now reproducible, matched-budget evaluation across heterogeneous test tasks rather than proliferation of named algorithms alone.

ACKNOWLEDGEMENTS

The authors acknowledge Universiti Teknikal Malaysia Melaka (UTeM), Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), and Universiti Malaysia Perlis (UniMAP) for academic and institutional support.

Ethical Approval

Ethical approval was not applicable because this study is a systematic review of published literature and does not involve human participants, human data, or animals.

Conflict Of Interest

The authors declare no conflict of interest.

Data Availability

The study-level coding matrix used for the synthesis is provided as Supplementary Table S1. It contains all 48 included primary studies and their coding for year, venue, authors, algorithm/strategy, architecture, task structure, objective structure, algorithm family, source/DOI, and verification basis. No confidential or human-subject data are used.

REFERENCES

1. Ahmed, B. S. (2016). Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. *Engineering Science and Technology, an International Journal*, 19(2), 737–753. <https://doi.org/10.1016/j.jestch.2015.11.006>
2. Ahuja, N., & Bhatia, P. K. (2023). An improved artificial bee colony optimization algorithm for test suite minimization. *International Journal of Advanced Computer Science and Applications*, 14(7). <https://doi.org/10.14569/IJACSA.2023.0140774>
3. Ahuja, N., & Bhatia, P. K. (2024). Test suite minimization through PSO. 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). <https://doi.org/10.1109/ICRITO61523.2024.10522114>
4. Anwar, Z., Afzal, H., Bibi, N., Abbas, H., Mohsin, A., & Arif, O. (2019). A hybrid-adaptive neuro-fuzzy inference system for multi-objective regression test suites optimization. *Neural Computing and Applications*, 31(11), 7287–7301. <https://doi.org/10.1007/s00521-018-3560-8>
5. Asthana, M., Gupta, K. D., & Kumar, A. (2020). Test suite optimization using lion search algorithm. In *Ambient Communications and Computer Systems* (pp. 77–90). Springer. https://doi.org/10.1007/978-981-15-1518-7_7
6. Bajaj, A., & Abraham, A. (2021). Prioritizing and minimizing the test cases using the dragonfly algorithms. *International Journal of Computer Information Systems and Industrial Management Applications*, 13, 62–71.
7. Bajaj, A., Sangwan, O. P., & Abraham, A. (2022). Improved novel bat algorithm for test case prioritization and minimization. *Soft Computing*, 26(22), 12393–12419. <https://doi.org/10.1007/s00500-022-07121-9>
8. Bharathi, M., & Sangeetha, V. (2018). Weighted rank ant colony metaheuristics optimization based test suite reduction in combinatorial testing for improving software quality. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). <https://doi.org/10.1109/ICCONS.2018.8663102>
9. Bharathi, M. (2022). Hybrid particle swarm and ranked firefly metaheuristic optimization-based software test case minimization. *International Journal of Applied Metaheuristic Computing*, 13(1), 1–20. <https://doi.org/10.4018/IJAMC.290534>
10. Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4), 571–583. <https://doi.org/10.1016/j.jss.2006.07.009>
11. Chintapalli, S. S., & Singh, S. P. (2019). Flower pollination algorithm for effective test case optimization in software testing. *International Journal of Engineering and Advanced Technology*, 9(1), 4711–4716. <https://doi.org/10.35940/ijeat.A1983.109119>
12. Choudhary, A., Agrawal, A. P., & Kaur, A. (2018). An effective approach for regression test case selection using Pareto based multi-objective harmony search. *Proceedings of the 11th International Workshop on Search-Based Software Testing*, 13–20. <https://doi.org/10.1145/3194718.3194722>
13. Deneke, A., Assefa, B. G., & Mohapatra, S. K. (2022). Test suite minimization using particle swarm optimization. *Materials Today: Proceedings*, 60(Part 1), 229–233. <https://doi.org/10.1016/j.matpr.2021.12.472>
14. Freeda, A. R., & Rajendran, S. P. (2024). Test suite optimization under multi-objective constraints for software fault detection and localization: Hybrid optimization based model. *Web Intelligence*, 22(2). <https://doi.org/10.3233/WEB-220131>

15. Gan, N., Wang, H., Zhao, Y., & Yang, F. (2021). Test case set reduction method based on GRE and ant colony algorithm. 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), 417–421.
16. Geng, J., Li, Z., Zhao, R., & Guo, J. (2016). Search based test suite minimization for fault detection and localization: A co-driven method. In Search Based Software Engineering (LNCS 9962, pp. 34–48). Springer. https://doi.org/10.1007/978-3-319-47106-8_3
17. Gupta, N., Sharma, A., & Pachariya, M. K. (2022). Multi-objective test suite optimization for detection and localization of software faults. Journal of King Saud University - Computer and Information Sciences, 34(6), 2897–2909. <https://doi.org/10.1016/j.jksuci.2020.01.009>
18. Habib, A. S., Khan, S. U. R., & Felix, E. A. (2023). A systematic review on search-based test suite reduction: State-of-the-art, taxonomy, and future directions. IET Software, 17(2), 93–136. <https://doi.org/10.1049/sfw2.12104>
19. Habib, A. S., Khan, S. U. R., Hussain, S., Ibrahim, N., Nisa, H. U., & Yousafzai, A. (2024). A similarity-based multi-objective test optimization technique using search algorithm. Systems and Soft Computing, 6, 200164. <https://doi.org/10.1016/j.sasc.2024.200164>
20. Jeyaprakash, S., & Alagarsamy, K. (2015). A distinctive genetic approach for test-suite optimization. Procedia Computer Science, 62, 427–434. <https://doi.org/10.1016/j.procs.2015.08.437>
21. Kabir, M. N., Ali, J., Alsewari, A. R. A., & Zamli, K. Z. (2017). An adaptive flower pollination algorithm for software test suite minimization. 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), 1–5. <https://doi.org/10.1109/EICT.2017.8275215>
22. Kader, M. A., Zamli, K. Z., Mohamad, A. B., & Mojumder, M. A. (2025). A two-phase adaptive hybrid SCA-PSO strategy for efficient multi-task test redundancy reduction. 2025 IEEE 9th International Conference on Software Engineering and Computer Systems (ICSECS), 46–51. <https://doi.org/10.1109/ICSECS65227.2025.11279040>
23. Kang, H. Y., Raja Ikram, R. R., Zamli, K. Z., & Emran, N. A. (2025). Exploring multifactorial techniques in rat swarm optimization: Preliminary results. Engineering, Technology & Applied Science Research, 15(3), 23430–23435. <https://doi.org/10.48084/etasr.10690>
24. Khan, S. U. R., Lee, S. P., Ahmad, R. W., Akhunzada, A., & Chang, V. (2016). A survey on test suite reduction frameworks and tools. International Journal of Information Management, 36(6), 963–975. <https://doi.org/10.1016/j.ijinfomgt.2016.05.025>
25. Khan, S. U. R., Lee, S. P., Javaid, N., & Abdul, W. (2018). A systematic review on test suite reduction: Approaches, experiment quality evaluation, and guidelines. IEEE Access, 6, 11816–11841. <https://doi.org/10.1109/ACCESS.2018.2809600>
26. Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report EBSE-2007-01.
27. Lakshminarayana, P., & SureshKumar, T. V. (2021). Automatic generation and optimization of test case using hybrid cuckoo search and bee colony algorithm. Journal of Intelligent Systems, 30(1), 59–72. <https://doi.org/10.1515/jisys-2019-0051>
28. Mekeng, A. M. R., Zamli, K. Z., Zarlis, M., Mohamad, M. A., & Darusalam, U. (2026). Tie-breaking in social network search algorithm for software test redundancy reduction optimization problem. 2026 International Seminar on Intelligent Business and Edge-Computing Research (ISIBER), 494–499. <https://doi.org/10.1109/ISIBER68248.2026.11470036>
29. Mohanty, S., Mohapatra, S. K., & Meko, S. F. (2020). Ant colony optimization (ACO-Min) algorithm for test suite minimization. In Progress in Computing, Analytics and Networking (pp. 55–63). Springer. https://doi.org/10.1007/978-981-15-2414-1_6
30. Mojumder, M. A., Qasymeh, M., Zamli, K. Z., Mistarihi, M., & Kader, M. A. (2025). A relay-based hybrid PSO-Jaya algorithm for efficient test redundancy reduction. 2025 IEEE 9th International Conference on Software Engineering and Computer Systems (ICSECS), 209–213. <https://doi.org/10.1109/ICSECS65227.2025.11279093>
31. Nachiyappan, S., Vimaladevi, A., & SelvaLakshmi, C. (2010). An evolutionary algorithm for regression test suite reduction. 2010 International Conference on Communication and Computational Intelligence (INCOCCI), 503–508.
32. Panigrahi, S. S., Sahoo, P. K., Sahu, B. P., Panigrahi, A., & Jena, A. K. (2021). Model-driven automatic paths generation and test case optimization using hybrid FA-BC. 2021 International Conference on

- Emerging Smart Computing and Informatics (ESCI), 263–268.
<https://doi.org/10.1109/ESCI50559.2021.9396999>
33. Rahman, M., Zamli, K. Z., & Kader, M. A. (2024). Improved sparrow search algorithm for test redundancy reduction. *Lecture Notes in Electrical Engineering*, 1212, 33–45.
https://doi.org/10.1007/978-981-97-3847-2_4
34. Rahman, M., Zamli, K. Z., Kader, M. A., Sidek, R. M., & Din, F. (2024). Comprehensive review on the state-of-the-arts and solutions to the test redundancy reduction problem with taxonomy. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 35(1), 62–87.
<https://doi.org/10.37934/araset.34.3.6287>
35. Reda, N., Hamdy, A., & Rashed, E. A. (2022). Multi-objective adapted binary bat for test suite reduction. *Intelligent Automation & Soft Computing*, 31(2), 781–797. <https://doi.org/10.32604/iasc.2022.019669>
36. Rizwan, A., Zamli, K. Z., Jilani, A. A. A., Ghorri, M. R., & Masood, Z. (2025). Multitask strategy for test redundancy reduction optimization. 2025 IEEE 9th International Conference on Software Engineering and Computer Systems (ICSECS), 394–399. <https://doi.org/10.1109/ICSECS65227.2025.11279026>
37. Sahoo, R. K., Mohapatra, D. P., & Patra, M. R. (2016). A firefly algorithm based approach for automated generation and optimization of test cases. *International Journal of Computer Sciences and Engineering*, 4(8), 54–58.
38. Selvakumar, S., Manikumar, T., Kumar, A. J. S., & Latha, L. (2013). An assimilated approach of concept analysis and particle swarm optimization algorithm for effective test suite minimization. 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).
<https://doi.org/10.1109/ICCIC.2013.6724231>
39. Sharma, A., Jadhav, A., Srivastava, P. R., & Goyal, R. (2010). Test cost optimization using tabu search. *Journal of Software Engineering and Applications*, 3(5), 477–486.
<https://doi.org/10.4236/jsea.2010.35054>
40. Sheikh, R., Babar, M. I., Butt, R., Abdelmaboud, A., & Eisa, T. A. E. (2023). An optimized test case minimization technique using genetic algorithm for regression testing. *Computers, Materials & Continua*, 74(3), 6789–6806. <https://doi.org/10.32604/cmc.2023.028625>
41. Sidek, R. M., Zamli, K. Z., & Kader, M. A. (2025a). Preliminary study of Brown Bear and Roulette Wheel Brown Bear for test redundancy reduction optimization problem. 2025 IEEE 9th International Conference on Software Engineering and Computer Systems (ICSECS), 57–61.
<https://doi.org/10.1109/ICSECS65227.2025.11279267>
42. Sidek, R. M., Zamli, K. Z., & Mohamed, R. (2025b). Multi-task test redundancy reduction optimization problem based on the Brown Bear Algorithm. *Proceedings of the 14th International Conference on Software and Computer Applications (ICSCA)*, 120–125. <https://doi.org/10.1145/3731806.3731865>
43. Sugave, S. R., Patil, S. H., & Reddy, B. E. (2017). DDF: Diversity Dragonfly Algorithm for cost-aware test suite minimization approach for software testing. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 701–707. <https://doi.org/10.1109/ICCONS.2017.8250554>
44. Sun, J.-Z., & Wang, S.-Y. (2010). A novel chaos discrete particle swarm optimization algorithm for test suite reduction. *The 2nd International Conference on Information Science and Engineering*, 1–4.
<https://doi.org/10.1109/ICISE.2010.5689888>
45. Suri, B., Mangal, I., & Srivastava, V. (2011). Regression test suite reduction using a hybrid technique based on BCO and genetic algorithm. *Special Issue of International Journal of Computer Science and Informatics*, II(1). <https://doi.org/10.47893/IJCSI.2013.1113>
46. Turner, A. J., White, D. R., & Drake, J. H. (2016). Multi-objective regression test suite minimisation for Mockito. In *Search Based Software Engineering (LNCS 9962, pp. 244–249)*. Springer.
https://doi.org/10.1007/978-3-319-47106-8_19
47. Verma, A. S., Choudhary, A., & Tiwari, S. (2020). Test case optimization using Butterfly Optimization Algorithm. 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 704–709. <https://doi.org/10.1109/Confluence47617.2020.9058334>
48. Wei, W., Cheng, W., Ye, L., Xia, S., Wang, Y., Xing, Y., & Wang, X. (2022). A study of test suite reduction based on Ant Lion Optimizer. *Tehnički vjesnik*, 29(1), 246–251. <https://doi.org/10.17559/TV-20210807090754>

49. Yamuç, A., Cingiz, M. Ö., Biricik, G., & Kalıpsız, O. (2017). Solving test suite reduction problem using greedy and genetic algorithms. 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 1–5. <https://doi.org/10.1109/ECAI.2017.8166445>
50. Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: A survey. *Software Testing, Verification and Reliability*, 22(2), 67–120. <https://doi.org/10.1002/stvr.430>
51. Zamli, K. Z., Hassin, M. H. M., & Al-Kazemi, B. (2015). tReductSA – test redundancy reduction strategy based on simulated annealing. *Communications in Computer and Information Science*, 513, 223–236. https://doi.org/10.1007/978-3-319-17530-0_16
52. Zamli, K. Z., Safieny, N., & Din, F. (2018). Hybrid test redundancy reduction strategy based on Global Neighborhood Algorithm and Simulated Annealing. *Proceedings of the 2018 7th International Conference on Software and Computer Applications*, 87–91. <https://doi.org/10.1145/3185089.3185146>
53. Zamli, K. Z., Kader, M. A., & Mekeng, A. M. R. (2024). A population division based multi-task sine cosine algorithm for test redundancy reduction optimization. *Proceedings of the 2024 10th International Conference on Computer Technology Applications (ICCTA)*, 94–102. <https://doi.org/10.1145/3674558.3674571>
54. Zamli, K. Z., Kader, M. A., & Sidek, R. M. (2025). Multi-task test redundancy reduction optimization problem based on the Sine Cosine Algorithm. *Lecture Notes in Networks and Systems*, 1364, 189–201. https://doi.org/10.1007/978-981-96-5220-4_16
55. Zhang, Y.-N., Yang, H., Lin, Z.-K., Dai, Q., & Li, Y.-F. (2017). A test suite reduction method based on novel quantum ant colony algorithm. 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 825–829. <https://doi.org/10.1109/ICISCE.2017.176>