



HIGHLIGHTS

- Systematic review of post 2021 Metaheuristic algorithms for multi hole drilling
- Comparative analysis of 39 studies focusing on ACO, Bees Algorithm, GA and GWO.
- Demonstrates that algorithm performance is highly dependent on problem scale
- Highlights physical validation and Reinforcement Learning as key future directions.

ARTICLE INFORMATION

Received Date August 7, 2026
Revision Date Sept 2, 2026
Accepted Date Sept 25, 2026
Publication Date Oct 6, 2026

Corresponding author :

Kürşat TANRIVER

kursat.tanriver@istun.edu.tr
+90 537 724 34 20

Cite this article :

N O. Gençay, Y. S. Demirel, K. Tanriveri M. Ay, "Tool Path Optimization Using Metaheuristic Algorithms: A Systematic Literature Review", *Journal of Science, Technology and Engineering Research*, No. 1, Oct 2026.
DOI: 10.53525/jster.2013127



Content of this journal is licensed under a
Creative Commons Attribution-Noncommercial
4.0 International License

Tool Path Optimization Using Metaheuristic Algorithms: A Systematic Literature Review

Necat Ozan Gençay¹  Yusuf Soner Demirel²  Kürşat Tanriver³  Mustafa Ay⁴ 

¹CNK Havacılık, İstanbul, Türkiye ORCID: 0009-0007-3154-4245, nogencay@gmail.com, +90 537 309 77 36

²CNK Havacılık, İstanbul, Türkiye ORCID:0009-0000-4353-1248 sdemirel@cnkltd.com, +90 542 306 82 21

³Istanbul Health and Technology University, Faculty of Engineering and Natural Sciences, Department of Mechatronics Engineering, İstanbul, Türkiye ORCID : 0000-0002-1723-4108

⁴Marmara University, Faculty of Technology, Department of Mechanical Engineering, İstanbul, Türkiye ORCID: 0000-0002-7672-1846 muay@marmara.edu.tr, +90 543 565 05 49

ABSTRACT

This study aims to systematically review the post-2021 literature on tool path optimization for multi-hole drilling processes, focusing on the metaheuristic algorithms most frequently applied to this combinatorial problem - namely Ant Colony Optimization (ACO), the Bees Algorithm, the Genetic Algorithm (GA), and the Grey Wolf Optimizer (GWO) - and to comparatively evaluate their relative performance, parameter-tuning burden, and hybridization trends across different problem scales. A systematic search was conducted through Google Scholar using keyword combinations related to tool path optimization, drilling sequence, and the corresponding algorithm families; studies published from 2021 onward that directly addressed hole or tool path optimization, or related Traveling Salesman Problem formulations, were included following a PRISMA-based selection process. A total of 39 application studies met the inclusion criteria and were classified according to the algorithm family employed, then synthesized within a comparative framework alongside nine founding papers used to define the underlying algorithms. Approximately 44% of the reviewed studies employed the Bees Algorithm or ACO, reflecting their maturity across both small-scale benchmarks and large-scale industrial scenarios, while the Grey Wolf Optimizer emerged as a rapidly growing alternative in energy-focused and laser micro-hole applications. No single algorithm consistently outperformed others across all problem scales: Simulated Annealing and the Bees Algorithm *performed strongly on small-to-medium regular problems, whereas Ant Colony Optimization and Dhouib-Matrix approaches proved more robust on large, irregular layouts, and hybridization of exploration and exploitation mechanisms emerged as the dominant research direction, with the best-performing hybrids reporting non-productive time reductions of up to 86.65%. Tool path optimization in multi-hole drilling remains an active, rapidly evolving field in which metaheuristic algorithms deliver measurable manufacturing efficiency gains; algorithm selection should be scale-dependent, and future research should prioritize physical validation and underexplored methods such as Cuckoo Search, the Shuffled Frog-Leaping Algorithm, and Reinforcement Learning.*

Keywords: Multi-Hole Drilling, Tool Path Optimization, Ant Colony Optimization, Bees Algorithm, Grey Wolf Optimizer

I. INTRODUCTION

Drilling is one of the most widely used manufacturing processes for machining metal and composite materials and plays a critical role across many industries, including automotive, aerospace, electronics, and general machine manufacturing. However, in the machining of multi-hole parts, a significant portion of total processing time is spent not on actual cutting, but on the non-productive distance the tool travels while moving from one hole to another (non-productive time / airtime); this non-productive time has been reported to account for as much as 70% of total processing time [1], [2], [3]. This ratio elevates the correct determination of drilling sequence from a mere technical detail to a strategic optimization problem that directly affects production cost, energy consumption, and machine utilization efficiency.

A wide variety of metaheuristic algorithms have been proposed in the literature to solve this problem efficiently. Ant Colony Optimization (ACO) [4], the Bees Algorithm [5], the Genetic Algorithm (GA) [6], the Grey Wolf Optimizer (GWO), Simulated Annealing (SA) [7], Particle Swarm Optimization (PSO) [8], Teaching-Learning-Based Optimization (TLBO) [9], Cuckoo Search [10], and the Shuffled Frog-Leaping Algorithm [11] are among the nature-inspired or evolutionary approaches most frequently employed in this domain. In recent years, next-generation approaches such as Reinforcement Learning and deterministic matrix-based methods (e.g., Dhouib-Matrix) have also begun to be applied to this problem class. What these algorithms have in common is their ability to produce high-quality solutions for combinatorial problems within acceptable time frames through decentralized, population-based search mechanisms.

In the literature, each of these algorithms has been addressed in different studies, at different problem scales, and through different validation methods (simulation, commercial CAM comparison, physical CNC/laser validation), yet the number of studies offering a comparative synthesis across algorithm families that encompasses the field as a whole remains limited. This study presents a systematic literature review addressing metaheuristic algorithms used for tool path optimization in multi-hole drilling processes in order to bridge this gap.

1.1 Problem Definition

The drilling path optimization problem is largely modeled in the literature as the Traveling Salesman Problem (TSP) and its variants; this formulation, in which holes are treated as "cities" and the path followed by the tool as a "tour," reduces the problem to a well-known instance of the combinatorial optimization domain [1], [2], [3], [13], [14], [15], [16]. Since the problem falls into the NP-hard class, exact solution methods (exhaustive enumeration, branch-and-bound, etc.) become computationally intractable as the number of holes increases, which drives the literature toward metaheuristic and swarm-intelligence-based approaches [17], [18].

Since holes in real industrial parts generally have different diameters and must be drilled with multiple tools in a specific sequence, the problem is often converted into a Precedence Constrained Traveling Salesman Problem (PCTSP) or similar sequence-constrained formulations [13], [19]. In some studies, the problem is instead addressed as a Multiple Traveling Salesman Problem (MTSP) in scenarios where multiple tools/machines operate simultaneously [20]. Recent studies have also begun to move the problem beyond the classical metaheuristic framework by modeling it as a Markov Decision Process (MDP) or a Reinforcement Learning environment [21], indicating that the field is increasingly opening up to new AI-based approaches. This diversity demonstrates that the problem cannot be reduced to a single standard formulation or a single algorithm family, and is instead highly application-dependent.

1.2 Aim and Contribution

In the existing literature, studies based on metaheuristic algorithms generally proceed as independent lines of research, with comparisons typically limited to two or three algorithms and to scales chosen by the proposing team. This situation makes it difficult to arrive at a generalizable picture of which algorithm family produces more reliable results at which problem scale and under which conditions.

The aim of this study is to systematically review the literature on tool path optimization in multi-hole drilling processes around Ant Colony Optimization, the Bees Algorithm, the Genetic Algorithm, the Grey Wolf Optimizer, and other current metaheuristic/hybrid

approaches, and to comparatively present the relative performance, parameter-tuning burden, and hybridization tendencies of these algorithm families across small/medium- and large-scale problems. The study's original contribution to the literature is threefold: (i) bringing together findings from 39 application studies published after 2021 within a single synthesis framework to evaluate the strengths and weaknesses of different algorithm families in the same context; (ii) classifying the reviewed studies according to the algorithm type employed and presenting a quantitative distribution to reveal where the field is concentrated; and (iii) identifying current research trends, application domains, and open research gaps in the field based on this evaluation.

II. METHOD AND PROCEDURE

2.1 Literature Search Strategy

The literature search for this study was conducted through the systematic collection and evaluation of academic sources related to the tool path optimization problem in multi-hole drilling processes. The search was primarily carried out via the Google Scholar database, using keyword combinations such as "tool path optimization," "multi-hole drilling," "drilling path optimization," "drilling sequence optimization," "ant colony algorithm drilling," "bees algorithm drill path," "genetic algorithm tool path," "grey wolf optimizer machining path," and "traveling salesman problem drilling." The publication date was restricted to 2021 and later in order to reflect current algorithmic developments.

2.2 Distribution of Reviewed Studies by Algorithm Type

The distribution of the 39 reviewed application studies according to the algorithm family employed is presented in Table 1. As shown in the Table 1, Bees Algorithm- and ACO-based studies alone account for approximately 44% of the 39 reviewed sources, confirming that these two algorithm families represent the most mature and frequently applied approaches to the multi-hole drilling path optimization problem. GWO has emerged as a rapidly rising alternative in recent years, particularly in energy-focused and laser micro-hole applications, while methods in the "Other Swarm Intelligence" category (Cuckoo Search, SFLA) along with Reinforcement Learning and Dhouib-Matrix approaches reflect the field's recent tendency to move beyond the classical metaheuristic framework.

Table 1. Distribution of Reviewed Studies by Algorithm Type

<i>Algorithm Family</i>	<i>Count</i>
<i>Ant Colony Optimization (ACO)</i>	8
<i>Bees Algorithm / Artificial Bee Colony (Bees/ABC)</i>	9
<i>Genetic Algorithm (GA)</i>	4
<i>Grey Wolf Optimizer (GWO)</i>	5
<i>Dhouib-Matrix (DM)</i>	2
<i>Simulated Annealing (SA)</i>	2
<i>Other Swarm Intelligence (Cuckoo Search, SFLA, hybrid SFLA-ACO)</i>	3
<i>Reinforcement Learning / AI-based Path Planning</i>	2
<i>Variable Neighborhood + Tabu Search (HVNTS)</i>	1
<i>Particle Swarm Optimization (PSO)</i>	1
<i>Teaching-Learning-Based Optimization (TLBO)</i>	1
<i>Joint Algorithm Review (PSO+ACO)</i>	1

2.3 Inclusion and Exclusion Criteria

The following criteria were applied when including sources in the review: (i) the study was published in 2021 or later; (ii) the study directly addressed hole/tool path optimization or a TSP/PCTSP/MTSP formulation directly related to this problem; (iii) the study employed ACO, the Bees Algorithm, GA, GWO, or another major metaheuristic algorithm used in this problem class as a method, or presented a comprehensive review of these algorithms; (iv) the study was published as a peer-reviewed journal article or conference paper. Based on these criteria, a total of 39 application studies were evaluated.

In addition, in order to define the algorithms discussed in Section 3 (Bees Algorithm, ACO, GA, SA, PSO, TLBO, Cuckoo Search, SFLA) accurately and in reference to primary sources, the

founding/seminal papers of these algorithms (published between 1993 and 2023, 9 in total) were exceptionally included in the reference list solely for theoretical definition purposes [4]-[12]. These 9 sources were not counted among the drilling/tool path application studies

that are the subject of the systematic review; they were used solely to provide conceptual background. One further reference [32] was likewise included outside the PRISMA-based selection process, used only as the conceptual/visual source for an original figure created in this study (Figure 5) and not counted as an application study. As a result, the reference list comprises a total of 49 sources.

2.4 Study Selection Process (PRISMA-Based)

The selection of studies included in this review followed a four-stage process consistent with the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework: identification, screening, eligibility assessment, and inclusion. The flow of records through these stages is summarized below.

Identification. An initial search conducted between July 10 and August 3, 2026, using the keyword combinations described in Section 2.1, returned a total of **224 records** from Google Scholar. **21 duplicates** were excluded, leaving **203 records** for screening.

Screening. The titles and abstracts of the 203 remaining records were screened against the inclusion criteria defined in Section 2.2. At this stage, **134 records** were excluded for reasons such as: falling outside the 2021–2026 publication window; addressing an unrelated domain (e.g., general path planning for mobile robots or unrelated scheduling problems with no connection to hole/tool path sequencing); or not employing a metaheuristic algorithm relevant to the scope of this review. This left **69 records** for full-text assessment.

Eligibility. The full texts of the remaining 69 records were assessed in detail. A further **30 records** were excluded at this stage for reasons including: absence of quantitative benchmark results (purely conceptual or theoretical proposals without simulation/experimental validation, apart from the single exception retained for methodological transparency, Taheri et al.); duplicate reporting of the same dataset across multiple publications by the same authors; or insufficient methodological detail to determine algorithm-specific outcomes.

Included. Following this process, **39 application studies** satisfied all inclusion criteria and were retained for the systematic review. As noted in Section 2.3, an additional **9 founding/seminal papers** on the core algorithms (Bees Algorithm, ACO, GA, GWO, SA, PSO, TLBO, Cuckoo Search, SFLA) were subsequently and exceptionally added to the reference list - outside the PRISMA-based selection process described above - solely to provide accurate, primary-source definitions of these algorithms in Section 3. This brings the total number of references in this study to 49.

The overall selection flow is summarized in Figure 1 below.

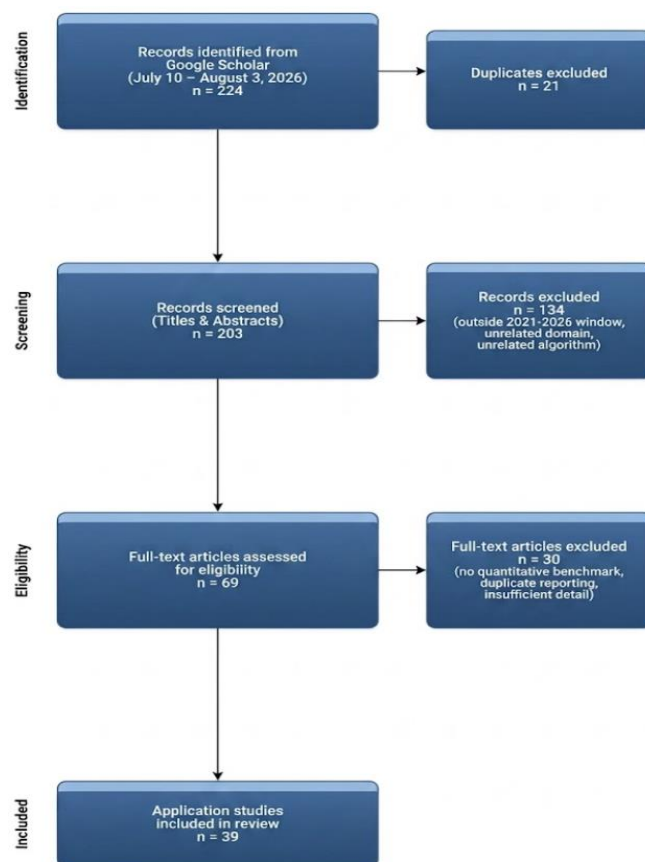


Figure 1. PRISMA flow diagram of the study identification, screening, eligibility, and inclusion process.

Table 2. Summary of Reviewed Articles

#	Author(s), Year	Algorithm	Test Scenario	Key Finding
1	Abidin et al. (2024)	Bees Algorithm	5x5-9x9 matrix	Competitive at small scale, declines at larger scale
2	Kamaruddin et al. (2021)	Bees Algorithm	14-hole PCB	Reached 280.0 mm optimum in 7/10 parameter sets
3	Rathod et al. (2024)	Enhanced BA (EBA)	14-hole PCB	Reached 280.0 mm optimum in all 10 parameter sets
4	Hoang et al. (2024)	ACO	34-52 hole sheet metal	32.77-34.77% shorter path vs. CAM
5	Trajković et al. (2026)	SA	25-hole multi-tool plate	13-47% path reduction
6	Rathod et al. (2023)	D-TLBO	14-158 hole, vs. SA	Outperformed SA at 158 holes
7	Rathod et al. (2022)	SA vs GA, PSO, ACO	5x5-11x11 matrix	SA best up to 9x9, then ACO
8	Olivari (2023)	ACO (AS/EAS/RAS+2-opt)	TSPLIB (eil51, eil101)	RAS+2-opt best result, lowest time
9	Yang, Huang & Ji (2026)	ACO+2-opt+elitist strategy	Rock drilling, up to 200 holes	7% improvement, crossings eliminated
10	Hoang et al. (2024)	GA	12-31 hole steel plate	24.63-11.45% reduction
11	Hamza et al. (2023)	BA + hybrid local search	TSPLIB MTSP	Reduced local optima trapping
12	Wan, Xu & Zuo (2023)	Reinforcement Learning (RL)	Complex cavity milling	27.33% advantage even at 3s
13	Abdullah et al. (2023)	ACO	9-operation part	Avg. 10.8% reduction in rapid travel
14	Wang et al. (2021)	HVNTS (VNS+Tabu Search)	5-axis EDM drilling (107 holes)	86.65% reduction in non-productive time
15	Hatem et al. (2021)	ACO+STEP-NC	13-hole STEP-NC part	16.96% reduction; time cut 254 -> 211s
16	Feng et al. (2022)	Improved ACO+NSGA-II	50-hole aluminum part	57.7% energy, 66.4% time reduction
17	Liu et al. (2024)	GWO review	Variant classification	Improved via hybrid/ensemble approaches
18	Makhadmeh et al. (2024)	GWO review	Recent developments	Robust metaheuristic skeleton
19	Pham et al. (2024)	BA application review	WEDM, disassembly, PCB, scheduling	New BA variants compiled
20	Sahin (2023)	CBA-NNM	24 TSPLIB (51-1002 cities)	<1% deviation up to 150 cities
21	Rathod et al. (2025)	CBA2 (2-parameter)	326-hole industrial panel	9.6% reduction, simpler tuning
22	Ibrahim et al. (2025)	ABC review	Variant/application survey	Strong in high-dimensional optimization
23	Taheri et al. (n.d.)	GA (theoretical)	Conceptual TSP-GA model	No numerical results, theoretical framework
24	Hatem et al. (2025)	GA	G-code optimization, open CNC	18% reduction, 5.3% better than ACO
25	Demir & Acar Vural (2024)	NSGA-II	Non-cutting path, physical CNC	~30% distance, ~7% cycle-time reduction
26	Zouari et al. (2024)	Dhouib-Matrix	20-2100 holes	Outperformed rivals in most scenarios
27	Jia et al. (2025)	Multi-objective GWO	14-hole drill-reaming	7.80% energy, 4.88% time reduction
28	Nasir et al. (2024/25)	GWO review	Energy systems (100+ papers)	Standard GWO 60% usage rate
29	Zhang et al. (2024)	Adaptive Discrete GWO	Laser micro-holes (90-649)	Up to 55.5% time reduction
30	Abidin et al. (2022)	PSO	10-150 hole PCB, physical CNC	Avg. 5% faster machining
31	Dhouib (2022)	Dhouib-Matrix-4 (DM4)	30-75 hole PCB	Shortest path in all scenarios

32	Zhang & Yang (2022)	Discrete Cuckoo Search (RW-DCS)	TSPLIB + glass-cutting path	38-59x more stable than rivals
33	Dalavi et al. (2022)	Modified SFLA	Rectangular matrix (4x5-20x20)	Shorter paths than GA and ACO
34	Nasir, Umer & Asgher (2022)	Hybrid SFLA-ACO	24-hole omega plate	5.9% better than commercial software
35	Mamdouh et al. (2025)	AI (Dijkstra+A*+RRT+2-opt)	Fully automated PCB drilling	13x efficiency vs. auto-router
36	Tanriver & Ay (2024)	Hybrid GA-ACO (GACO)	25-hole CFRP panel	3.16% improvement, 6.46% slower
37	Blum (2024)	Bibliometric review	25,993 Scopus documents (ACO)	China ~50% share, hybridization trend
38	Ang & Ng (2023)	BA+TRIZ operators	50-component PCB assembly	8.16% faster than best non-Bees
39	Priyadarshi & Kumar (2025)	PSO/ACO review	Case studies	Both algorithms have distinct strengths/weaknesses

2.5 Data Extraction and Quality Appraisal

For each of the 39 included application studies, the following information was extracted and tabulated (Table 2): the publishing author(s) and year, the primary algorithm or algorithm family employed, the test scenario (benchmark type and scale), and the key quantitative or qualitative finding reported. Because the reviewed studies employ markedly different benchmark problems - ranging from standard 14-hole PCB layouts to industrial panels with over 2000 holes - and different performance metrics (path-length reduction, processing-time reduction, energy consumption, and deviation from optimum), a narrative synthesis approach was adopted rather than a quantitative meta-analysis; studies were grouped first by algorithm family (Sections 3–4) and then compared qualitatively across problem scale to identify consistent performance patterns. To keep this scale-based comparison consistent throughout the review, three problem-scale bands are used: small scale (≤ 30 holes), medium scale (31–150 holes), and large scale (> 150 holes); these bands, rather than a fixed numeric cutoff, are applied qualitatively, since several reviewed studies test across a range that spans more than one band.

Given the applied engineering nature of this literature - in which standardized clinical-style risk-of-bias instruments (e.g., Cochrane RoB) are not directly applicable - the quality of the included studies was appraised qualitatively along three dimensions: (i) validation approach, distinguishing studies verified only in simulation from those additionally validated on physical CNC, laser, or EDM equipment; (ii) benchmark transparency, i.e., whether the test instance (hole count, layout, coordinates) and

comparison baseline (commercial CAM software, a named alternative algorithm, or a theoretical bound) were clearly specified; and (iii) reporting completeness, i.e., whether quantitative results were provided rather than purely conceptual claims. One study (Taheri et al.) that presented only a theoretical algorithmic framework without quantitative validation was nonetheless retained, as an explicitly noted exception, for its methodological transparency in framing the constrained TSP formulation; all other 38 application studies reported at least one quantitative performance metric.

III. THEORETICAL BACKGROUND

3.1 Travelling Salesman Problem and Its Relevance to Drilling Path Optimization

The Traveling Salesman Problem (TSP) is one of the most fundamental and extensively studied problems in combinatorial optimization. In its classical definition, given a set of cities and the distances between them, TSP seeks the shortest tour that visits each city exactly once and returns to the starting point. As the number of cities increases, the number of possible routes grows factorially, placing the problem in the NP-hard class; for example, for a problem with just 20 cities, the number of possible routes reaches approximately $19!$ (About 1.2×10^{17}), making it computationally intractable to solve within a reasonable time using exact methods [13], [22], [1], [2], [14].

The connection between the multi-hole drilling problem and TSP is direct: each hole location on a part is modeled as a "city," and the tool's movement from one hole to

another is modeled as a "traveling salesman's route"; the objective is to find the route that visits every hole exactly once while minimizing the total non-cutting/airtime distance [1], [2], [3], [15], [16]. Through this analogy, the rich body of metaheuristic solutions developed for TSP over decades (ACO, the Bees Algorithm, GA, SA, etc.) can be directly adapted to the drilling path optimization problem (Figure 2).

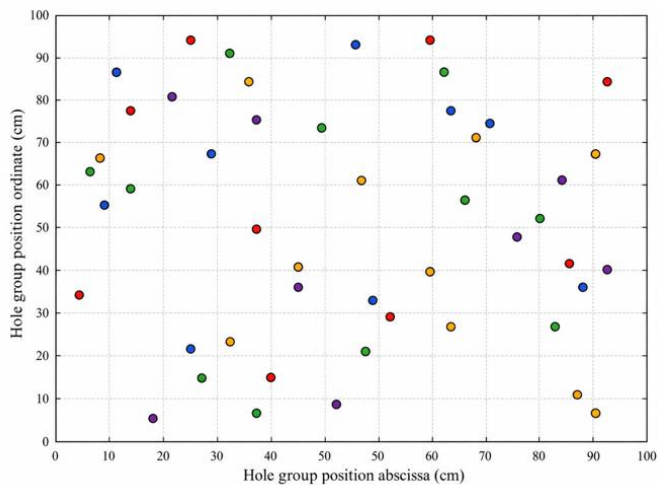


Figure 2. Author-created schematic representation of hole distribution and its modelling as a TSP, based on the problem described in [25].

However, in real industrial applications, the problem can become more complex than pure TSP. When holes have different diameters and must be drilled with multiple tools in a specific sequence, the problem is often converted into a Precedence Constrained Traveling Salesman Problem [13], [19], [14]. In scenarios where multiple tools or machines operate simultaneously, the Multiple Traveling Salesman Problem (MTSP) formulation is used instead [20]. While most studies adopt the standard Euclidean distance for distance calculations, some studies prefer the Chebyshev distance matrix on the grounds that it more realistically reflects the simultaneous X-Y axis movement of CNC/robot end-effectors [17], [23]. This diversity demonstrates that TSP in the multi-hole drilling literature does not correspond to a single fixed template but rather to a flexible mathematical framework that is shaped according to the application.

3.2 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a swarm intelligence algorithm proposed by Dorigo, Maniezzo, and Colnari in 1996 under the name "Ant System" (AS), inspired by the foraging behavior of ants finding the shortest path between a food source and their nest [4]. In nature, as ants move, they leave a chemical trail called pheromone; since

shorter paths are traversed more frequently, pheromone density on these paths increases over time, and the rest of the colony statistically tends to follow these denser pheromone trails. As a result, initially random ant movements converge around the shortest path through a collective, decentralized learning process [4]. The pheromone-based path selection mechanism of real ants, which forms the biological basis of ACO, is illustrated in Figure 3 [4].

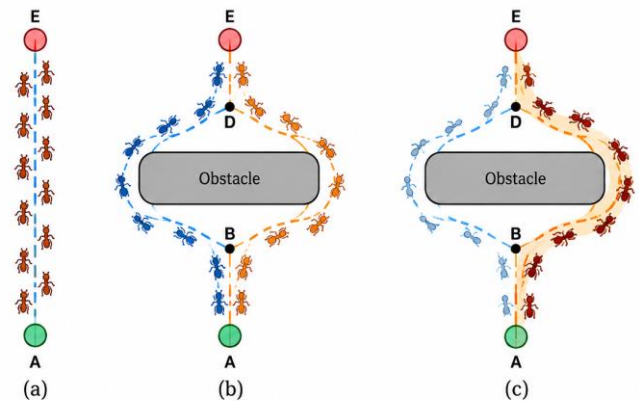


Figure 3. An example with real ants. (a) Ants follow a path between points A and E. (b) An obstacle is interposed; ants can choose to go around it following one of the two different paths with equal probability. (c) On the shorter path more pheromone is laid down. Author-created schematic representation of mechanism of ants, based on the concept described in [4].

ACO adapts this biological mechanism to combinatorial optimization problems using artificial agents (artificial ants) and digital pheromone matrices. In each iteration, the algorithm repeats the following steps: (i) each ant constructs a solution (route) using a probabilistic rule based on a weighted combination of pheromone intensity and heuristic/visibility information, (ii) pheromone trails are updated according to the quality of the constructed solutions (with better solutions receiving more pheromone), and (iii) old pheromone trails gradually evaporate over time to prevent premature convergence; a "tabu list" mechanism is also used to prevent nodes from being revisited within the same tour [4]. The original study proposed three variants - ant-cycle, ant-density, and ant-quantity [4]; subsequent literature has developed numerous additional variants, including Elitist Ant System (EAS), which grants extra pheromone reward to the best solution, Rank-based Ant System (RAS), which ranks ants and allows only the best to deposit pheromone, and Max-Min Ant System (MMAS), which bounds the search space between the best and worst solutions [17], [18].

When applied to the multi-hole drilling problem, ACO treats hole locations as "cities" and evaluates the routes constructed by ants as candidate tool paths; the shortest/most efficient routes accumulate more pheromone over time, guiding the algorithm's convergence toward these routes [13], [22], [24]. The most frequently criticized weakness of ACO in the literature is quality fluctuation and path crossings caused by randomness, particularly in large and irregular problems; this issue is typically addressed through local search operators such as 2-opt/Or-opt or adaptive elitist pheromone-update strategies [17], [18], [25].

An important distinction: Unlike genetic algorithms, ACO does not "evolve" a population (it does not use crossover/mutation); instead, it learns through indirect communication (stigmergy) via a shared memory (the pheromone matrix). This characteristic makes ACO a natural and effective choice particularly for graph/path-based problems (TSP, routing, scheduling), though it is less directly applicable to continuous optimization problems compared to methods such as the Bees Algorithm or GWO [26], [27].

3.3 Bees Algorithm/Artificial Bees Colony (Bees/ ABC)

The Bees Algorithm (BA) is another swarm intelligence method inspired by the foraging behavior of honeybee colonies. In nature, scout bees fly in random directions to discover flower patches in the surrounding area; upon returning to the hive, they perform a "waggle dance" whose duration and intensity are proportional to the quality (nectar richness) of the source found, thereby recruiting other worker bees to that area. Higher-quality sources receive more bees, while weaker sources are gradually abandoned [28]. The fundamental working principle of the algorithm is illustrated in Figure 4 [5].

BA translates this mechanism into two fundamental search processes: global search (exploration) is achieved through the random scanning of the search space by scout bees, while local search (exploitation) is achieved by directing a number of worker bees proportional to quality toward the best (elite) and good (best) sites found, allowing detailed scanning of the neighborhood of these sites. The balance between these two mechanisms enables the algorithm to both explore a wide area without getting stuck in local optima and conduct in-depth search in promising regions [1], [5].

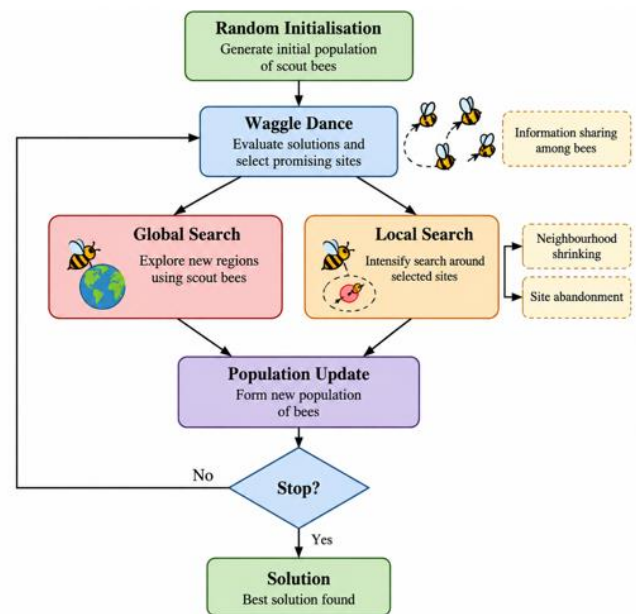


Figure 4. Author-created schematic representation of the basic Bees Algorithm workflow, based on the algorithm described in [5].

When applied to the multi-hole drilling problem, candidate hole sequences (candidate routes) are treated as "flower patches" for BA; the algorithm conducts an intensive local search in the neighborhood of sequences that yield the shortest route, using operators such as swap (exchanging the order of two holes), insertion (moving a hole to another position), or reversion (reversing a sub-sequence) [20], [29], [3]. The most frequently highlighted strength of BA in the literature is its ability to produce near-optimal results with high stability, particularly on small and medium-scale problems (e.g., the standard 14-hole PCB benchmark) [2], [3]. Conversely, the most notable weakness of the algorithm is that its classical form requires tuning a large number of parameters (a total of seven), and its performance relatively deteriorates as the search space grows (i.e., as the number of holes increases) [1], [5]. To address these two weaknesses, the literature has developed approaches that reduce the number of parameters to two [30], as well as variants such as the Enhanced Bees Algorithm (EBA), which employs multiple local search operators [3].

An important distinction: The Bees Algorithm is frequently confused with Artificial Bee Colony (ABC); although both are inspired by honeybee behavior, ABC uses a different architecture that divides bees into three roles - employed, onlooker, and scout - whereas the Bees Algorithm relies on the elite-site/best-site structure described above; in the literature, both methods are treated within the same swarm intelligence family and are sometimes used comparatively in the same studies [1], [31].

3.4 Genetic Algorithm (GA)

The Genetic Algorithm (GA) is an evolutionary computation method inspired by Darwin's principle of natural selection, operating not on the parameters themselves but on their coded representations (chromosomes, genes, alleles) [6]. The evolutionary cycle of the algorithm consists of the following steps: initialization, evaluation, selection (roulette wheel, tournament, etc.), crossover (single-point, two-point, uniform, etc.), mutation, and replacement [6]. The basic workflow of this evolutionary process is illustrated in Figure 5. In each generation, a fitness function that measures solution quality is evaluated; individuals with higher fitness gain an advantage in terms of the probability of being carried forward to the next generation. The literature emphasizes that the algorithm's success depends on the correct identification and recombination of "building blocks" - the sub-components of solutions - and that individually simple selection and mutation operators, when combined correctly, form powerful and efficient search systems [6].

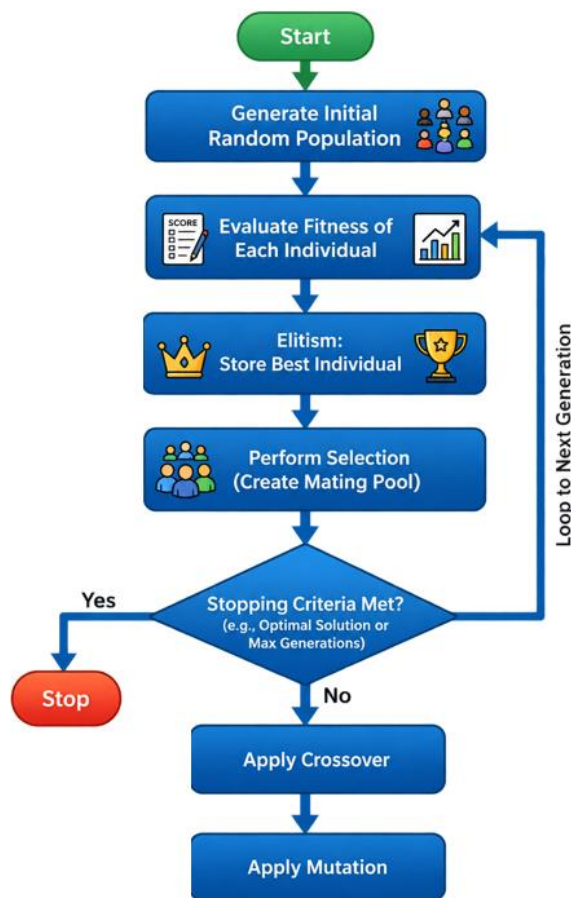


Figure 5. Basic flow chart of the Genetic Algorithm, created by the authors based on the concept described in [32].

When applied to the multi-hole drilling problem, each chromosome represents a hole sequence (route); the sum

of Euclidean distances between hole locations is used as the fitness function [19], [33]. The crossover operator typically combines sub-sequences of two parent routes to produce new routes, while the mutation operator randomly swaps the positions of two holes to preserve population diversity and prevent premature convergence [19], [34]. A key strength of GA highlighted in the literature is its flexible structure, which allows different constraints (tool changes, precedence order, collision avoidance) to be easily incorporated into the fitness function [33]; GA can also be naturally extended to multi-objective optimization -particularly with variants such as NSGA-II (Non-dominated Sorting Genetic Algorithm-II), which enables the simultaneous optimization of multiple objectives such as path length, smoothness, and energy consumption [25], [35].

The most frequently criticized weakness of GA is that computation time increases as population size and generation count grow, and it tends to converge more slowly than other swarm intelligence methods (particularly ACO and DM) on large-scale problems [19], [36].

3.5 Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a relatively new swarm intelligence algorithm proposed by Mirjalili and colleagues in 2014 [12], inspired by the hierarchical social structure and hunting strategy of grey wolves in nature. The leadership hierarchy within a wolf pack is divided into four levels: alpha (α) represents the best solution, beta (β) the second-best, and delta (δ) the third-best solution; all remaining individuals are designated omega (ω), and during the search process they perform encircling movements around the prey (optimum point) by tracking the positions of the alpha, beta, and delta wolves [26], [27], [37], [38]. This hierarchical structure of the GWO, consisting of alpha, beta, delta, and omega wolves, is illustrated in Figure 6.

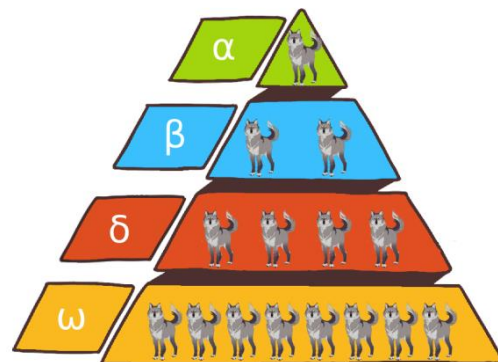


Figure 6. GWO hierarchical categories. Reproduced from [27] under the Creative Commons Attribution 4.0 International (CC BY 4.0) License.

The hunting process of GWO is modeled in three phases: tracking and approaching the prey, encircling the prey, and attacking the prey; these three phases are governed by a mathematical coefficient that controls the transition between global search (exploration) and local search (exploitation) [38], [26]. Through this mechanism, GWO explores a wide area early in the search process while focusing on precise local refinement around the best regions found as the search progresses.

When applied to multi-hole drilling and related precision machining problems, GWO has produced strong results particularly in energy-focused multi-objective optimization [37] and laser machining of irregular micro-hole patterns [39], scenarios where precision and efficiency are simultaneously important. To adapt to discrete problems, GWO's position-update mechanism is redefined using combinatorial operators such as Hamming-distance-based exchange, shift, and 2-opt instead of continuous space [39]. Review studies in the literature indicate that GWO has been rapidly adopted since 2014 due to its simple structure, small number of tunable parameters, and fast convergence, although its multi-objective versions have not yet matured to the level of ACO and BA [38], [27].

3.6 Other Metaheuristic and Hybrid Approaches

In addition to the four main algorithm families discussed above (ACO, Bees/ABC, GA, GWO), a wide variety of other metaheuristic methods have also been applied in the literature to the multi-hole drilling and related tool path optimization problem.

Simulated Annealing (SA) is a probabilistic global optimization method that mimics the physical process of a solid being slowly cooled and "freezing" into its minimum-energy configuration [7]. The algorithm proceeds using a finite set of states, a cost function, a neighborhood structure, and a gradually decreasing cooling schedule; its fundamental mechanism is that it accepts cost-increasing moves with a certain probability in order to avoid becoming trapped in local minima [7]. It has been theoretically shown that when a sufficiently slowly (logarithmically) decreasing cooling schedule is used, the algorithm is guaranteed to converge to the global optimum [7]. When applied to the multi-hole drilling problem, the hole sequence is formulated as a Traveling Salesman Problem [14], [16]; the literature reports that SA exhibits high stability particularly on small-to-medium-scale matrix problems (up to 81 holes), but struggles to maintain the quality-time balance as problem scale increases [16].

Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart in 1995, inspired by social behavior simulations of bird flocks and fish schools [8]. In the algorithm, candidate solutions flying through a multi-dimensional problem space are defined as "particles"; each particle updates its velocity and direction using simple mathematical operators and stochastic variables toward both its own best-known position (pbest) and the global best position found by the entire swarm so far (gbest) [8]. Since PSO was originally designed for continuous optimization, its application to discrete problems such as TSP requires a topological sorting procedure to convert particle positions into permutation sequences [40].

Teaching-Learning-Based Optimization (TLBO) was proposed by Rao, Savsani, and Vakharia in 2011, inspired by the teacher-student interaction in a classroom environment, and unlike other metaheuristic methods, it does not require algorithm-specific parameters that are difficult to tune (such as crossover rate or inertia weight) [9]. In the "Teacher Phase," the best solution in the population (the teacher) brings the overall knowledge level of the class closer to its own level, while in the "Learner Phase," students improve through random mutual interaction and knowledge exchange [9]. To be applicable to discrete problems such as TSP, standard TLBO is converted into "Discrete TLBO" using integer sequence encoding and custom crossover/mutation operators [15].

Dhouib-Matrix (DM) and its variants represent a relatively new approach that, unlike other nature-inspired methods, relies entirely on deterministic/semi-deterministic matrix calculations. The method scans the search space using a structural technique (DM-TSP1) and then exploits the identified regions using an original local search method called "Far-to-Near" (FtN); some variants additionally use tabu-list memory to prevent cyclical repetitions [36], [41]. The literature reports that DM variants produce markedly superior and stable results compared to nearly all other metaheuristic methods, particularly on very large-scale problems [36]. The general structure of the DM3 approach, including the DM-TSP2 procedure, the Far-to-Near local search, and the iterative modification of the parameter K, is illustrated in Figure 7.

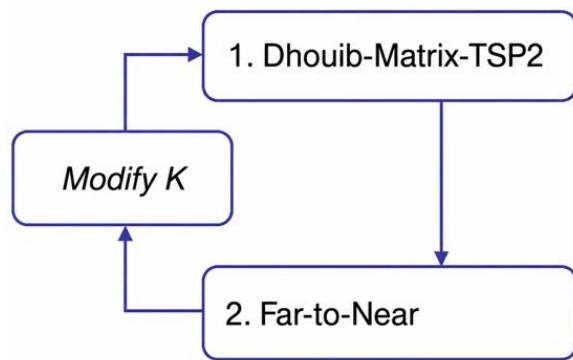


Figure 7. General structure of DM3, created by the authors based on the concept described in [36].

Cuckoo Search (CS) was proposed by Yang and Deb in 2009, blending the brood parasitism strategy of certain cuckoo species with a heavy-tailed random walk model known as "Lévy flight" [10]. The algorithm relies on three idealized rules: each cuckoo lays a solution in a randomly chosen nest, high-quality solutions carry over to the next generation, and the host bird may discover the foreign solution with a certain probability and abandon the nest [10]. Since the original CS was designed for continuous optimization, its application to discrete problems such as TSP requires replacing the Lévy flight mechanisms with discrete-space equivalents such as a local adjustment operator and discrete random walk; for large-scale problems, the method is typically combined with clustering techniques such as k-means to first partition the problem into subgroups [42]. The overall workflow of the discrete cuckoo search algorithm based on random walk and cluster analysis (RW-DCS) is illustrated in Figure 8.

The Shuffled Frog-Leaping Algorithm (SFLA), proposed by Eusuff, Lansey, and Pasha, is a metaheuristic inspired by the foraging behavior of frogs combined with the concept of cultural (memetic) information transfer [11]. The population is divided into subgroups called "memeplexes"; local search is performed within each memeplex, and memeplexes are periodically shuffled to enable global information sharing; a censorship mechanism that generates random new solutions is also included to prevent the spread of poor solutions [11]. To enhance the algorithm's search capacity against premature convergence, "Modified SFLA" variants incorporating additional parameters such as acceleration factors and inertia weight have been developed in the literature [43]. SFLA has also been used in hybrid SFLA-ACO structures, where the tasks of exploring the search space (SFLA) and determining the final sequence order (ACO) are decoupled [44].

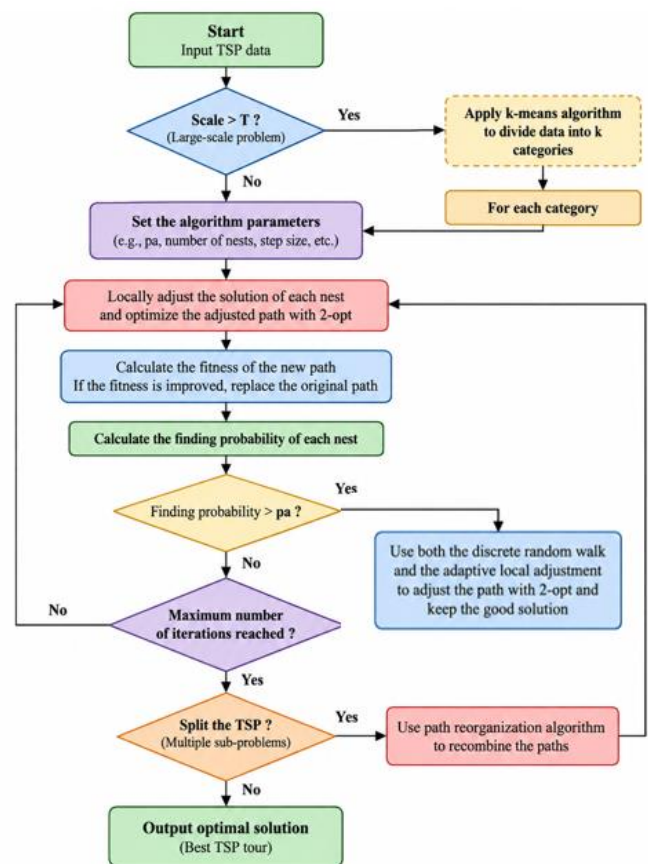


Figure 8. Author-created schematic representation of the RW-DCS algorithm workflow, based on the algorithm described in [42].

Reinforcement Learning (RL), unlike classical metaheuristic methods, is an AI approach that models the problem as a Markov Decision Process and relies on an agent learning the optimal policy (route) through trial-and-error via reward signals. When applied to tool path optimization, hybrid architectures are typically used that combine Q-learning for coarse-grained (macro) route decisions with the A* algorithm for fine-grained (micro) approach to the target [21]. Similarly, in fully automated PCB drilling systems, classical AI path-planning algorithms such as Dijkstra, A*, and Rapidly-exploring Random Trees (RRT) are used together with a 2-opt step that refines the resulting route [45].

The Variable Neighborhood Search and Tabu Search Hybrid (HVNTS) is a local search hybrid that combines the systematic neighborhood-switching strategy of Variable Neighborhood Search (VNS) with the forbidden-move memory of Tabu Search. It has been shown that, particularly in large-scale combinatorial problems involving numerous binary decision variables (e.g., 5-axis EDM drilling), non-productive time can be significantly

reduced when supported by specialized neighborhood-generation strategies [23].

Overall, the methods addressed in this section are represented in the literature by fewer studies compared to the four main algorithm families; this indicates both that these methods are still being newly adapted to the multi-hole drilling domain and that they present a relatively underexplored area for future research.

IV. FINDINGS AND DISCUSSION

4.1 ACO-Based Studies

Ant Colony Optimization stands out in the reviewed literature as one of the most frequently applied algorithm families for multi-hole drilling path optimization. Olivari [17] compared three ACO variants (AS, EAS, RAS) both individually and combined with 2-opt/Or-opt local search methods, showing that Rank-based Ant System combined with 2-opt produced the best overall result while also having the lowest computation time among the three algorithms. This finding indicates that variant selection alone is not sufficient within the ACO family, and that local search integration decisively affects the outcome. In direct drilling applications, Hoang et al. [13] tested standard ACO on 34- and 52-hole sheet-metal parts, achieving path-length reductions of 32.77% and 34.77%, respectively, compared to commercial CAM software. Similarly, Hatem et al. [24] integrated ACO into an open-architecture STEP-NC-based CNC control system, achieving a 16.96% path reduction and a decrease in processing time from 254 to 211 seconds on a 13-hole part.

Abdullah et al. [22] applied ACO beyond pure drilling to a mixed operation sequence (5 drilling + 4 pocket-machining operations), achieving an average 10.8% reduction in

rapid-travel across eight different pocket-machining strategies - demonstrating that ACO generalizes not only to hole sequencing but also to mixed operation sequencing.

The behavior of ACO in large-scale, irregular problems has been systematically examined by Yang, Huang, and Ji [18]. The authors compared six classical path-planning algorithms (ACO, nearest neighbor, minimum spanning tree, GA, SA, branch-and-bound) across four different hole-distribution patterns in a rock-drilling scenario, finding that GA, SA, and branch-and-bound were impractical at large scale, while ACO outperformed simpler heuristics but suffered from randomness and path-crossing issues. Their proposed hybrid method - strengthening ACO with 2-opt local search and dynamically adjusted elitist-strategy pheromone updating - reduced path-length variance by 26.4% for 80-hole random distributions and by 58.3% at the 200-hole large-scale case, while completely eliminating path crossings. This result confirms that ACO's fundamental weaknesses (randomness, crossing) can be systematically addressed through local search and adaptive pheromone strategies.

The hybridization trend is also evident in the GACO algorithm of Tanriver and Ay [46]; when ACO's pheromone-driven structure is combined with GA's crossover/mutation operators, a 3.16% improvement over standard ACO was achieved, though computation time increased by 6.46% - the authors found this trade-off reasonable when compared to the 0.56–16.6% range reported by other hybrid-ACO studies in the literature. The shortest tool paths obtained using the ACO and GACO algorithms are comparatively illustrated in Figure 9.

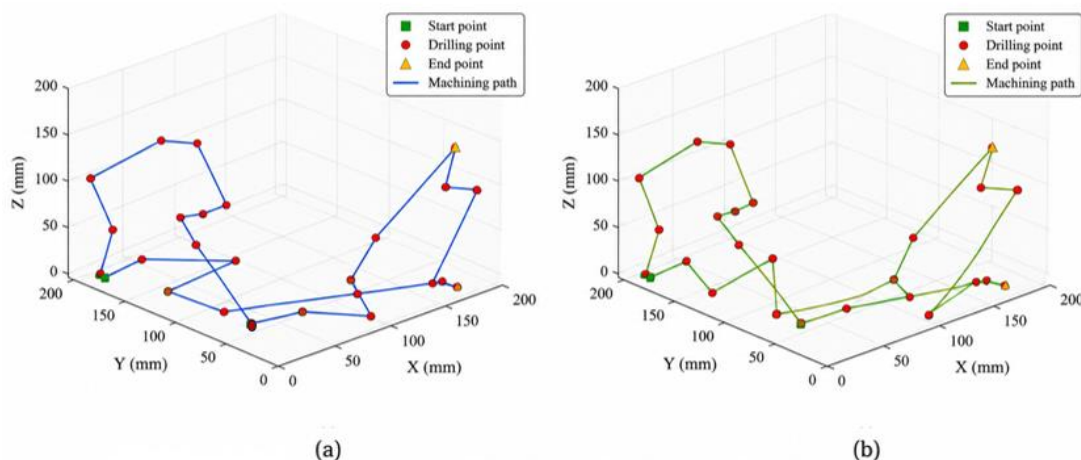


Figure 9. Author-created graphical representation of the shortest machining paths obtained using (a) ACO and (b) GACO, based on the results reported in [46].

Feng et al. [25] used ACO not in isolation but together with multi-objective NSGA-II optimization; while ACO alone achieved a 26.7% path-length reduction, applying it together with NSGA-II led to a massive reduction of 57.7% in energy consumption and 66.4% in processing time - demonstrating that route optimization and cutting-parameter optimization are complementary rather than competing improvements. The optimized tool path obtained using the improved ant colony algorithm is illustrated in Figure 10.

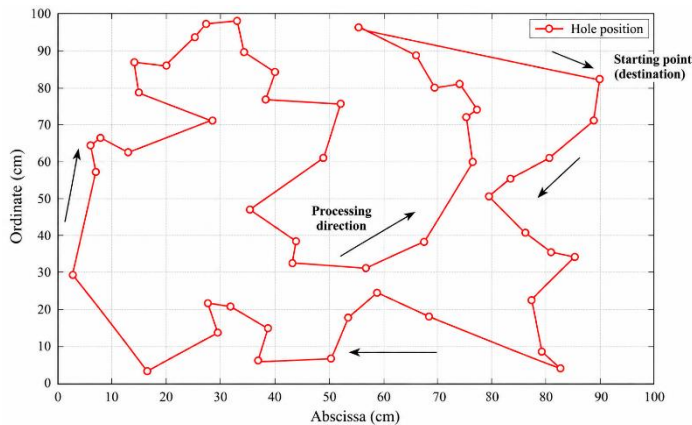


Figure 10. Author-created graphical representation of the optimized tool path obtained using the improved ant colony algorithm, based on the results reported in [25].

Blum's bibliometric review [47] confirms this trend at a broad literature scale, showing that post-2010 ACO research has shifted from producing new variants toward hybridization with machine learning and integer programming.

4.2 Bees Algorithm-Based Studies

The Bees Algorithm is the most represented algorithm family among the 39 reviewed sources and has produced strong results particularly on small-to-medium-scale PCB benchmark problems. Kamaruddin et al. [2] tested BA on the standard 14-hole PCB benchmark, reaching the optimal path length of 280.00 mm in 7 of 10 parameter sets, though it lagged behind the Bat Algorithm in terms of average path length and standard deviation - revealing BA's weakness in terms of stability. Abidin et al. [1] tested BA on 5x5, 7x7, and 9x9 matrix problems, finding competitive results at small scale, but performance falling behind other algorithms at the 9x9 scale, which they attributed to the algorithm's exploration mechanism weakening in large search spaces. The drill sequence obtained by the Bees Algorithm for the 9×9 problem is illustrated in Figure 11.

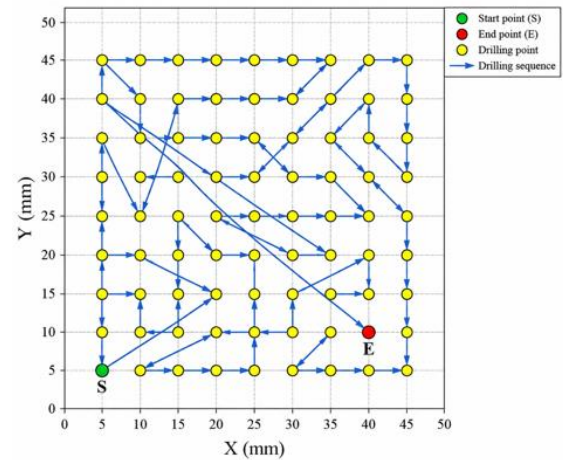


Figure 11. Author-created graphical representation of the drilling sequence obtained using the Bees Algorithm for the 9×9 problem, based on the results reported in [1].

Numerous improvements have been proposed in the literature to address this scale-dependent weakness. Hamza et al. [20] proposed a triple hybrid local search strategy targeting the cost increases at the start/end edges of routes, achieving over 21.5% cumulative improvement on MTSP benchmarks and outperforming advanced algorithms such as EACO and MACO even on the large-scale 1002-city problem. Sahin [22] supported population generation with a Nearest Neighbor method instead of random initialization (CBA-NNM) and used Multi-Insert instead of the Swap operator, achieving markedly superior results compared to classical CBA, with under 1% deviation up to 150 cities and 7.13% deviation at 1002 cities. Rathod et al. [3] proposed the Enhanced Bees Algorithm (EBA), which randomly selects among three local search operators (swap, insertion, reversion), reaching the theoretical optimum of 280.0 mm in all 10 parameter sets on the standard 14-hole PCB benchmark with a standard deviation of 0.0 mm - demonstrating that the stability issue observed in [2] was largely resolved by EBA.

Another major limitation of the Bees Algorithm is the high number of parameters in its classical form. Rathod et al. [30] tested the CBA2 approach, which reduces this number to two, on a 326-hole industrial punching panel, achieving a 9.6% reduction compared to commercial CAM - nearly matching the performance of the seven-parameter classical CBA, but with a much simpler tuning process. This finding confirms that parameter complexity can be reduced without significant performance loss. The electric panel geometry, the tool path generated by the commercial CAM software, and the optimized path obtained using CBA are comparatively illustrated in Figure 12.

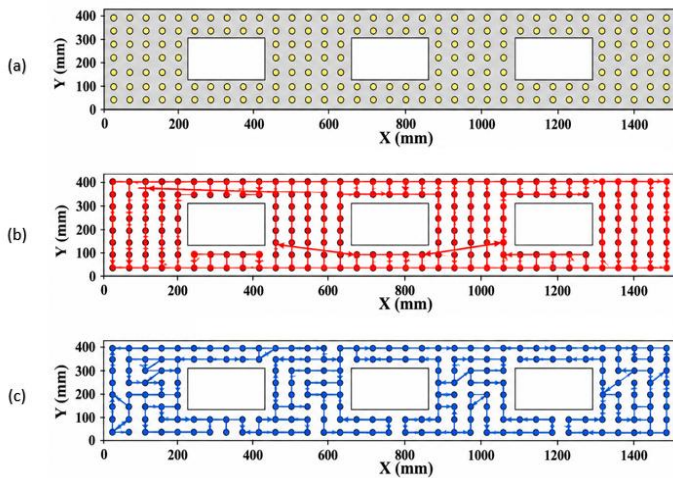


Figure 12. Author-created graphical representation of the 326-hole electric panel problem: (a) hole distribution, (b) tool path generated by the CAM software, and (c) best path obtained using CBA, based on the results reported in [30].

The Bees Algorithm has also been applied to combinatorial problems closely related to, though outside of, drilling, such as PCB assembly time. Ang and Ng [48] developed new local search operators inspired by TRIZ principles, achieving 8.16% faster assembly time than the best non-Bees solution in the literature on a 50-component PCB assembly line. The theoretical review by Castellani and Pham [5] and the application review by Pham et al. [28] confirm these findings at a broader scale, showing that BA has been successfully applied to various production problems, including WEDM parameter optimization, robotic disassembly-line balancing, and cloud-edge scheduling, though parameter-tuning burden remains the most frequently emphasized challenge across the literature. The ABC review by Ibrahim et al. [31] confirms that the closely related Artificial Bee Colony family exhibits a similar pattern, with local-optima trapping and insufficient exploitation being addressed through penalty methods and hybrid structures.

4.3 Genetic Algorithm-Based Studies

The Genetic Algorithm appears in the multi-hole drilling literature both as a direct application and as a theoretical/hybrid component. Hoang et al. [19] tested a GA-based method on 12- and 31-hole steel plates, achieving path-length reductions of 24.63% and 11.45%, respectively, compared to the commercial Catia CAM module, and emphasized that the resulting routes did not self-intersect and were therefore smoother. The authors noted that computation time increases with the number of holes but that this is negligible for serial production.

Hatem et al. [34] integrated GA directly into the G-code extraction and optimization process, developing software usable within an open-architecture CNC controller; achieving an 18% path reduction, and comparing this result with an ACO-based approach previously developed by the same team, reported that GA performed approximately 5.3% better. This direct comparison is notable as one of the rare instances in which GA outperforms ACO. In contrast, Taheri et al. [33] adapted GA at a theoretical level to a modified TSP formulation incorporating additional constraints such as tool changes and obstacle avoidance, but shared only the algorithmic framework and pseudo-code without numerical validation - this study constitutes a methodological exception among the 39 reviewed sources as the only one lacking quantitative results.

The most natural extension of GA is toward multi-objective optimization. Demir and Acar Vural [35] used Elitist Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) to simultaneously optimize path length (weight 0.9) and path smoothness (weight 0.1), achieving approximately 30% reduction in non-cutting distance and approximately 7% reduction in cycle time in experiments physically validated on a real CNC machine. The comparison of the original and optimized workpiece parameters in terms of non-cutting distance and cycle time is presented in Figure 13.

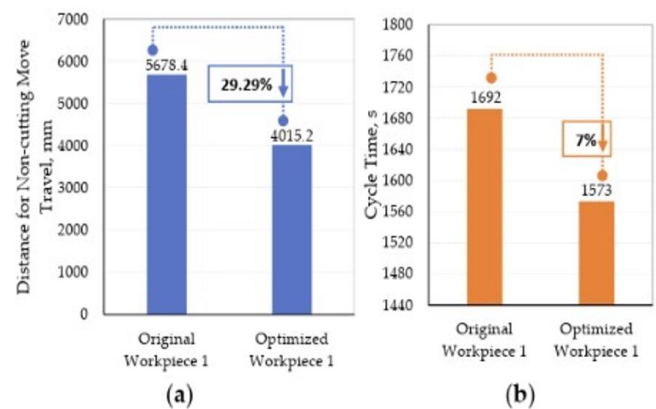


Figure 13. Comparison of original and optimized workplace parameters [35], reused under the CC BY 4.0 license (MDPI Applied Sciences).

Pareto analyses showed that NSGA-II produced a more diverse and dense set of optimal solutions compared to classical multi-objective GAs (MOGA). This finding confirms that the multi-objective advantage observed in the ACO+NSGA-II hybrid of Feng et al. [25] also holds for the GA family.

4.4 GWO-Based Studies

The Grey Wolf Optimizer stands out in the reviewed literature as the newest and most rapidly growing algorithm family. Jia et al. [37] used multi-objective GWO to simultaneously optimize energy consumption and processing time in hybrid drill-reaming machining, achieving 7.80% energy and 4.88%-time reduction compared to the traditional empirical strategy in physical validation, and also showed that GWO's 6.96% theoretical energy reduction outperformed TLBO's 5.17%. The optimized machining path obtained using the GWO algorithm is illustrated in Figure 14.

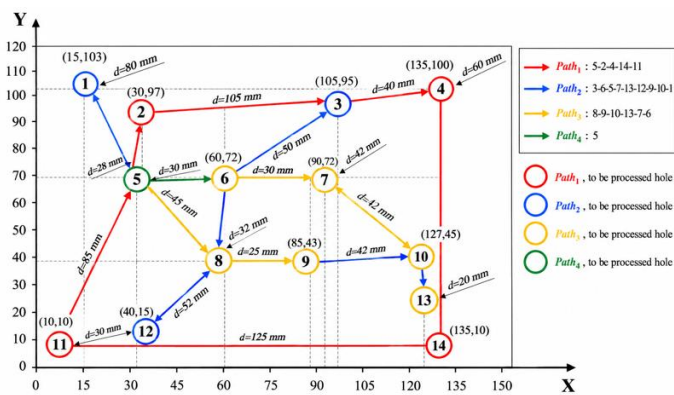


Figure 14. Author-created graphical representation of the GWO-optimized machining path, based on the results reported in [37].

Zhang et al. [39] applied Adaptive Discrete GWO (A-D-GWO) to the laser micro-hole drilling problem; adapted to discrete space using Hamming-distance-based operators, the algorithm outperformed GA, SA, ACO, PSO, and greedy algorithms across three different CAD files with 90–649 holes, achieving 75–80.5% reduction in jump time and up to 55.5% reduction in total processing time - among the highest improvement rates reported across the 39 reviewed sources.

This strong performance of GWO aligns with the algorithm's rapid rise in the broader literature. The energy-systems review by Nasir et al. [38] showed that the number of publications using GWO has increased steadily since 2014, with researchers preferring the Standard GWO variant 60% of the time and the Modified GWO variant 27% of the time. The reviews by Liu et al. [26] and Makhadmeh et al. [27] emphasize that GWO has been rapidly adopted due to its simple structure, low parameter requirements, and fast convergence, although its multi-objective versions have not yet matured to the level of ACO and BA - this finding, when considered together with

the successful application of GWO to a multi-objective problem in [37], indicates that the field is an active and developing research frontier.

4.5 Other Metaheuristic and Hybrid Studies (SA, PSO, TLBO, DM, Cuckoo Search, SFLA, RL)

Simulated Annealing stands out as a method that produces strong and stable results, particularly on medium-scale, regular matrix structures. Rathod et al. [16] compared SA with GA, PSO, and ACO across five different benchmarks, finding that SA showed the most stable and best performance up to the 9x9 matrix (81 holes), but fell behind ACO on the 11x11 matrix (121 holes). This scale-dependent transition point is consistent with the findings of Trajković et al. [14], who applied SA to a 25-hole multi-tool plate by decomposing it into three sub-problems, achieving path-length reductions ranging from 13% to 47% compared to FeatureCAM's default "minimize rapid distance" mode. The tool paths obtained using FeatureCAM and SA are comparatively illustrated in Figure 15.

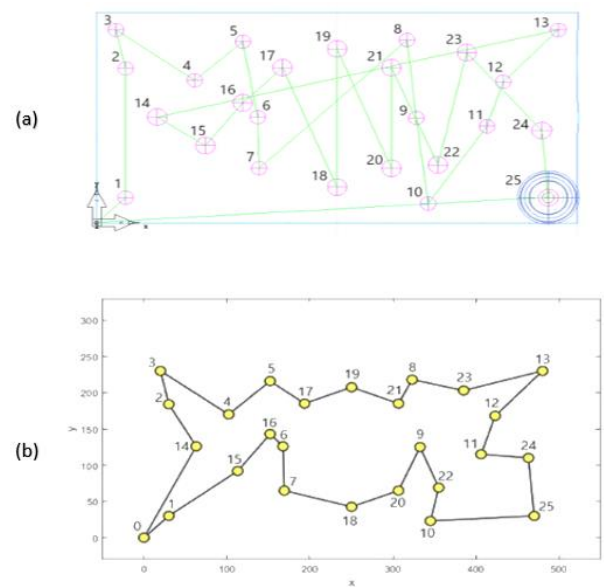


Figure 15. Author-created graphical representation of the tool paths obtained using (a) FeatureCAM and (b) SA, based on the results reported in [14].

Rathod et al. [15] compared Discrete Teaching-Learning-Based Optimization (D-TLBO) with SA; at the small 14-hole scale, both algorithms reached the same optimal result, while at the larger 158-hole scale, D-TLBO showed a clear advantage. This finding confirms the "SA is strong at small/medium scale but weakens at large scale" pattern observed in [16], specifically for D-TLBO as well. The comparative path-length distributions obtained by D-TLBO

and SA for the 158-hole workpiece are presented in Figure 16.

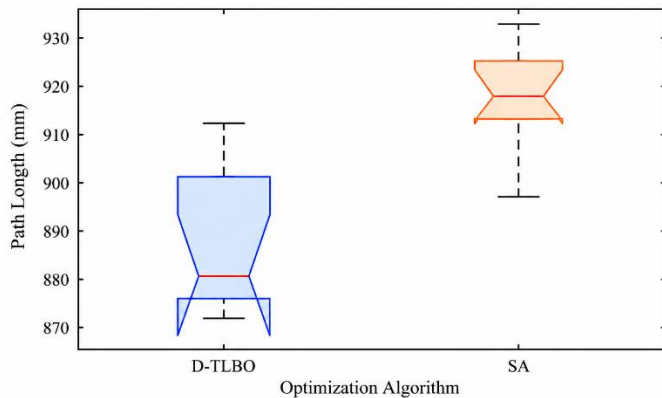


Figure 16. Author-created graphical representation of the path-length distributions obtained using D-TLBO and SA for the 158-hole workpiece, based on the results reported in [14].

Particle Swarm Optimization is one of the most extensively physically validated methods in the literature. Abidin et al. [40] physically tested routes produced by PSO on a real Haas VF-6 CNC milling machine across nine different scales ranging from 10 to 150 holes, achieving an average 5% faster machining time - a figure that rose to 11–12.6% at medium scale but declined to 4–5.5% at large scale due to the growth of the search space. The review by Priyadarshi and Kumar [49] evaluates PSO alongside ACO, confirming that both algorithms have distinct strengths and weaknesses depending on the scenario, and that modern modifications have successfully overcome these limitations.

The deterministic matrix-based Dhouib-Matrix (DM) approach stands out in the reviewed literature as the method demonstrating the most consistent superiority. Zouari et al. [36] compared DM variants against 11 competing algorithms across 13 different scenarios ranging from 20 to 2100 holes, reporting that in large-scale problems where standard GA showed a 231% deviation, DM's deviation remained close to zero - only in the largest circular layout with 2100 holes did a modified ACO variant achieve a marginal 0.21% advantage over DM3. Dhouib's own study [41] tested the DM4 technique on PCB problems with 30–75 holes, finding the shortest path in all scenarios and outperforming even its closest rival, hybrid CS-GA, by a 0.828% deviation margin. The shortest drilling path obtained using the DM4 method for the 75-hole PCB problem is illustrated in Figure 17.

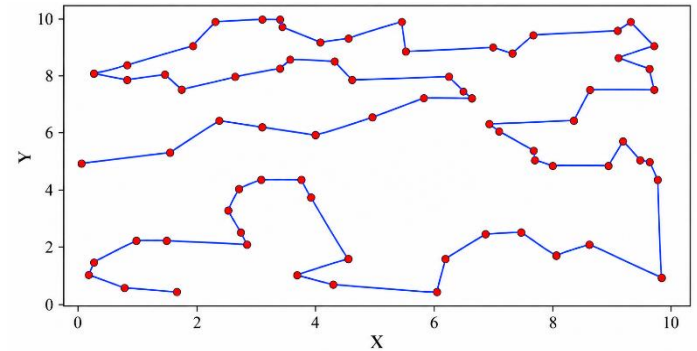


Figure 17. Author-created graphical representation of the shortest drilling path obtained using DM4 for the 75-hole PCB problem, based on the results reported in [41].

Cuckoo Search and the Shuffled Frog-Leaping Algorithm are the least-represented methods in the reviewed literature, though they have attracted recent interest. Zhang and Yang [42] developed a discrete Cuckoo Search variant (RW-DCS), achieving 38 to 59 times more stable results than competing algorithms both on TSPLIB benchmarks and in a real glass-cutting path application. Dalavi et al. [43] tested Modified SFLA on rectangular hole matrices, producing shorter paths than GA and ACO at the 4x5 and 11x11 scales, while lagging slightly behind ACO at the 5x5 scale. Nasir, Umer, and Asgher [44] proposed a hybrid combining SFLA's exploration strength with ACO's exploitation strength, achieving 11.1% improvement over the manual route and 5.9% improvement over commercial software on a 24-hole omega plate - indicating that the SFLA family, when hybridized with ACO, can mature to a level capable of outperforming commercial software.

Finally, next-generation AI-based approaches address this problem class from a different angle. Wan, Xu, and Zuo [21] were the first to model tool path optimization as a Markov Decision Process and proposed a Reinforcement Learning-based solution (Q-learning + A*), demonstrating superiority over GA, ACO, and PSO in both solution speed and quality, achieving a 27.33% performance advantage even under highly time-constrained conditions of just 3 seconds.

Mamdouh et al. [45] combined classical AI path-planning algorithms (Dijkstra, A*, RRT) with a 2-opt refinement step to develop a fully automated PCB drilling machine, achieving a 13-fold efficiency improvement in processing time compared to KiCAD's own auto-router. The HVNTS (VNS+Tabu Search) hybrid of Wang et al. [23] reduced non-productive time by 86.65% in a 5-axis EDM drilling process, demonstrating that classical local-search hybrids also remain competitive in this problem class.

4.6 Cross-Algorithm Synthesis: The Scale-Dependent Performance Pattern

Read across Sections 4.1–4.5, the findings converge on a consistent scale-dependent pattern, organized here using the small/medium/large problem-scale bands defined in Section 2.5. At small scale, the Bees Algorithm and Simulated Annealing dominate: BA reaches the theoretical optimum in 7 of 10 [2] to all 10 [3] parameter sets on the standard 14-hole PCB benchmark, and SA shows its most stable performance up to the 9x9 (81-hole) matrix [16]. At this scale, algorithm choice appears to matter less than parameter-tuning quality, since even the classical, unmodified forms of these algorithms perform competitively.

At medium scale, performance becomes markedly more differentiated and hybridization begins to dominate: D-TLBO overtakes SA once problem size reaches 158 holes [15], GA-based NSGA-II achieves a 30% non-cutting-distance reduction in physically validated CNC experiments [35], and Bees-Algorithm variants such as EBA extend their small-scale strength up to 326 holes [30]. GWO enters its most competitive range at the upper end of this band, delivering up to a 55.5% total processing-time reduction across three CAD files spanning 90–649 holes [39] - a range that extends from medium into large scale under the bands defined in Section 2.5 -among the highest improvement rates reported across the reviewed studies.

At large scale, the deterministic Dhouib-Matrix approach and ACO-hybrid strategies take over as the most consistently superior methods: DM maintains near-zero deviation from optimum even at 2100 holes, a scale at which standard GA's deviation reaches 231% [36], and ACO strengthened with 2-opt local search and adaptive elitist pheromone updating reduces path-length variance by 58.3% at the 200-hole scale while completely eliminating path crossings [18].

Two secondary patterns cut across this scale-dependent picture. First, hybridization is the most reliable lever for improving performance at any scale: every algorithm family in this review that was combined with a complementary mechanism (ACO+2-opt, ACO+NSGA-II, GA-ACO, SFLA-ACO, VNS+Tabu Search) outperformed its non-hybridized counterpart by a wide margin, with reported gains reaching 57.7% energy and 66.4% time for ACO+NSGA-II [25] and 86.65% non-productive-time reduction for the VNS+Tabu Search hybrid HVNTS [23]. Second, physical validation remains the exception rather than the rule: of the 39 reviewed studies, only a minority -

those applying PSO [40], GWO [37], NSGA-II [35], and Adaptive Discrete GWO [39] - were tested on real CNC, laser, or EDM equipment rather than in simulation alone, meaning that most of the improvement percentages summarized above should be read as simulation-based upper bounds rather than guaranteed industrial gains.

V. CONCLUSION AND RECOMMENDATION

This study systematically reviewed the post-2021 literature on tool path optimization in multi-hole drilling processes, drawing on a total of 49 sources comprising 39 application studies, 9 founding/theoretical references, and 1 figure-source reference.

Algorithm distribution and maturity level. Approximately 44% of the 39 reviewed application studies employ Bees Algorithm- or ACO-based approaches; these two algorithm families possess the most mature and diversified body of solutions, both on standard small-scale benchmarks (such as the 14-hole PCB) and in large-scale industrial scenarios. The Grey Wolf Optimizer has emerged as a rapidly rising alternative, particularly in energy-focused and laser micro-hole applications, while methods such as Dhouib-Matrix, Reinforcement Learning, Cuckoo Search, and the Shuffled Frog-Leaping Algorithm represent relatively new and underexplored research frontiers that reflect the field's tendency to move beyond the classical metaheuristic framework.

Scale-dependent performance pattern. One of the most consistently recurring findings across the study is that no algorithm demonstrates uniform superiority across all problem scales. Simulated Annealing and the Bees Algorithm produce strong and stable results on small-to-medium-scale, regular problems (up to 81 holes), with their performance relatively weakening as the number of holes increases (121 holes and above); in contrast, methods such as ACO, Dhouib-Matrix, and Adaptive Discrete GWO deliver more consistent results on large-scale, irregular problems (hundreds to thousands of holes). This pattern indicates that algorithm selection should be made conditionally, based on problem scale.

Hybridization as the dominant research direction. The majority of the reviewed studies move toward hybrid approaches that combine the exploration strength of one algorithm with the exploitation strength of another (e.g., GA-ACO, SFLA-ACO, VNS-Tabu Search, ACO+NSGA-II), rather than relying on a single "pure" algorithm. This trend has been consistently observed both in direct application studies and in bibliometric reviews (particularly in the ACO -

and ABC literature); hybridization has become the literature's primary strategy for addressing the chronic issues of parameter sensitivity and scale-dependent weakness in classical algorithms.

Validation approach. A significant portion of the reviewed studies have been tested only in simulation environments, with a more limited number of studies conducting physical validation on real CNC/laser/EDM machines. Examples that include physical validation (PSO applied to PCB drilling, GWO applied to drill-reaming, NSGA-II verified through CNC experiments, Adaptive Discrete GWO applied to laser micro-hole machining) largely confirm the industrial validity of simulation results, though the rate of physical validation across the literature should be increased.

Limitations and recommendations for future research. This review has several limitations. First, the search was largely restricted to Google Scholar and to publications from 2021 onward, meaning some influential earlier studies may have fallen outside the scope. Second, since most reviewed studies use different benchmark problems and different performance metrics, direct numerical comparison across algorithms has only been possible through a limited number of studies (particularly the comparative series by Rathod and colleagues). Accordingly, the following directions are recommended for future research: (i) systematic comparison of different algorithm families on standard, shared benchmark sets; (ii) increasing the rate at which simulation results are validated through physical CNC/laser/EDM experiments; (iii) more comprehensive adaptation of relatively underexplored methods such as Cuckoo Search, SFLA, and Reinforcement Learning to the multi-hole drilling problem; and (iv) broader adoption of multi-objective optimization studies that simultaneously address path length, processing time, energy consumption, and surface quality.

In conclusion, tool path optimization in multi-hole drilling processes remains an active and rapidly evolving research area in which metaheuristic algorithms provide concrete and measurable contributions to industrial manufacturing efficiency.

STATEMENT OF CONTRIBUTION RATE

Authors' contribution rates to the study are equal.

CONFLICTS OF INTEREST

They reported that there was no conflict of interest between the authors and their respective institutions.

RESEARCH AND PUBLICATION ETHICS

In the studies carried out within the scope of this article, the rules of research and publication ethics were followed.

REFERENCES

- [1] M. H. Z. Abidin, S. Kamaruddin, A. A. Malek, and N. A. Sukindar, "Application of the Bees Algorithm to Find Optimal Drill Path Sequence," in *Intelligent Manufacturing and Mechatronics, LNME*, 2024, pp. 3–16.
- [2] S. Kamaruddin, M. N. Rosdi, and N. A. Sukindar, "Optimization of Drilling Path Using the Bees Algorithm," *Manufacturing Technology*, vol. 21, no. 6, pp. 788–792, 2021.
- [3] V. P. Rathod, O. P. Yadav, S. P. Kadam, A. P. S. Rathore, and V. Lall, "An Enhanced Bees Algorithm for Multi-Hole Drill Tool Path Optimization," in *Proc. IEEM 2024*, 2024, pp. 262–266.
- [4] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. SMC-B*, vol. 26, no. 1, pp. 29–41, 1996.
- [5] M. Castellani and D. T. Pham, "The Bees Algorithm: A Gentle Introduction," in *Intelligent Production and Manufacturing Optimisation*, 2023, pp. 3–21.
- [6] K. Sastry, D. E. Goldberg, and G. Kendall, "Genetic Algorithms," in *Search Methodologies*, 2014, pp. 93–117.
- [7] D. Bertsimas and J. Tsitsiklis, "Simulated Annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.
- [8] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. ICNN'95*, 1995, pp. 1942–1948.
- [9] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [10] X. S. Yang and S. Deb, "Cuckoo Search via Lévy Flights," in *Proc. NaBIC 2009*, 2009, pp. 210–214.
- [11] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
- [12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [13] V. Q. Hoang, X. D. Pham, and M. S. Nguyen, "Prediction of Optimal Tool Path for Drilling Based on Ant Colony Algorithm," in *Proc. MMMS2022*, 2024, pp. 261–269.

- [14] A. Trajković, A. Stanković, G. Petrović, and V. Mitrović, "Multi tool hole drilling path optimization using simulated annealing," in *ICIST 2024 Proceedings*, M. Zdravković, M. Trajanović, N. Filipović, and Z. Konjović, Eds. 2024, pp. 143-151.
- [15] V. P. Rathod, S. P. Kadam, O. P. Yadav, and A. P. S. Rathore, "Discrete Teaching Learning-Based Optimization for Multi-Hole Drilling," in *Proc. INOCON 2023*, 2023, pp. 1–4.
- [16] V. P. Rathod, O. P. Yadav, and A. P. S. Rathore, "Comparative Study of Multi-hole Drill Path Optimization using Evolutionary Algorithms," in *Proc. IEEM 2022*, 2022, pp. 615–619.
- [17] L. Olivari, "Comparison of improved ACO algorithms for tool path optimization in multi-hole drilling," in *Proc. ICIL 2023*, 2023, pp. 74–78.
- [18] W. Yang, C. Huang, and P. Ji, "Ant colony algorithm optimized by local search and adaptive strategy for drilling sequence path planning," *J. Mech. Sci. Technol.*, vol. 40, no. 2, pp. 1291–1301, 2026.
- [19] V. Q. Hoang, N. T. Bui, X. D. Pham, and G. H. Vuong, "Optimization Toolpath Generation for Drill Hole on Three Axes CNC Machine Using Genetic Algorithms," in *Proc. MMMS2022*, 2024, pp. 415–425.
- [20] A. Hamza, A. Haj Darwish, and O. Rihawi, "A new local search for the bees algorithm to optimize multiple traveling salesman problem," *Intell. Syst. Appl.*, vol. 18, p. 200242, 2023.
- [21] Y. Wan, W. Xu, and T. Y. Zuo, "Tool Path Optimization for Complex Cavity Milling Based on Reinforcement Learning Approach," *IEEE Access*, vol. 11, pp. 66793–66807, 2023.
- [22] H. Abdullah, C. Law Boon Hui, and M. S. Zakaria, "Minimization of Machining Process Sequence Based on Ant Colony Algorithm," *J. Eng. Sci. Technol.*, vol. 18, no. 2, pp. 949–962, 2023.
- [23] J. Wang, X. C. Xi, L. Qin, Y. O. Zhang, and W. S. Zhao, "Non-productive time optimization for 5-axis EDM drilling using HVNTS algorithm," *Int. J. Prod. Res.*, vol. 59, no. 16, pp. 5068–5082, 2021.
- [24] N. Hatem, Y. Yusof, A. Z. A. Kadir, K. Latif, and M. M. Abedlhafe, "Optimization and execution of multiple holes-drilling operations based on STEP-NC," *Int. J. Adv. Manuf. Technol.*, vol. 114, pp. 2031–2043, 2021.
- [25] C. Feng, X. Chen, J. Zhang, Y. Huang, and Z. Qu, "Minimizing the energy consumption of hole machining," *Int. J. Adv. Manuf. Technol.*, vol. 121, pp. 215–228, 2022.
- [26] Y. Liu, A. As'array, M. K. Hassan, A. A. Hairuddin, and H. Mohamad, "Review of the grey wolf optimization algorithm: variants and applications," *Neural Comput. Appl.*, vol. 36, pp. 2713–2735, 2024.
- [27] S. N. Makhadmeh, M. A. Al-Betar, I. Abu Doush, M. A. Awadallah, S. Kassaymeh, S. Mirjalili, and R. A. Zitar, "Recent Advances in Grey Wolf Optimizer," *IEEE Access*, vol. 12, pp. 23028–23055, 2024.
- [28] D. T. Pham, Y. Song, Y. Laili, A. Ismail, N. Hartono, M. Packianather, and M. Castellani, "The Bees Algorithm and Its Applications in Production and Manufacturing," *Int. J. Adv. Prod. Res.*, 2024.
- [29] M. Sahin, "Solving TSP by using combinatorial Bees algorithm with nearest neighbor method," *Neural Comput. Appl.*, vol. 35, pp. 1863–1879, 2023.
- [30] V. P. Rathod, O. P. Yadav, S. P. Kadam, and A. P. S. Rathore, "Combinatorial Bees Algorithm with 2-Parameters for Punching Path Optimization," in *Proc. IEEM 2025*, 2025.
- [31] A. O. Ibrahim, E. M. E. Elfadel, I. A. T. Hashem, H. J. Syed, M. A. Ismail, A. H. Osman, and A. Ahmed, "The Artificial Bee Colony Algorithm: A Comprehensive Survey," *Arch. Comput. Methods Eng.*, vol. 32, pp. 3499–3533, 2025.
- [32] S. N. Sivanandam and S. N. Deepa, "Genetic algorithms," in *Introduction to genetic algorithms*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 15–37.
- [33] S. M. Taheri, M. Sasaki, K. Matsushita, H. Ngetha, and S. Ito, "A Study of Different Optimization Methods of Tool Path Machines," n.d.
- [34] N. Hatem, M. M. Abedlhafe, A. D. Hassan, N. H. J. Al Hassan, Y. Yusof, Y. Saif, and I. Maznah, "Develop Software to Extract and Optimize Drilling G-Code Using GA," *J. Eur. Syst. Autom.*, vol. 58, no. 6, pp. 1305–1313, 2025.
- [35] G. Demir and R. Acar Vural, "Non-Cutting Moving Toolpath Optimization with NSGA-II," *Applied Sciences*, vol. 14, no. 11, p. 4471, 2024.
- [36] A. Zouari, D. Souhail, F. Lehyani, and J. C. Sá, "Tool Path Length Optimization in Drilling Operations: A Comparative Study," in *Proc. FAIM 2023*, 2024, pp. 293–301.
- [37] S. Jia, S. Wang, J. Song, S. Li, A. Li, Q. Cao, and Z. S. Li, "Energy modelling and multi-objective tool path optimisation," *Int. J. Prod. Res.*, vol. 63, no. 24, pp. 10309–10335, 2025.
- [38] M. Nasir, A. Sadollah, S. Mirjalili, S. A. Mansouri, M. Safaraliev, and A. R. Jordehi, "A Comprehensive Review on Applications of Grey Wolf Optimizer in Energy Systems," *Arch. Comput. Methods Eng.*, vol. 32, pp. 2279–2319, 2024/2025.

- [39] T. Zhang, H. Hu, Y. Liang, X. Liu, Y. Rong, C. Wu, G. Zhang, and Y. Huang, "A novel path planning approach... adaptive discrete grey wolf optimizer," *Comput. Ind. Eng.*, vol. 193, p. 110320, 2024.
- [40] N. W. Z. Abidin, N. Salim, M. F. F. Ab. Rashid, N. M. Z. N. Mohamed, A. N. M. Rose, and A. Mokhtar, "Experimental of Multi-holes Drilling Toolpath Using PSO and CAD-CAM Software on PCB," in *Recent Trends in Mechatronics*, 2022, pp. 197–204.
- [41] S. Dhouib, "Finding the Shortest Holes Drilling Path via Dhouib-Matrix-4," in *Mechatronics and Automation Technology*, 2022, pp. 396–401.
- [42] Z. Zhang and J. Yang, "A discrete cuckoo search algorithm for TSP," *Comput. Ind. Eng.*, vol. 169, p. 108157, 2022.
- [43] A. M. Dalavi, P. J. Pawar, T. P. Singh, and A. Gomes, "Optimal Drilling Sequences Using Modified SFLA," *Int. J. Ind. Eng. Prod. Res.*, vol. 33, no. 4, pp. 1–10, 2022.
- [44] M. Nasir, M. Umer, and U. Asgher, "Application of hybrid SFLA and ACO to omega plate for drilling process planning," *Archives for Technical Sciences*, vol. 26, no. 1, pp. 1–12, 2022.
- [45] M. Mamdouh, A. Khalid, R. Mahmoud, O. Desouki, and S. O. Abdellatif, "A fully automated drilling machine for PCBs with superior path optimization," *Sci. Rep.*, vol. 15, no. 40367, 2025.
- [46] K. Tanriver and M. Ay, "Efficient Path Planning for Drilling Processes: The Hybrid Approach of a Genetic Algorithm and Ant Colony Optimisation," *Transactions of FAMENA*, vol. 48, no. 3, pp. 125–140, 2024.
- [47] C. Blum, "Ant colony optimization: A bibliometric review," *Phys. Life Rev.*, vol. 51, pp. 87–95, 2024.
- [48] M. C. Ang and K. W. Ng, "Minimising Printed Circuit Board Assembly Time Using the Bees Algorithm with TRIZ-Inspired Operators," in *Intelligent Production and Manufacturing Optimisation – The Bees Algorithm Approach*, Springer Series in Advanced Manufacturing, D. T. Pham and N. Hartono, Eds. Switzerland: Springer Nature Switzerland AG, 2023, p. 25–41.
- [49] R. Priyadarshi and R. R. Kumar, "Evolution of Swarm Intelligence: PSO and ACO Approaches," *Arch. Comput. Methods Eng.*, vol. 32, pp. 3609–3650, 2025.