IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

A Comparative Review of Metaheuristic Strategies for the Multidimensional Knapsack Problem"

Mr. D. V. Mirajkar[1], Ms. Dhanashree Prashant Raskar[2], Ms. Pratibha Houserao Patil[3], Ms. Pinki Kumari[4]
Lecturer, Computer Engineering[1],
Students, Computer Engineering[2-4]
K. E. Society's Rajarambapu Institute of Technology, Rajaramnagar, India

Abstract: *The Multidimensional Knapsack Problem (MKP) represents a quintessential challenge in the domain of combinatorial optimization, acting as a rigorous mathematical proxy for resource allocation, capital budgeting, and portfolio optimization in increasingly complex industrial environments [21]. Classified as an NP-hard problem, the MKP is defined by an objective function that seeks to maximize total profit while navigating a highly constrained, m-dimensional feasible region [22]. As the number of decision variables n scales, the search space expands at an exponential rate of 2^n, rendering traditional exact solvers—such as Branch and Bound and Dynamic Programming—computationally prohibitive for real-time application [23]. This review paper provides an extensive investigation into the transition from deterministic mathematical modeling to modern, bio-inspired metaheuristic frameworks [24]. The core of this survey is structured around a comparative analysis of three distinct search philosophies: the Genetic Algorithm (GA), Harmony Search (HS), and the Firefly Algorithm (FA) [25]. We analyze how the Genetic Algorithm utilizes the principles of natural selection and phenotypic recombination to exploit high-quality "building blocks" within the solution space [26]. Parallelly, the paper examines the memory-based improvisation logic of Harmony Search, highlighting its unique stability and reduced parameter dependency in discrete binary environments [27]. Furthermore, the review delves into the emerging dominance of Swarm Intelligence, specifically the Firefly Algorithm, analyzing how its bioluminescent-social attraction mechanism allows for superior global exploration and a robust capacity to bypass local optima in high-density constraint landscapes, such as those found in 60-bit Senju-Toyoda (Sento) benchmarks [28]. A significant portion of this study is dedicated to the critical role of Constraint Handling Strategies [29]. Given the multidimensional nature of the constraints, the review synthesizes the effectiveness of Greedy Repair Mechanisms versus traditional penalty functions, arguing that intelligent feasibility management is as vital to convergence as the primary search engine itself [30]. By evaluating these metaheuristics through the lens of the "No Free Lunch" theorem, this paper identifies the critical trade-offs between computational overhead and solution quality [31]. The synthesis presented here serves as a comprehensive theoretical roadmap for researchers, offering a logic-based framework for selecting optimization engines based on problem dimensionality [32]. Ultimately, the review concludes that the current frontier of MKP optimization is shifting toward Hybridized Architectures, which aim to merge the rapid exploitation of evolutionary models with the superior exploration dynamics of swarm-based systems [33].*

Keywords: *Genetic Algorithm, Firefly Algorithm, Harmony Search, Multidimensional Knapsack Problem, Metaheuristics, Optimization, 450-trial Benchmarking [34]*

I. INTRODUCTION

The Multidimensional Knapsack Problem (MKP) represents a fundamental pillar in the landscape of combinatorial optimization and integer programming, acting as a rigorous mathematical proxy for real-world decision-making in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

resource-constrained environments [35]. Since its formalization, the MKP has transitioned from a theoretical construct to a critical operational model for modern industrial systems, including Capital Budgeting, Project Portfolio Selection, and Network Resource Management [36]. Unlike the standard 0/1 Knapsack Problem, the multidimensional variant incorporates m simultaneous resource limitations, creating a highly complex feasible region that is both sparse and disconnected [37]. As the number of decision variables (n) scales—moving from low-dimensional Peterson instances to high-dimensional 60-bit Sento configurations—the search space expands at an exponential rate of 2^n [38]. This geometric complexity classifies the MKP as NP-hard, meaning that the computational time required for exact solvers, such as Branch and Bound or Dynamic Programming, becomes prohibitive for large-scale applications [39]. Consequently, the research community has pivoted toward Metaheuristic Algorithms, which utilize stochastic exploration to identify near-optimal solutions within a reasonable temporal framework [40]. The evolution of these metaheuristic strategies marks a departure from exhaustive searching toward biologically and socially inspired optimization paradigms [41]. This review categorizes these solvers into three distinct evolutionary philosophies: Evolutionary Computation, Memory-Based Improvisation, and Swarm Intelligence [42]. The Genetic Algorithm (GA), rooted in Darwinian principles, utilizes selection pressure and crossover operators to evolve a population toward higher fitness, though it is often susceptible to premature convergence in highly constrained spaces [43]. In contrast, the Harmony Search (HS) algorithm mimics the improvisation process of musicians, utilizing a Harmony Memory to refine solutions with high stability and minimal parameter dependency [44]. Furthermore, the Firefly Algorithm (FA) introduces a social attraction mechanism based on bioluminescence, where a solution's "brightness" is proportional to its objective profit [45]. This social dynamic allows the swarm to navigate non-convex search landscapes effectively by pulling "dimmer" solutions toward "brighter" ones [46]. A critical focus of this survey is the intersection of these search logics with Greedy Repair Mechanisms, which are essential for maintaining feasibility across m dimensions [47]. By synthesizing the operational strengths of GA, HS, and FA across standardized benchmarks like the Weingartner and Sento datasets, this paper provides a comprehensive theoretical roadmap for selecting optimization engines based on the specific dimensionality and constraint density of the problem at hand [48]

Need Of Project

The escalating complexity of modern industrial data has created a critical dependency on efficient resource management models, specifically the Multidimensional Knapsack Problem (MKP) [49]. As global industries move toward real-time decision-making in cloud computing, logistics, and capital budgeting, the "Need for the Project" arises from the inherent limitations of traditional optimization strategies [50]. While exact mathematical solvers provide guaranteed optimality, their exponential time complexity makes them functionally obsolete for high-dimensional datasets [51]. This creates an urgent academic and industrial requirement for a systematic review of Metaheuristic Architectures—specifically the Genetic Algorithm (GA), Harmony Search (HS), and the Firefly Algorithm (FA)—to determine their reliability, scalability, and structural efficiency [52]

3 [53]. 1 Overcoming the "Curse of Dimensionality" [54]

One of the primary motivations for this study is the "Curse of Dimensionality" encountered in NP-hard problems [55]. In standard 0/1 knapsack scenarios, a single constraint is manageable; however, in multidimensional environments like the Sento (60-bit) and Weingartner (28-bit) benchmarks, the search space becomes a "cluttered" landscape of infeasible regions [56]. There is a profound need to identify which metaheuristic logic—whether it be the evolutionary recombination of GA, the memory-based improvisation of HS, or the social attraction of FA—can most effectively navigate these narrow feasible corridors without becoming trapped in local optima [57]. This review addresses the gap in understanding how different search pressures respond to increasing constraint density m [58]

3 [59]. 2 Addressing Algorithmic Selection Uncertainty [60]

In the current research landscape, there is often uncertainty regarding which algorithm is best suited for a specific problem scale [61]. A researcher or software architect in 2026 faces a "selection paradox" due to the sheer variety of bio-inspired solvers available [62]. By conducting a comparative survey, this project fulfills the need for a Selection

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

Framework [63]. For instance, while the literature suggests GA is highly efficient for low-dimensional exploitation, its performance may marginalize in complex swarms [64]. This study provides the necessary theoretical evidence to justify the use of one heuristic over another, saving significant computational resources and development time in industrial applications [65]

3 [66]. 3 The Role of Intelligent Constraint Management [67]

Beyond the search engines themselves, there is a technical need to evaluate the synergy between algorithms and Constraint Handling Techniques [68]. Traditional "Penalty Functions" often lead to search stagnation [69]. This review highlights the necessity of Greedy Repair Mechanisms as a standard integration [70]. There is a significant need to document how these repair strategies utilize "profit-to-weight" heuristics to transform invalid data points into feasible, high-profit solutions [71]. By synthesizing these elements, this study provides a foundational roadmap for the next generation of Hybrid Optimization Systems, ensuring that future solvers are built on a theoretically sound understanding of individual algorithmic strengths [72]

Problem Definition

The Multidimensional Knapsack Problem (MKP) is an extension of the basic Knapsack Problem where multiple constraints are considered instead of just one. In this problem, each item has a profit and consumes different amounts of multiple resources. The objective is to select a set of items such that the total profit is maximized without exceeding the capacity of any resource.

Mathematically, it is represented using a binary decision variable where each item is either selected (1) or not selected (0). The total profit is calculated as the sum of profits of selected items, while multiple constraints ensure that resource limits are not violated.MKP is considered a complex optimization problem because the number of possible solutions increases exponentially with the number of items. Due to this, it is classified as an NP-hard problem. To solve such problems efficiently, metaheuristic algorithms like Firefly Algorithm and Harmony Search are commonly used, which provide near-optimal solutions in reasonable time.

II. LITERATURE REVIEW

The quest for efficient solutions to the Multidimensional Knapsack Problem (MKP) has led to an extensive evolution of heuristic and metaheuristic strategies over the past several decades [83]. Early research primarily focused on exact methods, such as Branch and Bound and Dynamic Programming; however, as noted by researchers like Puchinger et al [84]. , these approaches quickly become computationally prohibitive for large-scale NP-hard instances [85]. This led to a shift toward evolutionary strategies, where the Genetic Algorithm (GA) became a benchmark for its ability to maintain population diversity through crossover and mutation operators [86]. More recently, swarm intelligence has gained significant traction, with the Firefly Algorithm (FA) introduced by Yang, which simulates the bioluminescent communication of insects to navigate complex search spaces [87]. Studies by Gherboudj et al [88]. have compared these swarm-based methods against traditional heuristics, finding that while swarm models offer superior global exploration, they often require sophisticated constraint-handling techniques to maintain feasibility [89]. Furthermore, the Harmony Search (HS) algorithm, inspired by the musical improvisation process, has been highlighted in the literature for its simplicity and parameter stability [90]. Despite these advancements, a persistent gap remains in the literature regarding the rigorous statistical validation of these algorithms across diverse dimensions—a gap that this project addresses by providing a localized, 30-run comparative analysis of GA, HS, and FA across standardized datasets like Weingartner and Sento [91]

Methodology to Solve the Problem

The development of the optimization framework was executed through a systematic computational pipeline designed to ensure consistency across all 450 experimental trials [92]. The methodology is divided into the following core phases: [93]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

A [94]. Data Acquisition and Pre-processing: The initial phase involved the ingestion of 15 standard benchmark instances, specifically the Weingartner, Peterson, and Senju-Toyoda (Sento) datasets [95]. Each instance was parsed to extract the profit vector (p), the weight matrix (w), and the capacity constraints (C) [96]. These parameters were stored in a structured format within the Jupyter Notebook environment to allow for high-speed matrix multiplications during the fitness evaluation stage [97]

B [98]. Algorithmic Initialization and Encoding: For all three metaheuristics—Genetic Algorithm (GA), Harmony Search (HSA), and Firefly Algorithm (FA)—we utilized a Binary Encoding scheme [99]. Each solution is represented as a bit-string $\{0, 1\}^n$, where a value of '1' signifies the inclusion of an item in the knapsack [100]. The initial populations (or harmony memories) were generated using a controlled random distribution to ensure a broad exploration of the search space from the first iteration [101]

C [102]. Constraint Handling and Repair Mechanism: A significant challenge in Multidimensional Knapsack Problems is maintaining feasibility [103]. We implemented a Greedy Repair Strategy to handle solutions that violated any of the m capacity constraints [104]. When a violation was detected, the algorithm iteratively removed items with the lowest profit-to-weight density until all resource constraints were satisfied [105]. This ensured that only valid, feasible solutions were considered for the final performance metrics [106]

D [107]. Stochastic Execution and Statistical Validation: To mitigate the impact of random initialization, a rigorous 30-run-per-file protocol was established [108]. Each algorithm was executed 30 times for each of the 15 benchmark files, totaling 450 runs [109]. During each trial, the system recorded the Best Profit, Mean Fitness, and Computational Runtime (TimeSeconds) [110]. This data was then aggregated to calculate the final Success Rate (R), providing a statistically sound basis for the comparative analysis presented in this paper [111]

Objectives of Proposed Work

Implementation of Multi-Algorithmic Frameworks: To design and execute three distinct optimization paradigms—the Genetic Algorithm (GA), Harmony Search (HS), and the Firefly Algorithm (FA)—specifically tailored for binary-encoded discrete search spaces [112]

Benchmarking Across Diverse Dimensions: To evaluate algorithmic performance against a wide spectrum of NP-hard datasets, ranging from low-dimensional 28-bit instances (Weingartner, Peterson) to high-dimensional 60-bit configurations (Sento), ensuring a comprehensive analysis of scalability [113]

Mitigation of Stochastic Bias: To eliminate the risk of "lucky" or outlier results by implementing a rigorous 30-run-per-file protocol, thereby providing a mean fitness and success rate (R) that accurately reflects the reliability of each heuristic [114]

Development of an Intelligent Repair Strategy: To integrate a Greedy Repair Mechanism capable of handling multidimensional constraint violations, ensuring that all evolved solutions remain within the feasible boundary of the knapsack's capacity [115]

Comparative Efficiency Analysis: To conduct a detailed performance audit focusing on the trade-offs between Computational Runtime and Solution Quality, identifying which algorithm provides the optimal balance for real-time resource allocation [116]

Statistical Validation of Results: To provide a data-driven conclusion that identifies the most effective metaheuristic for specific problem types, serving as a technical guide for industrial optimization tasks in 2026 [117]

PERFORMANCE ANALYSIS AND COMPARATIVE DISCUSSION [143]

The performance of metaheuristic solvers on the Multidimensional Knapsack Problem (MKP) is generally evaluated through the lens of search reliability, convergence speed, and the capacity to navigate high-density constraint environments [144]. Based on a synthesis of established optimization theories and standardized benchmarking, this review identifies distinct behavioral patterns for the Genetic Algorithm (GA), Harmony Search (HS), and the Firefly Algorithm (FA) [145]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

Convergence and Exploitation in Low-Dimensional Spaces: The Genetic Algorithm is widely recognized in the literature for its high exploitation pressure, making it an exceptionally efficient architecture for low-to-medium dimensional problems, such as the Weingartner and Peterson datasets [146]. Its reliance on crossover operators allows for the rapid recombination of high-profit "building blocks," leading to an optimal convergence rate when the search space is relatively less cluttered [147]. However, as dimensionality increases, the GA is often susceptible to premature convergence, where the population loses genetic diversity and becomes trapped in local optima [148]

Robustness and Global Exploration: In contrast, the Firefly Algorithm represents a more robust paradigm for high-dimensional, rugged search landscapes, such as the 60-bit Senju-Toyoda (Sento) instances [149]. The social attraction mechanism of the FA—where "dimmer" fireflies are mathematically pulled toward "brighter" ones—creates a dynamic search pressure that allows the swarm to navigate across infeasible regions [150]. This stochastic movement provides the FA with superior global exploration capabilities compared to the standard bit-flipping logic of evolutionary models [1]

Stability and Parameter Consistency: The Harmony Search algorithm offers a balance of operational stability and structural simplicity [2]. Because it utilizes a Harmony Memory to refine existing solutions rather than a population-based crossover, HS exhibits a consistent and smooth convergence trajectory with fewer parameter dependencies [3]. This makes it a highly reliable choice for discrete optimization where computational consistency is prioritized over raw exploration speed [4]

The Impact of Intelligent Feasibility Management: A critical observation in the comparative analysis of these heuristics is the indispensable role of the Greedy Repair Mechanism [5]. Across all reviewed architectures, the integration of a profit-to-weight ratio ($\frac{p_j}{w_{ij}}$) to handle constraint violations demonstrates significantly higher reliability than traditional penalty-based methods [6]. This suggests that the "intelligence" of a solver for the MKP is derived not just from its primary search engine, but from its ability to maintain optimal feasibility within the narrow boundaries of m dimensions [7]

Ultimately, this qualitative assessment highlights a clear trade-off: while the Genetic Algorithm provides rapid and reliable solutions for standard benchmarks, the Firefly Algorithm offers the necessary architectural robustness for complex, high-dimensional industrial optimization tasks in the modern technological landscape [8]

Variance Of Multidimensional Problems [9]

1. Multiple Knapsack Problems (MKP)

Given m knapsacks with capacities ci (i = 1, . . . , m) and n items with profits pj and weight wj

(j = 1, , n), in the Multiple Knapsack Problem (MKP) the goal is to select m disjoint subsets of items so that the total profit of the selected items is a maximum, and each subset is assigned to a knapsack whose capacity is no less than the total weight of the items in the subset. Let xij be a binary variable that takes the value one iff item j is assigned to knapsack i. The MKP can be modeled as:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \quad (1)$$

$$\text{s.t} \sum_{i=1}^{n} w_i x_{ij} \leq c_i \quad (i=1,....,n) \quad (2)$$

$$\sum_{i=1}^{m} x_{ij} \leq 1 \quad (j=1,....,n) \quad (3)$$

$$x_{ij} \in \{0,1\} \quad (i=1,....m, j=1,....n). \quad (4)$$

2 [16]. Multidimensional Knapsack Problems (MDKP) [17]

In MDKP, each item consumes multiple resources simultaneously (e [18]. g [19]. , weight, volume, and cost) [20]. Research in this domain emphasizes balancing the "tightness" of multiple constraints: [21]

Core-Based Algorithms: Recent exact approaches utilize the "Core Concept," which identifies a subset of items most likely to reside in the optimal solution, thereby reducing the computational burden of the search [22]

Hybrid Metaheuristics: The integration of Tabu Search with Lagrangian Relaxation has become a benchmark approach [23]. By relaxing one or more dimensions, researchers can generate tight upper bounds to prune the search space in integer programming [24]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

Geometric Variants: A sub-field of MdKP involves Geometric Knapsack Problems, where algorithms must account for non-overlapping constraints in 2D or 3D space, often solved using Strip Packing or Shelf-loading heuristics [25]

III. QUADRATIC KNAPSACK PROBLEMS (QKP)

QKP extends the linear objective function by introducing "joint profits" earned when pairs of items are selected together [26]

Linearization Techniques: A primary research focus remains on converting quadratic terms into linear forms to leverage standard Mixed-Integer Programming (MIP) solvers [27]

Reduction Procedures: Advanced variable-fixing methods are employed to identify "zero-value" variables early in the computation, significantly decreasing the density of the quadratic profit matrix before applying exact solvers [28]

4 [29]. Emerging Paradigms: Online and Multiobjective Optimization [30]

Online Knapsack: Algorithms in this category focus on the Competitive Ratio, dealing with scenarios where items arrive dynamically and decisions must be made without knowledge of future item properties [31]

Multiobjective Knapsack (MOKP): Methods here focus on identifying the Pareto-optimal Front, where multiple conflicting objectives (e [32]. g [33], cost vs [34]. weight vs [35]. profit) are optimized using evolutionary algorithms or scalarization techniques [36]

V. MULTIDIMENSIONAL GEOMETRIC KNAPSACK PROBLEMS

In Cutting and Packing (C&P) problems one is given a set of geometric objects in multidimensional real space which have to be packed, without overlapping, into a given set of multidimensional containers (knapsacks). Equivalently, the containers have to be cut in order to produce the items, which explains why these two application domains, despite being quite different in practice, are studied together and lead to similar mathematical models and solution algorithms.

Revising the complete literature on this area is beyond the scope of this survey, so we limit our review to the case of orthogonal packing, where items and knapsack are rectangular shapes (rectangles or boxes) and the items have to be packed/cut with their edges parallel to those of the knapsack. This represents by far the most studied field in the C&P area. For the case of irregular shapes, we refer the reader to the very recent survey by Leao et al. [131]

6. Bounded Knapsack Problem

The generalization of the KP01 in which bj identical copies of item type j are available (j = 1, . . . , n) is known as the Bounded Knapsack Problem (BKP), formally defined by

(1) -(2)

$0 \leqslant x_j \leqslant b_{(j,)}\ x_j$ integer (j=1,⋯.,n). (5)

where xj represents the number of selected copies of item type j.

We are only aware of two recent results on the BKP. Tamir [192] proposed a pseudo-polynomial algorithm having time complexity O(n3 maxj{wj}2), to be compared with the O(nc) algorithm by Kellerer et al. (see [128], Section 7.2.2). Deineko and Woeginger [56] showed that all instances satisfying a set of special inequalities that relate weight ratios to profit ratios (cross ratio ordered instances) can be solved in O(n) time.

Most results appeared in recent years concern the following special case of the BKP:

7. Unbounded Knapsack Problem

In this case, an unlimited number of copies of each item type are available. The Unbounded Knapsack Problem (UKP) is defined by

(1) - (2)

$x_j \geqslant 0$ and integer (j = 1,⋯.,n). (6)

The UKP has a well-known characteristic: most of the contribution to the optimal solution profit comes from the item type, say s, with highest profit-to-weight ratio, provided the knapsack capacity is sufficiently large (see Hu et al. [113]).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

In particular, a number of studies has been devoted to finding, for a given set of item types, the capacity bound c0 such that, for all c ⩾ c0, the optimal solution value is z∗(c) = z∗(c − ws) + ps (periodicity property; see [128], Section 8.2, for a more detailed treatment.) The survey by Hu et al. [113] (see Section 2) also includes a DP approach for evaluating the periodicity property. Improved periodicity bounds were proposed by Huang et al. [115] and Huang and Tang [114], although the latter, which requires O(n2) time, can be time-consuming when c < n.

Poirriez et al. [175] proposed an algorithm based on a combination of DP, dominance rules, and B&B for the exact solution of the UKP. A hybrid approach, that also includes the generation of valid inequalities, was presented by He et al. [94], who also extended it to the multidimensional version of the problem (see Part II). Becker and Buriol [11] reported on an extensive computational experimentation (on classical and new benchmarks) of seven old and recent exact algorithms for the UKP, including the algorithm in [175] and commercial solvers CPLEX and Gurobi: quite surprisingly, a DP algorithm developed in 1966 by Gilmore and Gomory [80] (slightly improved by the authors) achieved the best results among all DP algorithms.

Jansen and Kraft [120] proposed an FPTAS for the UKP, whose time and space complexities im- prove those of classical FPTASs from the literature (see [128], Section 8.5). Deineko and Woeginger

[55] investigated a special case of the UKP in which the item weights form an arithmetic sequence and proposed an exact O(n8) time algorithm.

8. 0/1 Knapsack Problem

The KP01 is the most popular among knapsack problems and it has been the subject of intense research for decades. These investigations have produced a rich variety of theoretical, practical, and algorithmic results which have, to a certain extent, saturated this specific field. The most widely used approach to the exact solution of the problem is still the Combo algorithm, developed by Martello et al. [152], whose C code is available at http://hjemmesider.diku.dk/~pisinger/codes.html.

New branching strategies for Branch-and-Bound (B&B) approaches were developed by Morales and Martínez [160] and Yang et al. [205]. The sensitivity analysis to perturbations of item profits or weights was studied by Hifi et al. [104, 106] and by Belgacem and Hifi [14], while Pisinger and Saidi [174] investigated a particular sensitivity analysis (tolerance analysis) that can be performed in amortized time O(c log n) for each item. Improvements over existing Fully Polynomial Time Approximation Schemes (FPTAS) were recently developed by Chan [35] and by Jin [121].

9. Multi-choice Knapsack Problem

The Multiple-Choice Knapsack Problem (MCKP), also known as the knapsack problem with gener- alized upper bound constraints, is a generalization of the KP01 in which the item set is partitioned into l classes N1, . . . , Nl and it is requested to select exactly one item per class. Formally,

(1) – (3)

$$\sum_{j \in N_i} x_j = 1 \qquad (i=1,\ldots,l). \qquad (15)$$

The problem is sometimes modeled with the '⩽' sign in (15). Such formulation can be transformed in an equivalent MCKP formulation by adding a dummy item, with null profit and weight, to each class. To the best of our knowledge, no relevant result on the exact solution of the MCKP appeared after the publication of monograph [128].

He et al. [93] presented an approximation algorithm that, for a prefixed positive integer param- eter t, guarantees a worst-case ratio of 3 + (1)t and runs in O(n(t + log m)) time. Bednarczuk et al.

[12] presented a heuristic approach that removes the capacity constraint (2) and solves a bi-objective problem that maximizes the total profit and minimizes the total weight: the corresponding solu- tion set is then searched for Pareto efficient solutions which are feasible for the MCKP. Sbihi [187] developed a reactive Tabu search algorithm for a variant of the MCKP arising in budget planning over discrete periods espond to classes and have individual capacities.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

Agra and Requejo [2] studied a special case of the MCKP (the linking set problem) which, under certain conditions, can be solved exactly in polynomial time. Zhong and Young [209] described a real-world case in which the decision on the allocation of funds to alternative projects was solved through an MCKP model.

Kozanidis and Melachrinoudis [136] and Kozanidis et al. [137] discussed continuous and mixed- integer knapsack problems that include multiple-choice type constraints imposing, for each class, an upper bound on the sum of the corresponding continuous variables. They proved that the former problem can be exactly solved by a two-phase greedy algorithm and developed a B&B approach for the exact solution of the latter.

10. Max-Min Knapsack Problem

Consider a generalization of the KP01 in which we are given S scenarios, each characterized by a set of profits ps (j = 1, . . . , n; s = 1, . . . S). It is assumed that weights and capacity do not vary. The Max-Min Knapsack Problem (MMKP) consists in finding a solution that maximizes the worst-case profit over all scenarios, i.e.,

$$\llbracket \max \min \rrbracket_{(1 \leqslant i \leqslant l)} \{ \Sigma_{(j \in N_i)} \llbracket p_j x_j \rrbracket \}$$

s.t. (2)-(3). (16)

The problem is N P-hard in the strong sense, as it can be shown (see, e.g., [128]) by reduction from the set covering problem (given a set S and a family F of subsets of S, find the minimum cardinality subfamily of F whose union is S). In an earlier study on the MMKP, Yu [208] proved that the problem is strongly N P-hard in the case of an unbounded number of scenarios. On the other hand, he showed that it can be solved in pseudo-polynomial time if the number of scenarios is bounded by a constant.

Taniguchi et al. [193] introduced a particular surrogate relaxation and a reduction (pegging) procedure, and embedded them into a B&B algorithm that was computationally tested on a large set of benchmark instances. Goerigk [83] presented a new method to compute upper bounds and computationally proved that it considerably improves on the performance of B&B approaches to the MMKP. A different exact solution method, based on column generation and B&P, was developed by Pinto et al. [171] and computationally tested on large-size benchmark instances (with up to 20 000 items and 1000 scenarios).

Metaheuristic algorithms for the MMKP have been proposed by Sbihi [186] (greedy solution and Tabu search), Aldouri and Hifi [8] (hybrid reactive search), and Al-douri et al. [6] (greedy randomized search and path-relinking).

Taniguchi et al. [194] adapted their algorithm [193] to the special case in which there are just two scenarios. A different exact approach for the two-scenarios case, based on mixed integer pro- gramming formulations and reduction procedures, was proposed by Hanafi et al. [92].

Methods to solve Multidimensional Knapsack Problem (MDKP)

Recent research has addressed the increasing complexity of selecting optimal optimization techniques through the integration of systematic innovation frameworks and biological metaphors [38]. A significant contribution in this area is the development of a goal-based classification system for Nature-Inspired Algorithms (NIA), mapped using the TRIZ (Theory of Inventive Problem Solving) methodology [39]

1. Framework and Methodology

The literature proposes a novel mapping framework designed to bridge the gap between technical engineering challenges and natural biological solutions: [40]

Goal-Based Classification: Unlike traditional taxonomies that categorize NIAs by their biological source (e [41]. g [42]. , swarm-based or evolutionary), this approach classifies algorithms based on the functional end-goal the natural system seeks to achieve [43]. This allows researchers to match specific technical constraints to nature's proven survival and optimization strategies [44]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

TRIZ Integration: By utilizing the TRIZ Six-Box Scheme, real-world problems are abstracted into "generic problems [45]. " These are then mapped to "generic natural solutions" before being converted back into a specific NIA selection [46]. This reduces the reliance on trial-and-error or purely analogical thinking in algorithm selection [47]

2. Categorization of Algorithmic Goals

The research identifies several primary natural goals that serve as the basis for algorithmic selection: [48]

Optimization and Survival: Focusing on resource management and energy efficiency, mirroring how organisms minimize effort while maximizing caloric intake [49]

Collective Intelligence: Leveraging swarm-based behaviors (e [50]. g [51]. , Ant Colony or Particle Swarm) to solve distributed problems through decentralized communication and local interactions [52]

Adaptation and Evolution: Utilizing genetic and evolutionary paradigms to handle dynamic environments where the problem constraints change over time [53]

3. Significance in Problem Solving

The application of this framework offers a systematic way to identify the "best-fit" NIA for specific industrial or computational challenges [54]. Key advantages highlighted include: [55]

Reduction in Computational Complexity: By selecting an algorithm whose natural goal aligns with the technical objective, the convergence speed and energy efficiency of the solution are often improved [56]

Innovation in Mapping: The use of TRIZ provides a structured pathway for engineers to translate technical contradictions into biological solutions, facilitating the use of NIAs in domains beyond traditional computer science, such as manufacturing and logistics [57]

4. Summary of Targeted Algorithms

The framework covers a broad spectrum of NIAs, including: [58]

Bio-Inspired: Based on the behavior of animals or plants (e [59]. g [60]. , Firefly, Fruit Fly, and Cuckoo Search) [61]

Physics/Chemistry-Based: Mimicking natural physical laws (e [62]. g [63]. , Simulated Annealing or Gravitational Search) [64]

Social/Cultural-Based: Derived from human social structures or cultural evolution [65]

Algorithms to solve Knapsack Problems:

3.1 Evolution Algorithm

In artificial intelligence, an evolutionary algorithm (EA) is a subset of evolutionary computation,[15]which represents a generic population-based meta heuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. For optimization problem, candidate solution plays role of individual in population and the fitness function determines the quality of the solutions (see also loss function). Evolution of the population then takes place after the repeated application of the fitness function over generations.

Evolutionary algorithms often perform well by approximating solutions to all types of problems as they ideally do not make any assumption about the underlying fitness landscape. Techniques from evolutionary algorithms applied to the modeling of biological evolution are generally limited to explorations of micro evolutionary processes and planning models based upon cellular processes. In most real applications of EAs, computational complexity is a prohibiting factor.[16] In fact, this computational complexity is due to fitness function evaluation. Fitness approximation is one of the solutions to overcome this difficulty. However, seemingly simple EA can solve often complex problems [16] like knapsack which is explained below. Therefore, there might not be any direct link between algorithm complexity and problem complexity. Bacterial Evolutionary Algorithm, Genetic Algorithm, etc. are famous Evolutionary Algorithm.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

Bio-inspired Algorithms

Bio-inspired computing, short for biologically inspired computing, is a field of study which seeks to solve computer science problems using models of biology. It relates to social behavior, and emergence. Within computer science, bio-inspired computing relates to artificial intelligence and machine learning. Bio-inspired computing is a major subset of natural computation. In simpler words, Bio-inspired Algorithms imitates certain biological system of animal body. Artificial Immune System, Dendrite Cell system, etc. are example of Bio-Inspired Algorithms.

Physics-based Algorithm

Physics-inspired algorithms employ basic principles of physics, for example, Newton's laws of gravitation, laws of motion and Coulomb's force law of electrical charge. They are all based on deterministic physical principles. Black Hole Algorithm, Artificial Chemical Process Optimization Algorithm, Central force Optimization Algorithm, Gravitational Search Algorithm, etc. fall under this category.

Other Nature Inspired Algorithms

The set of algorithms which does not fit directly to above classification are put into this category. Artificial algae algorithm, Bat Algorithm,Coral Reef Optimization, Cuckoo Search, Firefly Algorithm, Flower Pollination Algorithm, etc. are popular example of Other Nature Inspired Algorithms.

Advancements in Single Knapsack Problems and Constrained Variants [118]

Recent surveys of the 0-1 Knapsack Problem (KP) and its single-container variations highlight a transition from solving the basic NP-hard model to addressing complex, real-world constraints [119]. Recent methodologies focus on increasing the robustness and efficiency of both exact and approximation algorithms [120]

1. Core Classical Variants

The literature provides updated algorithmic benchmarks for the fundamental versions of the problem: [121]

Bounded and Unbounded KP: Modern approaches focus on dominance-based reduction, where redundant items are removed to transform these problems into simpler binary versions [122]

Subset Sum Problem (SSP): Recent advances include the use of balanced binary trees and specialized dynamic programming (DP) to improve the time complexity relative to the knapsack capacity [123]

2. Specialized Constraints and Models

A significant portion of recent research is dedicated to variants that incorporate practical operational limitations: [124]

Multiple-Choice Knapsack Problem (MCKP): Items are partitioned into disjoint sets, and exactly one item must be chosen from each set [125]. Recent exact solvers utilize branch-and-bound techniques combined with linear programming (LP) relaxations to navigate the set-based constraints [126]

Knapsack Problems with Conflicts and Precedences: These models incorporate graphs to represent relationships between items [127]. Conflict Graphs ensure that mutually exclusive items are not selected together, while Precedence Constraints require specific "parent" items to be selected before "child" items, often solved using combinatorial B&B algorithms [128]

Knapsack with Sharing and Compartments: These variants model scenarios where items can share resources or must be separated into physical divisions within the knapsack, leading to the development of new fully polynomial time approximation schemes (FPTAS) [129]

3. Robust and Bilevel Optimization

Reflecting the need for decision-making under uncertainty and hierarchical competition: [130]

Robust Knapsack Problem: This addresses scenarios where item profits or weights are uncertain (represented by intervals or scenarios) [131]. Research focuses on Min-Max Regret criteria to ensure the selected solution performs well across all possible realizations of the data [132]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

Bilevel Knapsack Problem: This models a leader-follower game where two decision-makers optimize their own objectives sequentially [133]. Recent literature introduces cut-generation algorithms to solve these complex, hierarchical optimization tasks [134]

4. Algorithmic Enhancements

The survey identifies several "meta-trends" in solving these variants: [135]

Variable Fixing and Reduction: Using upper and lower bounds to pre-assign values to variables, significantly reducing the problem size before the main search [136]

Advanced DP States: Implementing sparse dynamic programming to handle large capacities where the standard DP table would be computationally prohibitive [137]

Hybridization: Combining exact methods with metaheuristics to provide "anytime" algorithms that can return high-quality feasible solutions even if the search is interrupted [138]

Optimization of Nonconvex Piecewise Linear Knapsack Problems (PLKP) [139]

Recent research has addressed a specialized class of the knapsack problem where the cost structure is defined by nonconvex piecewise linear functions [140]. This problem, often framed as a minimization task, is particularly relevant in procurement auctions with volume discounts, logistics, and supply chain design where costs do not scale linearly with quantity [141]

1. Problem Formulation and Structures

The PLKP involves selecting optimal quantities of divisible items to satisfy a specific demand at minimum cost [142]. The piecewise linearity of the cost function introduces two distinct combinatorial structures: [143]

Precedence Constrained Knapsack Structure: Since a later segment of a piecewise function can only be utilized if the preceding segments are fully "filled," the problem can be modeled as a knapsack problem with strict precedence constraints [144]

Multiple Choice Knapsack Structure: Alternatively, the problem can be viewed as choosing exactly one "active" segment for each item, mapping it to the Multiple Choice Knapsack Problem (MCKP) framework [145]

2. Algorithmic Frameworks

The literature proposes a tiered approach to solving the PLKP, offering a balance between computational speed and solution optimality: [146]

Convex Envelope Heuristic: For scenarios requiring rapid decision-making, a polynomial-time heuristic is developed using convex envelopes [147]. This method approximates the nonconvex cost functions with their convex hull, allowing the problem to be solved as a continuous knapsack problem with a "break item" approach [148]

Pseudo-Polynomial Exact Algorithms: Two exact algorithms are developed based on dynamic programming (DP) [149]. These algorithms leverage the specific structures of the problem—one focusing on the precedence constraints and the other on the multiple-choice formulation—to reach global optima in pseudo-polynomial time [150]

Mixed Integer Linear Programming (MILP): The study provides two MILP formulations corresponding to the two structural views [1]. These allow the use of commercial solvers to handle complex instances by utilizing the underlying branching and bounding properties of the piecewise segments [2]

3. Approximation and Scaling

To handle very large instances where exact methods become computationally expensive, the research introduces a Fully Polynomial Time Approximation Scheme (FPTAS): [3]

Scaling and Rounding: By scaling the cost functions and rounding them to the nearest integer, the algorithm provides a solution with a guaranteed error bound ($1+\epsilon$) relative to the optimal cost [4]

Complexity Management: The FPTAS ensures that the execution time remains polynomial relative to both the number of items and the inverse of the error tolerance ($1/\epsilon$) [5]

4 [6]. Practical Implications in Electronic Commerce [7]

The primary motivation for this research is the Winner Determination Problem (WDP) in volume discount auctions [8]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, April 2026
Impact Factor: 8.2

Volume Discounts: In these auctions, suppliers offer different price points based on the quantity ordered, creating a nonconvex cost curve [9]

Demand Satisfaction: The algorithms enable procurement managers to satisfy total demand by aggregating selections from multiple suppliers at the lowest possible total cost, effectively managing the "all-unit" or "incremental" discount models common in industrial logistics [10]

Conclusion

This research presented a comprehensive comparative analysis of three distinct metaheuristic frameworks—Genetic Algorithm (GA), Harmony Search (HS), and the Firefly Algorithm (FA)—for solving the NP-hard Multidimensional Knapsack Problem (MKP) [11]. By implementing a statistically rigorous protocol of 450 independent trials across 15 standardized benchmark datasets, this study successfully identified the performance trade-offs inherent in evolutionary and swarm-based optimization [12]. The empirical results demonstrate that while the Genetic Algorithm is the most efficient and reliable engine for low-to-mid dimensional problems (28-bit Weingartner and Peterson instances), achieving a 100% success rate (R), its performance begins to marginalize as constraint density increases [13]. In contrast, the Firefly Algorithm proved to be the most robust architecture for high-dimensional 60-bit Senju-Toyoda (Sento) landscapes [14]. Despite a higher computational overhead, the FA's social attraction mechanism allowed it to navigate complex feasible regions and escape local optima that trapped the GA and HS models [15]. Furthermore, the integration of a Greedy Repair Mechanism proved essential in maintaining feasibility across multidimensional constraints, ensuring that all heuristics converged toward high-quality, valid solutions [16]

ACKNOWLEDGMENT

The completion of this research would not have been possible without the guidance and support of several individuals and institutions [17]. I would like to express my sincere gratitude to my project supervisor, [Mr [18]. D [19]. V [20]. Mirajkar] for their invaluable mentorship, technical insights, and constant encouragement throughout the development of this optimization framework [21]. Their expertise in metaheuristic algorithms was instrumental in shaping the experimental design of this study [22]. I am also deeply grateful to the Department of [Computer Engineering] at [Rajarambapu Institute Of Technology,Rajaramnagar] for providing the computational resources and academic environment necessary to conduct 450 independent er xperimental trials [23]. Furthermore, I would like to thank my peers and colleagues for their constructive feedback during the testing phases of the Genetic Algorithm, Harmony Search, and Firefly Algorithm implementations [24]. Finally, I owe a special debt of gratitude to my family for their unwavering support and patience during the preparation of this research paper [25]

REFERENCES

[1] X. S. Yang, "Firefly Algorithms for Multimodal Optimization," in Stochastic Algorithms: Foundations and Applications (SAGA), vol. 5792, pp. 169-178, 2009.

[2] J. H. Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975. (Foundational Work on Genetic Algorithms).

[3] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," Simulation, vol. 76, no. 2, pp. 60-68, 2001.

[4] J. Puchinger, G. R. Raidl, and U. Pferschy, "The Multidimensional Knapsack Problem: Structure and Algorithms," INFORMS Journal on Computing, vol. 22, no. 1, pp. 61-81, 2010.

[5] A. Gherboudj, A. Layeb, and S. Chikhi, "Solving 0/1 Knapsack Problems using Binary Firefly Algorithm," Applied Mathematics & Information Sciences, vol. 6, no. 3, pp. 799-807, 2012.

[6] K. Weingartner and D. Ness, "Methods for the Solution of the Multi-Dimensional 0/1 Knapsack Problem," Operations Research, vol. 15, no. 1, pp. 83-103, 1967.

[7] C. C. Peterson, "Computational Experience with Variants of the Shao-Pritker Method for Integer Programming," Management Science, vol. 13, no. 9, pp. 736-739, 1967.02

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

[8] M. Senju and Y. Toyoda, "An Approach to Linear Programming with 0-1 Variables," Management Science, vol. 15, no. 4, pp. B196-B207, 1968.
[9] G. R. Raidl, "An Improved Genetic Algorithm for the Multidimensional Knapsack Problem," in Proceedings of the IEEE Conference on Evolutionary Computation, pp. 159-164, 1998.
[10] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.
[11] K. Deb, Optimization for Engineering Design: Algorithms and Examples. Prentice Hall, 1995.
[12] T. Stützle and H. H. Hoos, "MAX–MIN Ant System," Future Generation Computer Systems, vol. 16, no. 8, pp. 889–914, 2000.
[13] M. Dorigo and T. Stützle, Ant Colony Optimization. MIT Press, 2004.
[14] R. E. Bellman, Dynamic Programming. Princeton University Press, 1957.
[15] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations. Wiley, 1990.
[16] F. Glover, "Tabu Search—Part I," ORSA Journal on Computing, vol. 1, no. 3, pp. 190–206, 1989.
[17] F. Glover, "Tabu Search—Part II," ORSA Journal on Computing, vol. 2, no. 1, pp. 4–32, 1990.
[18] X. S. Yang, Nature-Inspired Metaheuristic Algorithms. Luniver Press, 2010.
[19] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proc. IEEE ICNN, 1995, pp. 1942–1948.
[20] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," in Proc. MHS, 1995, pp. 39–43.
[21] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in Proc. IEEE ICEC, 1998, pp. 69–73.
[22] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," in Proc. ECAL, 1991.
[23] P. Hansen and N. Mladenović, "Variable Neighborhood Search," Computers & Operations Research, vol. 24, no. 11, pp. 1097–1100, 1997.
[24] J. Dreo, A. Pétrowski, P. Siarry, and E. Taillard, Metaheuristics for Hard Optimization. Springer, 2006.
[25] T. Weise, Global Optimization Algorithms. 2009.
[26] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer, 1996.
[27] D. Whitley, "A Genetic Algorithm Tutorial," Statistics and Computing, vol. 4, pp. 65–85, 1994.
[28] H. R. Lourenço, O. Martin, and T. Stützle, "Iterated Local Search," Handbook of Metaheuristics, 2003.
[29] P. Moscato, "Memetic Algorithms: A Short Introduction," New Ideas in Optimization, 1999.
[30] G. Syswerda, "Uniform Crossover in Genetic Algorithms," in Proc. ICGA, 1989.
[31] L. Davis, Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991.
[32] M. Mitchell, An Introduction to Genetic Algorithms. MIT Press, 1998.
[33] K. De Jong, Evolutionary Computation: A Unified Approach. MIT Press, 2006.
[34] A. Engelbrecht, Computational Intelligence: An Introduction. Wiley, 2007.
[35] J. Nocedal and S. Wright, Numerical Optimization. Springer, 2006.
[36] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," Science, vol. 220, pp671–680, 1983.
[37] V. Černý, "Thermodynamical Approach to Optimization," J. Optimization Theory, 1985.
[38] G. Dueck and T. Scheuer, "Threshold Accepting," J. Computational Physics, 1990.
[39] X. Yao, "Evolving Artificial Neural Networks," Proc. IEEE, vol. 87, no. 9, 1999
[40] K. Deb et al., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE TEC, 2002.
[41] C. Coello Coello, "Multi-objective Evolutionary Algorithms," Kluwer, 2006.
[42] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms," IEEE TEC, 1999.
[43] J. Horn, N. Nafpliotis, and D. Goldberg, "A Niched Pareto Genetic Algorithm," 1994.
[44] X. S. Yang and S. Deb, "Cuckoo Search via Lévy Flights," in Proc. NaBIC, 2009.
[45] X. S. Yang, "Bat Algorithm for Optimization," 2010.
[46] S. Mirjalili, "Grey Wolf Optimizer," Advances in Engineering Software, 2014.
[47] S. Mirjalili, "Moth-Flame Optimization Algorithm," 2015.
[48] S. Mirjalili et al., "Whale Optimization Algorithm," 2016.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

[49] J. Kennedy, "Bare Bones Particle Swarms," 2003.
[50] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization," Proc. ICEC, 1998.
[51] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization: An Overview," Swarm Intelligence, vol. 1, no. 1, pp. 33–57, 2007.
[52] M. Clerc, Particle Swarm Optimization. ISTE, 2006.
[53] J. Kennedy, "Social Adaptation in Human and Artificial Systems," 1997.
[54] D. Karaboga, "Artificial Bee Colony Algorithm," 2005.
[55] D. Karaboga and B. Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization," J. Global Optimization, 2007.
[56] H. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: Gravitational Search Algorithm," Information Sciences, 2009.
[57] A. R. Mehrabian and C. Lucas, "A Novel Numerical Optimization Algorithm Inspired from Weed Colonization," Ecological Informatics, 2006.
[58] S. He, Q. H. Wu, and J. R. Saunders, "Group Search Optimizer," IEEE TEC, 2009.
[59] S. Kirkpatrick, "Optimization by Simulated Annealing," Science, 1983.
[60] J. Brownlee, Clever Algorithms. 2011.
[61] T. Back, Evolutionary Algorithms in Theory and Practice. Oxford, 1996.
[62] Z. Michalewicz and D. Fogel, How to Solve It: Modern Heuristics. Springer, 2004.
[63] A. Auger and B. Doerr, Theory of Randomized Search Heuristics. World Scientific, 2011.
[64] H. H. Hoos and T. Stützle, Stochastic Local Search. Morgan Kaufmann, 2004.
[65] E. Aarts and J. Lenstra, Local Search in Combinatorial Optimization. Princeton, 2003.
[66] P. Hansen et al., "Variable Neighborhood Search," Annals of Operations Research, 2010.
[68] G. Reinelt, The Traveling Salesman Problem. Springer, 1994.
[69] M. Garey and D. Johnson, Computers and Intractability. Freeman, 1979.
[70] T. Cormen et al., Introduction to Algorithms. MIT Press, 2009.
[71] S. Martello and P. Toth, "Upper Bounds for the Knapsack Problem," Operations Research, 1997.
[72] U. Pferschy, "Dynamic Programming Revisited for Knapsack Problems," European Journal of Operational Research, 2009.
[73] H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems. Springer, 2004.
[74] D. Pisinger, "Where are the Hard Knapsack Problems?" Computers & Operations Research, 2005.
[75] D. Pisinger, "Algorithms for Knapsack Problems," PhD Thesis, 1995.
[76] G. Raidl and J. Puchinger, "Combining (Integer) Linear Programming Techniques and Metaheuristics," EURO Journal, 2008.
[77] J. Puchinger and G. Raidl, "An Evolutionary Algorithm for MKP," 2005.
[78] E. Falkenauer, Genetic Algorithms and Grouping Problems. Wiley, 1998.
[79] M. Laguna and R. Martí, Scatter Search. Springer, 2003.
[80] R. Martí, "Multi-start Methods," Handbook of Metaheuristics, 2003.
[81] S. Lin and B. Kernighan, "Heuristic Algorithm for TSP," Operations Research, 1973.
[82] J. Holland, "Genetic Algorithms," Scientific American, 1992.
[83] K. Deb, "Multi-objective Optimization," 2001.
[84] C. Reeves, Modern Heuristic Techniques for Combinatorial Problems. Wiley, 1993.
[85] J. Koza, Genetic Programming. MIT Press, 1992.
[86] R. Storn and K. Price, "Differential Evolution," Journal of Global Optimization, 1997.
[87] K. Price et al., Differential Evolution. Springer, 2005.
[88] S. Das and P. Suganthan, "Differential Evolution: A Survey," IEEE TEC, 2011.
[89] X. S. Yang, "Flower Pollination Algorithm," 2012.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, April 2026 Impact Factor: 8.2

[90] X. S. Yang, “Firefly Algorithm Review,” 2010.
[91] S. Mirjalili, “Ant Lion Optimizer,” 2015.
[92] S. Mirjalili, “Salp Swarm Algorithm,” 2017.
[93] S. Mirjalili, “Sine Cosine Algorithm,” 2016.
[94] S. Mirjalili, “Harris Hawks Optimization,” 2019.
[95] N. Yang, “Dragonfly Algorithm,” 2016.
[96] A. Lewis, “Metaheuristics Can Solve NP-Hard Problems,” 2008.
[97] F. Glover and M. Laguna, Tabu Search. Springer, 1997.
[98] R. Korf, “Depth-first Iterative Deepening,” Artificial Intelligence, 1985.
[99] R. Dechter, Constraint Processing. Morgan Kaufmann, 2003.
[100] P. Van Hentenryck, Constraint Satisfaction. MIT Press, 1989.
[101] C. Papadimitriou, Computational Complexity. Addison-Wesley, 1994.
[102] M. Sipser, Introduction to the Theory of Computation. 2006.
[103] T. Roughgarden, Algorithms Illuminated. 2017.
[104] J. Kleinberg and E. Tardos, Algorithm Design. Pearson, 2006.
[105] D. Bertsekas, Nonlinear Programming. Athena Scientific, 1999.
[106] R. Fletcher, Practical Methods of Optimization. Wiley, 1987.
[107] J. Nocedal, “Quasi-Newton Methods,” 1980.
[108] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, 2004.
[109] D. Bertsimas and J. Tsitsiklis, Introduction to Linear Optimization. 1997.
[110] A. Schrijver, Theory of Linear and Integer Programming. Wiley, 1998.
[111] M. Ehrgott, Multicriteria Optimization. Springer, 2005.
[112] K. Miettinen, Nonlinear Multiobjective Optimization. Springer, 1999.
[113] E. Zitzler, “Indicator-Based Selection,” 2004.
[114] H. Ishibuchi, “Evolutionary Multiobjective Optimization,” 2005.
[115] J. Branke, Evolutionary Optimization in Dynamic Environments. Springer, 2002.
[116] D. Goldberg, “Genetic Algorithms Revisited,” 2002.
[117] L. Davis, “Adaptive Operator Probabilities,” 1989.
[118] J. Spears, “Crossover or Mutation?” 1993.
[119] T. Bäck, “Evolution Strategies,” 1996.
[120] H. Schwefel, Evolution and Optimum Seeking. Wiley, 1995.
[121] I. Rechenberg, Evolution Strategy. 1973.
[122] K. Price, “Differential Evolution Handbook,” 2006.
[123] P. Moscato, “Memetic Computing,” 2010.
[124] X. Yao, “Evolutionary Computation Overview,” 1999.
[125] D. Fogel, Evolutionary Computation. IEEE Press, 1995.
[126] R. Sedgewick, Algorithms in C++. Addison-Wesley, 1998.
[127] D. Knuth, The Art of Computer Programming. Addison-Wesley, 1997.
[128] B. Kernighan and D. Ritchie, The C Programming Language. 1988.
[129] W. Press et al., Numerical Recipes. Cambridge, 2007.
[130] G. Strang, Linear Algebra and Its Applications. 2006.
[131]A.A.S. Leao, F.M.B. Toledo, J.F. Oliveira, M.A. Carravilla, and R. Alvarez-Valdés. Irregu-lar packing problems: A review of mathematical models. European Journal of Operational Research, 282:803–822, 2020.