Journal of
*Composites Science*

MDPI

*Article*

# Multi-Hole Drilling Tool Path Planning and Cost Management through Hybrid SFLA-ACO Algorithm for Composites and Hybrid Materials

**Nasir Mehmood [1,*], Muhammad Umer [1] and Umer Asgher [2,3]**

[1] Sir Syed CASE Institute of Technology, Business and Engineering Management Department, Islamabad 44220, Pakistan
[2] Quality Assurance & NUST International Office Directorate (QA & NIO Dte), National University of Science and Technology (NUST), Islamabad 44000, Pakistan
[3] Department of Robotics and Artificial Intelligence (AI), School of Mechanical and Manufacturing Engineering (SMME), National University of Science and Technology (NUST), Islamabad 44000, Pakistan
* Correspondence: nasir.nust17@gmail.com

**Abstract:** In the process of drilling multiple holes in composites and hybrid materials, almost 70% of the time is consumed in tool traveling and tool changing. Recently, researchers have focused on this consumption of time for optimization of the tool path. A literature review revealed the following research gap: little work has been performed on the hybridization of metaheuristics. In the present study, the hybridization of SFLA and ACO metaheuristic algorithms is carried out, which is based on this research gap. The hybridization of SFLA and ACO metaheuristic algorithms provides originality and novelty in this study. The main objective of this study is to minimize the tool path in drilling problems. The proposed algorithm was applied to five benchmark multi-hole drilling problems and one industrial problem from the literature. The outcome of this work is evaluated with the results of dynamic programming (DP), ACO, an immune-based evolutionary approach (IA), and a modified SFLA for five benchmark problems. The accuracy of the results was improved by 2.27% using the proposed hybrid algorithm, indicating that the proposed algorithm is superior to DP, ACO, IA, and the modified SFLA. Additionally, the results of the proposed hybrid algorithm for an example industrial problem from the literature were compared with those of the SFLA and modified SFLA. The proposed algorithm reduced the total cost by 6.17% and 3.76% compared with the SFLA and modified SFLA, respectively. Thus, the efficacy of the proposed hybrid algorithm was confirmed, along with its applicability.

**Keywords:** manufacturing management; process planning; industrial management; engineering management; production planning; artificial intelligence

check for updates

**Citation:** Mehmood, N.; Umer, M.; Asgher, U. Multi-Hole Drilling Tool Path Planning and Cost Management through Hybrid SFLA-ACO Algorithm for Composites and Hybrid Materials. *J. Compos. Sci.* **2022**, *6*, 364. https://doi.org/10.3390/jcs6120364

Academic Editors: Nithesh Naik and B. Shivamurthy

Received: 30 September 2022
Accepted: 29 November 2022
Published: 2 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Making a circular primary hole in a work piece through a drill bit is called the drilling process. Reaming, boring, and tapping are drilling processes performed on the primary hole. Drilling is the most important machining process, as it is required in most manufacturing industries for making different products. The drilling tool travel and switch time comprise 70% of the drilling time [1]. Importantly, the tool travel time and switch time are non-productive and do not contribute to the drilling process; hence, they do not add value to jobs. Thus, the optimization of these parameters has attracted the researchers' and the practitioners' attention for optimization. The literature review carried out showed that little research had been performed on the hybridization of different metaheuristics; therefore, in this study, the objective of time minimization was achieved through the hybridization of the shuffled frog-leaping algorithm (SFLA) and the ant colony optimization (ACO) algorithm. The SFLA uses a random approach and does not construct the frog step by step.

The SFLA-ACO algorithm uses a probabilistic approach instead of a random approach, which has produced better results. The proposed hybrid algorithm was compared with other metaheuristic algorithms. Minimizing the tool travel time reduces the manufacturing time and, thus, decreases overhead costs and lead time. Thus, it helps industrialists and manufacturers reduce product costs, which is crucial for survival in the global market environment of cutthroat competition.

It has been found that most of the researchers have used metaheuristic algorithms. The algorithms include GA, ACO, PSO, and SA. Most of the researchers have modified the basics of these algorithms for optimization. In very rare cases, two algorithms have been hybridized. This is the research gap that has been found in the available literature. This research gap compelled the author to work on it. After having thoroughly studied the pros and cons of the SFLA and ACO algorithms, it has been found that these algorithms can be hybridized for better results. This was the real motivation behind this study. In this study, these two algorithms have been successfully hybridized, which shows the novelty of this study.

## 2. Drilling Problems and the Literature Review

Drilling problems are grouped into circular, rectangular, constraint problems, and multi-surface problems, or a combination of any of these, keeping in view the nature of the job pattern. In circular and rectangular drilling problems, the holes are located in a circular and rectangular pattern, respectively. Most of the time, the distance between holes is the same in such problems. In cases of path constraint problems, some design feature of the work piece does not allow the drill bit to move directly from one hole to the other hole posing a path constraint. When more than one surface of the work piece is required to be drilled is involved then such a problem is termed as multi-surface drilling problems. Requirements of the drilling tool suggest drilling problems 'classification as multi-tool (MT) and single-tool hole drilling problems (ST). If the work piece requires only one type of hole or the holes diameter is the same and the hole can be drilled with a single tool, then this problem is considered a single-tool hole drilling problem. The work piece requires a different diameter of holes; in this case, more than one drilling tool will be required, and the problem will be categorized as multi-tool hole drilling problems. It is further divided into multi-tool hole drilling with sequence-dependent drilling times (MTseq) and multi-tool hole drilling with precedent constraints (MTpc). Drilling costs are evaluated in terms of time taken in the drilling process and the wear to the tool. Tool traveling time, tool switching time, and time is taken in the actual drilling of the holes constitute the complete drilling process. For the time taken by the complete drilling process, all these times are summed up. Mainly, there are three types of distances are used in the literature, which are given below. $d_{euclidean}$ is used when the spindle moves in both the x and y directions. $d_{rectilinear}$ is used when the spindle can move only in one direction. $d_{chebysh}$ uses modulus on the coordinates.

$$d_{euclidean} = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2} \quad (1)$$

$$d_{rectillinear} = |a_1 - a_2| + |a_1 - a_2| \quad (2)$$

$$d_{chebysh} = \max(|a_1 - a_2|, |b_1 - b_2|) \quad (3)$$

## 3. Literature Review

For a literature review, papers published over the past 10 years were selected and critically reviewed.

In [2], ACO was used to optimize the drilling tool path for a rectangular hole pattern. Three case studies were examined, and the results were compared with those of a genetic algorithm (GA). The modified ACO outperformed the GA. In [3], the parallel implementation of ACO was employed for optimization and improvement of the operations sequence for a printed circuit board (PCB) that achieves the shortest tool path. The results of the same problem were also achieved by using MasterCAM. When both results were compared,

it was found that ACO achieved a shorter drilling time than MasterCAM for all the case studies. For such problems, metaheuristic approaches are more useful for optimization than commercially available design software. In [4], ACO was used for drilling-route optimization with a PCB, and the results were better than particle swarm optimization (PSO). In [5], a metaheuristic algorithm called the magnetic optimization algorithm (MOA) was proposed for PCB tool path optimization. The MOA was applied to relatively small-scale problems, and compared with PSO and ACO, the results obtained were satisfactory and encouraging. In [6], a biogeography-based optimization (BBO) algorithm was proposed, which is used to solve the multi-tool drilling problem for non-productive tool airtime and tool switch time optimization. The results obtained using the BBO algorithm were comparable to those achieved using the GA and ACO. The paper suggested that real-world problems can also be solved using this algorithm successfully. In [7], a PCB design software-produced image was used to determine the hole coordinates, which were used by the nearest-neighbor algorithm (NNA) for path planning and optimization. The NNA was used to optimize the drilling problem for up to 103 holes, and the results were satisfactory. In [8], a GA was used to optimize the drilling tool path. The results obtained using the GA were significantly better than those obtained using the commercially available software ArtCAM, indicating the applicability of metaheuristic algorithms to combinatorial optimization problems. In [9], the concept of integrating computer-aided design and computer-aided manufacturing (CAM) through STEP-data files was introduced to automatically generate process planes, from data files to finished products. Numerical control (NC) codes are generated from the geometric information of the part and later inspected by the coordinates measurement machine (CMM). In [10], the intelligent water drop (IWD) algorithm was applied to a PCB problem. The IWD algorithm converged rapidly and produced efficient results. It outperformed PSO and ACS concerning the convergence and the number of iterations. In [11], hole-making in an injection mold was studied, and the PSO algorithm was used to determine the shortest tool travel path. Because this was a multi-tool problem, the tool switch time and the tool traveling time were included in the objective function. A cost matrix indicated significant improvements. In [12], a new approach called the simulated Kalman filter approach (SKFA) was used for PCB hole drilling tool path optimization, and it achieved good convergence without many iterations. The SKFA was compared with PSO, ACS, and cuckoo search (CS), and the results were encouraging. In [13], the technique is based on the shortest-path search algorithm (SPSA). It was applied to a standard PCB problem with 14 holes requiring a single tool. Compared with PSO and the firefly algorithm (FA), it achieved encouraging results. In [14] tackles the injection-mold problem using PSO and the SFLA. The problem has a multi-tool precedent constraint, and the objective function includes both the tool traveling and changing time. The minimum costs in terms of time achieved by the two algorithms were identical. The study indicated that both algorithms are effective for large-scale injection-mold problems. In [15], the shortest drilling path was generated by the cncKad program. The results of cncKad were compared with those of ACO and GA for the same problem. The ACO solution was better than the G codes produced through the commercially available software cncKad. In [16], the optimal foraging algorithm (OFA), inspired by animal behavioral ecology, was proposed. The objective is to determine the optimal sequence of drilling operations for minimizing manufacturing costs. Four real-world problems were solved using the OFA, and the results were compared with those of the GA, PSO, ACO, and differential evolution for validation, indicating that they were satisfactory. In [17], a GA was applied to drill 28 holes in a work piece. The results of the GA were compared with those of the commercially available CAM system, Autodesk Inventor. For tool-path optimization, the GA outperformed Autodesk Inventor. In [18], a newly developed algorithm called the Bat Algorithm (BA) was used for tool-path sequencing and optimization. The BA is nature-inspired and imitates the preying and hunting characteristics of bats. The BA results were compared with those of the GA, PSO, ACO, and artificial bee colony (ABC) algorithms for four example problems. The computational speed and length achieved using the BA were better than those of the other algorithms.

In [19], neural networks (NNs) were applied to tool-path optimization. The proposed method can be applied to up to 100 nodes. It uses an NN and a divide-and-conquer strategy to solve large-scale TSP problems. In [20], the hybrid variable neighborhood tabu search (HVNTS) algorithm was used for electrical discharge machining (EDM) drilling. A six-hole problem for EDM drilling was selected, and the results indicated that the HVNTS could significantly reduce the tool travel time with a reasonable computational time. In [21], discrete teaching–learning-based optimization (DTLBO) was used to optimize the sequence of the tool traveling time for the multi-hole drilling problem. The results of DTLBO were better than those of the commercially available software, CAMotics.

## 4. Hybridization of SFLA and ACO

The expansion of the proposed hybrid algorithm will be explained in this section.

### *4.1. Concept of SFLA*

Shuffled complex evolution algorithms and PSO are the basis of the SFLA. These two methods are combined to design better metaheuristics for solving discrete and combinatorial optimization problems. In SFLA the shuffling process provides the search for global optima whereas the random approach ensures the local search. At the start of the SFLA, an initial population of frogs P is generated randomly, i.e., P = [X1, X2 ... Xn], where X1 represents the first frog and n represents the total number of frogs in the initial population. After having generated the frogs, these are ordered as per their performance. The arrangement of frogs is performed frog from higher-performance frogs to lower-performance frogs, i.e., O = [X5, X1, X3, X4, X7, X10]. Here O is the array that keeps the frogs in higher to lower performance frogs. The order shows that X5 is a better performance frog than X1, and X1 is a better performance frog than X3. In the next step, the number of memeplexes is defined. This number of memeplexes is dictated by the complexity of the problem. The more complex problem suggests a greater number of memeplexes, and vice versa. After the ordering, the ordered frogs are allocated to all the memeplexes in such a way that X5 is allocated to memeplex 1, X1 is allocated to memeplex 2, and X3 is allocated to memeplex 3. Suppose that there are only three memplexes defined, then X4 will be allocated to memplex 1, X7 will be allocated to memeplex 2, and X10 will be allocated to memplex 3. In this way, the allocation continues until all the frogs are allocated to the memeplexes. This is represented as P = m × n, where m represents the number of memplexes, and n represents the number of frogs in each memplex. Suppose that we have a population of frogs of P = 15, then the number of memplexes will be 3 i.e., m = 3, and the number of frogs in each memeplex will be 5, i.e., n = 5. The memplex allocation is as follows:

$$\text{Memplex 1 } M_1 = (X_5, X_4 \dots) \tag{4}$$

$$\text{Memplex 2 } M_2 = (X_1, X_7 \dots) \tag{5}$$

$$\text{Memplex 3 } M_3 = (X_3, X_{10} \dots) \tag{6}$$

After allocation of frogs to memeplexes, a local search in each memplex is performed, and the worst solution is improved. The repetition of this iterative process is carried out for a predefined number. Subsequently, the frogs are ordered from high to low performance, and the memplex allocation is conducted again using a predefined method. The entire process is repeated until the frogs converge; thus, the global optimum is achieved.

### *4.2. Concept of ACO*

Dorigo and his colleagues introduced the term ACO and proposed the ACO algorithm in 1991, which finds the shortest path through the indirect communication of ants called stigmergy. This indirect communication is because of the pheromones, which each ant drops while it moves in search of food. The other ants smell the pheromone and follow it; hence, they remain on track. The ACO algorithm constructs the solution step by step and it improves the solution after each iteration. It provides an optimal solution that is based

on probability. It can be used both for discrete and continuous problems. In production scheduling, both job shop and flexible job shop scheduling problems can be solved by using ACO. The ACO has been applied successfully to the traveling salesman problem, vehicle routing problem, task allocation of heterogeneous unmanned aerial vehicles, and open shop scheduling problem. The possibility of an ant $k$ positioned at hole $i$ traveling to hole $j$ is as follows:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} \quad (7)$$

In Equation (7), pheromone trails and heuristic information are represented by $\tau_{ij}$ and $\eta_{ij}$, respectively. The pheromone quantity is controlled by $\alpha$ where as $\beta$ controls the heuristic information.

If $\alpha = 0$, the solution only uses the heuristic information, and the pheromone quantity is zero. In this case, the probability of the closest ant is higher than the other ants for selecting the next node.

If $\beta = 0$, the solution does not use heuristic information and only depends on the pheromone quantity.

### *4.3. Development of the Proposed Hybrid SFLA-ACO Algorithm*

Many hybrid algorithms have been used in the literature for production planning and control. In the proposed algorithm, SFLA and ACO algorithms have been hybridized to improve the performance by improving the accuracy of the results.

The exploration of the design space is carried out by designing the different memeplex structures, which is a strong area of SFLA. The global optima are based on this exploration by using different memplexes. In the proposed algorithm, it is assumed that P1 and P2 are those holes that are maximum apart. There are two approaches or paths from P1 to reach P2. The remaining points are located on these two approaches or paths. After the selection of P1 and p2, the number of memplexes is defined. This number depends on the number of holes in the multi-hole drilling problem. The larger number of holes in a problem will suggest more memplexes and vice versa. The basic structure of a memplex is designed, and the remaining holes are allocated to the memplexes. If the total number of holes (N) in the drilling problem is even, then, excluding P1 and P2, the remaining holes (P) will be divided equally between the two paths available between P1 and P2. If the total number of holes (N) in the drilling problem is odd, then excluding P1 and P2, the remaining holes (P) will be divided between two paths available between P1 and P2 such that one path will have one more hole than the other. This way, the memplexes are defined to provide the exploration of the search space.

The local search is carried out by the ACO metaheuristic. The allocation of each hole in the path is performed by using Equation (7). Equation (7) defines a probability of each hole for the next allocation. This probability is a function of the tool travel distance from the present hole to all other non-allocated holes and the pheromone quantity. In order to avoid pheromone accumulation, which may cause the algorithm to stick in the local optima, pheromone evaporation is carried out at an appropriate rate. The probability for the nearer hole is more as compared to the farther hole, and vice versa, keeping in view the relative distances. A complete path is constructed step by step in this way until all the holes are allocated to the path. Once a path is completed, a solution is ready. The tour cost is computed and compared with the convergence criterion. If it is satisfied, the process ends, as the optimal result has been achieved. If not, then again, a local search is carried out. This is a repetitive process that is carried out for a predetermined number of iterations until the convergence criterion is met. In this way, the exploitation process for the local search is carried out with the help of the ACO algorithm by constructing the solution based on the probabilities of the non-visited holes. The flow diagram below, as Figure 1 explains the complete process of the hybrid SFLA-ACO algorithm.

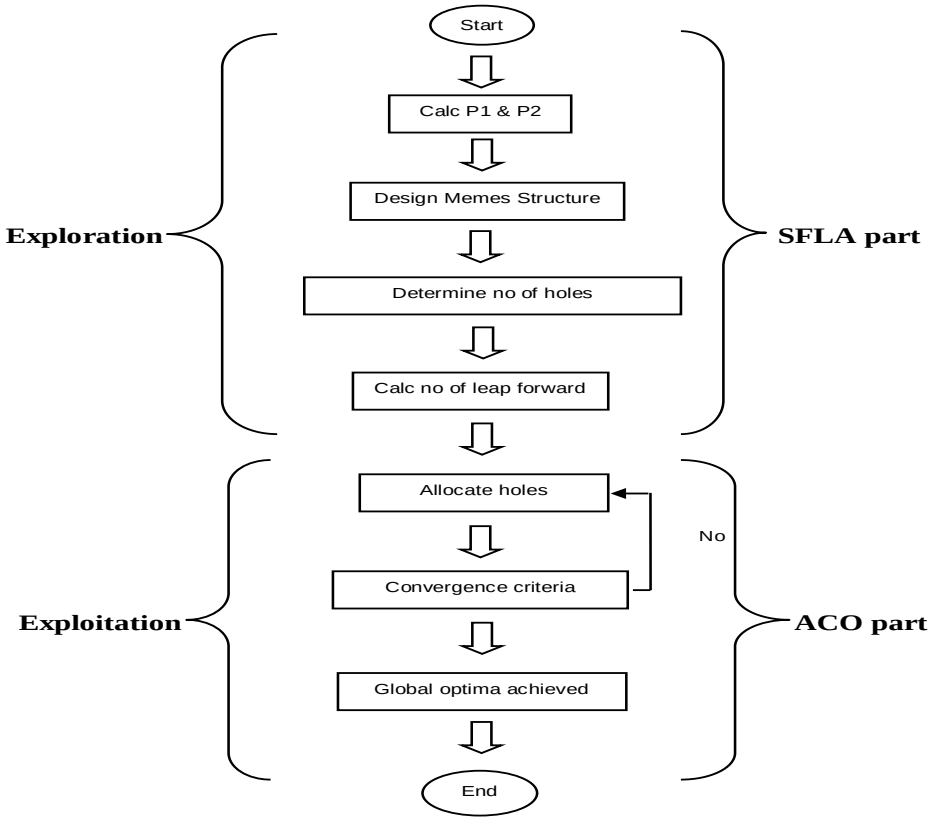


**Figure 1.** Flowchart of the hybrid SFLA-ACO algorithm.

## 5. Hybrid SFLA-ACO—Application to Drilling Benchmark Problems

This section presents the benchmark problems, the results obtained, and a comparison of the results with those of other metaheuristics.

### *5.1. Problem Definition*

The proposed hybrid algorithm was applied to five benchmark problems for computational experiments. For these problems, the number of holes was 5, 10, 15, 20, and 25 [22]. The complexity of the problems increased exponentially with the increasing number of holes, as evident by the design space, which was 120, $3.6 \times 10^6$, $1.3 \times 10^{12}$, $2.4 \times 10^{18}$, and $1.6 \times 10^{25}$ for 5, 10, 15, 20, and 25 holes, respectively. The following assumptions were made in solving the problems.

a. The tool returns to its initial position after visiting all holes once;
b. The number of rows in each problem is $\sqrt{J}$, where J represents the total number of holes;
c. The gap between holes in each direction is 2 cm;
d. The location of the Jth hole is shown in Figure 2.

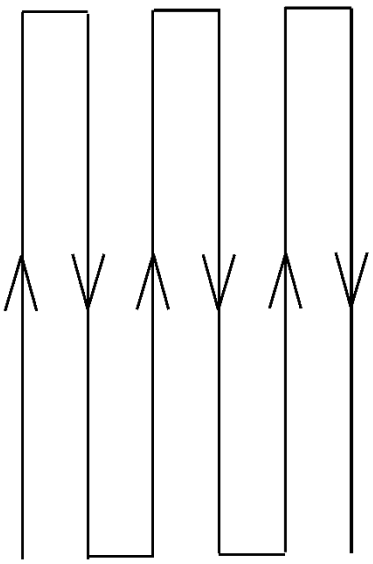

**Figure 2.** Finding the position of the Jth hole in the work piece.

For instance, if the number of holes in the work piece is 10, the number of rows is four as shown in Figure 3 below.

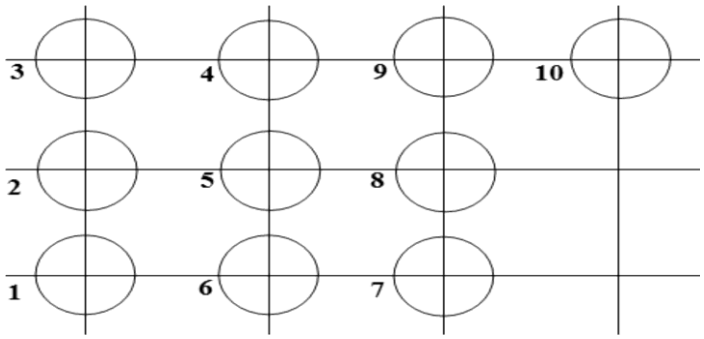


**Figure 3.** Holes locations for the problems with 10 holes.

The distance matrix obtained for the 25-hole problem is presented in Table 1.

**Table 1.** Euclidean distance matrix for the 25-hole problem.

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 2 | 4 | 6 | 8 | 8.24 | 6.32 | 4.47 | 2.82 | 2 | 4 | 4.47 | 5.65 |
| **2** | 2 | 0 | 2 | 4 | 6 | 6.32 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 4 | 4.47 |
| **3** | 4 | 2 | 0 | 2 | 4 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 5.65 | 4.47 | 4 |
| **4** | 6 | 4 | 2 | 0 | 2 | 2.82 | 2 | 2.82 | 4.47 | 6.32 | 7.21 | 6.65 | 4.47 |
| **5** | 8 | 6 | 4 | 2 | 0 | 2 | 2.82 | 4.47 | 6.32 | 8.24 | 8.94 | 7.21 | 5.65 |
| **6** | 8.24 | 6.32 | 4.47 | 2.82 | 2 | 0 | 2 | 4 | 6 | 8 | 8.24 | 6.32 | 4.47 |
| **7** | 6.32 | 4.47 | 2.82 | 2 | 2.82 | 2 | 0 | 2 | 4 | 6 | 6.32 | 4.47 | 2.82 |
| **8** | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 4 | 2 | 0 | 2 | 4 | 4.47 | 2.82 | 2 |
| **9** | 2.82 | 2 | 2.82 | 4.47 | 6.32 | 6 | 4 | 2 | 0 | 2 | 2.82 | 2 | 2.82 |
| **10** | 2 | 2.82 | 4.47 | 6.32 | 8.24 | 8 | 6 | 4 | 2 | 0 | 2 | 2.82 | 4.47 |
| **11** | 4 | 4.47 | 5.65 | 7.21 | 8.94 | 8.24 | 6.32 | 4.47 | 2.82 | 2 | 0 | 2 | 4 |
| **12** | 4.47 | 4 | 4.47 | 5.65 | 7.21 | 6.32 | 4.47 | 2.82 | 2 | 2.82 | 2 | 0 | 2 |
| **13** | 5.65 | 4.47 | 4 | 4.47 | 5.65 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 4 | 2 | 0 |
| **14** | 7.21 | 5.65 | 4.47 | 4 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 6.32 | 6 | 4 | 2 |
| **15** | 8.94 | 7.21 | 5.56 | 4.47 | 4 | 2 | 2.82 | 4.47 | 6.32 | 8.24 | 8 | 6 | 4 |

**Table 1.** *Cont.*

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **16** | 10 | 8.46 | 7.21 | 6.32 | 6 | 4 | 4.47 | 5.65 | 7.21 | 8.94 | 8.24 | 6.32 | 4.47 |
| **17** | 8.46 | 7.21 | 6.32 | 6 | 6.32 | 4.47 | 4 | 4.47 | 5.65 | 7.21 | 6.32 | 4.47 | 2.82 |
| **18** | 7.21 | 6.32 | 6 | 6.32 | 7.21 | 5.65 | 4.47 | 4 | 4.47 | 5.65 | 4.47 | 2.82 | 2 |
| **19** | 6.32 | 6 | 6.32 | 7.21 | 8.46 | 7.21 | 5.65 | 4.47 | 4 | 4.47 | 2.82 | 2 | 2.82 |
| **20** | 6 | 6.32 | 7.21 | 8.46 | 10 | 8.94 | 7.21 | 5.65 | 4.47 | 4 | 2 | 2.82 | 4.47 |
| **21** | 8 | 8.24 | 8.94 | 10 | 11.31 | 10 | 8.46 | 7.21 | 6.32 | 6 | 4 | 4.47 | 5.65 |
| **22** | 8.24 | 8 | 8.24 | 8.94 | 10 | 8.46 | 7.21 | 6.32 | 6 | 6.32 | 4.47 | 4 | 4.47 |
| **23** | 8.94 | 8.24 | 8 | 8.24 | 8.94 | 7.21 | 6.32 | 6 | 6.32 | 7.21 | 5.65 | 4.47 | 4 |
| **24** | 10 | 8.94 | 8.24 | 8 | 8.24 | 6.32 | 6 | 6.32 | 7.21 | 8.46 | 7.21 | 5.65 | 4.47 |
| **25** | 11.31 | 10 | 8.94 | 8.24 | 8 | 6 | 6.32 | 7.21 | 8.46 | 10 | 8.94 | 7.21 | 6.65 |
| | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | |
| **1** | 7.21 | 8.94 | 10 | 8.46 | 7.21 | 6.32 | 6 | 8 | 8.24 | 8.94 | 10 | 11.31 | |
| **2** | 5.62 | 7.21 | 8.46 | 7.21 | 6.32 | 6 | 6.32 | 8.24 | 8 | 8.24 | 8.94 | 10 | |
| **3** | 4.47 | 5.65 | 7.21 | 6.32 | 6 | 6.32 | 7.21 | 8.94 | 8.24 | 8 | 8.24 | 8.94 | |
| **4** | 4 | 4.47 | 6.32 | 6 | 6.32 | 7.21 | 8.46 | 10 | 8.94 | 8.24 | 8 | 8.24 | |
| **5** | 4.47 | 4 | 6 | 6.32 | 7.21 | 8.46 | 10 | 11.31 | 10 | 8.94 | 8.24 | 8 | |
| **6** | 2.82 | 2 | 4 | 4.47 | 5.65 | 7.21 | 8.94 | 10 | 8.46 | 7.21 | 6.32 | 6 | |
| **7** | 2 | 2.82 | 4.47 | 4 | 4.47 | 5.65 | 7.21 | 8.46 | 7.21 | 6.32 | 6 | 6.32 | |
| **8** | 2.82 | 4.47 | 5.65 | 4.47 | 4 | 4.47 | 5.65 | 7.21 | 6.32 | 6 | 6.32 | 7.21 | |
| **9** | 4.47 | 6.32 | 7.21 | 5.65 | 4.47 | 4 | 4.47 | 6.32 | 6 | 6.32 | 7.21 | 8.46 | |
| **10** | 6.32 | 8.24 | 8.94 | 7.21 | 5.65 | 4.47 | 4 | 6 | 6.32 | 7.21 | 8.46 | 10 | |
| **11** | 6 | 8 | 8.24 | 6.32 | 4.47 | 2.82 | 2 | 4 | 4.47 | 5.65 | 7.21 | 8.94 | |
| **12** | 4 | 6 | 6.32 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 4 | 4.47 | 5.65 | 7.21 | |
| **13** | 2 | 4 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 5.65 | 4.47 | 4 | 4.47 | 5.65 | |
| **14** | 0 | 2 | 2.82 | 2 | 2.82 | 4.47 | 6.32 | 7.21 | 5.65 | 4.47 | 4 | 4.47 | |
| **15** | 2 | 0 | 2 | 2.82 | 4.47 | 6.32 | 8.24 | 8.94 | 7.21 | 5.65 | 4.47 | 4 | |
| **16** | 2.82 | 2 | 0 | 2 | 4 | 6 | 8 | 8.24 | 6.32 | 4.47 | 2.82 | 2 | |
| **17** | 2 | 2.82 | 2 | 0 | 2 | 4 | 6 | 6.32 | 4.47 | 2.82 | 2 | 2.82 | |
| **18** | 2.82 | 4.47 | 4 | 2 | 0 | 2 | 4 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | |
| **19** | 4.47 | 6.32 | 6 | 4 | 2 | 0 | 2 | 2.82 | 2 | 2.82 | 4.47 | 6.32 | |
| **20** | 6.32 | 8.24 | 8 | 6 | 4 | 2 | 0 | 2 | 2.82 | 4.47 | 6.32 | 8.24 | |
| **21** | 7.21 | 8.94 | 8.24 | 6.32 | 4.47 | 2.82 | 2 | 0 | 2 | 4 | 6 | 8 | |
| **22** | 5.65 | 7.21 | 6.32 | 4.47 | 2.82 | 2 | 2.82 | 2 | 0 | 2 | 4 | 6 | |
| **23** | 4.47 | 5.65 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 4 | 2 | 0 | 2 | 4 | |
| **24** | 4 | 4.47 | 2.82 | 2 | 2.82 | 4.47 | 6.32 | 6 | 4 | 2 | 0 | 2 | |
| **25** | 4.47 | 4 | 2 | 2.82 | 4.47 | 6.32 | 8.24 | 8 | 6 | 4 | 2 | 0 | |

### *5.2. Formulation of Optimization Model and Results*

In multi-hole drilling problems, the costs involved are tool wear costs, tool switching costs, drilling costs, and tool traveling costs. In this study, only the tool travel cost was considered for optimization, as it consumed almost 70% of the total time. Moreover, the tool travel cost is non-productive and does not add value to the manufactured part. For these reasons, the tool travel cost was selected for optimization. Tool traveling cost is directly proportional to the non-productive traveling distance between the holes. In this study, the Euclidean distance is employed to calculate the distance between the hole positions. The non-productive tool traveling time $L_{ab}$ when the tool bit of the lathe machine moves from the current hole position "a" to the proceeding hole position "b" is calculated by Equation (1), in which Euclidian distance is used.

The proposed work is implemented using MATLAB and used Intel (R) Core (TM) 2 Dou CPU 17500 @ 2.20 GHz. The results are shown in Figure 4.

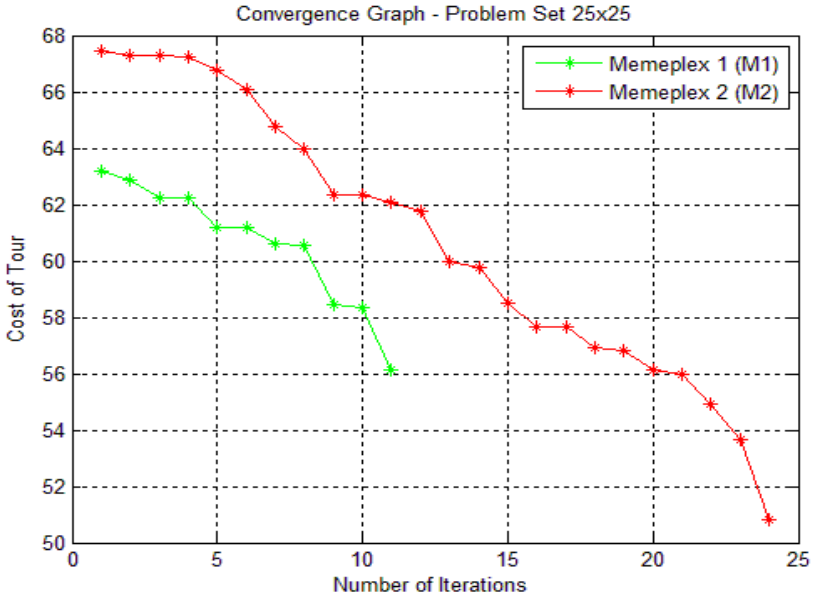


**Figure 4.** MATLAB graph shows the results of the 25-hole problem.

Figure 4 displays the cost of the tour and the number of iterations on the x-axis and the y-axis, respectively. The number of memeplexes used in this problem is two. i.e., M1 and M2. The range of results for M1 is from 63 to 56, and the number of iterations performed for M1 is 11. This means 63 is the worst solution and 56 is the best solution for the MI. Likewise, the range of results for M2 is from 67 to 50.82. This means that 67 is the worst solution and 50.82 is the best solution for the M2. The cost of 50.82 is considered the optimal cost, which has been calculated in M2 in less than 25 iterations. The optimal sequence with this optimal cost is 1, 2, 3, 4, 5, 6, 7, 8, 13, 18, 17, 14, 15, 16, 25, 24, 23, 22, 21, 20, 19, 12, 11, 10, 9, and 1. The computational time is 22.72 s for this problem.

### *5.3. Comparison of Results and Discussion*

A comparison between the results of the proposed hybrid algorithm and the results of the following four metaheuristic algorithms: DP, ACO, an immune-based evolutionary approach (IA), and a modified SFLA is carried out for evaluation of performances. Table 2 compares the traveling cost and path sequence performance among dynamic programming (DP), ACO, the IA, the modified SFLA, and the proposed hybrid algorithm.

The proposed algorithm outperformed DP, ACO, and IA. For the 5-hole problem, the total distance traveled by the spindle in DP, ACO, and the IA was 12, and the total distance traveled by the modified SFLA, and the proposed algorithm was 10.82. Thus, the proposed algorithm outperformed DP, ACO, and IA by 9.9%. For the 10-hole problem, the total distance traveled by DP, ACO, and the IA was 24, and the total distance traveled by the modified SFLA, and the proposed algorithm was 21.64. Thus, the proposed algorithm outperformed DP, ACO, and IA by 9.9%. For the 15-hole problem, the total distance traveled by DP, ACO, and the IA was 32, and the total distance traveled by the modified SFLA, and the proposed algorithm was 30.82. Thus, the proposed algorithm outperformed DP, ACO, and IA by 3.7%. For the 20-hole problem, the total distance traveled by DP, ACO, the IA, the modified SFLA, and the proposed algorithm was 40. Thus, the five algorithms exhibited similar performance. For the 25-hole problem, the DP algorithm was not applied. The total distance traveled by ACO, the IA, and the modified SFLA was 52, and the total distance traveled by the proposed algorithm was 50.82. Thus, the proposed algorithm outperformed ACO, the IA, and the modified SFLA by 2.3%. Overall, the results indicate that the proposed algorithm is more accurate than the other metaheuristic algorithms. These results show not only the efficacy of the hybrid SFLA-ACO algorithm but also show the edge over other metaheuristics with which it is compared.

**Table 2.** Comparison of results.

| Number of Holes | Objective Functions/Best Sequence | | | | |
|---|---|---|---|---|---|
| | **DP [22]** | **ACO [22]** | **IA [23]** | **Modified SFLA [24]** | **Proposed Algorithm** |
| 5 | 12<br>{1, 2, 3, 4, 5, 1} | 12<br>{1, 2, 3, 4, 5, 1} | 12<br>{1, 2, 3, 4, 5, 1} | 10.82<br>{1, 2, 3, 5, 4, 1} | 10.82<br>{2, 1, 4, 5, 3, 2} |
| 10 | 24<br>{1, 2, 3, 4, 5, 8, 9, 10, 7, 6, 1} | 24<br>{1, 2, 3, 4, 5, 8, 9, 10, 7, 6, 1} | 24<br>{1, 2, 3, 4, 9, 10, 8, 5, 7, 6, 1} | 21.64<br>{1, 2, 3, 4, 5, 9, 10, 8, 7, 6, 1} | 21.64<br>{1, 2, 3, 4, 9, 10, 6, 7, 6, 5, 1} |
| 15 | 32<br>{1, 2, 3, 4, 5, 8, 9, 10, 11, 14, 15, 13, 12, 7, 6, 1} | 32<br>{1, 2, 3, 4, 5, 8, 9, 10, 11, 14, 15, 13, 12, 7, 6, 1} | 32<br>{1, 2, 3, 4, 9, 10, 15, 14, 13, 12, 11, 7, 8, 5, 6, 1} | 30.82<br>{1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 10, 9, 8, 1} | 30.82<br>{1, 2, 3, 4, 5, 6, 11, 12, 13, 14, 15, 9, 10, 7, 8, 1} |
| 20 | 40<br>{1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 20, 19, 14, 15, 18, 17, 16, 9, 8, 1} | 40<br>{1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 20, 19, 14, 15, 18, 17, 16, 9, 8, 1} | 40<br>{1, 2, 4, 5, 12, 13, 20,19, 14, 15, 18, 17, 16, 9, 10, 11, 6, 7, 8, 1} | 40<br>{1, 7, 2, 3, 4, 5, 6, 11, 12, 10, 15, 14, 13, 20, 19, 18, 17, 16, 8, 9, 1} | 40<br>{1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 20, 19, 14, 15, 18, 17, 16, 9, 8, 1} |
| 25 | NA | 52<br>{1, 10, 11, 20, 21, 22, 19, 18, 17, 14, 15, 16, 25, 24, 23, 13, 12, 9, 8, 7, 6, 5, 4, 3, 2, 1} | 52<br>{1, 2, 3, 4, 5, 6, 15, 16, 25, 24, 17, 7, 14, 13, 8, 9, 12, 19, 18, 23, 22, 21, 20, 11, 10, 1} | 52<br>{1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 24, 25, 23, 21, 22, 18, 19, 20, 11, 10, 1} | 50.82<br>{1, 2, 3, 4, 5, 6, 7, 8, 13, 18, 17, 14, 15, 16, 25, 24, 23, 22, 21, 20, 19, 12,11, 10, 9, 1} |

## 6. Hybrid SFLA-ACO—Application to an Industrial Problem from the Literature

In this section, the problem statement, formulation of the optimization model, and results obtained are presented, along with a comparison of the results of the proposed model with those of the SFLA and modified SFLA.

### *6.1. Problem Statement*

The hybrid algorithm was applied to a problem from the literature [25]. Diagrams of the work piece for this case study are shown in Figures 5 and 6. The thickness of the sample was 80 mm.

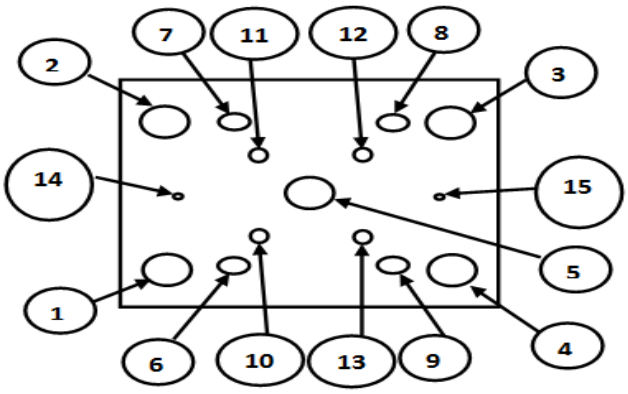


**Figure 5.** Location of holes.

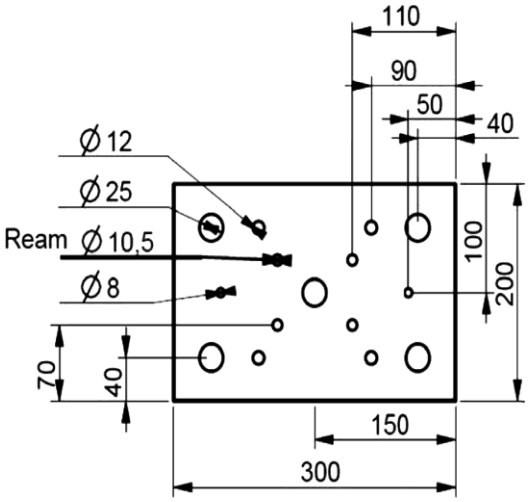


**Figure 6.** Diameters and coordinates of holes.

In the example part, 15 holes with four different diameters had to be drilled. The number of holes with each diameter was provided. For example, two Ø8-diameter holes had to be drilled. The tool requirements and the number of operations were calculated, as shown in Table 3. The total number of operations was 47, and the maximum number of operations was required for the diameter of Ø25; five tools were needed. Table 4 presents the details of the tools and the tool diameters required for hole-making in the work piece. For example, tool 1 had an 8-mm diameter, and the total number of operations was 15.

**Table 3.** Details of holes and tool requirements.

| | Dia | No. of Holes | Tool Req | Ops/Hole | Total Ops | Tool | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | Ø8 | 2 | 1 | 1 | 2 | 2 | - | - | - | - | - | - |
| 2 | Ø10.5 | 4 | 1, 2, 7 | 3 | 12 | 4 | 4 | - | - | - | - | 4 |
| 3 | Ø12 | 4 | 1, 3 | 2 | 8 | 4 | - | 4 | - | - | - | - |
| 4 | Ø25 | 5 | 1, 3, 4, 5, 6 | 5 | 25 | 5 | - | 5 | 5 | 5 | 5 | - |
| Total | 4 | 15 | 7 | | 47 | 15 | 4 | 9 | 5 | 5 | 5 | 4 |

**Table 4.** Details of tools and tool diameters.

| Tool Type | Drill | | | | | | Reamer | Total |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 7 |
| Tool dia (mm) | 8 | 10 | 12 | 16 | 20 | 25 | 10.5 | 7 |
| Ops | 15 | 4 | 9 | 5 | 5 | 5 | 4 | 47 |

In this problem, it was assumed that whenever the tool is switched for drilling, the time needed for the switching is always 2 s for the computer numerical control (CNC) machine and that the cost per unit tool switch time (a) is PKR. 25/min, and the cost per unit of non-productive traveling time (b) is PKR. 25/min. When tool d is used on hole e (Cde), the combined tool wear cost and machining or drilling hole costs are identical for all operations (PKR. 338).

### *6.2. Formulation of Optimization Model and Results Obtained*

For optimization, all the costs, i.e., the tool traveling cost, the tool switching cost, the tool wear cost, and the machining cost, must be minimized. In this problem, the combined tool and machining cost $C_{de}$ is constant (PKR. 338). Each tool switch takes 2 s and costs PKR. 25/min. The non-productive tool traveling cost is expended when the tool moves

from its present location to the next drilling location. It is directly proportional to the non-productive traveling distance between the holes. In this problem, it is assumed that a two-axis machine travels only in one direction. Thus, the rectilinear distance between the holes is used. The non-productive tool traveling time when the tool bit of the machine moves from the present hole location (d) to the next hole location (e) is calculated by Equation (2), which uses the rectilinear motion of the spindle.

The hybrid algorithm was used for the optimization, and the results were encouraging. The tool–hole combination sequences obtained using the proposed algorithm are presented in Table 5.

**Table 5.** Optimal sequences obtained using the proposed hybrid algorithm.

| ⟶ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Tool–hole combination | 8-H1 | 12-H1 | 16-H1 | 20-H1 | 25-H1 | 8-H6 | 12-H6 | 8-H10 | 10-H10 |
| Tool–hole combination | 10.5-H10 | 8-H13 | 10-H13 | 10.5-H13 | 8-H9 | 12-H9 | 8-H4 | 12-H4 | 16-H4 |
| Tool–hole combination | 20-H4 | 25-H4 | 8-H15 | 8-H3 | 12-H3 | 16-H3 | 20-H3 | 25-H3 | 8-H8 |
| Tool–hole combination | 12-H8 | 8-H12 | 10-H12 | 10.5-H12 | 8-H5 | 12-H5 | 16-H5 | 20-H5 | 25-H5 |
| Tool–hole combination | 8-H11 | 10-H11 | 10.5-H11 | 8-H7 | 12-H7 | 8-H2 | 2-H2 | 16-H2 | 20-H2 |
| Tool–hole combination | 25-H2 | 8-H14 | | | | | | | |

The sequence of holes for the optimal path was as follows: 1, 6, 10, 13, 9, 4, 15, 3, 8, 12, 5, 11, 7, 2, and 14. The total non-productive distance traveled by the spindle was 830 mm, and the total number of tool switches was 45.

### *6.3. Comparison of Results and Discussion*

In [25], the aforementioned problem was solved using the SFLA and modified SFLA. The results are presented in Tables 6 and 7, respectively.

**Table 6.** Optimal sequences obtained using the SFLA.

| ⟶ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Tool–hole combination | 8-H1 | 12-H1 | 16-H1 | 20-H1 | 25-H1 | 8-H2 | 12-H2 | 16-H2 | 20-H2 |
| Tool–hole combination | 25-H2 | 8-H3 | 12-H3 | 8-H4 | 16-H3 | 20-H3 | 25-H3 | 12-H4 | 16-H4 |
| Tool–hole combination | 20-H4 | 25-H4 | 8-H5 | 12-H5 | 16-H5 | 20-H5 | 25-H5 | 8-H6 | 12-H6 |
| Tool–hole combination | 8-H8 | 12-H8 | 8-H9 | 8-H10 | 10-H10 | 10.5-H10 | 8-H7 | 8-H11 | 10-H11 |
| Tool–hole combination | 8-H12 | 10-H12 | 10.5-H12 | 12-H9 | 8-H13 | 10-H13 | 8-H15 | 10.5-H13 | 10.5-H11 |
| Tool–hole combination | 12-H7 | 8-H14 | | | | | | | |

**Table 7.** Optimal sequences obtained using the modified SFLA.

| ⟶ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Tool–hole combination | 8-H1 | 12-H1 | 16-H1 | 20-H1 | 25-H1 | 8-H2 | 12-H2 | 16-H2 | 20-H2 |
| Tool–hole combination | 25-H2 | 8-H3 | 12-H3 | 16-H3 | 20-H3 | 25-H3 | 8-H4 | 12-H4 | 16-H4 |
| Tool–hole combination | 20-H4 | 25-H4 | 8-H5 | 12-H5 | 16-H5 | 20-H5 | 25-H5 | 8-H6 | 12-H6 |
| Tool–hole combination | 8-H10 | 8-H11 | 8-H | 8-H8 | 12-H8 | 8-H9 | 12-H9 | 10-H10 | 10.5-H10 |
| Tool–hole combination | 10-H11 | 8-H12 | 10-H12 | 8-H15 | 8-H13 | 10-H13 | 10.5-H13 | 10.5-H12 | 10.5-H11 |
| Tool–hole combination | 12-H7 | 8-H14 | | | | | | | |

Table 6 shows the sequence of holes for the optimized path obtained using the SFLA. The drilling sequence for the holes was as follows: 1, 2, 3, 4, 3, 4, 5, 6, 8, 9, 10, 7, 11, 12, 9, 13, 15, 13, 11, 7, and 14. The total non-productive distance traveled by the spindle was 2350 mm, and the total number of tool switches was 44.

Table 7 shows the sequence of holes for the optimized path obtained using the modified SFLA. The sequence of holes for the optimized path was 1, 2, 3, 4, 5, 6, 10, 11, 7, 8, 9, 10, 11, 12, 15, 13, 12, 11, 7, and 14. The total non-productive distance traveled by the spindle was 1920 mm, and the total number of tool switches was 40. The difference between Tables 6 and 7 is the difference in the approach of the local search, as the local search is modified in order to avoid premature convergence.

The results obtained using the SFLA and the modified SFLA in [25] were compared with those of the proposed hybrid algorithm, as shown in Table 8. Here, $L_{de}$ represents the

non-productive tool travel time needed to move the spindle from present point d to next point e in the rectilinear direction, and $S_{de}$ represents the time needed for switching the present tool in the spindle with the new tool required for drilling.

**Table 8.** Comparison of results.

| Method | $C_{de}$ (PKR) | $b \times l_{de}$ (PKR) | $a \times S_{de}$ (PKR) | Total Processing Cost (PKR) |
|---|---|---|---|---|
| SFLA | 338 | 52.07 | 32.25 | 422.32 |
| Modified SFLA | 338 | 43.75 | 30 | 411.75 |
| Hybrid SFLA-ACO | 338 | 20.75 | 37.5 | 396.25 |

In comparison, it has been observed that the tool wear cost and the cost of actual drilling or hole-making are the same for all the algorithms, as per the assumption made at the start. The total cost of non-productive time traveled by the spindle for the SFLA, the modified SFLA, and the proposed hybrid algorithm was PKR. 52.075, 43.75, and 20.75, respectively. Thus, the modified SFLA improved the non-productive tool traveling time by 16% in comparison with the results of the SFLA. The proposed hybrid algorithm improved the non-productive tool traveling time by 60.15% and 50.57% compared with the SFLA and modified SFLA, respectively. The total tool switch costs were PKR.32.25, 30, and 37.50 for the SFLA, the modified SFLA, and the proposed hybrid algorithm, respectively. The complete processing costs (including the hole-making cost, tool traveling cost, and tool switching cost) for the SFLA, the modified SFLA, and the proposed hybrid algorithm were PKR.422.325, 411.75, and 396.25, respectively. Thus, the modified SFLA improved the total cost by 2.5% compared with the SFLA. The hybrid SFLA-ACO algorithm improved the total cost by 6.17% and 3.76% compared with the SFLA and modified SFLA, respectively. As indicated by the results in Table 8, the proposed hybrid algorithm outperformed both the SFLA and the modified SFLA with regard to the total cost.

## 7. Conclusions

For 5-, 10-, and 15-hole benchmark problems, the proposed hybrid algorithm outperformed DP, ACO, and IA. For a 20-hole problem, the results of DP, ACO, IA, the modified SFLA, and the proposed algorithm were similar. For a 25-hole problem, the proposed algorithm outperformed DP, ACO, IA, and the modified SFLA. It can be concluded that the proposed algorithm is superior to DP, ACO, IA, and the modified SFLA. Additionally, the proposed algorithm outperformed the SFLA and the modified SFLA for industrial problems from the literature. The managerial implications of the proposed algorithm are positive, as it achieves better results than other metaheuristics. Better process planning and cost reductions are the main managerial implications. As drilling is the main machining process, which is frequently required in different parts manufacturing, therefore the scope of the proposed algorithm for multi-hole drilling problems is broad in manufacturing industries. The improved results signify the importance of the hybridization of metaheuristic algorithms. Researchers should investigate the possibilities of hybridization of different algorithms for the improvement and optimization of results. The results of this study indicate that the proposed algorithm is applicable to complex drilling tool path optimization problems. Additionally, the proposed algorithm can be applied to drilling tool path optimization problems in the real-world manufacturing industry, and industrial management can benefit from the hybridization of metaheuristic algorithms. The application of the proposed algorithm to real-world industrial problems will not only validate its efficacy but also highlight its limitations.

**Author Contributions:** Conceptualization, methodology, algorithm and its validation, formal analysis—N.M. Supervision, project management—M.U. The rigorous testing, manuscript evaluation and corrections—U.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Merchant, M.E. World trends and prospects in manufacturing technology. *Int. J. Veh. Des.* **1985**, *6*, 121–138.
2. Abbas, A.T.; Aly, M.F.; Hamza, K. Optimum drilling path planning for a rectangular matrix of holes using ant colony optimisation. *Int. J. Prod. Res.* **2011**, *49*, 5877–5891. [CrossRef]
3. Medina-Rodriguez, N.; Montiel-Ross, O. Tool path optimization for computer numerical control machines based on parallel ACO. *Eng. Lett.* **2012**, *20*, 8.
4. Saealal, M.S.; Abidin, A.F.; Adam, A.; Mukred, J.; Khalil, K.; Yusof, Z.M.; Ibrahim, Z.; Nordin, N. An ant colony system for routing in PCB holes drilling process. *Int. Symp. Innov. Manag. Inf. Prod.* **2012**, *3*, 50–56.
5. Ismail, M.M.; Othman, M.A.; Sulaiman, H.A.; Meor Said, M.A.; Misran, M.H.; Ramlee, R.A.; Sinnappa, M.; Zakaria, Z.; Ahma, B.H.; Abd Aziz, M.Z.A.; et al. Route planning analysis in holes drilling process using magnetic optimization algorithm for electronic manufacturing sector. *World Appl. Sci. J.* **2013**, *21*, 91–97.
6. Tamjidy, M.; Paslar, S.; Baharudin, B.; Hong, T.S.; Ariffin, M.K.A. Biogeography based optimization (BBO) algorithm to minimise non-productive time during hole-making process. *Int. J. Prod. Res.* **2015**, *53*, 1880–1894. [CrossRef]
7. Alwis, P.L.S.C.; Premarathna, A.S.; Fonseka, Y.P.; Samarasinghe, S.M.; Wijayakulasooriya, J.V. Automated printed circuit board (PCB) drilling machine with efficient path planning. In *SAITM Res Symposium on Engineering Advancements*; SAITM: Malabe, Sri Lanka, 2014.
8. Ahmad, M.; Beddu, S.; binti Itam, Z.; Alanimi, F.B.I. State of the art compendium of macro and micro energies. *Adv. Sci. Technol. Res. J.* **2019**, *13*, 88–109. [CrossRef]
9. Borkar, B.R.; Puri, Y.M.; Kuthe, A.M. Deshpande PS. Automatic CNC part programming for through hole drilling. *Procedia Mater. Sci.* **2014**, *5*, 2513–2521. [CrossRef]
10. Srivastava, P.R. A cooperative approach to optimize the printed circuit boards drill routing process using intelligent water drops. *Comput. Electr. Eng.* **2015**, *43*, 270–277. [CrossRef]
11. Dalvi, A.M.; Pawar, P.J.; Sing, T.P. Optimization of hole making operations for injection mould using particle swarm optimization algorithm. *Int. J. Ind. Eng. Comput.* **2015**, *6*, 433–444. [CrossRef]
12. Aziz, N.H.A.; Ab Aziz, N.A.; Ibrahim, Z.; Razali, S.; Abas, K.H.; Mohamad, M.S. A kalman filter approach to PCB drill path optimization problem. In Proceedings of the 2016 IEEE Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, 16–18 December 2016.
13. Daadoo, M. Path optimization for computer numerical control printed circuit boards in holes drilling process-case study. *Int. J. Eng. Technol.* **2016**, *6*, 365–377.
14. Dalavi, A.M. Optimal sequence of hole-making operations using particle swarm optimization and shuffled frog leaping algorithm. *Eng. Rev.* **2016**, *36*, 187–196.
15. Nguyen, T.T.; Pham, H.T.; Nguyen, T.H. Tool path optimization in CNC punching machine for sheet metal manufacturing. In Proceedings of the 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, 21–23 July 2017; pp. 381–386.
16. Zhang, W.B.; Zhu, G.Y. Drilling path optimization by optimal foraging algorithm. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2847–2856. [CrossRef]
17. Saravanan, M. *Tool Path optimization by Genetic algorithm for Energy Efficient Machining*; SWANSEA Printing Technology Ltd.: Swansea, UK, 2018.
18. Diyaley, S.; Burman Biswas, A.; Chakraborty, S. Determination of the optimal drill path sequence using bat algorithm and analysis of its optimization performance. *J. Ind. Prod. Eng.* **2019**, *36*, 97–112. [CrossRef]
19. Fok, K.-Y.; Ganganath, N.; Cheng, C.-T.; Iu, H.H.C.; Chi, K.T. Tool-path optimization using neural networks. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5.
20. Wang, J.; Xi, X.C.; Qin, L.; Zhang, Y.Q.; Zhao, W.S. Non productive time optimization for 5-axis EDM drilling using HVNTS algorithm. *Int. J. Prod. Res.* **2021**, *59*, 5068–5082. [CrossRef]
21. Garcia, H.R.; Romero, J.S.; Gomis, H.M.; Rao, R.V. Parallel implementation of metaheuristics for optimizing tool path computation on CNC machining. *Comput. Ind.* **2020**, *123*, 103322. [CrossRef]
22. Ghaiebi, H.; Solimanpur, M. An ant algorithm for optimization of hole-making operations. *Comput. Ind. Eng.* **2007**, *52*, 308–319. [CrossRef]
23. Hsieh, Y.C.; Lee, Y.C.; You, P.S. Using an effective immune based evolutionary approach for the optimal operation sequence of hole-making with multiple tools. *J. Comput. Inf. Syst.* **2011**, *7*, 411–418.

24. Dalavi, A.M.; Pawar, P.J.; Singh, T.P. Sequence optimization of hole-making operations for injection mould using shuffled frog leaping algorithm with modification. *Manag. Prod. Eng. Rev.* **2018**, *9*, 71–78.
25. Dalavi, A.M.; Pawar, P.J.; Singh, T.P. Tool path planning of hole-making operations in ejector plate of injection mould using modified shuffled frog leaping algorithm. *J. Comp. Des. Eng.* **2016**, *3*, 266–273. [CrossRef]